Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 205 (2005) 292–314

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

An approximate two-dimensional Riemann solver for hyperbolic systems of conservation laws

Vincent Guinot [*]

Laboratoire Hydrosciences, Hydrosciences Montpellier, UMR 5569, Université Montpellier 2, Maison des Sciences, de l'Eau – MSE, 34095 Montpellier Cedex 5, France

Received 19 August 2004; received in revised form 9 November 2004; accepted 10 November 2004
Available online 21 December 2004

Abstract

A two-dimensional Riemann solver is proposed for the solution of hyperbolic systems of conservation laws in two dimensions of space. The solver approximates the solution of a so-called angular two-dimensional Riemann problem as the weighted sum of the solutions of one-dimensional Riemann problems. The weights are proportional to the aperture of the regions of constant state. The two-dimensional solver is used to determine the solution of the equations at the cell vertices. The intercell fluxes are estimated using a linear combination between the point solutions at the cell vertices and the solutions of the one-dimensional problems at the centers of the cell interfaces. Besides allowing the computational time step to be increased the method gives more accurate results and is less sensitive to the anisotropy induced by the computational grid.
© 2004 Elsevier Inc. All rights reserved.

Keywords: Conservation laws; Shock waves; Godunov-type schemes; Riemann solver; Source term discretization; Shallow-water equations

1. Introduction

The existing finite volume-based techniques for multidimensional hyperbolic systems of conservation laws can roughly be classified into the following four categories:

1. Time splitting techniques [22,5,7] consist in solving successively the one-dimensional versions on the equations for each direction of space. Such techniques are easily implemented on structured (and more particularly Cartesian) grids in that they allow a wide range of one-dimensional numerical schemes to be

[*] Tel.: +33 4 67 14 90 56; fax: + 33 4 67 14 47 74.
E-mail address: guinot@msem.univ-montp2.fr.

doi:10.1016/j.jcp.2004.11.006

used without modification. This is most convenient when implementing TVD schemes, the TVD properties of which are easier to formulate and analyze in one dimension of space than in several dimensions.

2. Locally one-dimensional techniques, better known as finite volume techniques, consist in solving the governing equations in the local coordinate system attached to the interfaces between the computational cells [25,19,14]. The explicit versions of such techniques are characterized by a more restrictive stability constraint than the time splitting technique. Simple numerical tests on the linear advection equation advection show that strongly divergent flows may induce the failure of the technique when the wave propagation speeds are not computed properly, unless specific divergence correction techniques are used [12]. Such divergence correction techniques however break the local one-dimensional character of the approach. The use of locally rotated reference frames with respect to the grid has also been proposed in an attempt to eliminate the bias introduced by the grid [20,6] while retaining the simplicity of the one-dimensional Riemann problem solution.
3. Wave propagation-based techniques use a decomposition of the wave propagation paths along preferential directions. In the wave-splitting technique the wave propagation path is accounted for not only in the normal direction but also in the direction tangent to the cell interfaces. The method, applied to scalar conservation laws in [1], is generalized to hyperbolic systems in [18,17,8]. The importance of the flux gradients in the direction tangent to the interfaces to the accuracy of the solution is pointed out in [21,4]. The wave splitting approach is easy to implement on Cartesian grids but is quite time-consuming. This is because the solution of an $m \times m$ system requires m invariants to be computed in each direction of space, thus leading to m^N averaging operations for a problem in N dimensions. In the case of the two-dimensional shallow water equations ($m = 3$, $N = 2$, see [14] for the detailed implementation and a comparison with the previous two approaches), this leads to three times as many operations as in the case of wave splitting or the classical finite volume technique. The problem becomes more acute when the algorithm is used on unstructured grids of arbitrary shape because of the time-consuming character of the averaging operations over the domains of dependence of the cell interfaces.
4. Mixed finite differences-finite volume techniques. Such techniques use the point solutions of the multidimensional Riemann problems at the cell vertices to provide an interpolation of the fluxes along the interfaces between the computational cells. They lead to a stability criterion similar to that of the time splitting technique. A two-dimensional solver based on a linear superposition of one-dimensional Riemann problems was recently proposed in [3] and applied to the two-dimensional Euler equations of gas dynamics. In [13] a two-dimensional Riemann solver was presented for the solution of the two-dimensional shallow water equations. This solver makes use of the multidimensional extension of the characteristic formulation of the conservation laws (also called bicharacteristics formulation, see [24,23,9,15] for examples from various fields of physics) to convert the two-dimensional Riemann problems at the cell corners into equivalent, one-dimensional Riemann problems in the direction normal to the cell interfaces. The solver presented in [13] has the drawbacks that (i) it is formulated only for structured grids and (ii) the equivalent, one-dimensional Riemann problem (hence the solution) depends to some extent on the direction in which the equivalent Riemann problem is to be defined. Therefore, the number of equivalent Riemann problems to be defined at a given vertex is equal to the number of interfaces to which this vertex belongs, which makes the procedure time-consuming.

This paper presents the details of a two-dimensional Riemann solver that does not exhibit the directional bias of the solver presented in [13]. In contrast with the solver devised in [3] the solution is sought as the combination of solutions of one-dimensional problems along the bisectors of the angular sectors of constant state. Section 2 indicates how the solution of the two-dimensional solver at the cell vertices should be used to compute the fluxes between the computational cells within the finite volume approach. Section 3 presents the details of the two-dimensional solver and provides the specific formulations for structured grids. Section 4 presents the options retained for source term discretization. Section 5 provides an application example to the 3×3 system of the two-dimensional shallow water equations. Section 6 is devoted to concluding remarks.

2. Outline of the approach

2.1. Solution principle

The proposed solvers are designed for the solution of two-dimensional hyperbolic systems of conservation laws that can be written in the following vector form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S}, \tag{2.1}$$

where **U** is the conserved vector variable, **F** and **G** are the fluxes in the x- and y-directions, respectively, and **S** is the source term. Eq. (2.1) can be rewritten in non-conservation form as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y} = \mathbf{S}, \tag{2.2}$$

where **A** and **B** are the Jacobian matrices of **F** and **G** with respect to **U**, respectively. The system (2.1) is said to be hyperbolic if any linear combination of **A** and **B** has m real eigenvalues and m linearly independent eigenvectors, m being the size of the system. In the present paper, the conservation part of Eq. (2.1) is discretized as

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{A_i} \sum_{j \in N(i)} \left(\mathbf{F}_{i,j} n_{i,j}^{(x)} + \mathbf{G}_{i,j} n_{i,j}^{(y)} \right) w_{i,j}, \tag{2.3}$$

where $\mathbf{U}_i^n$ is the average value of **U** over the cell i at the time level n, A_i is the area of the cell i, $N(i)$ is the set of neighbor cells if the cell i, $w_{i,j}$ is the width of the interface (i,j) between the cells i and j, $n_{i,j}^{(x)}$ and $n_{i,j}^{(y)}$ are the x- and y-components of the normal unit vector of the interface (i,j), oriented from i to j, and Δt is the computational time step. Assuming that $\mathbf{U}_i^n$ is known, $\mathbf{U}_i^{n+1}$ is computed by estimating the values of **F** and **G** at the interfaces (i,j). The fluxes are usually approximated as:

$$\left. \begin{aligned} \mathbf{F}_{i,j} &= \mathbf{F}(\mathbf{U}_{i,j}^{n+1/2}), \\ \mathbf{G}_{i,j} &= \mathbf{G}(\mathbf{U}_{i,j}^{n+1/2}), \end{aligned} \right\} \tag{2.4}$$

where $\mathbf{U}_{i,j}^{n+1/2}$ is the average value of **U** over the interface (i, j) between the time levels n and $n + 1$. $\mathbf{U}_{i,j}^{n+1/2}$ is the solution of the Riemann problem in the direction normal to the interface (i, j), defined as

$$\mathbf{U}_{i,j}(\eta, t^n) = \begin{cases} \mathbf{U}_{i,j,\mathrm{L}} & \text{for } \eta < 0, \\ \mathbf{U}_{i,j,\mathrm{R}} & \text{for } \eta > 0, \end{cases} \tag{2.5}$$

where η is the coordinate normal to the interface. The left and right states $\mathbf{U}_{i,j,\mathrm{L}}$ and $\mathbf{U}_{i,j,\mathrm{R}}$ of the Riemann problem are determined form the distribution of the variable **U** in the cells i and j. In the proposed approach the fluxes are computed using the solution of the Riemann problem (2.5) at the middle of the interfaces and the solution of two-dimensional Riemann problems at the vertices of the computational cells (Fig. 1). In a first option a linearization is performed on the conserved variable **U:**

$$\left. \begin{aligned} \mathbf{F}_{i,j} &= \mathbf{F}\left[(\mathbf{U}_{k_1}^{n+1/2} + \mathbf{U}_{k_2}^{n+1/2})\beta + (1 - 2\beta)\mathbf{U}_{i,j}^{n+1/2} \right], \\ \mathbf{G}_{i,j} &= \mathbf{G}\left[(\mathbf{U}_{k_1}^{n+1/2} + \mathbf{U}_{k_2}^{n+1/2})\beta + (1 - 2\beta)\mathbf{U}_{i,j}^{n+1/2} \right], \end{aligned} \right\} \tag{2.6}$$

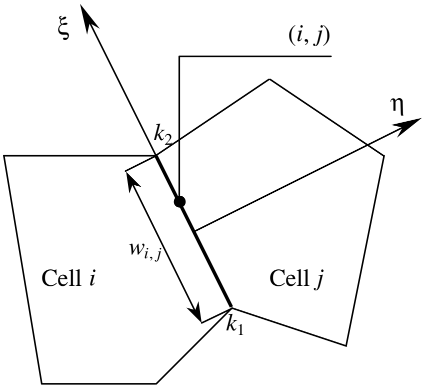


Fig. 1. Definition sketch for the computation of the intercell fluxes.

where the subscripts k_1 and k_2 denote the point values at the two vertices k_1 and k_2 of the interface (*i, j*) and β is a weighting coefficient between 0 and 1/2. In a second option the linearization is performed over the fluxes **F** and **G**:

$$\left.\begin{aligned} \mathbf{F}_{i,j} &= \left[\mathbf{F}(\mathbf{U}_{k_1}^{n+1/2}) + \mathbf{F}(\mathbf{U}_{k_2}^{n+1/2})\right]\beta + (1-2\beta)\mathbf{F}(\mathbf{U}_{i,j}^{n+1/2}), \\ \mathbf{G}_{i,j} &= \left[\mathbf{G}(\mathbf{U}_{k_1}^{n+1/2}) + \mathbf{G}(\mathbf{U}_{k_2}^{n+1/2})\right]\beta + (1-2\beta)\mathbf{G}(\mathbf{U}_{i,j}^{n+1/2}), \end{aligned}\right\} \tag{2.7}$$

Note that under the assumption of a piecewise linear distribution of **U** (or **F** and **G**) along the interface β must be set to 1/4, and to 1/6 if a parabolic distribution is assumed. β = 1/3 corresponds to an equal weighting between the values at the cell corners and at the middle of the interfaces. β = 1/2 does not appear as a valid option in that it gives full weight to the values at the vertices and none to the value at the center of the interface. The key point of the present algorithm is the estimate of the solutions $\mathbf{U}_{k_1}^{n+1/2}$ and $\mathbf{U}_{k_2}^{n+1/2}$ of the two-dimensional Riemann problems at the vertices k_1 and k_2 This aspect is detailed in Section 3.

2.2. Overview of the algorithm

1. Reconstruct the variable **U** over the computational cells. The reconstructed variable over the cell *i* is denoted by $\tilde{\mathbf{U}}_i^n(x,y)$.
2. The reconstructed profiles over the computational cells are used to form to sets of Generalized Riemann problems. The first set is that of the Riemann problems in the direction normal to the interfaces between the computational cells. The second set is that of the two-dimensional, angular Riemann problems at the cell vertices. How such two-dimensional Riemann problems are defined and solved is the subject of Section 3.
3. The one-dimensional Riemann problems are solved at the cell interfaces. Their solution provides the set of values $\mathbf{U}_{i,j}^{n+1/2}$ at the centers of the interfaces (*i, j*). Solving the two-dimensional Riemann problems at the cell vertices *k* provides the values $\mathbf{U}_k^{n+1/2}$.
4. The fluxes $\mathbf{F}_{i,j}$ and $\mathbf{G}_{i,j}$ are computed using Eq. (2.6) or (2.7) and the solution is advanced in time using Eq. (2.3).

3. Two-dimensional Riemann problems

3.1. Principle of the solution method

A two-dimensional angular Riemann problem is defined as the following initial-value problem in the polar coordinate system ($\rho\theta$):

$$\left.\begin{aligned}&\tfrac{\partial \mathbf{U}}{\partial t}+\tfrac{\partial \mathbf{F}}{\partial x}+\tfrac{\partial \mathbf{G}}{\partial y}=0,\\ &\mathbf{U}(\rho,\theta,t^n)=\begin{cases}\mathbf{U}_{\mathrm{R}}^{(p)} & \text{for } \theta\in]\theta_{p-1},\theta_p[\\ \mathbf{U}_{\mathrm{L}}^{(p)} & \text{for } \theta\in]\theta_{p-1}+\pi,\theta_p+\pi[\end{cases}\quad (p=1,\ldots,P),\end{aligned}\right\}\tag{3.1}$$

where P is the number of pairs of regions of constant state, and ρ and θ are the radius and the angle from a reference point and direction, respectively. In what follows the x-axis is taken as reference direction (Fig. 2). The purpose of the present section is not to determine the solution of the Riemann problem (3.1) at all points of space but only at the location of the initial discontinuity, that is in x, $y = 0$ in a Cartesian coordinate system or $\rho = 0$ in the polar coordinate system. The approximate solution is sought in the form

$$\mathbf{U}(0,\theta,t>t^n)=\int_0^{\pi}\phi(\chi)\mathbf{U}^{(p)}(0,0,t>t^n)\,\mathrm{d}\chi,\tag{3.2}$$

where $\phi(\theta)$ is a weighting coefficient and $\mathbf{U}^{(p)}(0,0,t)$ is the solution of the one-dimensional Riemann problem defined as

$$\left.\begin{aligned}&\tfrac{\partial \mathbf{U}}{\partial t}+\tfrac{\partial \mathbf{F}}{\partial x}+\tfrac{\partial \mathbf{G}}{\partial y}=0,\\ &\mathbf{U}^{(p)}(\eta^{(p)},\xi^{(p)},t^n)=\begin{cases}\mathbf{U}_{\mathrm{L}}^{(p)} & \text{for } \eta^{(p)}<0,\\ \mathbf{U}_{\mathrm{R}}^{(p)} & \text{for } \eta^{(p)}>0,\end{cases}\end{aligned}\right\}\tag{3.3}$$

where $(\eta^{(p)},\xi^{(p)})$ is the coordinate system obtained by rotating the coordinate system (x,y) by an angle $(\theta_{p-1} + \theta_p)/2$ (see Fig. 2). The expression of $\phi(\chi)$ is easily determined by noticing that ϕ is independent of the choice of the origin for χ. Therefore $\phi(\chi)$ is a constant. Assuming a uniform initial distribution $\mathbf{U}(\rho,\theta,t^n)$, the solution is constant with respect to time, equal to the initial constant distribution and Eq. (3.2) leads to

$$\int_0^{\pi}\phi\,\mathrm{d}\chi=1.\tag{3.4}$$

Therefore $\phi = 1/\pi$ and Eq. (3.2) becomes

$$\mathbf{U}(0,\theta,t>t^n)=\frac{1}{\pi}\int_0^{\pi}\mathbf{U}_{\rho}(\eta,\chi,t>t^n)\,\mathrm{d}\chi.\tag{3.5}$$

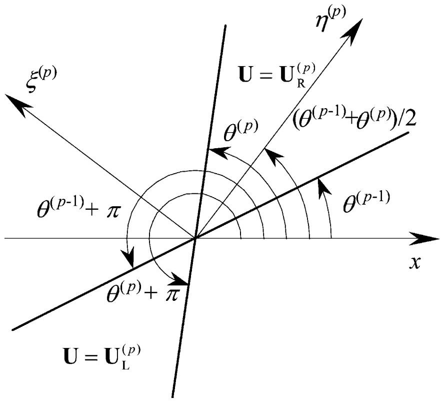

$\eta^{(p)}$
$\xi^{(p)}$
$\mathbf{U}=\mathbf{U}_{\mathrm{R}}^{(p)}$
$\theta^{(p)}$
$(\theta^{(p-1)}+\theta^{(p)})/2$
$\theta^{(p-1)}+\pi$
$\theta^{(p-1)}$
x
$\theta^{(p)}+\pi$
$\mathbf{U}=\mathbf{U}_{\mathrm{L}}^{(p)}$

Fig. 2. Definition sketch of the two-dimensional Riemann problem.

Substituting Eq. (3.1) into Eq. (3.5) yields

$$\mathbf{U}(0,\theta,t > t^n) = \sum_{p=1}^{P} \frac{\theta_p - \theta_{p-1}}{\pi} \mathbf{U}^{(p)}(0,0,t > t^n), \quad \theta \in [0,2\pi[. \tag{3.6}$$

The one-dimensional Riemann problem (3.3) is solved by rewriting Eq. (3.3) in the local coordinate system $(\eta^{(p)},\xi^{(p)})$:

$$\left.\begin{array}{l} \frac{\partial \mathbf{U}^{(p)}}{\partial t} + \frac{\partial \mathbf{F}^{(p)}}{\partial \eta^{(p)}} + \frac{\partial \mathbf{G}^{(p)}}{\partial \xi^{(p)}} = 0, \\ \mathbf{U}^{(p)}(\eta^{(p)},\xi^{(p)},t^n) = \begin{cases} \mathbf{U}_{\mathrm{L}}^{(p)} & \text{for } \eta^{(p)} < 0, \\ \mathbf{U}_{\mathrm{R}}^{(p)} & \text{for } \eta^{(p)} > 0, \end{cases} \end{array}\right\} \tag{3.7}$$

where the superscript (p) denotes the expressions of the fluxes in the local coordinate system. Note that in this formulation the gradient of the transverse flux $\mathbf{G}^{(p)}$ is considered as a source term. The solution of Eq. (3.7) is computed in two steps. The first step consists in solving the Riemann problem in the η direction and assigning the result to $\mathbf{U}^{(p)}$

$$\mathbf{U}^{(p)} = \mathbf{U}_{\eta}^{(p)}, \tag{3.8}$$

where $\mathbf{U}_{\eta}^{(p)}$ is the solution of the following one-dimensional Riemann problem in the η direction:

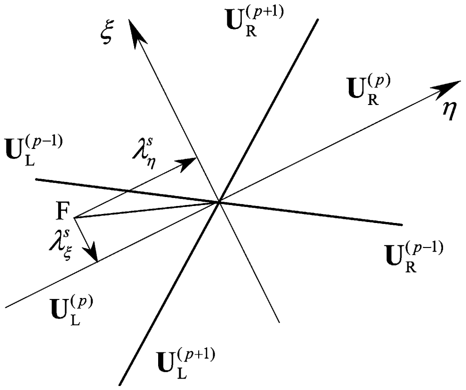
$$\left.\begin{array}{l} \frac{\partial \mathbf{U}^{(p)}}{\partial t} + \frac{\partial \mathbf{F}^{(p)}}{\partial \eta^{(p)}} = 0, \\ \mathbf{U}^{(p)}(\eta^{(p)},\xi^{(p)},t^n) = \begin{cases} \mathbf{U}_{\mathrm{L}}^{(p)} & \text{for } \eta^{(p)} < 0, \\ \mathbf{U}_{\mathrm{R}}^{(p)} & \text{for } \eta^{(p)} > 0, \end{cases} \end{array}\right\} \tag{3.9}$$

$\mathbf{U}_{\eta}^{(p)}$ is computed using any standard, exact or approximate Riemann solver and used as a basis to construct a Riemann problem in the ξ direction. Following the wave splitting approach [17] the states of the transverse Riemann problem are characterized using the wave propagation speeds computed from $\mathbf{U}_{\eta}^{(p)}$. The Riemann problem in the ξ direction is defined as:

$$\left.\begin{array}{l} \frac{\partial \mathbf{U}^{(p)}}{\partial t} + \frac{\partial \mathbf{G}^{(p)}}{\partial \xi^{(p)}} = 0, \\ \mathbf{U}^{(p)}(\eta^{(p)},\xi^{(p)},t^n) = \begin{cases} \mathbf{U}_{\xi,\mathrm{L}}^{(p)} & \text{for } \xi^{(p)} < 0, \\ \mathbf{U}_{\xi,\mathrm{R}}^{(p)} & \text{for } \xi^{(p)} > 0, \end{cases} \end{array}\right\} \tag{3.10}$$

where the states $\mathbf{U}_{\xi,\ \mathrm{S}}{}^{(p)}$ (S = L,R) are defined as the states across the interface to which the foot of the characteristic issued from ($\eta = 0$, $\xi = 0$) is closer (see Fig. 3):

$$\begin{array}{l} \mathbf{U}_{\xi,\mathrm{L}}^{(p)} = \begin{cases} \mathbf{U}_{\mathrm{L}}^{(p+1)} & \text{if } \lambda_{\eta}^{s} > 0 \text{ and } \lambda_{\xi}^{s} > 0, \\ \mathbf{U}_{\mathrm{L}}^{(p)} & \text{if } \lambda_{\eta}^{s} > 0 \text{ and } \lambda_{\xi}^{s} < 0, \\ \mathbf{U}_{\mathrm{R}}^{(p-1)} & \text{if } \lambda_{\eta}^{s} < 0 \text{ and } \lambda_{\xi}^{s} > 0, \\ \mathbf{U}_{\mathrm{R}}^{(p)} & \text{if } \lambda_{\eta}^{s} < 0 \text{ and } \lambda_{\xi}^{s} < 0, \end{cases} \\ \mathbf{U}_{\xi,\mathrm{R}}^{(p)} = \begin{cases} \mathbf{U}_{\mathrm{L}}^{(p)} & \text{if } \lambda_{\eta}^{s} > 0 \text{ and } \lambda_{\xi}^{s} > 0, \\ \mathbf{U}_{\mathrm{L}}^{(p-1)} & \text{if } \lambda_{\eta}^{s} > 0 \text{ and } \lambda_{\xi}^{s} < 0, \\ \mathbf{U}_{\mathrm{R}}^{(p)} & \text{if } \lambda_{\eta}^{s} < 0 \text{ and } \lambda_{\xi}^{s} > 0, \\ \mathbf{U}_{\mathrm{R}}^{(p+1)} & \text{if } \lambda_{\eta}^{s} < 0 \text{ and } \lambda_{\xi}^{s} < 0, \end{cases} \end{array} \tag{3.11}$$



ξ
$\mathbf{U}_{\mathrm{R}}^{(p+1)}$
$\mathbf{U}_{\mathrm{R}}^{(p)}$
η
$\mathbf{U}_{\mathrm{L}}^{(p-1)}$
λ_η^s
F
λ_ξ^s
$\mathbf{U}_{\mathrm{R}}^{(p-1)}$
$\mathbf{U}_{\mathrm{L}}^{(p)}$
$\mathbf{U}_{\mathrm{L}}^{(p+1)}$

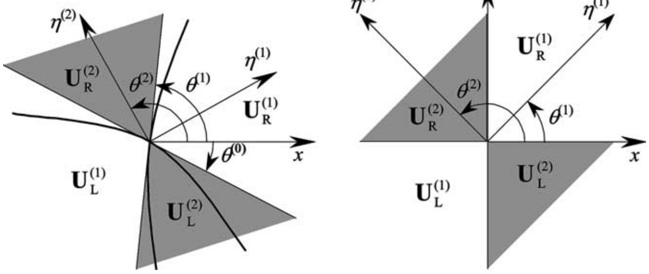
Fig. 3. Definition sketch for the characterization of the Riemann problem in the ξ-direction. In the present example the upwind trajectory is closer to the interface between the left states of the zones $p-1$ and p, therefore the left and right states of the problem are set to $\mathbf{U}_{\mathrm{L}}^{(p)}$ and $\mathbf{U}_{\mathrm{L}}^{(p-1)}$, respectively.

where the eigenvalues λ_η^s and λ_ξ^s are the η and ξ wave speeds, respectively. In most equations of fluid dynamics, λ_η^s and λ_ξ^s are equal to the sum of the propagation speed of the pressure waves and the flow velocities in the η and ξ directions, respectively. Since the function of Eq. (3.11) is to provide information about the location of the domain of dependence, the η and ξ velocities provide a sufficiently correct approximation of λ_η^s and λ_ξ^s. This has been confirmed by numerical experiments carried out on the shallow water equations. The final solution $\mathbf{U}^{(p)}$ is the solution of the Riemann problem (3.10) and (3.11). Note that this second Riemann problem should not be solved if the domain of dependence of the origin ($\eta = 0$, $\xi = 0$) is located entirely in the domain p. If it is the case the solution of the Riemann problem in the ξ direction cannot influence the final solution at ($\eta = 0$, $\xi = 0$). This particular restriction ensures that the exact solution is obtained in the case of a scalar conservation law, as shown in Section 3.3.

3.2. A particular case: curvilinear and cartesian grids

In the particular case of a curvilinear grid the two-dimensional angular Riemann problem is a two zone problem ($P = 2$, see Fig. 4). The angles are defined by the local tangents at the intersection between the grid lines. The estimate of the transverse flux is simplified by noticing that:

$$\left.\begin{aligned}\mathbf{U}_{\xi,\mathrm{L}}^{(1)} &= \mathbf{U}_{\xi,\mathrm{R}}^{(1)},\\ \mathbf{U}_{\xi,\mathrm{L}}^{(2)} &= \mathbf{U}_{\xi,\mathrm{R}}^{(2)}.\end{aligned}\right\} \tag{3.12}$$



$\eta^{(2)}$
$\mathbf{U}_{\mathrm{R}}^{(2)}$
$\theta^{(2)}$
$\theta^{(1)}$
$\eta^{(1)}$
$\mathbf{U}_{\mathrm{R}}^{(1)}$
$\theta^{(0)}$
x
$\mathbf{U}_{\mathrm{L}}^{(1)}$
$\mathbf{U}_{\mathrm{L}}^{(2)}$
$\eta^{(2)}$
y
$\eta^{(1)}$
$\mathbf{U}_{\mathrm{R}}^{(1)}$
$\mathbf{U}_{\mathrm{R}}^{(2)}$
$\theta^{(2)}$
$\theta^{(1)}$
x
$\mathbf{U}_{\mathrm{L}}^{(1)}$
$\mathbf{U}_{\mathrm{L}}^{(2)}$

Fig. 4. Two-dimensional Riemann problems for a general curvilinear (left) and a Cartesian grid (right).

Therefore the determination of the states $\mathbf{U}_{\xi,\mathrm{S}}^{(p)}$ of the Riemann problem in the x-direction is conditioned only by the sign of λ_ξ^s. The symmetry considerations of Eq. (3.12) lead to the following simplification of Eq. (3.11):

$$\begin{aligned}\mathbf{U}_{\xi,\mathrm{L}}^{(1)} &= \begin{cases}\mathbf{U}_\mathrm{L}^{(2)} & \text{if } \lambda_\xi^s > 0,\\ \mathbf{U}_\eta^{(1)} & \text{if } \lambda_\xi^s < 0,\end{cases}\\ \mathbf{U}_{\xi,\mathrm{R}}^{(1)} &= \begin{cases}\mathbf{U}_\eta^{(1)} & \text{if } \lambda_\xi^s > 0,\\ \mathbf{U}_\mathrm{R}^{(2)} & \text{if } \lambda_\xi^s < 0.\end{cases}\end{aligned} \tag{3.13}$$

3.3. Application to the scalar laws on Cartesian grids

The algorithm above can be further simplified when scalar laws are to be solved on Cartesian grids. Consider the two-dimensional Riemann problem at the vertex ($i - 1/2$, $j - 1/2$) between the four cells ($i - 1$, $j - 1$), ($i - 1$, j), (i,$j - 1$) and (i,j), to be solved for the scalar law

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0, \tag{3.14}$$

that can be rewritten in characteristic form as

$$\frac{\partial U}{\partial t} + \lambda_x \frac{\partial U}{\partial x} + \lambda_y \frac{\partial U}{\partial y} = 0 \tag{3.15}$$

with:

$$\left.\begin{aligned}\lambda_x &= \tfrac{\mathrm{d}F}{\mathrm{d}U},\\ \lambda_y &= \tfrac{\mathrm{d}G}{\mathrm{d}U}.\end{aligned}\right\} \tag{3.16}$$

Assume that both λ_x and λ_y are positive. Then the domain of dependence of the vertex ($i - 1/2$, $j - 1/2$) is located in the cell ($i - 1$, $j - 1$) and the exact solution is known

$$U(x_{i-1/2}, y_{j-1/2}, t > t^n) = U_{i-1,j-1}^n. \tag{3.17}$$

From Eq. (3.6), the solution is given by

$$U(x_{i-1/2}, y_{j-1/2}, t > t^n) = \frac{U^{(1)} + U^{(2)}}{2}, \tag{3.18}$$

where $U^{(1)}$ and $U^{(2)}$ are the solutions of the Riemann problems in the coordinate systems ($\eta^{(1)}$,$\xi^{(1)}$) and ($\eta^{(2)}$,$\xi^{(2)}$), respectively.

The Riemann problem for the first pair of regions of constant state is characterized by the following values:

$$\left.\begin{aligned}\theta^{(0)} &= -\pi/4,\\ \theta^{(1)} &= \pi/4,\\ U_\mathrm{L}^{(1)} &= U_{i-1,j},\\ U_\mathrm{R}^{(1)} &= U_{i,j-1},\\ U_{\xi,\mathrm{L}}^{(1)} &= U_{i-1,j-1},\\ U_{\xi,\mathrm{R}}^{(1)} &= U_\eta^{(1)}.\end{aligned}\right\} \tag{3.19}$$

The solution $U_\eta^{(1)}$ of the Riemann problem in the $\eta^{(1)}$-direction is equal to $U_{\mathrm{L}}^{(1)}$ if the projection of the velocity vector (λ_x,λ_y) onto the axis $\eta^{(1)}$ is positive, and equal to $U_{\mathrm{R}}^{(1)}$ otherwise. Since the domain of dependence of the solution lies in the cell $(i-1, j-1)$ it does not belong to the first region of constant state and the Riemann problem in the direction $\xi^{(1)}$ must be solved. The solution of this problem is $U_{\xi,\mathrm{L}}^{(1)}$, that is $U_{i-1,j-1}$. The final solution for the first pair of regions of constant state is therefore

$$U^{(1)} = U_{i-1,j-1}. \tag{3.20}$$

The Riemann problem for the second pair of regions of constant state is characterized by the following values:

$$\left.\begin{aligned} \theta^{(1)} &= \pi/4, \\ \theta^{(2)} &= 3\pi/4, \\ U_{\mathrm{L}}^{(2)} &= U_{i-1,j-1}, \\ U_{\mathrm{R}}^{(2)} &= U_{i,j}. \end{aligned}\right\} \tag{3.21}$$

The projection of the velocity vector (λ_x,λ_y) onto the axis $\eta^{(2)}$ being positive, the solution $U_\eta^{(2)}$ of the Riemann problem in the $\eta^{(2)}$ direction is equal to $U_{\mathrm{L}}^{(2)}$. Since the domain of dependence of the solution lies in the cell $(i-1, j-1)$ it lies entirely in the second pair of regions of constant state and no Riemann problem needs to be solved in the direction $\xi^{(2)}$. Therefore $U^{(2)}$ remains equal to

$$U^{(2)} = U_{i-1,j-1}. \tag{3.22}$$

Substituting Eqs. (3.20) and (3.22) into Eq. (3.18) yields

$$U(x_{i-1/2},y_{j-1/2},t > t^n) = U_{i-1,j-1}. \tag{3.23}$$

Therefore the exact solution is obtained. The reasoning above is valid whatever the signs of λ_x and λ_y.

4. Discretization of source terms

Two types of source terms are considered in the present section. Source terms of the first type, denoted by $\mathbf{S}_1$, are source terms that depend only on the local value of both the flow variable $\mathbf{U}$ and some vector flow parameter ϕ_1 Source terms of the second type, denoted by $\mathbf{S}_2$, depend on the local value of the flow variable and on the gradient in a flow parameter ϕ_2 The source term $\mathbf{S}$ is the sum of the two source terms:

$$\begin{aligned} &\mathbf{S} = \mathbf{S}_1 + \mathbf{S}_2, \\ &\mathbf{S}_1 = \mathbf{S}_1(\mathbf{U},\phi_1), \\ &\mathbf{S}_2 = \mathbf{S}_2(\mathbf{U},\nabla\phi_2) = \mathbf{s}(\mathbf{U})\nabla\phi_2. \end{aligned} \tag{4.1}$$

These two types of source terms are treated using two different approaches.

Source terms of the first kind are discretized using time splitting. The equations to be solved reduce to the following vector ordinary differential equations:

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} = \mathbf{S}_1(\mathbf{U},t), \tag{4.2}$$

that can be discretized using any standard finite difference technique. For the shallow water equations used as an example in Section 5, the following explicit linearization is proposed [14]

$$\frac{\partial \mathbf{U}}{\partial t} = \frac{\mathbf{S}_1(\mathbf{U}_i^n)}{\mathbf{U}_i^n}\mathbf{U}, \tag{4.3}$$

where the ratio $\mathbf{S}_1/\mathbf{U}_i^n$ denotes the vector formed by the component-to-component ratio of $\mathbf{S}_1$ and $\mathbf{U}_i^n$. Eq. (4.3) has the following analytical solution:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n \exp\left[\frac{\mathbf{S}_1(\mathbf{U}_i^n)}{\mathbf{U}_i^n}\Delta t\right]. \tag{4.4}$$

This solution is unconditionally stable because the sign of $\mathbf{S}_1$ is opposite to that of the velocity components.

Source terms of the second type are accounted for using a technique similar to the source term upwinding approach [2,11]. It is reminded that the original source term upwinding technique consists in splitting the source term computed at the interface between adjacent computational cells into two parts, to be assigned to the left- and right-hand cells, respectively. The splitting is carried out in such a way that balancing the source term with the fluxes allow equilibrium conditions to be preserved (i.e., water initially at rest should remain at rest), which makes the solution oscillation-free and contributed to the overall solution stability. This technique cannot be used directly here because the solution of the two-dimensional Riemann problems (3.1) used in Eq. (2.6) or (2.7) are not sought for the fluxes **F** and **G**, but for the conserved variable **U**. The upwinding technique is adapted to the computation of **U** as follows. The one-dimensional problems with source term is to be solved:

$$\left.\begin{aligned}
&\tfrac{\partial \mathbf{U}}{\partial t} + \tfrac{\partial \mathbf{F}(\mathbf{U},\phi_2)}{\partial x} = \mathbf{s}(\mathbf{U},\phi_2)\tfrac{\partial \phi_2}{\partial x},\\
&\mathbf{U}(x,t^n) = \begin{cases} \mathbf{U}_\mathrm{L} & \text{for } x<0,\\ \mathbf{U}_\mathrm{R} & \text{for } x>0,\end{cases}\\
&\phi_2(x) = \begin{cases} \phi_{2,\mathrm{L}} & \text{for } x<0,\\ \phi_{2,\mathrm{R}} & \text{for } x>0.\end{cases}
\end{aligned}\right\} \tag{4.5}$$

This problem is transformed into an equivalent Riemann problem where the parameter ϕ_2 is constant, equal to $\phi_{2,\mathrm{LR}}$ The assumption of a constant ϕ_2 yields a zero source term and the problem (4.5) becomes:

$$\left.\begin{aligned}
&\tfrac{\partial \mathbf{U}}{\partial t} + \tfrac{\partial \mathbf{F}(\mathbf{U},\phi_{2,\mathrm{LR}})}{\partial x} = 0,\\
&\mathbf{U}(x,t^n) = \begin{cases} \mathbf{U}_\mathrm{L} & \text{for } x<0,\\ \mathbf{U}_\mathrm{R} & \text{for } x>0.\end{cases}
\end{aligned}\right\} \tag{4.6}$$

Following the approach of source term upwinding, the following estimate is proposed for $\phi_{2,\mathrm{LR}}$:

$$\phi_{2,\mathrm{LR}} \approx \frac{\lambda^-\phi_{2,\mathrm{L}} - \lambda^+\phi_{2,\mathrm{R}}}{\lambda^- - \lambda^+}, \tag{4.7}$$

where λ^- and λ^+ are the minimum wave speed on the left-hand side of the interface and the maximum wave speed on the right-hand side of the interface, respectively. The Riemann problem (4.6) is solved using any standard, exact or approximate Riemann solver. The difference with classical approaches is that the problem is not solved for **F** but for **U**. The value of the solution **U** at the interface is denoted by $\mathbf{U}_\mathrm{LR}$. Two different estimates of the fluxes are used for the balance in the left- and right-hand cells. The flux $\mathbf{F}_{\mathrm{LR,S}}$ (S = L,R) used for the balance in the cell on the side S of the interface between the cells L and R is estimated as

$$\mathbf{F}_{\mathrm{LR,S}} = \mathbf{F}(\mathbf{U}_\mathrm{LR},\phi_{2,\mathrm{S}}),\quad \mathrm{S} = \mathrm{L,R}. \tag{4.8}$$

The fluxes $\mathbf{F}_{i,j}$ and $\mathbf{G}_{i,j}$ in Eq. (2.3) are therefore estimated as:

$$\left.\begin{aligned}
&\mathbf{F}_{i,j} \approx \mathbf{F}(\mathbf{U}_{i,j}^{n+1/2},\phi_{2,i}),\\
&\mathbf{G}_{i,j} \approx \mathbf{G}(\mathbf{U}_{i,j}^{n+1/2},\phi_{2,i}),
\end{aligned}\right\} \tag{4.9}$$

where $\phi_{2,i}$ is the value of the parameter ϕ_2 in the cell i and $\mathbf{U}_{i,j}^{n+1/2}$ is the solution of the Riemann problem at the interface (i,j), with the constant parameter $\phi_{2,i,j}$ estimated as in Eq. (4.7).

5. Application example: the shallow water equations

5.1. Governing equations

The shallow water equations are written in the form (2.1) with the following definitions of **U**, **F**, **G** and **S**

$$\mathbf{U}=\begin{bmatrix} h \\ uh \\ vh \end{bmatrix},\quad \mathbf{F}=\begin{bmatrix} uh \\ u^2h+gh^2/2 \\ uvh \end{bmatrix},\quad \mathbf{G}=\begin{bmatrix} vh \\ uvh \\ v^2h+gh^2/2 \end{bmatrix},\quad \mathbf{S}=\begin{bmatrix} 0 \\ (S_{f,x}-S_{0,x})gh \\ (S_{f,y}-S_{0,y})gh \end{bmatrix}, \tag{5.1}$$

where g is the gravitational acceleration, h is the water depth, u and v are the flow velocities in the x- and y-directions, respectively, $S_{0,x}$ and $S_{0,y}$ are the bottom slopes in the x- and y-directions, respectively, and $S_{f,x}$ and $S_{f,y}$ are the friction slopes in the x- and y-directions, respectively. $\mathbf{U}^{(p)}$, $\mathbf{F}^{(p)}$ and $\mathbf{G}^{(p)}$ are given by

$$\mathbf{U}^{(p)}=\begin{bmatrix} h \\ u^{(p)}h \\ v^{(p)}h \end{bmatrix},\quad \mathbf{F}^{(p)}=\begin{bmatrix} u^{(p)}h \\ u^{(p)^2}h+gh^2/2 \\ u^{(p)}v^{(p)}h \end{bmatrix},\quad \mathbf{G}^{(p)}=\begin{bmatrix} v^{(p)}h \\ u^{(p)}v^{(p)}h \\ v^{(p)^2}h+gh^2/2 \end{bmatrix}, \tag{5.2}$$

where $u^{(p)}$ and $v^{(p)}$ are the components of the velocity in the $\eta^{(p)}$- and $\xi^{(p)}$-directions, respectively:

$$\left.\begin{aligned} u^{(p)} &= u\cos\tfrac{\theta^{(p)}+\theta^{(p-1)}}{2}+v\sin\tfrac{\theta^{(p)}+\theta^{(p-1)}}{2}, \\ v^{(p)} &= -u\sin\tfrac{\theta^{(p)}+\theta^{(p-1)}}{2}+v\cos\tfrac{\theta^{(p)}+\theta^{(p-1)}}{2}. \end{aligned}\right\} \tag{5.3}$$

The source term is expressed as the sum of source terms of the first and second types as in Eq. (4.1). The source term of the first type corresponds to the friction term

$$\mathbf{S}_1=\begin{bmatrix} 0 \\ ghS_{f,x} \\ ghS_{f,y} \end{bmatrix}=\begin{bmatrix} 0 \\ -\frac{g}{K^2}\frac{(u^2+v^2)^{1/2}}{h^{1/3}}u \\ -\frac{g}{K^2}\frac{(u^2+v^2)^{1/2}}{h^{1/3}}v \end{bmatrix}, \tag{5.4}$$

where K is Strickler's friction coefficient. In the notation (4.1) the parameter ϕ_1 is the friction coefficient K. The source term of the second type is the source term due to bottom slope

$$\mathbf{S}_2=\begin{bmatrix} 0 \\ -ghS_{0,x} \\ -ghS_{0,y} \end{bmatrix}=\begin{bmatrix} 0 \\ -gh\frac{\partial z_b}{\partial x} \\ -gh\frac{\partial z_b}{\partial y} \end{bmatrix}, \tag{5.5}$$

where z_b (the parameter ϕ_2 in the notation (4.1)) is the bottom elevation. In the computational examples presented hereafter, the Riemann solver used for the solution of the one-dimensional problems (3.9) and (3.10) is the HLLC approximate solver [16,26]. It is recalled that the HLLC solver uses the assumption of two discontinuities propagating at speeds λ^- and λ^+ for the determination of the first two components of the flux. A balance over two control volumes containing the discontinuities yields the following relationships:

$$\left.\begin{aligned} (\mathbf{U}_\mathrm{L}-\mathbf{U}_\mathrm{LR})\lambda^- &= \mathbf{F}_\mathrm{L}-\mathbf{F}_\mathrm{LR}, \\ (\mathbf{U}_\mathrm{R}-\mathbf{U}_\mathrm{LR})\lambda^+ &= \mathbf{F}_\mathrm{R}-\mathbf{F}_\mathrm{LR}, \end{aligned}\right\} \tag{5.6}$$

where $\mathbf{F}_S = \mathbf{F}(\mathbf{U}_S)$ (S = L,R). In contrast with the classical approach where the flux $\mathbf{F}_{LR}$ is calculated directly, the system (5.6) is solved for $\mathbf{U}_{LR}$

$$\mathbf{U}_{LR} = \frac{\lambda^+\mathbf{U}_R - \lambda^-\mathbf{U}_L + \mathbf{F}_L - \mathbf{F}_R}{\lambda^+ - \lambda^-}. \tag{5.7}$$

The wave speeds λ^- and λ^+ used in Eqs. (5.7) and (4.7) are estimated as proposed in [10]:

$$\left.\begin{aligned} \lambda^- &= -\max(|u|_L + c_L, |u|_R + c_R), \\ \lambda^+ &= \max(|u|_L + c_L, |u|_R + c_R), \end{aligned}\right\} \tag{5.8}$$

where $c = (gh)^{1/2}$ is the propagation speed of the pressure waves in still water. The third component of $\mathbf{U}$ is estimated as proposed in [26]:

$$v_{LR} = \begin{cases} v_L & \text{if } u_{LR} > 0, \\ v_R & \text{if } u_{LR} < 0. \end{cases} \tag{5.9}$$

5.2. Tests on Cartesian grids

The performance of the two-dimensional solver is illustrated by the classical circular dambreak test case on a Cartesian grid. The parameters of the test case are shown in Table 1. The radius of the circular dam is chosen as a non-integer multiple of the cell size in order to minimize grid artifacts that usually result in diamond-shaped domains. Fig. 5 shows how the initial condition is represented on the 1 m × 1 m grid adopted for the simulation. Given the coarseness of the grid, the shape of the dam is octagonal rather than perfectly

Table 1
Parameters of the circular dambreak test case

Symbol	Meaning	Value
g	Gravitational acceleration	9.81 m^2/s
h_{in}	Initial water depth inside the circular region	10 m
h_{out}	Initial water depth outside the circular region	1 m
r_0	Radius of the circular region	10.5 m
β	Weighting coefficient in Eqs. (2.6) and (2.7)	1/6, 1/4 and 1/3
Δx	Cell size in the x-direction	1 m
Δy	Cell size in the y-direction	1 m

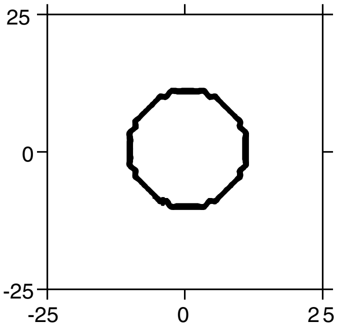


Fig. 5. Circular dambreak problem. Representation of the initial region of constant state on the 1 m × 1 m computational grid. x- and y-coordinates in metres.

circular. Therefore the numerical solution should not be expected to be perfectly circular. Such a test however is fairly representative of real-world, engineering hydrodynamic simulations, where limited computational resources and time constraints often precludes the use of refined grids.

Figs. 6–9 show the numerical results obtained at $t = 1$ s and $t = 3$ s for the original Godunov scheme with various values of the coefficient β in Eqs. (2.6) and (2.7). The solutions are compared to the semi-analytical solution of the shallow water equations in polar coordinates, to the solution given by the time-splitting algorithm and the classical finite volume approach with a one-dimensional solver. Note that the analytical solution is interpolated on the grid used in the simulations. Table 2 gives the maximum values of the water depth (obtained at the center of the circular region) at $t = 1$ s. Table 3 gives the minimum and maximum values of the water depths and the subsequent height of the rarefaction and shock wave at $t = 3$ s.

Owing to the sequential treatment of the x- and y-directions the solution profile given by the time-splitting technique is not the same along x and y (Figs. 6(b) and 8(b)). Although the rarefaction wave tends to smear along the grid diagonals compared to the main grid directions, the shock wave can be seen to be fairly isotropic (Figs. 7(a) and 9(a)).

When the classical finite volume approach is used, the main grid directions are treated in an equivalent way but a strong anisotropy can be observed between the main grid direction and the diagonals. In plan

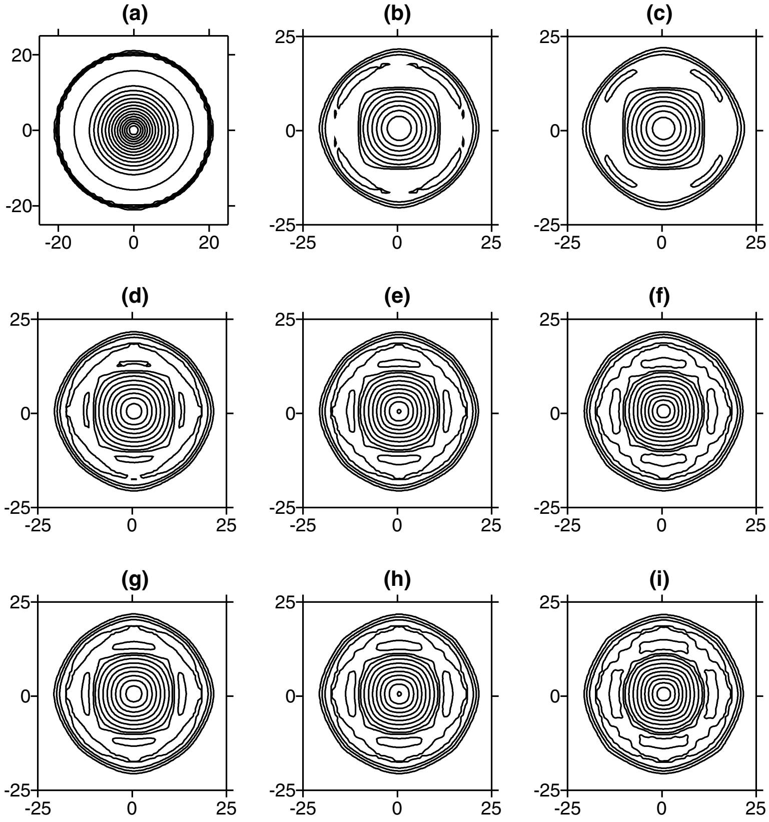


Fig. 6. Circular dambreak. Plan view of the semi-analytical (a) and numerical solutions (b–i) for the water level at $t = 1$ s. Solution obtained using the time-splitting algorithm (b), the finite volume algorithm with a one-dimensional solver (c) and the proposed solver (d–i). Fluxes estimated using Eq. (2.6) and $\beta = 1/6$ (d), $\beta = 1/4$ (e), $\beta = 1/3$ (f), and with Eq. (2.7) and $\beta = 1/6$ (g), $\beta = 1/4$ (h), $\beta = 1/3$ (i). Contour line spacing 0.5 m.

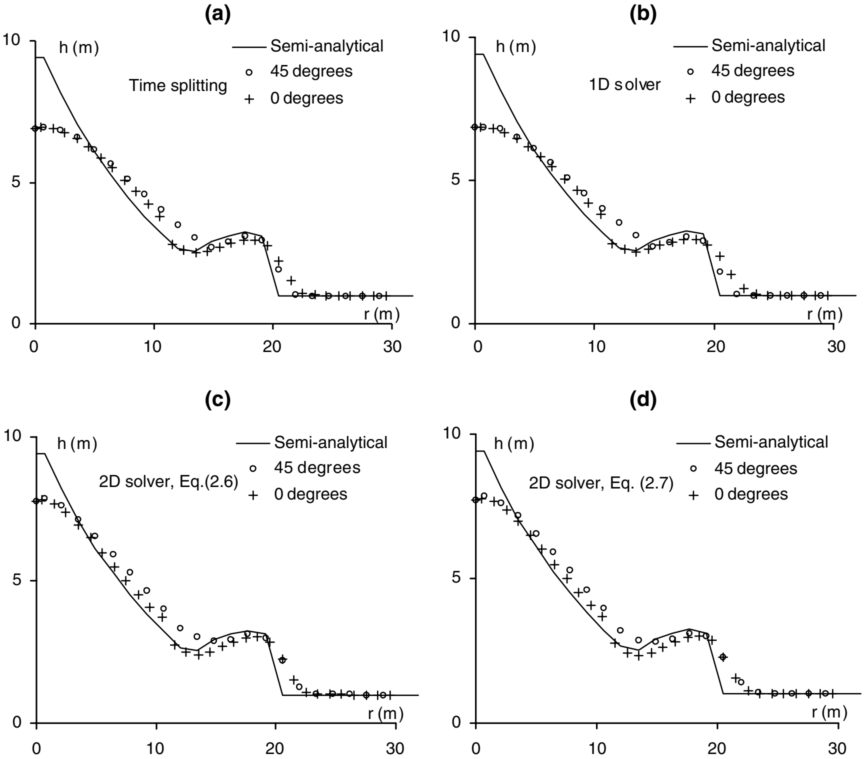


Fig. 7. Circular dambreak. Cross-sections of the semi-analytical and computed water depths along the diagonal and the main grid directions at $t = 1$ s. Solutions given by the time splitting approach (a), the finite volume approach with a one-dimensional solver (b), the finite volume approach with the two-dimensional solver using $\beta = 1/3$ in Eq. (2.6) (c) and Eq. (2.7) (d).

view (Figs. 6(c) and 8(c)), this results in a diamond-shaped shock wave and in a square-shaped rarefaction wave (central contour lines in Fig. 6(c)). Cross-sections along the main grid direction and the grid diagonal indicate that the anisotropy of the rarefaction wave is equivalent to that given by the time splitting approach and that the shock wave exhibits more anisotropy than in the time splitting solution (Figs. 7(b) and 9(b)).

The two-dimensional solver produces more isotropic solutions. The most influential factor in the quality of he solution appears to be the value of β, while the choice of the integration formula (Eq. (2.6) or (2.7)) plays a minor role. Values of β between 1/4 and 1/3 allow the circular shape of the central rarefaction wave to be restored and the octagonal symmetry of the initial condition to be preserved in the shock wave (Figs. 6(e),(f),(h) and (i)). This is confirmed by the cross-sections in the grid diagonal and the main grid directions (Figs. 7(c) and (d)). It can also be seen that the maximum value of the water depth at $t = 1$ s is much better preserved when the two-dimensional solver is used. This gain in accuracy cannot be attributed only to the increase in the computational time step because the time splitting technique that is subject to the same stability constraint gives poorer results (Fig. 7(a)). These results are confirmed by Figs. 8 and 9 that show the results obtained at $t = 3$ s.

Figs. 10 and 11 show the results obtained at $t = 1$ and 3s using the second-order MUSCL scheme in combination with a one-dimensional solver and the proposed two-dimensional solver. The MUSCL reconstruction is the original, one-dimensional reconstruction [27]. A multidimensional linear reconstruction such as that described in [8] was also experimented and did not lead to significant improvements in the quality of

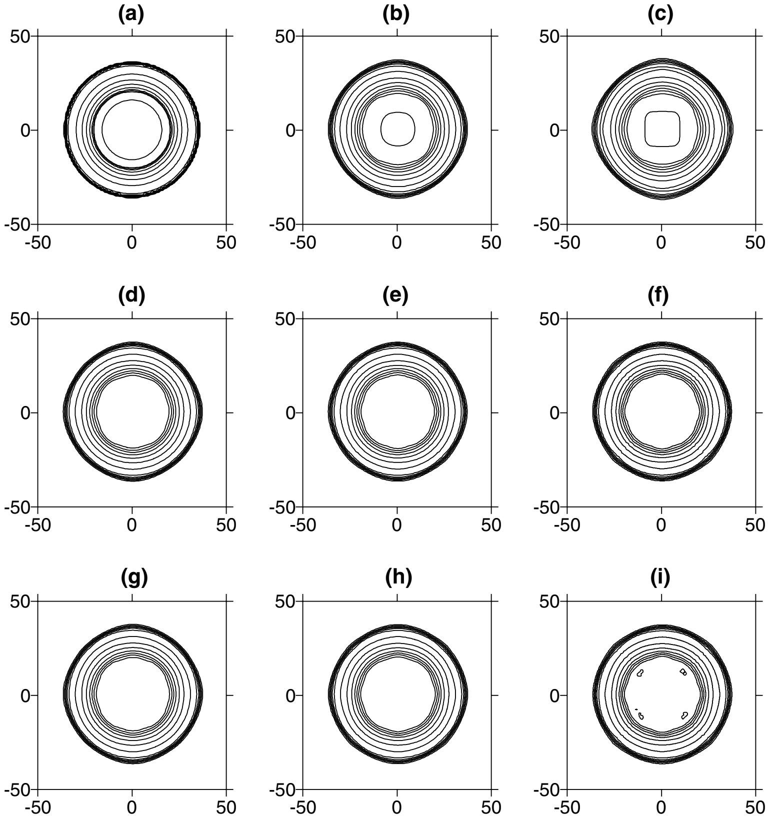


Fig. 8. Circular dambreak. Plan view of the semi-analytical (a) and numerical solutions (b–i) for the water level at $t = 3$ s. Solution obtained using the time-splitting algorithm (b), the finite volume algorithm with a one-dimensional solver (c) and the proposed solver (d–i). Fluxes estimated using Eq. (2.6) and $\beta = 1/6$ (d), $\beta = 1/4$ (e), $\beta = 1/3$ (f), and with Eq. (2.7) and $\beta = 1/6$ (g), $\beta = 1/4$ (h), $\beta = 1/3$ (i). Contour line spacing 0.2 m.

the solution. The two-dimensional solver is used with $\beta = 1/3$. The comparison with Figs. 6–11 and the numerical values in Tables 2 and 3 show that (i) using the MUSCL scheme in combination with a one-dimensional solver allows the minimum, maximum and amplitude of the waves to be better preserved than when the original Godunov scheme is used, but that (ii) the two-dimensional solver with the Godunov technique gives more isotropic solutions for comparable wave amplitudes, while (iii) optimal accuracy is achieved when the MUSCL reconstruction is used with the two-dimensional solver.

5.3. Circular dambreak on anisotropic triangular grid

The second test case consists of the same circular dambreak simulation carried out on an anisotropic triangular grid. The grid is generated from the Cartesian grid used in the previous test by cutting the 1 m × 1 m cells along the diagonal (Fig. 12). Fig. 13 shows how the initial condition is represented on the triangular grid. As in the previous test the initial condition not to be perfectly circular due to the coarseness of the grid. Fig. 14 is a plan view of the water depths computed using a one-dimensional and the two-dimensional Riemann solver with $\beta = 1/3$ in Eq. (2.6). Fig. 15 shows cross-sections in the x-direction and along the SW–NE diagonal. Note that the drawing of the contour lines and the representation of

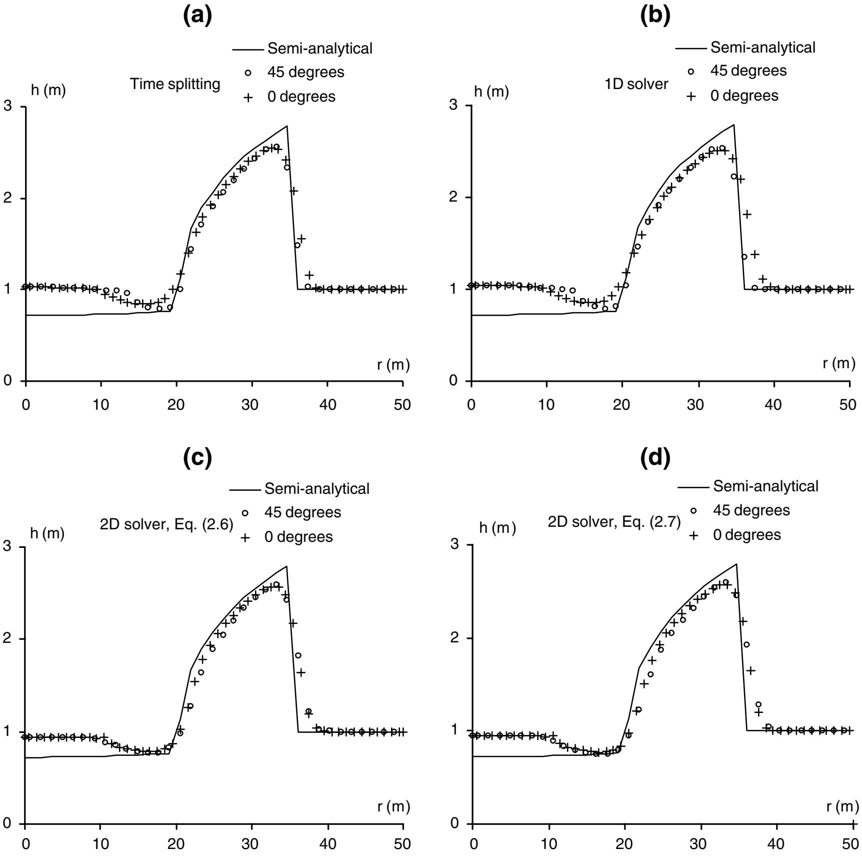


Fig. 9. Circular dambreak. Cross-sections of the semi-analytical and computed water depths along the diagonal and the main grid directions at $t = 3$ s. Solutions given by the time splitting approach (a), the finite volume approach with a one-dimensional solver (b), the finite volume approach with the two-dimensional solver using $\beta = 1/3$ in Eq. (2.6) (c) and Eq. (2.7) (d).

Table 2
Circular dambreak: properties of the solutions at $t = 1$ s

Scheme	Solver	β	Maximum h value (m)	Figures
Semi-analytical	–	–	9.83	6(a), 7
Godunov + time splitting	1D	–	6.97	6(b), 7(a)
Godunov	1D	–	6.89	6(c), 7(b)
	2D, Eq. (2.6)	1/6	7.30	6(d)
		1/4	7.55	6(e)
		1/3	7.85	6(f), 7(c)
	2D, Eq. (2.7)	1/6	7.31	6(g)
		1/4	7.54	6(h)
		1/3	7.84	6(i), 7(d)
MUSCL	1D	–	7.61	10(a), 11(a)
	2D, Eq. (2.6)	1/3	8.35	10(b), 11(b)

Table 3
Circular dambreak: properties of the solutions at $t = 3$ s

Scheme	Solver	β	Minimum h value (m)	Maximum h value (m)	Δh (m)	Figures
Semi-analytical	–	–	0.72	2.80	2.02	8(a), 9
Godunov + time splitting	1D	–	0.79	2.57	1.78	8(b), 9(a)
Godunov	1D	–	0.79	2.55	1.76	8(c), 9(b)
	2D, Eq. (2.6)	1/6	0.78	2.63	1.85	8(d)
		1/4	0.77	2.64	1.87	8(e)
		1/3	0.76	2.66	1.90	8(f), 9(c)
	2D, Eq. (2.7)	1/6	0.77	2.63	1.86	8(g)
		1/4	0.75	2.64	1.89	8(h)
		1/3	0.74	2.65	1.91	8(i), 9(d)
MUSCL	1D	–	0.80	2.69	1.89	10(c), 11(c)
	2D, Eq. (2.6)	1/3	0.75	2.69	1.94	10(d), 11(d)

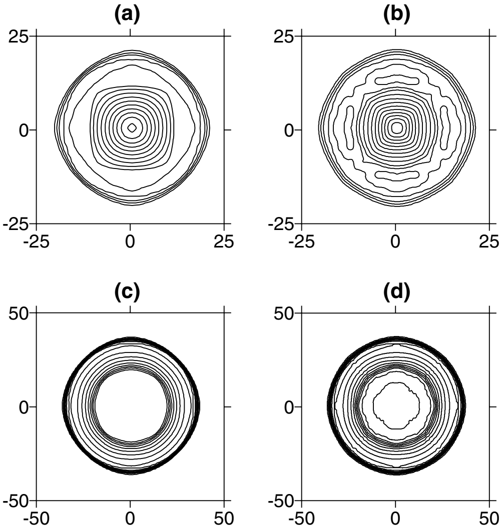


Fig. 10. Circular dambreak. Plan view of the water levels computed at $t = 1$ s (top) and $t = 3$ s (bottom) with the MUSCL scheme in combination with the one-dimensional solver (left) and the two-dimensional solver (right). The two-dimensional solver is used with $\beta = 1/3$.

the solution along the x- and diagonal directions requires that the solution be interpolated first on a square grid. Since the gravity centers of the computational cells are not aligned along x and y, the interpolation generates artifacts in the representation of the solution. For this reason the solution obtained using the one-dimensional software (Fig. 14(b)) exhibits oscillations that are not observed on the square grid (Fig. 6(c)). In spite of this artifact, it can be seen from Fig. 15 that the two-dimensional solver contributes to improve the quality of the numerical solution compared to the one-dimensional solver in that the top of the rarefaction wave is better represented at $t = 1$ s (see Table 4 for the numerical values) and that it yields a better location of the shock for both $t = 1$ s and $t = 3$ s. Indeed, in the solution obtained using the one-dimensional solver, the foot of the shock is consistently one cell behind that of the analytical solution (Figs.

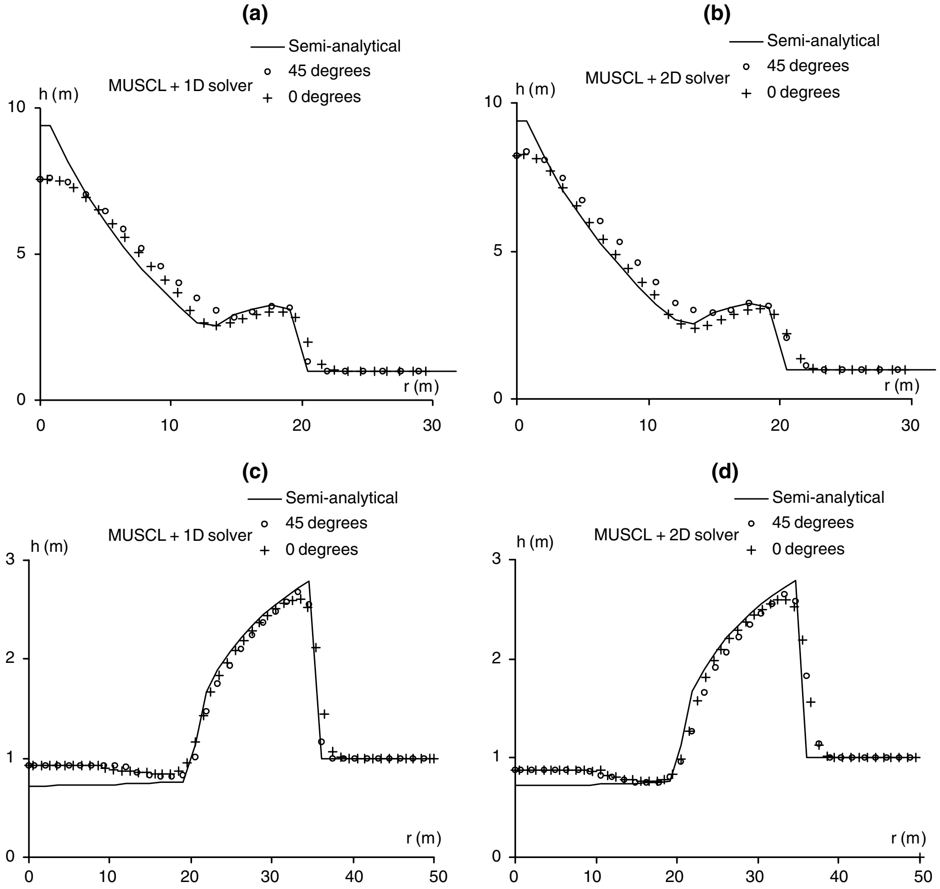


Fig. 11. Circular dambreak. Cross-sections of the water levels computed at $t = 1$ s (top) and $t = 3$ s (bottom) using the MUSCL scheme in combination with the one-dimensional solver (left) and the two-dimensional solver (right). The two-dimensional solver is used with $\beta = 1/3$.

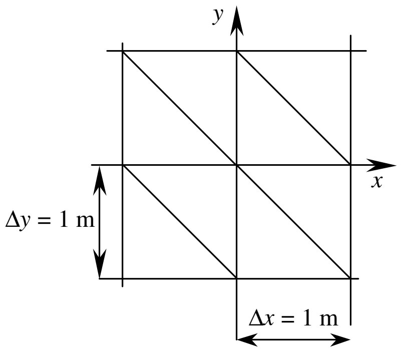


Fig. 12. Circular dambreak on anisotropic triangular grid. Definition sketch of the grid.

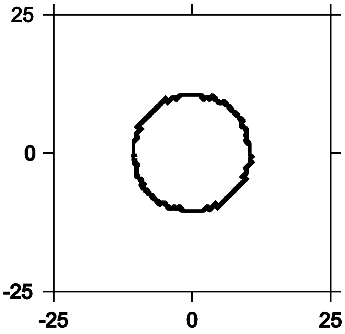


Fig. 13. Circular dambreak on anisotropic triangular grid. Representation of the initial region of constant state on the computational grid (right). x- and y-coordinates in metres.

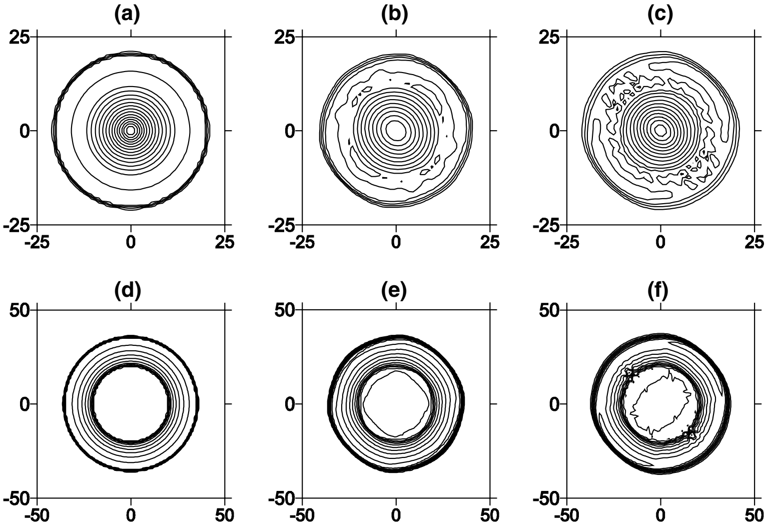


Fig. 14. Circular dambreak on anisotropic triangular grid. Plan view of the semi-analytical and numerical solutions for the water level at t = 1 s (top) and t = 3 s (bottom). Semi-analytical solution (left), numerical solution using the one-dimensional solver (middle) and two-dimensional solver with β = 1/3 in Eq. (2.6) (right). Contour line spacing 0.5 m (top) and 0.2 m (bottom).

15(a) and (c)), which is not the case for the solution given by the two-dimensional solver (Figs. 15(b) and (d)). Although the solution seems to collapse in the x-direction (cross-shaped dots in Fig. 15(d)), a manual exploration of the result file showed that this apparent profile smearing is the result of the interpolation onto the square grid used for the graphical representation (see Table 5).

5.4. Circular dambreak over a bottom step

The third test case illustrates the capability of the proposed approach to handle source terms arising from variations in the bottom level. In this test a circular wave propagates in still water over a bottom step in the x-direction. Two simulations are shown here. In the first simulation the step is positive (i.e., the wave propagates from deep water into shallow water). In the second simulation the bottom step is negative (i.e.,

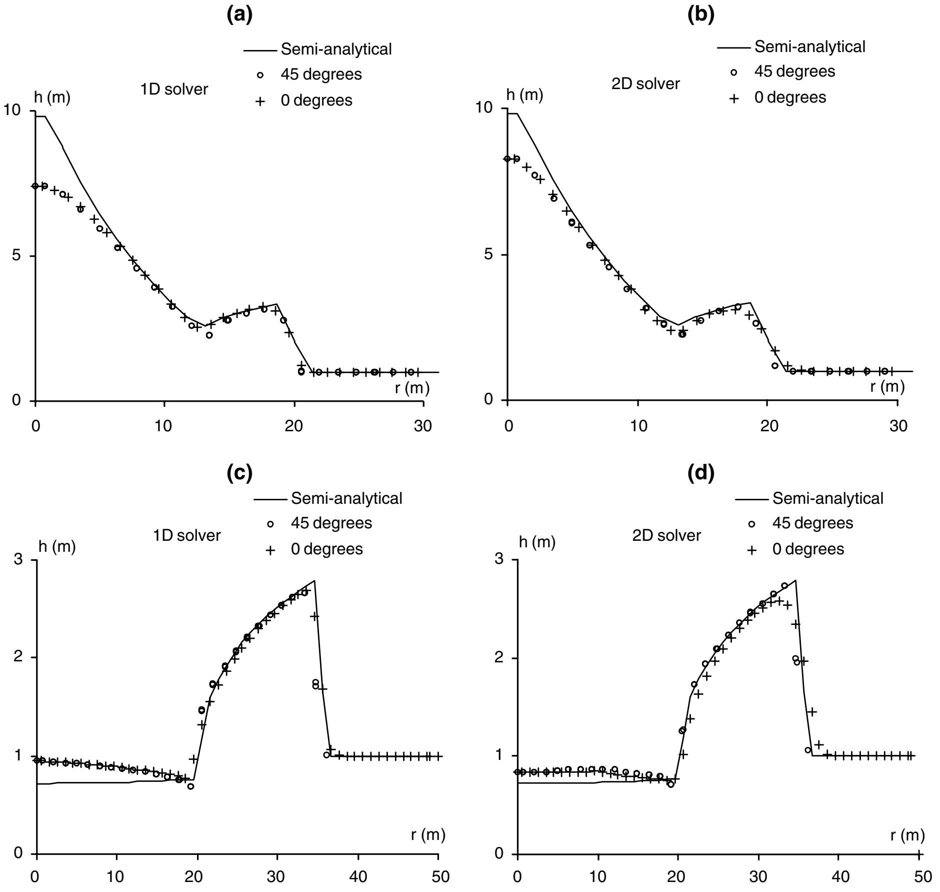


Fig. 15. Circular dambreak on anisotropic triangular grid. Cross-sections of the semi-analytical and computed water depths along the diagonal and the main grid directions at $t = 1$ s (top) and $t = 3$ s (bottom).

Table 4
Circular dambreak on anisotropic triangular grid: properties of the solutions at $t = 1$ s

Scheme	Solver	β	Minimum h value (m)	Maximum h value (m)	Δh (m)	Figures
Semi-analytical	–	–	1	9.83	8.83	14(a), 15(a), 15(b)
Godunov	1D	–	1	7.43	6.43	14(b), 15(a)
Godunov	2D, Eq. (2.6)	1/3	1	8.28	7.28	14(c), 15(b)

Table 5
Circular dambreak on anisotropic triangular grid: properties of the solutions at $t = 3$ s

Scheme	Solver	β	Minimum h value (m)	Maximum h value (m)	Δh (m)	Figures
Semi-analytical	–	–	0.72	2.80	2.08	14(c), 15(c), 15(d)
Godunov	1D	–	0.61	2.69	2.08	14(e), 15(c)
Godunov	2D, Eq. (2.6)	1/3	0.66	2.74	2.08	14(f), 15(d)

Table 6
Parameters of the circular dambreak test case

Symbol	Meaning	Value
g	Gravitational acceleration	9.81 m^2/s
z_{in}	Initial water level inside the circular region	10 m
z_{out}	Initial water level outside the circular region	1 m
r_0	Radius of the circular region	10.5 m
x_0	Abscissa of the center of the circular dam	0 m
x_s	Abscissa of the bottom step	−20 m
y_0	Ordinate of the center of the circular dam	0 m
$z_{b,\text{L}}$	Bottom level on the left-hand side of the step	0 m (simulation 1) −20 m (simulation 2)
$z_{b,\text{R}}$	Bottom level on the right-hand side of the step	−20 m (simulation 1) 0 m (simulation 2)
β	Weighting coefficient in Eqs. (2.6) and (2.7)	1/4
Δx	Cell size in the x-direction	1 m
Δy	Cell size in the y-direction	1 m

the wave propagates from shallow water into deep water). Table 6 provides the data used for the simulations. The same isotropic triangular grid made of equilateral triangles of width 1 m is used for both simulations. In order to test the capability of the method to handle perturbations arising from irregularities in the geometry the basis of the equilateral triangles is oriented along the x-axis. Therefore the discretized bottom level follows a sawtooth profile in plan view (Fig. 16). Fig. 17 shows a plan view of the computed free surface elevations at t = 3 s for Simulation 1. Fig. 18 shows a plan view of the computed free surface elevations at t = 1.5 s for Simulation 2. In each case the solution given by the two-dimensional solver with β = 1/3 is plotted with the solution given by the classical one-dimensional solver. As shown by the contours the two-dimensional solver allows the shocks to be much better resolved, with thinner and higher fronts, than does the one-dimensional solver. Moreover the contour lines are smooth and no oscillation is observed in the neighborhood of the step. Although some grid interpolation artifact can be observed in the solutions, the regular nature of the grid leads to a much better representation of the initial condition and a smoother and more isotropic behavior of the solution than when Cartesian grids or anisotropic triangular grids are used.

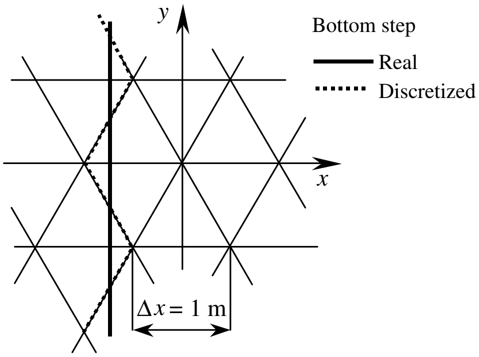


Fig. 16. Circular dambreak on an isotropic triangular grid with bottom step. Definition sketch of the computational grid and representation of the bottom step on the grid.

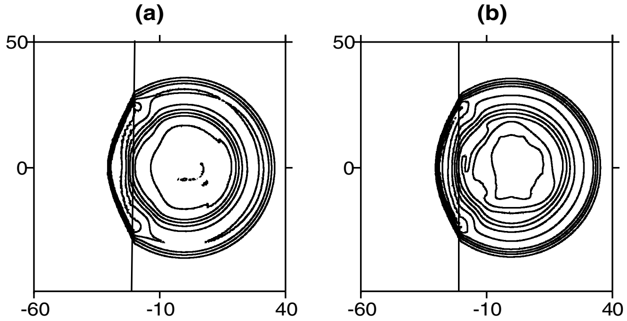


Fig. 17. Simulation 1. Free surface elevations computed using the one-dimensional (a) and two-dimensional (b) solver at $t = 1.5$ s. Contour line spacing 0.25 m.

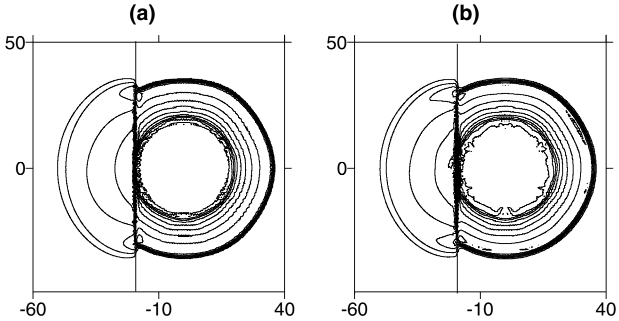


Fig. 18. Simulation 2. Free surface elevations computed using the one-dimensional (a) and two-dimensional (b) solver at $t = 3$ s. Contour line spacing 0.25 m.

6. Concluding remarks

The numerical examples provided in Section 5 show that the two-dimensional Riemann solver may contribute to improve the quality of the solutions of hyperbolic conservation laws. The solutions obtained using the two-dimensional Riemann solver are generally more oscillatory than those obtained using classical one-dimensional solvers. This oscillatory behavior is induced by the solution of the transverse Riemann problem. Numerical experiments carried out during the development of the method have shown that solving the angular Riemann problems in the η-direction only and ignoring the gradients in the ξ-direction introduces a strong numerical diffusion. Restoring the influence of the transverse gradients allows the transverse numerical diffusion to be reduced to a large extent, at the expense of the smoothness of the solution in the neighborhood of shocks and steep fronts. In the light of the computational examples and other numerical experiments not presented here, the following conclusions can be drawn about the performance of the proposed solver: (i) the solver may contribute to reduce the anisotropy in the numerical solutions; (ii) optimal isotropy is achieved for $\beta = 1/4$ or $\beta = 1/3$; (iii) Eqs. (2.6) and (2.7) yield very similar results; (iv) using the two-dimensional solver in combination with a first-order scheme may lead to more accurate solutions than a one-dimensional solver with a second-order scheme; (v) this is not entirely explained by the larger time step allowed by the two-dimensional solver because the time splitting approach, that uses the same stability constraint, produces less accurate results than the proposed solver. The improved computational quality is mainly attributed to the fact that the interpolation formulae (2.6) and (2.7) account for the transverse variation of the sought variables (and therefore of the fluxes) along the interfaces between the computational cells, which is not accounted for by classical, local one-dimensional methods.

References

[1] J.B. Bell, C.N. Dawson, G.R. Shubin, An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions, J. Comput. Phys. 74 (1988) 1–24.

[2] A. Bermudez, E. VasquezCendron, Upwind methods for hyperbolic conservation laws with source terms, Comp. Fluid 23 (1994) 1049–1071.

[3] M. Brio, A.R. Zakharian, G.M. Webb, Two-dimensional Riemann solver for Euler equations of gas dynamics, J. Comput. Phys. 167 (2001) 177–195.

[4] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, J. Comput. Phys. 87 (1990) 171–200.

[5] M. Crandall, A. Majda, The method of fractional steps for conservation laws, Numer. Math. 34 (1980) 285–314.

[6] A. Dadone, B. Grossmann, Characteristic-based, rotated upwind scheme for the Euler equations, AIAA J. 30 (1992) 2219–2226.

[7] W. Dai, P.R. Woodward, Extension of the piecewise parabolic method to multidimensional ideal magnetohydrodynamics, J. Comput. Phys. 115 (1994) 485–514.

[8] W. Dai, P.R. Woodward, A second-order unsplit Godunov scheme for two- and three-dimensional Euler equations, J. Comput. Phys. 134 (1997) 261–281.

[9] A. Daubert, O. Graffe, Quelques aspects des écoulements presque horizontaux à deux dimensions en plan et non permanents, application aux estuaires, La Houille Blanche 8 (1967) 847–860 (in French).

[10] S.F. Davis, Simplified second-order Godunov-type methods, SIAM J. Sci. Statistic. Comput. 9 (1988) 445–473.

[11] P. Garcia-Navarro, M.E. Vasquez-Cendron, On numerical treatment of the source terms in the shallow water equations, Comp. Fluid 29 (2000) 951–979.

[12] V. Guinot, Linear advection modelling: the issue of divergent flows, J. Hydroinform. 2 (2000) 113–121.

[13] V. Guinot, Riemann solvers and boundary conditions for two-dimensional shallow-water simulations, Int. J. Numer. Meth. Fluid 41 (2003) 1191–1219.

[14] V. Guinot, Godunov-type Schemes: An Introduction for Engineers, Elsevier, Amsterdam, 2003.

[15] N. Katopodes, T. Strelkoff, Two-dimensional shallow-water wave models, J. Eng. Mech. Div., ASCE 105 (1979) 317–334.

[16] A. Harten, P.D. Lax, B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1983) 35–61.

[17] R.J. LeVeque, Wave propagation algorithms for multidimensional hyperbolic systems, J. Comput. Phys. 131 (1997) 327–353.

[18] R.J. LeVeque, High resolution finite volume methods on arbitrary grids via wave propagation, J. Comput. Phys. 78 (1988) 36–63.

[19] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, Cambridge, 2002.

[20] D.W. Levy, K.G. Powell, B. Van Leer, Use of a rotated Riemann solver for the two-dimensional Euler equations, J. Comput. Phys. 106 (1993) 201–214.

[21] J. Saltzmann, An unsplit 3D upwind method for hyperbolic conservation laws, J. Comput. Phys. 115 (1994) 153–168.

[22] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.

[23] H. Sauerwein, Numerical calculations of multidimensional and unsteady flows by the method of characteristics, J. Comput. Phys. 1 (1967) 406–432.

[24] H. Sauerwein, The method of characteristics for the three-dimensional unsteady magnetofluid dynamics of a multi-component medium, J. Fluid Mech. 25 (1966) 17–41.

[25] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer, Berlin, 1999.

[26] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL-Riemann solver, Shock Waves 4 (1994) 25–34.

[27] B. Van Leer, Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection, J. Comput. Phys. 23 (1977) 276–299.