INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2003; **41**:1191–1219 (DOI: 10.1002/fld.488)

Riemann solvers and boundary conditions for two-dimensional shallow water simulations

Vincent Guinot*,†

Université Montpellier 2, Case courrier MSE, 34095 Montpellier Cedex 5, France

SUMMARY

Most existing algorithms for two-dimensional shallow water simulations treat multi-dimensional waves using wave splitting or time splitting. This often results in anisotropy of the computed flow. Both wave splitting and time splitting are based on a local decomposition of the multi-dimensional problem into one-dimensional, orthogonal problems. Therefore, these algorithms handle boundary conditions in a very similar way to classical one-dimensional algorithms. This should be expected to trigger a dependence of the number of boundary conditions on the direction of the flow at the boundaries. However, most computational codes based on alternate directions do not exhibit such sensitivity, which seems to contradict the theory of existence and uniqueness of the solution. The present paper addresses these issues. A Riemann solver is presented that aims to convert two-dimensional Riemann problems into a one-dimensional equivalent Riemann problem (ERP) at the interfaces between the computational cells. The ERP is derived by applying the theory of bicharacteristics at each end of the interface and by performing a linear averaging along the interface. The proposed approach is tested against the traditional one-dimensional approach on the classical circular dambreak problem. The results show that the proposed solver allows the isotropy of the solution to be better preserved. Use of the two-dimensional solver with a first-order scheme may give better results than use of a second-order scheme with a one-dimensional solver. The theory of bicharacteristics is also used to discuss the issue of boundary conditions. It is shown that, when the flow is subcritical, the number of boundary conditions affects the accuracy of the solution, but not its existence and uniqueness. When only one boundary condition is to be prescribed, it should not be the velocity in the direction parallel to the boundary. When two boundary conditions are to be prescribed, at least one of them should involve the component of the velocity in the direction parallel to the boundary. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: multi-dimensional conservation laws; shallow water flow; Riemann problem; Godunov-type schemes; boundary conditions

1. INTRODUCTION

Substantial effort has been devoted over the past 20 years to the development of computational techniques for multi-dimensional flow simulation, in particular in the field of finite volumes for systems of hyperbolic conservation laws. The earliest techniques for multi-dimensional flow

[*] Correspondence to: V. Guinot, Université Montpellier 2, Case courrier MSE, 34095 Montpellier Cedex 5, France.
[†] E-mail: guinot@msem.univ-montp2.fr

Received 17 April 2002
Revised 18 November 2002

computations were based either on time splitting or on the so-called finite volume approach. When used for the solution of multi-dimensional hyperbolic systems, time splitting [1] is also called 'alternate directions'. It consists of solving the multi-dimensional problem by applying one-dimensional techniques in each spatial direction successively. It has been used on structured as well as on unstructured grids. The finite volume method (see e.g. Reference [2]) consists of solving local one-dimensional problems in the direction normal to the interfaces between the computational cells. However, these two methods are well-known to be inaccurate when the flow is oblique to the grid or strongly divergent.

Multi-dimensional methods were developed at the end of the 1970s and at the beginning of the 1980s for multi-dimensional scalar transport (see e.g. References [3–5]). Attempts to generalise such methods have led to the wave splitting approach [6] and the front tracking approach in multiple dimensions [7, 8]. These methods allow the computational time step to be increased compared to the finite volume approach, but some of them are time-consuming (in particular front tracking) and are still based on the decomposition of the waves into one-dimensional wave patterns. Recently, multi-dimensional Riemann solvers have been designed for the solution of the two-dimensional Euler equations [9]. These solvers also use the superimposition of one-dimensional patterns.

As early as the 1960s, genuine two-dimensional approaches [10] had been developed for the solution of the shallow water equations. They were based on the theory of bicharacteristics (see e.g. Reference [11] for detailed consideration of the theory) in a finite difference context [10, 12]. In this type of method, the solution at a given point is determined by solving differential relationships along three selected characteristic lines passing through the point. The flow variables at the feet of the bicharacteristics are interpolated between the grid points. The problem with this approach is that each particular choice for the orientation of the bicharacteristics lines leads to a different interpolation and subsequently to a different numerical solution. Since there is no particular reason why any particular orientation should be preferred for the bicharacteristics, there is an infinity of possible numerical solutions. This problem is hardly addressed in the publications mentioned above.

Another issue is that of boundary conditions. When time splitting, finite volume and wave splitting approaches are used, the multi-dimensional problem is reduced to a set of one-dimensional problems. In most systems of conservation laws of fluid dynamics, the transverse momentum term (or velocity) is a Riemann invariant with a propagation speed equal to the longitudinal (in-line) flow velocity. Consequently, the number of conditions to be prescribed at a boundary is a function of the direction of the flow. For subcritical shallow water flow, two boundary conditions are needed if the flow is entering the domain, while only one is needed if the flow is leaving the domain. However, in many industrial two-dimensional computational codes (based on local one-dimensional treatment of the flow), the number of boundary conditions (be it 1, 2 or 3) is fixed in advance for the whole length of the simulation, regardless of the flow direction. There is, therefore, an apparent contradiction between the theory and its practical implementation into simulation software. Since the number of boundary conditions is directly related to the existence and uniqueness of solutions, one may wonder why this contradiction does not lead to systematic failure of the simulations.

The present paper aims to address these issues. Section 2 briefly presents the theory of bicharacteristics and its application to the set of equations that describe shallow water flow in two dimensions. Section 3 shows how the bicharacteristics approach can be used to solve two-dimensional Riemann problems for the shallow water equations. Section 4 compares some computational results obtained using the proposed approach to results obtained using some classical approaches. Section 5 addresses the issue of boundary conditions and shows that the apparent contradiction mentioned above is actually not a contradiction, but reflects a possible lack of accuracy in the discretization of boundary conditions. Section 6 provides concluding remarks.

2. THE BICHARACTERISTIC APPROACH

2.1. The theory of bicharacteristics

The theory of bicharacteristics for multi-dimensional hyperbolic systems has been detailed in a number of publications (see e.g. Reference [11] for a theoretical approach and [12, 10, 13] for applications to shallow water simulations) and will not be detailed here. The present section focuses on the conservation part of the shallow water equations, assuming that the source terms that account for friction and bottom slope are taken into account in a further step via time splitting.

Under the assumption of horizontal bed and frictionless motion, the shallow water equations can be written in conservation form as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{0} \tag{1a}$$

$$\mathbf{U} = \begin{bmatrix} h \\ q_x \\ q_y \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} q_x \\ q_y^2/h + gh^2/2 \\ q_x q_y r/h \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} q_y \\ q_x q_y/h \\ q_y^2/h + gh^2/2 \end{bmatrix} \tag{1b}$$

where g is gravity, h is the water depth, and q_x and q_y are the discharges per unit width in the x- and y-directions, respectively. The characteristic form of Equation (1a) is

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y} = \mathbf{0} \tag{2}$$

where $\mathbf{A} = \partial \mathbf{F}/\partial \mathbf{U}$ and $\mathbf{B} = \partial \mathbf{G}/\partial \mathbf{U}$ are the Jacobian matrices of the fluxes with respect to the flow variable. It can be shown (see Reference [13] for more details) that, from any point M in the phase space (x, y, t), two surfaces $t = \phi(x, y)$ are issued, in which the following relationship is satisfied:

$$\left| \mathbf{I} - \mathbf{A}\frac{\partial \phi}{\partial x} - \mathbf{B}\frac{\partial \phi}{\partial y} \right| = 0 \tag{3}$$

where $\mathbf{I}$ is the identity matrix. These surfaces in the phase space are the two-dimensional extension of the characteristic lines in one dimension. They represent the boundaries of the domains of influence of the solution. Differential relationships can be defined along these surfaces that allow the solution to be fully determined from the initial conditions. The first surface, called the χ-surface hereafter, is restricted to a single line, the direction of which in the phase space is given by the vector:

$$\mathbf{t}_\chi = \begin{bmatrix} u \\ v \\ \mathbf{1} \end{bmatrix} \tag{4}$$

where $u = q_x/h$ and $v = q_y/h$ are the velocities in the x- and y-directions, respectively. It corresponds to the set of differential relationships

$$\left.\begin{aligned} \mathrm{d}x/\mathrm{d}t = u \\ \mathrm{d}y/\mathrm{d}t = v \end{aligned}\right\} \tag{5}$$

Along this surface, the following relationship is valid:

$$-\sin\chi \frac{\mathrm{d}u}{\mathrm{d}t} + \cos\chi \frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{2}{t}\frac{\partial c}{\partial \chi} \tag{6}$$

where $c = (gh)^{1/2}$ is the celerity of waves in still water and χ is the angle with a direction of reference (say, x). The time t commences from the time at which the initial conditions are known. The second surface, called the θ-surface, is a cone formed by circles expanding uniformly from the central χ-surface at a speed c. It is given by the set of vectors:

$$\mathbf{t}_\theta = \begin{bmatrix} u + c\cos\theta \\ v + c\sin\theta \\ 1 \end{bmatrix}, \quad \theta \in [0, 2\pi] \tag{7}$$

It corresponds to the set of differential relationships:

$$\left.\begin{aligned} \mathrm{d}x/\mathrm{d}t &= u + c\cos\theta \\ \mathrm{d}y/\mathrm{d}t &= v + c\sin\theta \end{aligned}\right\} \tag{8}$$

Along the line defined by the tangent vector $\mathbf{t}_\theta$, the following relationship is valid:

$$\cos\theta \frac{\mathrm{d}u}{\mathrm{d}t} + \sin\theta \frac{\mathrm{d}v}{\mathrm{d}t} + 2\frac{\mathrm{d}c}{\mathrm{d}t} = \frac{\sin\theta}{t}\frac{\partial u}{\partial\theta} - \frac{\cos\theta}{t}\frac{\partial v}{\partial\theta} \tag{9}$$

Figure 1 shows the χ- and θ-surfaces in the phase space (x, y, t). The point from which the surfaces issue is taken as the origin of the co-ordinate system. The χ-surface is defined by the following parametric equations:

$$\begin{aligned} x_\chi &= ut \\ y_\chi &= vt \\ t_\chi &= t \end{aligned} \tag{10}$$

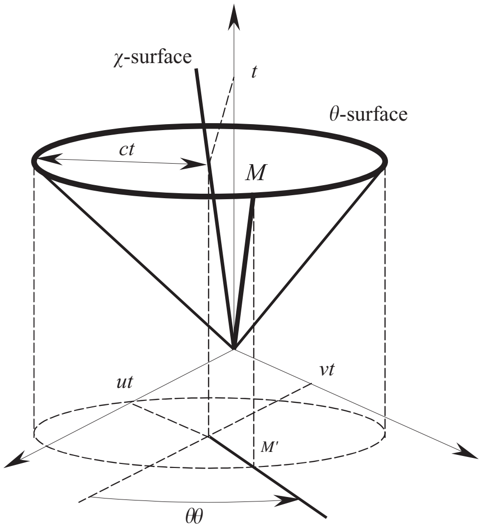

Figure 1. The χ- and θ-surfaces in the phase space.

while the θ-surface is defined by the following two-parameter family:

$$\left.\begin{aligned} x_\theta &= ut + ct\cos\theta \\ y_\theta &= vt + ct\sin\theta \\ t_\theta &= t \end{aligned}\right\} \tag{11}$$

Note that, for a one-dimensional problem, the partial derivatives with respect to χ and θ vanish in Equations (6) and (9). For a one-dimensional problem along $x, \theta = 0$ and Equations (6) and (9) simplify into the well-known one-dimensional characteristics formulation

$$\left.\begin{aligned} \frac{\mathrm{d}v}{\mathrm{d}t} &= 0 \text{ along } \frac{\mathrm{d}x}{\mathrm{d}t} = u \\ \frac{\mathrm{d}}{\mathrm{d}t}(u - 2c) &= 0 \text{ along } \frac{\mathrm{d}x}{\mathrm{d}t} = u - c \\ \frac{\mathrm{d}}{\mathrm{d}t}(u + 2c) &= 0 \text{ along } \frac{\mathrm{d}x}{\mathrm{d}t} = u + c \end{aligned}\right\} \tag{12}$$

2.2. Problems commonly associated with the bicharacteristic approach

Assume that the initial state of the variable $\mathbf{U}$ is known for all points of space at time $t = 0$. The purpose is to determine $\mathbf{U}$ at point $M(0, 0, t_M)$. To do so, relationships (6) and (9) are used along the χ- and θ-surfaces between times $t = 0$ and $t = t_M$. It can be proved [13] that the system of differential relationships is well-posed in the two following cases: (i) one differential relationship (6) written along the χ-surface and two differential relationships (9) along the θ-surface for two angles θ and $\theta + \pi$; or (ii) three relationships (9) written for three angles

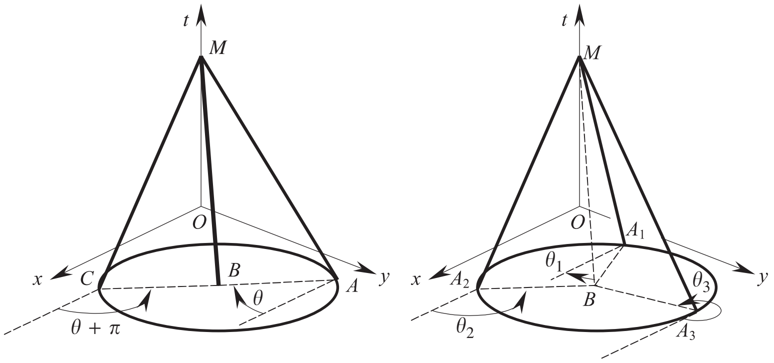

Figure 2. Two options for problem well-posedness. Left: one differential relationship along the χ-surface, two relationships on a diameter of the θ-surface. Right: three relationships along the θ-surface.

θ_1, θ_2 and θ_3. In each case, a 3×3 system of algebraic equations can be written along the lines $[A_1M], [A_2M]$ and $[A_3M]$. Figure 2 illustrates the two possible options.

Assuming that the values of h, q and r at known at the points A_1, A_2 and A_3, the 3×3 system of algebraic equations above can be solved and $\mathbf{U}$ is fully determined at the point M. This method was applied in the 1960s [10] and 1970s [12] to finite differences. In these applications, $\mathbf{U}_{A_1}, \mathbf{U}_{A_2}$ and $\mathbf{U}_{A_3}$ were interpolated from the discrete flow variables at the neighbouring computational points. Since the purpose is to apply the method in the framework of finite volumes, $\mathbf{U}_{A_1}, \mathbf{U}_{A_2}$ and $\mathbf{U}_{A_3}$ should now be determined from the reconstruction of the flow variables within the cells. The difficulty in the application of this method arises from the arbitrary character of the choice of the points A_1, A_2 and A_3.

Consider two situations where, owing to different combinations of the variables in the neighbouring cells, the reconstructed flow variables within a cell are identical everywhere, except from a region of limited size that will be denoted by Ω. Assume that part of Ω belongs to the domain of dependence of M (i.e. the circle in Figure 2). Since the two reconstructions differ over part of the domain of influence of M, the final solutions $\mathbf{U}_M$ should also differ. However, if none of the points A_1, A_2, A_3 belongs to Ω, the difference between the two reconstructions will not appear from the differential relationships and the numerical solution will be exactly the same in both cases. The present approach aims to propose a solution to this problem.

3. SOLUTION OF TWO-DIMENSIONAL RIEMANN PROBLEMS

3.1. Principle of the approach

It is aimed to derive an approach that allows for the solution of the system of hyperbolic PDEs (1) using a classical finite-volume approach over Cartesian grids. The evolution of $\mathbf{U}$ from one time step to the next is given by

$$\mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^{n} + (\mathbf{F}_{i-1/2,j} - \mathbf{F}_{i+1/2,j})\frac{\Delta t}{\Delta x} + (\mathbf{G}_{i,j-1/2} - \mathbf{G}_{i,j+1/2})\frac{\Delta t}{\Delta y} \tag{13}$$

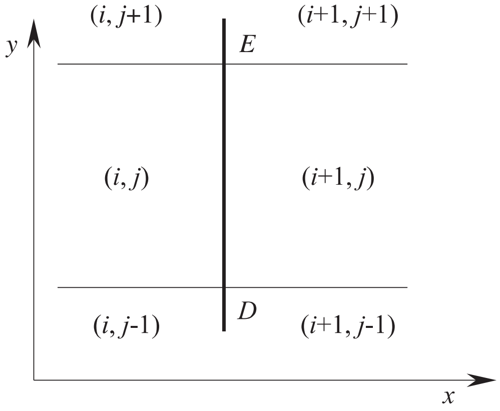

Figure 3. Determination of the one-dimensional ERP in the x-direction from two two-dimensional Riemann problems at the cell corners D and E.

where $\mathbf{U}^n_{i,j}$ is the average value of $\mathbf{U}$ over the cell (i,j), $\mathbf{F}_{i-1/2,j}$ is the flux at the interface between the cells $(i-1,j)$ and (i,j), and $\mathbf{G}_{i,j-1/2}$ is the flux at the interface between the cells $(i,j-1)$ and (i,j). Δt is the computational time step, and Δx and Δy are the cell sizes in the x- and y-directions, respectively. The fluxes $\mathbf{F}$ and $\mathbf{G}$ must be computed from the values $\mathbf{U}^n$ of $\mathbf{U}$ at the time level n in the cells that surround (i,j).

The proposed solution approach was derived on the basis of the following remarks:

(a) As mentioned in Section 2.1, any method where the differential relationships are solved along arbitrarily selected bicharacteristics is bound to be inaccurate because it may fail to capture fully the information available over the domain of dependence of the solution. This problem is solved in the proposed approach by integrating three equations of the type of Equation (9) with respect to θ over three intervals that cover the entire domain of dependence.

(b) Since many (exact or approximate) solvers are available for the one-dimensional Riemann problem, it is convenient to convert the multi-dimensional Riemann problem into an equivalent, one-dimensional one that can be solved using standard techniques. This is achieved by defining the domain of integration as a function of the normal of the interface for which the equivalent one-dimensional Riemann problem must be defined.

(c) Consider the interface $[DE]$ between two computational cells (i,j) and $(i+1,j)$ (Figure 3). A two-dimensional Riemann problem can be defined at the point D using all the cells of which D is a corner. Similarly, a two-dimensional Riemann problem can be defined at E from the states in all the cells of which E is a corner. In general, these two 2-dimensional Riemann problems will lead to two different equivalent one-dimensional Riemann problems. The contribution of these two Riemann problems should be taken into account in the flux computation across $[DE]$. The proposed approach uses a piecewise linear interpolation between the Riemann problems at the corners of the interface and the one-dimensional problem that can be defined at the centre of the interface.

The three steps are detailed below. For the sake of simplicity, they are detailed for Cartesian grids, but the method can be extended to unstructured grids. The principle of the solver is presented for subcritical flow in Sections 3.2–3.5 and extended to supercritical situations in Section 3.6.

3.2. Derivation of the integral relationships

The present section shows how to integrate relationships (9) over the whole domain of dependence of the solution so as to provide algebraic equations that can be solved for $\mathbf{U}_M$. The first step consists of integrating Equation (9) over three domains $[\theta_1,\theta_2]$, $[\theta_2,\theta_3]$ and $[\theta_3,\theta_4]$ that cover the entire domain of dependence as shown in Figure 2, so that the influence of the initial state is taken into account over the whole domain of dependence. This is the case if $\theta_4=\theta_1+2\pi$. The choice of the triplet $(\theta_1,\theta_2,\theta_3)$ is discussed in the next subsection. Note that it could also have been envisaged to integrate two relationships (9) over $[\theta_1,\theta_2]$ and $[\theta_2,\theta_1+2\pi]$ and one relationship (6) over $[\theta_1,\theta_1+2\pi]$, but the latter leads to a trivial equality of the type $0=0$ and cannot be used.

The initial variables are assumed to be constant over each of the intervals $[\theta_1,\theta_2],[\theta_2,\theta_3]$ and $[\theta_3,\theta_1+2\pi]$:

$$\mathbf{U}=\mathbf{U}^{(k)}=\begin{bmatrix} h^{(k)} \\ q_x^{(k)} \\ q_y^{(k)} \end{bmatrix} \quad \text{for } \theta\in[\theta_k,\theta_{k+1}] \tag{14}$$

where $h^{(k)},q_x^{(k)},q_y^{(k)}$ are obtained from variable reconstructions over the computational cells of which M is a corner. In the present notation, $M=D$ or $M=E$. The following notation is used: the variable $\mathbf{U}$ is sought for the time $t=\Delta t$. Its distribution over space in form (14) is assumed to be known at time $t=0$. The point at which $\mathbf{U}$ is sought is taken as the origin of the space co-ordinates. Its location $(0,0,\Delta t)$ in the phase space is denoted by M. The χ-surface passing at M intersects the $(x$–$y)$ plane at the point F in the phase space. On the θ-surface, a characteristic is characterized by the angle θ that its projection $[AM]$ on the $(x$–$y)$ plane forms with the direction x. A is the intersection of the characteristic with the $(x$–$y)$ plane. The (known) flow variables at the point A are denoted with the subscript A and the (sought) values at M are without subscript. The notation adopted is illustrated by Figure 4. Integrating (9) over $[\theta_1,\theta_2]$ gives

$$(\sin\theta_{k+1}-\sin\theta_k)\frac{\mathrm{d}u}{\mathrm{d}t}+(\cos\theta_k-\cos\theta_{k+1})\frac{\mathrm{d}v}{\mathrm{d}t}+2(\theta_{k+1}-\theta_k)\frac{\mathrm{d}c}{\mathrm{d}t}$$
$$=\int_{\theta_k}^{\theta_{k+1}}\left(\frac{\sin\theta}{t}\frac{\partial u}{\partial\theta}-\frac{\cos\theta}{t}\frac{\partial v}{\partial\theta}\right)\mathrm{d}\theta \tag{15}$$

The differentials $\mathrm{d}/\mathrm{d}t$ are eliminated by integrating Equation (15) between $t=0$ and Δt, that is, between A and M on the θ-surface. The integration yields the following equality:

$$(\sin\theta_{k+1}-\sin\theta_k)(u-u_A)+(\cos\theta_k-\cos\theta_{k+1})(v-v_A)+2(\theta_{k+1}-\theta_k)(c-c_A)$$
$$=\int_0^{\Delta t}\int_{\theta_k}^{\theta_{k+1}}\left(\frac{\sin\theta}{t}\frac{\partial u}{\partial\theta}-\frac{\cos\theta}{t}\frac{\partial v}{\partial\theta}\right)\mathrm{d}\theta\,\mathrm{d}t \tag{16}$$

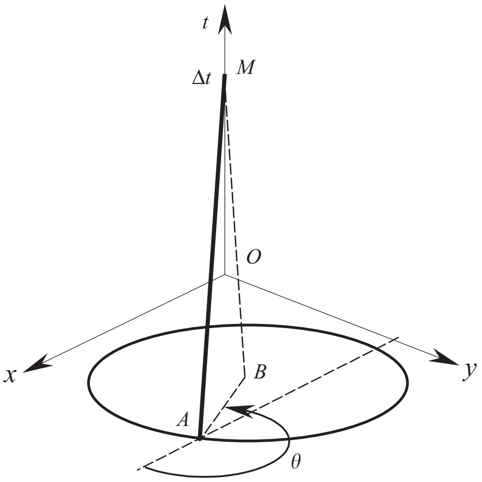

Figure 4. Definition sketch for the integration of the differential relationships along the bicharacteristics. The circle in the $(x–y)$ plane indicates the domain of dependence of the solution at M.

The right-hand side of Equation (16) is estimated as follows. The partial derivatives $\partial u/\partial\theta$ and $\partial v/\partial\theta$ are expressed as functions of the partial derivatives along x and y:

$$\left.\begin{aligned}\frac{\partial u}{\partial\theta}&=\frac{\partial u}{\partial x}\frac{\partial x}{\partial\theta}+\frac{\partial u}{\partial y}\frac{\partial y}{\partial\theta}=0\\ \frac{\partial v}{\partial\theta}&=\frac{\partial v}{\partial x}\frac{\partial x}{\partial\theta}+\frac{\partial v}{\partial y}\frac{\partial y}{\partial\theta}=0\end{aligned}\right\}\tag{17}$$

Substituting Equation (17) into Equation (16) leads to an expression of the form

$$\alpha_k u+\beta_k v+\gamma_k c=\delta_k\tag{18}$$

where the coefficients are given by

$$\left.\begin{aligned}\alpha_k&=\sin\theta_{k+1}-\sin\theta_k\\ \beta_k&=\cos\theta_k-\cos\theta_{k+1}\\ \gamma_k&=2(\theta_{k+1}-\theta_k)\\ \delta_k&=\alpha_k u_A+\beta_k v_A+\gamma_k c_A\end{aligned}\right\}\tag{19}$$

Writing three relationships (18) for $k=1,2,3$ and solving the resulting system for (u,v,c) allows $\mathbf{U}$ to be determined in a unique way at $t=\Delta t$.

3.3. Conversion to a one-dimensional Riemann problem

This subsection explains how relationships (18) can be used to convert the two-dimensional Riemann problem to a one-dimensional ERP in a Cartesian grid. The principle is explained

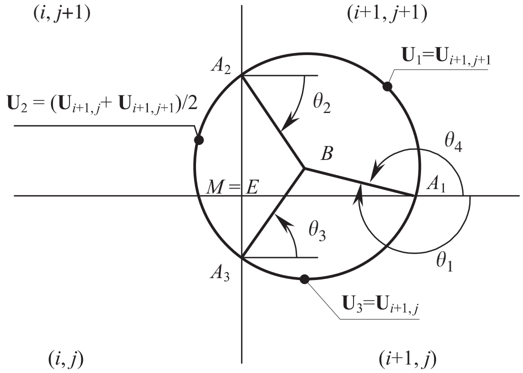

Figure 5. Determination of the right state of the ERP. Convention for the indexing of the angles θ_k.

for the cell corner E of Figure 3. This corner belongs to the cells (i,j), $(i+1,j)$, $(i+1, j+1)$, $(i,j+1)$. The method can be applied to determine ERPs in the x- and y-directions at all corners of the cells.

The angles θ_k and the values $\mathbf{U}^{(k)}$ are determined using a generalization of the method that is classically used to determine the ERP from a generalized Riemann problem in one dimension. The details of the classical method can be found in a number of references such as [14, 15]. The two-dimensional Riemann problem is denoted by the three states $(\mathbf{U}^{(1)},\mathbf{U}^{(2)},\mathbf{U}^{(3)})$ and the equivalent problem in x is denoted by $(\mathbf{V}_\mathrm{L},\mathbf{V}_\mathrm{R})$. Figure 5 shows the convention used for the numbering of the angles for the right state of the ERP. Simple trigonometric considerations lead to the following formulae:

$$\left.\begin{aligned}\theta_1 &= \arcsin(v/c)-\pi\\ \theta_2 &= -\arccos(-u/c)\\ \theta_3 &= \arccos(-u/c)\\ \theta_4 &= \theta_1+2\pi\end{aligned}\right\} \tag{20}$$

Assuming first that the flow is continuous, the solution $\mathbf{V}_M$ of the equivalent Riemann problem at the point M is given by

$$\mathbf{V}_M=\sum_{p=1}^{3} a^{(p)}\,W_M^{(p)} \tag{21}$$

where the quantities $W^{(p)}$ are the Riemann invariants along x and the coefficients $a^{(p)}$ are assumed to be constant through appropriate linearization. The Riemann invariants satisfy the following relationships:

$$\mathrm{d}W^{(p)}=0 \quad \text{along} \quad \frac{\mathrm{d}x}{\mathrm{d}t}=\lambda^{(p)} \tag{22}$$

where $\lambda^{(p)}$ is the pth eigenvalue of the matrix $\mathbf{A}$ in Equation (2). The summation in Equation (21) is separated according to the signs of the eigenvalues

$$\mathbf{V}_M = \sum_{\lambda^{(p)} \leqslant 0} a^{(p)} W_M^{(p)} + \sum_{\lambda^{(p)} > 0} a^{(p)} W_M^{(p)} \tag{23}$$

The states $\mathbf{V}_\mathrm{L}$ and $\mathbf{V}_\mathrm{R}$ can be expressed in a similar way

$$\left.\begin{aligned} \mathbf{V}_\mathrm{L} &= \sum_{\lambda^{(p)} \leqslant 0} a^{(p)} W_\mathrm{L}^{(p)} + \sum_{\lambda^{(p)} > 0} a^{(p)} W_\mathrm{L}^{(p)} \\ \mathbf{V}_\mathrm{R} &= \sum_{\lambda^{(p)} \leqslant 0} a^{(p)} W_\mathrm{R}^{(p)} + \sum_{\lambda^{(p)} > 0} a^{(p)} W_\mathrm{R}^{(p)} \end{aligned}\right\} \tag{24}$$

From Equation (22), the following equalities can be stated:

$$W_M^{(p)} = \begin{cases} W_\mathrm{R}^{(p)} & \text{if } \lambda^{(p)} \leqslant 0 \\ W_\mathrm{L}^{(p)} & \text{if } \lambda^{(P)} > 0 \end{cases} \tag{25}$$

Similarly, Equation (15) can be rewritten in the form of Equation (22):

$$dJ^{(k)} = 0 \quad \text{along} \begin{cases} \mathrm{d}x/\mathrm{d}t = u^{(k)} + c^{(k)} \cos\theta \\ \mathrm{d}y/\mathrm{d}t = v^{(k)} + c^{(k)} \sin\theta \\ \theta \in [\theta_k, \theta_{k+1}] \end{cases} \tag{26}$$

where $J^{(k)} = \alpha_k u^{(k)} + \beta_k v^{(k)} + \gamma_k c^{(k)}$ is the two-dimensional generalization of the classical one-dimensional Riemann invariant, integrated over the part of the θ-surface defined by Equation (26). The solution $\mathbf{U}_M$ at the point M is given by

$$\mathbf{U}_M = \sum_{p=1}^{3} b^{(p)} J_M^{(p)} \tag{27}$$

where the coefficients $b^{(p)}$ are also assumed to be constant through an appropriate linearization. The summation in Equation (26) is carried out according to the location of the foot of the bicharacteristics. The first region, defined by the arc $[A_1A_2]$ is located in the cell $(i+1, j+1)$ as shown in Figure 5. The second region, defined by the arc $[A_2A_3]$, covers the cells $(i, j+1)$ and (i, j). The third region, defined by $[A_3A_1]$, covers the cell $(i+1, j)$. Since it is required that the one-dimensional ERP $(\mathbf{V}_\mathrm{L}, \mathbf{V}_\mathrm{R})$ should be equivalent to the two-dimensional problem $(\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)})$, $\mathbf{V}_M$ must be equal to $\mathbf{U}_M$ and therefore

$$\sum_{\lambda^{(p)} \leqslant 0} a^{(p)} W_\mathrm{R}^{(p)} + \sum_{\lambda^{(p)} > 0} a^{(p)} W_\mathrm{L}^{(p)} = \sum_{p=1}^{3} b^{(p)} J_M^{(p)} \tag{28}$$

Since the purpose is to determine the right state of the Riemann problem, only the characteristics coming from the right-hand side of the interface should be considered. Therefore, a sufficient condition for Equation (28) to be satisfied is

$$\sum_{\lambda^{(p)} \leqslant 0} a^{(p)} W_\mathrm{R}^{(p)} = b^{(1)} J_M^{(1)} + b^{(3)} J_M^{(3)} \tag{29}$$

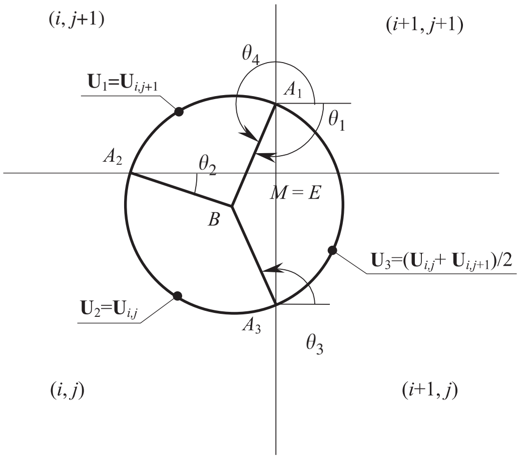

Figure 6. Angles and variables for the determination of the left state of the ERP.

This means that the invariant $J^{(2)}$ for the right-hand side of the interface can be taken equal to any arbitrary value, since it does not contribute to the final solution. The simplest possible choice is to use the average of **U** over the right-hand side of the interface, that is

$$J^{(2)} = J^{(2)}\left(\frac{\mathbf{U}_{i+1,j} + \mathbf{U}_{i+1,j+1}}{2}\right) \tag{30}$$

For similar reasons, the missing invariants $W^{(p)}$ for the equivalent Riemann problem can be taken equal to any arbitrary value. There again, the simplest possible choice consists of taking them equal to $W^{(p)}(\mathbf{U}_{i+1,j})$. Equation (29) eventually becomes

$$\mathbf{V}_{\mathrm{R}} = b^{(1)}J^{(1)}(\mathbf{U}_{i+1,j+1}) + b^{(2)}J^{(2)}\left(\frac{\mathbf{U}_{i+1,j} + \mathbf{U}_{i+1,j+1}}{2}\right) + b^{(3)}J^{(3)}(\mathbf{U}_{i+1,j}) \tag{31}$$

From a practical point of view, $\mathbf{V}_{\mathrm{R}}$ is obtained by solving a system of three equations (18), by choosing the angles as indicated by the sketch in Figure 5, using the flow variables of the cell $(i+1, j+1)$ to compute the coefficients of the second equation. A similar reasoning for the left state leads to choosing the angles as indicated by the sketch in Figure 6. The values of the variable for the first equation (17) must be taken from the cell $(i, j+1)$ and from the cell (i, j) for the second equation. For the third equation, the average of **U** is used again

$$J^{(3)} = J^{(3)}\left(\frac{\mathbf{U}_{i,j} + \mathbf{U}_{i,j+1}}{2}\right) \tag{32}$$

3.4. Interpolation along the cell edge

Another two-dimensional Riemann problem can be defined at the lower corner of the interface $(i+1/2, j)$. It leads to another ERP in the x-direction for this lower corner. In general, this ERP will be different from the ERP of the upper corner. Moreover, there exists a region about the centre of the interface where, owing to the limitation on the Courant number, the domain of dependence of the solution is mainly located in the cells (i, j) and $(i+1, j)$. In this region, the

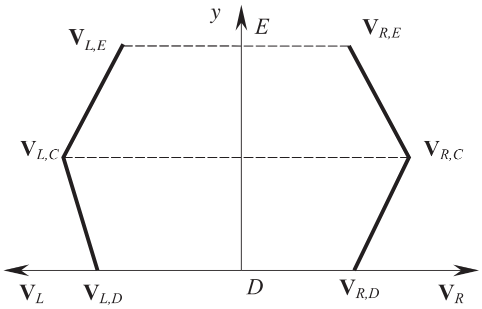

Figure 7. Piecewise linear interpolation of the states of the Riemann problem along the cell interface.

Riemann problem is one-dimensional. For each cell interface parallel to the y-axis, there are three Riemann problems: an ERP $(\mathbf{V}_{\mathrm{L},D}, \mathbf{V}_{\mathrm{R},D})$ at the point D, a Riemann problem $(\mathbf{V}_{\mathrm{L},C}, \mathbf{V}_{\mathrm{R},C})$ at the centre defined in the classical one-dimensional fashion, and an ERP $(\mathbf{V}_{\mathrm{L},E}, \mathbf{V}_{\mathrm{R},E})$ at the point E. The proposed method makes the assumption of a piecewise linear distribution of the states of the Riemann problem along y (see Figure 7). Finally, the x-flux across the interface will be calculated by solving the Riemann problem $(\mathbf{U}_{i+1/2,j,\mathrm{L}}, \mathbf{U}_{i+1/2,j,\mathrm{R}})$. The left and right states of this Riemann problem are obtained by averaging the piecewise linear distribution on each side of the interface:

$$\left.\begin{aligned} \mathbf{U}_{i+1/2,j,\mathrm{L}} &= \tfrac{1}{4}\,\mathbf{V}_{\mathrm{L},D} + \tfrac{1}{2}\,\mathbf{V}_{\mathrm{L},C} + \tfrac{1}{4}\,\mathbf{V}_{\mathrm{L},E} \\ \mathbf{U}_{i+1/2,j,\mathrm{R}} &= \tfrac{1}{4}\,\mathbf{V}_{\mathrm{R},D} + \tfrac{1}{2}\,\mathbf{V}_{\mathrm{R},C} + \tfrac{1}{4}\,\mathbf{V}_{\mathrm{R},E} \end{aligned}\right\} \tag{33}$$

3.5. *Simplification of the approach*

To summarize, the computation of the fluxes across an interface $(i+1/2, j)$ involves the following steps:

(1) For the southern end D of the interface, compute the angles θ_k, $k=1,\ldots,4$ from Equation (20) and assign the $\mathbf{U}_k$, $k=1,\ldots,3$ as indicated in Section 2.2 for the left state of the ERP.
(2) Determine the coefficients of Equations (18) for $k=1,2,3$ and solve the resulting system for $\mathbf{U}$. This yields the equivalent state $\mathbf{V}_{\mathrm{L},D}$.
(3) Repeat steps 1 and 2 for the right state of the Riemann problem. This gives the equivalent state $\mathbf{V}_{\mathrm{R},D}$.
(4) Repeat steps 1–3 for the determination of the ERP at the northern end E of the interface.
(5) Define the one-dimensional Riemann problem $(\mathbf{V}_{\mathrm{L},C}, \mathbf{V}_{\mathrm{R},C})$ at the centre of the interface using the classical one-dimensional approach.
(6) Perform the averaging of the three Riemann problems as indicated in Section 3.4 to obtain the averaged problem $(\mathbf{U}_{i+1/2,j,\mathrm{L}}, \mathbf{U}_{i+1/2,j,\mathrm{R}})$.
(7) Solve this problem using any convenient, approximate or exact solver and compute the flux at the interface.

Steps 1 and 2 must be repeated four times for the left and right states of the two ERPs to be determined. In step 2, the solution of the 3×3 system of algebraic equations proves to be very time-consuming in the overall process. From experience, the computational burden can be reduced by approximately one-third if the following approximation is made. It is assumed that the centre of the domain of dependence is located on the corner of the cell. Then, the angles θ_k are given by

$$\left.\begin{aligned}
\theta_1 &= -\pi/2\\
\theta_2 &= 0\\
\theta_3 &= \pi/2\\
\theta_4 &= 3\pi/2
\end{aligned}\right\}\ \text{for the state } L \tag{34}$$

For the left state, the set of relationships (18) becomes

$$\left.\begin{aligned}
u_{\mathrm{L}} - v_{\mathrm{L}} + \frac{\pi}{2}c_{\mathrm{L}} &= u_1 - v_1 + \frac{\pi}{2}c_1\\
u_{\mathrm{L}} + v_{\mathrm{L}} + \frac{\pi}{2}c_{\mathrm{L}} &= u_2 + v_2 + \frac{\pi}{2}c_2\\
-2u_{\mathrm{L}} + \pi c_{\mathrm{L}} &= -2u_3 + \pi c_3
\end{aligned}\right\} \tag{35}$$

The solution of system (35) is

$$\left.\begin{aligned}
u_{\mathrm{L}} &= \tfrac{1}{4}(u_1 + u_2 + 2u_3) + \tfrac{1}{4}(v_2 - v_1) + \frac{\pi}{8}(c_1 + c_2 - 2c_3)\\
v_{\mathrm{L}} &= \tfrac{1}{2}(u_2 - u_1) + \tfrac{1}{2}(v_1 + v_2) + \frac{\pi}{4}(c_2 - c_1)\\
c_{\mathrm{L}} &= \frac{1}{2\pi}(u_1 + u_2 - 2u_3) + \frac{1}{2\pi}(v_2 - v_1) + \tfrac{1}{4}(c_1 + c_2 + 2c_3)
\end{aligned}\right\} \tag{36}$$

where $\mathbf{U}_1 = \mathbf{U}_{i,j+1}$, $\mathbf{U}_2 = \mathbf{U}_{i,j}$ and $\mathbf{U}_3 = (\mathbf{U}_{i,j} + \mathbf{U}_{i,j+1})/2$. Conversely, for the right state, the angles are given by

$$\left.\begin{aligned}
\theta_1 &= -\pi\\
\theta_2 &= -\pi/2\\
\theta_3 &= \pi/2\\
\theta_4 &= \pi
\end{aligned}\right\} \tag{37}$$

This leads to the system

$$\left.\begin{aligned}
-u_{\mathrm{R}} - v_{\mathrm{R}} + \frac{\pi}{2}c_{\mathrm{R}} &= -u_1 - v_1 + \frac{\pi}{2}c_1\\
2u_{\mathrm{R}} + \pi c_{\mathrm{R}} &= 2u_2 + \pi c_2\\
-u_{\mathrm{R}} + v_R + \frac{\pi}{2}c_{\mathrm{R}} &= -u_3 + v_3 + \frac{\pi}{2}c_3
\end{aligned}\right\} \tag{38}$$

The solution of system (38) is

$$\left.\begin{aligned}
u_{\mathrm{R}} &= \tfrac{1}{4}(u_1 + 2u_2 + u_3) + \tfrac{1}{4}(v_1 - v_3) + \frac{\pi}{8}(-c_1 + 2c_2 - c_3)\\
v_{\mathrm{R}} &= \tfrac{1}{2}(u_1 - u_3) + \tfrac{1}{2}(v_1 + v_3) + \frac{\pi}{4}(c_3 - c_1)\\
c_{\mathrm{R}} &= \frac{1}{2\pi}(-u_1 + 2u_2 - u_3) + \frac{1}{2\pi}(v_3 - v_1) + \tfrac{1}{4}(c_1 + 2c_2 + c_3)
\end{aligned}\right\} \qquad (39)$$

where, $\mathbf{U}_1 = \mathbf{U}_{i+1,j+1}$, $\mathbf{U}_2 = (\mathbf{U}_{i+1,j} + \mathbf{U}_{i+1,j+1})/2$ and $\mathbf{U}_3 = \mathbf{U}_{i+1,j}$.

The same approach can be used to transform the same two-dimensional Riemann problem into an ERP along y for the computation of the fluxes across the interface $(i, j+1/2)$, that is the interface between the cells (i, j) and $(i, j+1)$. Denoting the states on the bottom and on the top sides of the interface by $\mathbf{V}_{\mathrm{B}}$ and $\mathbf{V}_{\mathrm{T}}$, respectively, it is easy to check that the following relationships are valid:

$$\left.\begin{aligned}
u_{\mathrm{B}} &= \tfrac{1}{2}(u_1 + u_3) + \tfrac{1}{2}(v_3 - v_1) + \frac{\pi}{4}(c_3 - c_1)\\
v_{\mathrm{B}} &= \tfrac{1}{4}(u_3 - u_1) + \tfrac{1}{4}(v_1 + 2v_2 + v_3) + \frac{\pi}{8}(c_1 - 2c_2 + c_3)\\
c_{\mathrm{B}} &= \frac{1}{2\pi}(-u_1 + u_3 + v_1 - 2v_2 + v_3) + \tfrac{1}{4}(c_1 + 2c_2 + c_3)
\end{aligned}\right\} \qquad (40)$$

where $\mathbf{U}_1 = \mathbf{U}_{i+1,j}$, $\mathbf{U}_2 = (\mathbf{U}_{i,j+1} + \mathbf{U}_{i+1,j+1})/2$ and $\mathbf{U}_3 = \mathbf{U}_{i,j}$, and

$$\left.\begin{aligned}
u_{\mathrm{T}} &= \tfrac{1}{2}(u_1 + u_2) + \tfrac{1}{2}(v_1 - v_2) + \frac{\pi}{4}(c_2 - c_1)\\
v_{\mathrm{T}} &= \tfrac{1}{4}(u_1 - u_2) + \tfrac{1}{4}(v_1 + v_2 = 2v_3) + \frac{\pi}{8}(-c_1 - c_2 + 2c_3)\\
c_{\mathrm{T}} &= \frac{1}{2\pi}(-u_1 + u_2 - v_1 - v_2 + 2v_3) + \tfrac{1}{4}(c_1 + c_2 + 2c_3)
\end{aligned}\right\} \qquad (41)$$

where $\mathbf{U}_1 = \mathbf{U}_{i+1,j+1}$, $\mathbf{U}_2 = \mathbf{U}_{i,j+1}$ and $\mathbf{U}_3 = (\mathbf{U}_{i,j} + \mathbf{U}_{i+1,j})/2$.

3.6. Extension to supercritical conditions

The relationships above were derived under the assumption that the flow is subcritical. However, they are also valid for supercritical situations. When the flow is supercritical, the entire domain of dependence of the solution is located on one side of the interface. Assume for convenience that it is located on the left side of the interface. Then, the solution is entirely determined by the left state of the Riemann problem. Therefore, Equation (30) can still be used for the right state of the ERP. In the determination of the left state of the ERP, the region $[\theta_3, \theta_4]$ is now located on the left-hand side of the interface. Therefore, the values of u, v and c used for the third relationship (18) should be obtained from the average $\mathbf{U}_3 = (\mathbf{U}_{i,j} + \mathbf{U}_{i+1,j})/2$. This is precisely the choice that has been made in Equation (32). The same reasoning can be applied to a supercritical flow going from right to left.

3.7. Stability constraint

The proposed approach is similar to the wave splitting approach proposed by LeVeque [6, 8] in that the value of the flow variable is determined using the initial state over the entire (two-dimensional) domain of dependence of the solution. The numerical solution is stable provided that the entire domain of dependence of the solution does not cross more than one computational cell in each direction of space. In other words, the maximum of the Courant numbers for all the waves in each direction of space should not exceed unity. This condition can be written as

$$\max\left(\frac{|u-c|\Delta t}{\Delta x}, \frac{|u|\Delta t}{\Delta x}, \frac{|u+c|\Delta t}{\Delta x}, \frac{|v-c|\Delta t}{\Delta y}, \frac{|v|\Delta t}{\Delta y}, \frac{|v+c|\Delta t}{\Delta y}\right) \leqslant 1 \tag{42}$$

Note that the stability constraint attached to the classical finite volume approach with one-dimensional solvers is more restrictive, as it imposes that the sum of the maximum Courant numbers in each direction of space should not exceed unity:

$$\max\left(\frac{|u-c|\Delta t}{\Delta x}, \frac{|u|\Delta t}{\Delta x}, \frac{|u+c|\Delta t}{\Delta x}\right) + \max\left(\frac{|v-c|\Delta t}{\Delta y}, \frac{|v|\Delta t}{\Delta y}, \frac{|v+c|\Delta t}{\Delta y}\right) \leqslant 1 \tag{43}$$

The proposed approach has the advantage over the more classical finite volume approach that larger time steps can be used, thus minimizing numerical diffusion and its undesirable effects on the computational solution.

4. DISCUSSION: BOUNDARY CONDITIONS

This section aims to clarify the issue of boundary conditions in two-dimensional simulations. As mentioned in Section 1, there is an apparent contradiction between the theory of existence and uniqueness of solutions in a one-dimensional context and the practical implementations in many software packages for shallow water simulations. In particular, there exist a number of commercially available software packages for two-dimensional simulations where the numerical algorithms use direction splitting (via alternate directions or wave splitting). In these packages, the boundaries are also treated in a dimensionally split fashion. One would therefore expect the number of required boundary conditions to change depending on whether the flow is entering or leaving the domain. In particular, a subcritical incoming flow would require two boundary conditions and a subcritical outflowing one should be expected to require only one boundary condition. Still, many of the packages above can handle the computation with only one boundary condition when the flow is entering the domain. Some other packages can handle situations where the flow is leaving the domain, although two, or three, boundary conditions are being prescribed. The purpose of this section is to demonstrate that this apparent contradiction with the one-dimensional theory is definitely acceptable in the framework of two-dimensional flow. The problem is first investigated in the framework of finite differences, that is the framework used by most commercially available packages. This is the subject of Section 4.1. Section 4.2 deals with the implications for the proposed approach. Section 4.3 illustrates these considerations with the discretization of boundary conditions at a boundary parallel to the y-direction.

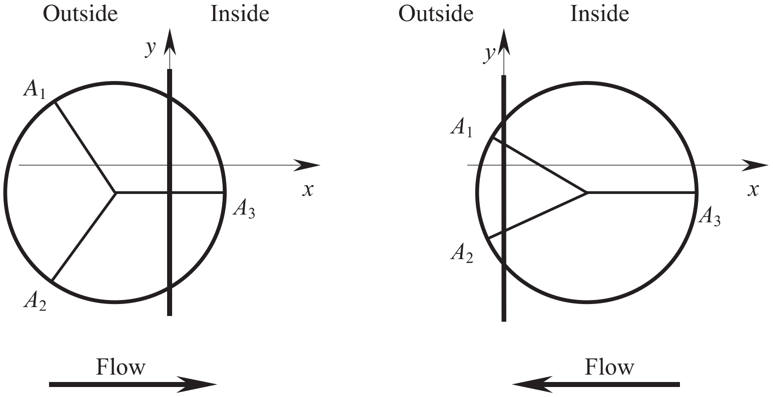

Figure 8. Two boundary conditions prescribed under subcritical conditions. Inflow (left) and outflow (right).

4.1. Finite difference formulations

The case where two boundary conditions are prescribed is examined first. Prescribing two boundary conditions is equivalent to imposing that the feet of two of the bicharacteristics should lie outside the computational domain and the third one should be located inside the computational domain. Figure 8 illustrates such a situation for a boundary parallel to the y-axis. The sketch on the left refers to a subcritical inflow and the sketch on the right represents subcritical outflow.

When the flow is entering the domain, most of the domain of dependence (circle in the sketch) is located outside of the computational domain and the feet A_1 and A_2 of the outer characteristics can be located sufficiently far apart from each other to ensure optimal accuracy of the discretization.

If the flow is leaving the domain, most of the domain of dependence of the solution is located inside the computational domain and the region available for the choice of A_1 and A_2 is narrower. This induces two sources of inaccuracy:

(1) The short distance between A_1 and A_2 in the y-direction leads to inaccurate estimates of the y-gradients, which may result in a numerical diffusion (that is usually associated with small Courant numbers).
(2) There is only one point (A_3) to account for the influence of the inner part of the computational domain on the solution. However, for an outflow the inner part of the domain should be the most influential one, since it contains most of the domain of dependence of the solution. The result is a lack of balance between the influence of the inner and outer parts of the computational domain.

Still, it is possible to prescribe two boundary conditions even though the flow is leaving the computational domain, but the discretization can be expected to yield inaccurate results. The reasons are that it does not reflect the respective influence of the inner and outer parts of the domain on the final solution, and because the discretized gradients along the boundary are most likely to be inaccurate. From symmetry considerations, similar conclusions can be

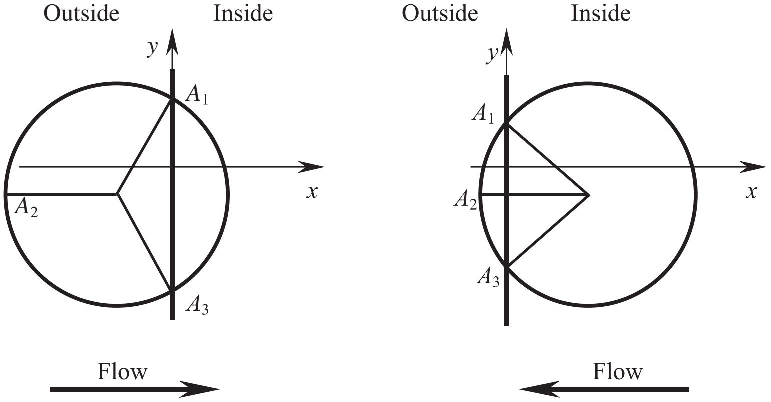

Figure 9. Two boundary conditions prescribed under subcritical conditions. Inflow (left) and outflow (right).

derived about the case where only one boundary condition is prescribed for an incoming flow. Note that in both cases, three boundary conditions could be supplied by locating the three points outside the computational domain. It could also be chosen not to prescribe any condition at all, by locating the three points inside the computational domain. This is a typical way of discretizing non-reflective boundary conditions (see e.g. Reference [2]).

4.2. Proposed approach

The implications for the proposed approach are very similar to those derived in Section 5.1. It is assumed again that two boundary conditions are prescribed. The locations of the three regions used for relationships (18) are shown in Figure 9. The two boundary conditions are prescribed along $[A_1A_2]$ and $[A_2A_3]$. As illustrated by the sketch on the right, prescribing two boundary conditions for a subcritical outflow leads to an imbalance between the contributions from the boundaries and the inner part of the computational domain. Indeed, most of the domain of dependence of the solution is located inside the computational domain, but its contribution to the final solution is accounted for by only one relationship (18). Moreover, the narrow shapes of the regions $[A_1A_2]$ and $[A_2A_3]$ may also cause inaccuracy in the discretization of the gradients along y. Consequently, optimal accuracy of the discretization is achieved by using two boundary conditions for a subcritical inflow and one condition for a subcritical outflow.

4.3. Examples of discretized boundary conditions

The present section shows how the proposed approach can be applied to the discretization of boundary conditions. For the sake of clarity, only a left-hand boundary parallel to the y-direction is considered, but the methodology can be applied in the same way to boundaries parallel to the x-direction.

Consider first the case where only one boundary condition is to be prescribed at the boundary. This condition could be a water depth, x- or y-velocity or discharge, or a relationship between two or more of these variables. The purpose is to determine the value $\mathbf{U}_{1/2,j+1/2}$

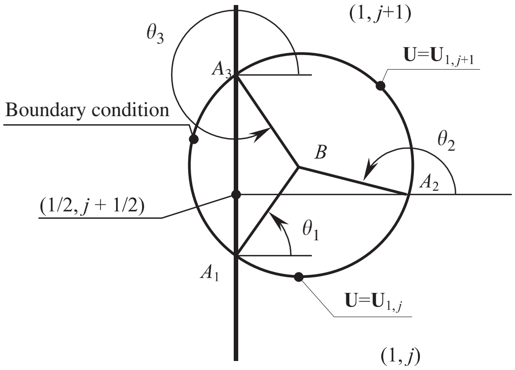

Figure 10. Definition sketch for the application of the boundary conditions.

of $\mathbf{U}$ at the point $(1/2, j+1/2)$ of the boundary between the two cells $(1, j)$ and $(1, j+1)$ (Figure 10). The values of $\mathbf{U}$ at the corners of the boundary cells will be used to determine the boundary fluxes using Equation (33) exactly as for internal cells. As explained in the previous section, relationships (18) are used over two arcs $[\theta_1, \theta_2]$ and $[\theta_2, \theta_3]$ that cover the cells $(1, j)$ and $(1, j+1)$, respectively (Figure 10). The missing third condition is supplied by the boundary condition. As for internal interfaces, the procedure can be simplified by using the following approximation for the angles θ_k:

$$\left.\begin{aligned}\theta_1 &= \pi/2\\ \theta_2 &= \pi\\ \theta_3 &= 3\pi/2\end{aligned}\right\} \tag{44}$$

Noticing that $\mathbf{U} = \mathbf{U}_{1,j}$ and $\mathbf{U} = \mathbf{U}_{1,j+1}$ over the arcs $[\theta_1, \theta_2]$ and $[\theta_2, \theta_3]$, respectively, and substituting Equation (44) into Equations (18), (19), leads to the following system:

$$\left.\begin{aligned}-u_{1/2,j+1/2} + u_{1,j} + v_{1/2,j+1/2} - v_{1,j} + (c_{1/2,j+1/2} - c_{1,j})\pi = 0\\ -u_{1/2,j+1/2} + u_{1,j+1} - v_{1/2,j+1/2} + v_{1,j+1} + (c_{1/2,j+1/2} - c_{1,j+1})\pi = 0\end{aligned}\right\} \tag{45}$$

Subtracting the first from the second equation (45) yields the following equality:

$$v_{1/2,j+1/2} = \frac{-u_{1,j} + u_{1,j+1}}{2} + \frac{v_{1,j} + v_{1,j+1/2}}{2} + \pi\frac{c_{1,j} - c_{1,j+1}}{2} \tag{46}$$

The immediate implication of Equation (46) is that the velocity v cannot be prescribed at the boundary because v is entirely determined from the values inside the domain. Adding the two equalities in Equation (45) yields the following condition:

$$-u_{1/2,j+1/2} + \pi c_{1/2,j+1/2} + \frac{u_{1,j} + u_{1,j+1}}{2} - \frac{v_{1,j} - v_{1,j+1}}{2} - \pi\frac{c_{1,j} + c_{1,j+1}}{2} = 0 \tag{47}$$

Prescribing u or c (or a relationship between the two) allows Equation (48) to be solved uniquely for $u_{1/2,j+1/2}$ and $c_{1/2,j+1/2}$. If a value c_b of c is to be prescribed at the boundary, solving Equation (47) for $u_{1/2,j+1/2}$ gives:

$$u_{1/2,j+1/2} = \frac{u_{1,j} + u_{1,j+1}}{2} + \frac{-v_{1,j} + v_{1,j+1}}{2} + \pi\left(c_b - \frac{c_{1,j} + c_{1,j+1}}{2}\right) \tag{48}$$

Conversely, if a value u_b of the velocity u is to be prescribed at the boundary, solving Equation (47) for $c_{1/2,j+1/2}$ leads to

$$c_{1/2,j+1/2} = \frac{2u_b - u_{1,j} - u_{1,j+1}}{2\pi} + \frac{v_{1,j} - v_{1,j+1}}{2\pi} + \frac{c_{1,j} + c_{1,j+1}}{2} \tag{49}$$

Note that the impossibility for v to be prescribed at the boundary does not occur only for simplification (44). It can be checked that this situation occurs when the angles θ_k satisfy the following relationships:

$$\left.\begin{aligned} \theta_2 &= \pi \\ \theta_3 &= 2\pi - \theta_1 \end{aligned}\right\} \tag{50}$$

which is always the case when the centre B of the domain of dependence is located on the interface between the cells $(1,j)$ and $(1,j+1)$. This occurs, in particular, when the direction of the flow is perpendicular to the boundary. In this case Equations (18) and (19) yield the following system:

$$\left.\begin{aligned} (\sin\theta_2 - \sin\theta_1)(u_{1/2,j+1/2} - u_{1,j}) - (\cos\theta_2 - \cos\theta_1)(v_{1/2,j+1/2} - v_{1,j}) \\ + 2(\theta_2 - \theta_1)(c_{1/2,j+1/2} - c_{1,j}) = 0 \\ (\sin\theta_3 - \sin\theta_2)(u_{1/2,j+1/2} - u_{1,j+1}) - (\cos\theta_3 - \cos\theta_2)(v_{1/2,j+1/2} - v_{1,j+1}) \\ + 2(\theta_3 - \theta_2)(c_{1/2,j+1/2} - c_{1,j+1}) = 0 \end{aligned}\right\} \tag{51}$$

Equation (50) leads to:

$$\left.\begin{aligned} \sin\theta_2 - \sin\theta_1 &= \sin\theta_3 - \sin\theta_2 \\ \cos\theta_2 - \cos\theta_1 &= -(\cos\theta_3 - \cos\theta_2) \\ \theta_2 - \theta_1 &= \theta_3 - \theta_2 \end{aligned}\right\} \tag{52}$$

Subtracting the second from the first equation (51) and substituting Equation (52) gives:

$$v_{1/2,j+1/2} = \frac{u_{1,j} - u_{1,j+1}}{1 + \cos\theta_1}\frac{\sin\theta_1}{2} + \frac{v_{1,j} + v_{1,j+1}}{2} + \frac{\pi - \theta_1}{1 + \cos\theta_1}(c_{1,j} - c_{1,j+1}) \tag{53}$$

which shows that v is determined entirely by the values inside the domain. Therefore, it cannot be prescribed at the boundary.

Consider now the case where two conditions are to be prescribed at the boundary. In this case, the differential relationship (9) is integrated over the arc $[\theta_1, \theta_3]$:

$$\int_{\theta_1}^{\theta_2} [(u_{1/2,j+1/2} - u_{1,j})\cos\theta + (v_{1/2,j+1/2} - v_{1,j})\sin\theta + 2(c_{1/2,j+1/2} - c_{1,j})]\,\mathrm{d}\theta$$
$$+ \int_{\theta_2}^{\theta_3} [(u_{1/2,j+1/2} - u_{1,j+1})\cos\theta + (v_{1/2,j+1/2} - v_{1,j+1})\sin\theta$$
$$+ 2(c_{1/2,j+1/2} - c_{1,j+1})]\,\mathrm{d}\theta = 0 \tag{54}$$

Using the approximation given by Equation (44) leads to Equation (48) again. Since $v_{1/2,j+1/2}$ does not appear in Equation (48), it must be specified via at least one of the two boundary conditions, otherwise leading to non-uniqueness of the solution. In order to ensure the closure of Equation (48) and uniqueness of the solution the remaining boundary condition should be a prescribed value of u, a prescribed value of c or a relationship between u and c. The solution is given by Equation (48) or by Equation (47) depending on the nature of the boundary condition. It can be shown again that this requirement on the nature of the boundary conditions is necessary when Equation (50) is verified.

The considerations above can be summarized as follows: if only one boundary condition is to be prescribed, it should involve the celerity (or the water depth) and/or the component of the velocity in the direction normal to the boundary. Attempting to specify the component of the velocity in the direction parallel to the boundary may lead to non-existence of the solution for some configurations of the flow field. If two boundary conditions are to be prescribed, at least one of them has to involve the component of the velocity parallel to the boundary, otherwise leading to non-uniqueness of the solution for some configurations of the flow field.

5. COMPUTATIONAL EXAMPLES

5.1. Circular dambreak simulations

The performance of the proposed solver is compared to that of alternative solvers using the classical circular dambreak test. The initial conditions are the following. The water depth is equal to 1 m everywhere in the computational domain, except for a circular region the radius of which is 5 m, where the depth is equal to 30 m. The horizontal dimensions of the computational domain are 100 m × 100 m. Owing to the non-zero depth around the circular dam and the large initial depth in the reservoir, a supercritical shock appears.

The problem was solved using two schemes combined with two solvers. In all cases, the time step was set equal to the maximum permissible value that satisfies the stability constraint (see Section 3.7). The first scheme is the classical, first-order Godunov scheme and the second scheme is the discontinuous profile method (DPM) that yields second-order accuracy on smooth profiles [14] and has the advantage of better capturing discontinuities than the usual MUSCL reconstruction. The two solvers used were the classical one-dimensional solver (that yields the stability constraint given by Equation (43)) and the proposed one (that yields the stability constraint given by Equation (42)), with the simplification described in Section 3.5. Figures 11–14 show the results obtained on a square grid with a cell width

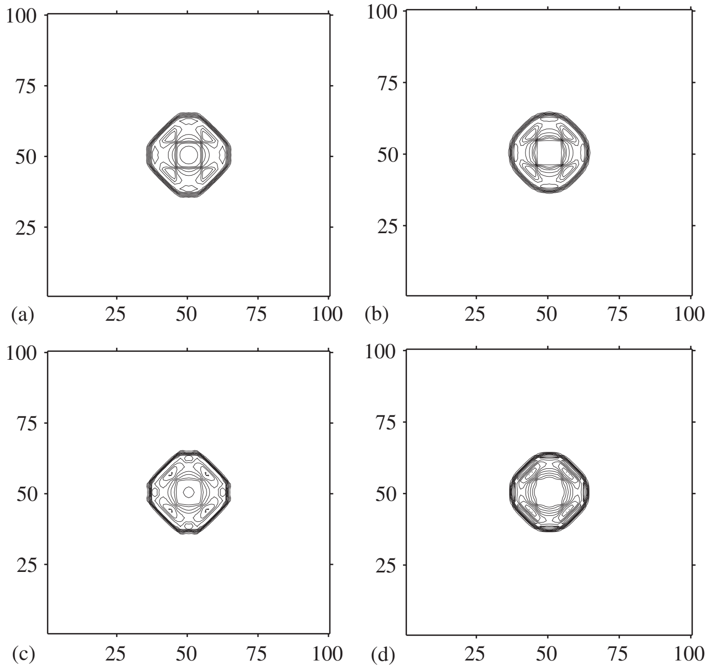

Figure 11. Circular dambreak. Contour lines of the water depths after 0.5 s. (a) Godunov scheme with one-dimensional solver. (b) Godunov scheme with proposed solver. (c) DPM scheme with one-dimensional solver. (d) DPM scheme with proposed solver.

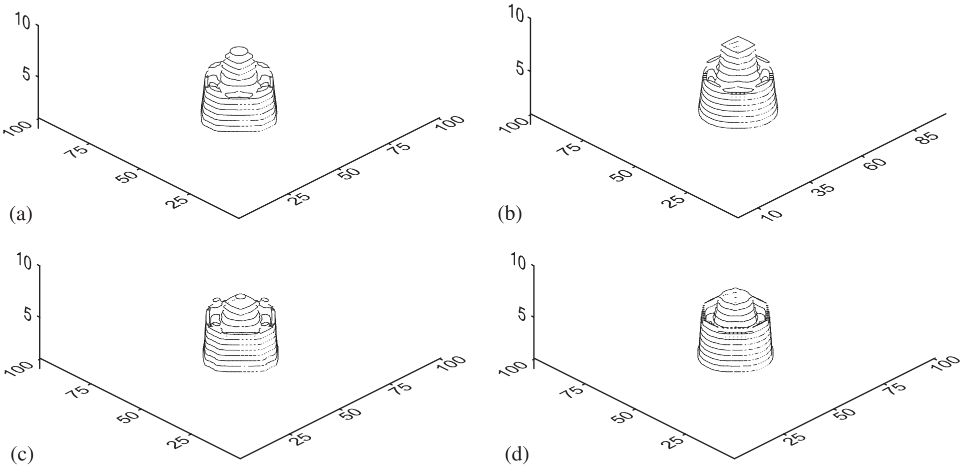

Figure 12. Circular dambreak. Perspective view of the water depths after 0.5 s. (a) Godunov scheme with one-dimensional solver. (b) Godunov scheme with proposed solver. (c) DPM scheme with one-dimensional solver. (d) DPM scheme with proposed solver.

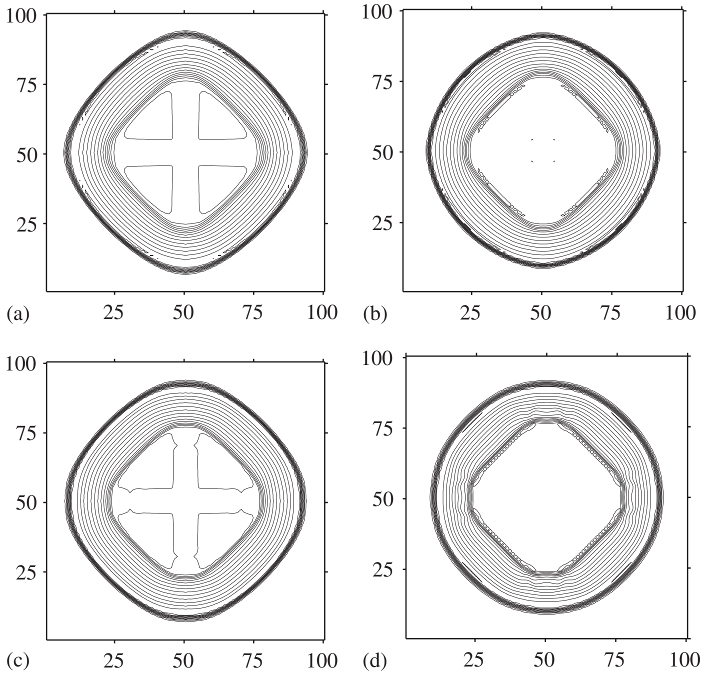

Figure 13. Circular dambreak. Contour lines of the water depths after 3 s. (a) Godunov scheme with one-dimensional solver. (b) Godunov scheme with proposed solver. (c) DPM scheme with one-dimensional solver. (d) DPM scheme with proposed solver.

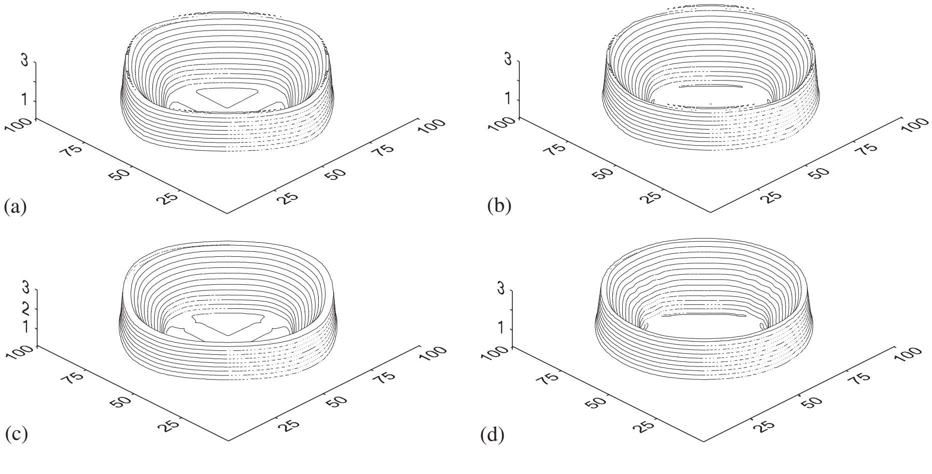

Figure 14. Circular dambreak. Perspective view of the water depths after 3 s. (a) Godunov scheme with one-dimensional solver. (b) Godunov scheme with proposed solver. (c) DPM scheme with one-dimensional solver. (d) DPM scheme with proposed solver.

of 1 m. In each figure, sketch (a) shows the results obtained using the Godunov scheme with the classical one-dimensional solver. (b) Presents the results of the Godunov scheme in conjunction with the two-dimensional Riemann solver. (c) Shows the combination of the DPM scheme with the one-dimensional solver and (d) displays the water depths obtained from the DPM and the two-dimensional Riemann solver. Figures 11 and 12 present the results for $t = 0.5$ s, while Figures 13 and 14 show the results at $t = 3$ s after the breaking of the dam. From these results, it is clear that the proposed solver allows the isotropy of the solution to be preserved better than if the one-dimensional solver is used. Owing to the coarse resolution of the initial depth profile, the isotropy of the solution is not satisfactory in any of the four cases at $t = 0.2$ s. Still, the proposed two-dimensional solver yields better computed profiles than the one-dimensional one. In particular, the polarization of the shock front along the diagonal of the grid is dramatically reduced by the two-dimensional solver. From Figure 13, it appears that the tail of the rarefaction wave (inner contours in the profiles) still exhibits some anisotropy at further times, but the shock wave is better resolved by the two-dimensional solver. The conclusion that can be drawn from this experiment is that, when multi-dimensional flows are involved, a first-order scheme coupled with a multi-dimensional solver may better preserve the isotropy of the solution than a second-order scheme that uses a one-dimensional solver. This conclusion, also stated in Reference [9], is confirmed by many other experiments that cannot be presented here owing to lack of space.

Figure 15 compares the evolution of the water depth at the centre of the dam with the (semi-analytical) theoretical solution. When the two-dimensional solver is used the computed evolution is closer to the theoretical solution, regardless of the order of the scheme. In all cases, the time at which the water level at the centre of the dam starts decreasing is underestimated in the computation. This is easily explained by the coarseness of the grid that does not allow the circular symmetry of the initial condition to be preserved in the computation (remember that the cell dimensions are 1 m × 1 m for a dam radius of 5 m). On the computational grid the average distance from the border to the centre is smaller than 5 m. As a result the rarefaction wave starting from the border of the dam reaches the centre earlier in the computation than in the theoretical solution. The use of the two-dimensional solver allows the discrepancy between the computation and the analytical solution to be reduced. This is because the two-dimensional solver allows larger time steps to be used, hence leading to a reduction of the amount of numerical diffusion in the computed solution. The rarefaction wave is less affected by artificial spreading when the two-dimensional solver is used and the time at which the wave reaches the centre of the dam is better represented in the computational solution.

Table I compares the computational costs of the various approaches as a ratio of CPU times to the CPU time needed by the one-dimensional, first-order scheme. When used with the first-order scheme, the two-dimensional solver yields a relative increase of only 40% in the CPU compared to the one-dimensional solver. Although the two-dimensional solver requires more computations per time step, its enhanced stability allows larger computational time steps to be adopted, thus limiting the total number of elementary operations to be performed throughout the simulation. In contrast, the second-order scheme with the one-dimensional solver requires 2.4 times as much total CPU time as the first-order scheme. It requires 4 times as much CPU time when used with the two-dimensional solver. The total CPU time needed by the second-order scheme with the two-dimensional solver is 85% larger than with the one-dimensional solver.

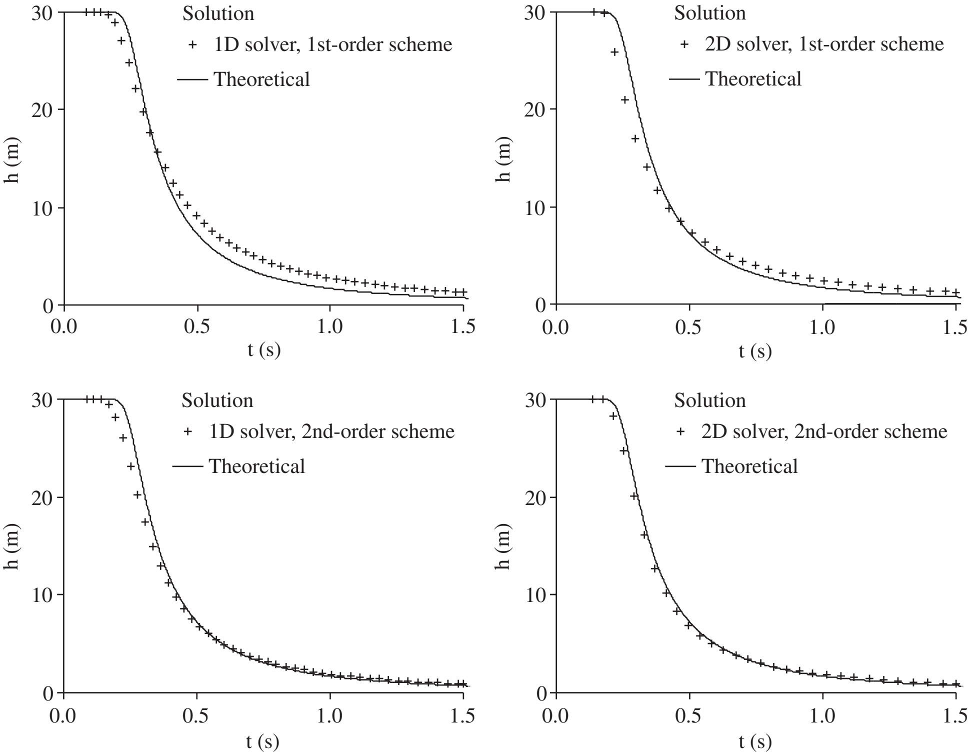

Figure 15. Circular dambreak. Comparison of the computed water depth at the centre of the dam as a function of time with the theoretical solution.

Table I. Computational cost of the various approaches for the circular dambreak test case (ratios are expressed to the total CPU time of the first-order scheme with the one-dimensional solver).

Scheme accuracy	Solver	CPU time ratio
First-order	one-dimensional	1.0
First-order	two-dimensional	1.4
Second-order	one-dimensional	2.4
Second-order	two-dimensional	4.0

5.2. Boundary conditions: wave reflections

The purpose of this test is to compare the computational results obtained using the options presented in Section 4 for the treatment of boundary conditions. The configuration of the test is illustrated by Figure 16. The horizontal dimensions of the computational domain are 100 m × 100 m. The water is initially at rest, with a uniform initial depth of 1 m. Sinusoidal oscillations are imposed on the free surface at a point located 10 m away from the eastern and northern boundaries. The amplitude and period of the oscillations are 0.5 m and 6 s,

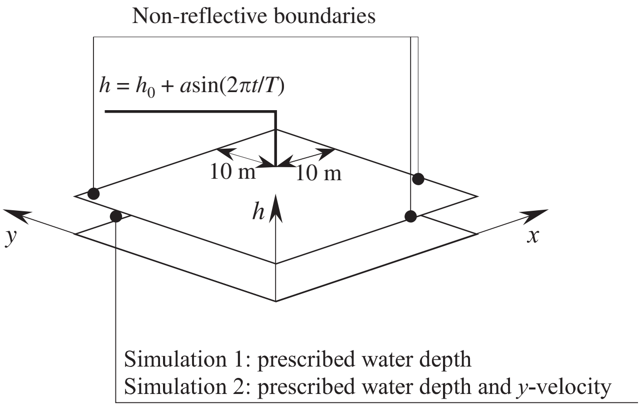
Non-reflective boundaries
$h = h_0 + a\sin(2\pi t/T)$
10 m
10 m
h
y
x
Simulation 1: prescribed water depth
Simulation 2: prescribed water depth and y-velocity

Figure 16. Definition sketch of the second test case.

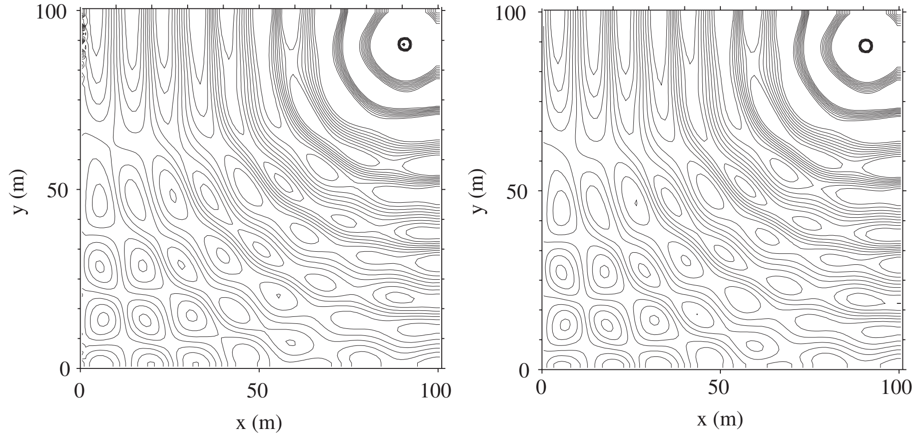

Figure 17. Computed water depths at $t = 1500$ s using one boundary condition (left) and two boundary conditions (right). The contour line equidistance is 2.5×10^{-4} m. The fields are truncated between $h = 0.99$ and 1.01 m for the sake of clarity.

respectively. These oscillations propagate freely into the domain and reach the boundaries. All boundaries are non-reflective, except for the western boundary, where two of the options proposed in Section 4 are applied. In a first simulation, a water depth of 1m alone is prescribed at the boundary. In a second simulation, both a water depth of 1 m and a y-velocity of 0 m/s are prescribed at the boundary.

Figure 17 shows the computed water depths using the two options at $t = 1500$ s after the beginning of the simulation. Figure 18 shows the difference between the two computed fields. Figure 19 shows the computed depths as a function of time at three points along the boundary. Clearly, the results obtained using one and two boundary conditions differ significantly. It will be noticed that the largest difference between the two computations is observed near the north-

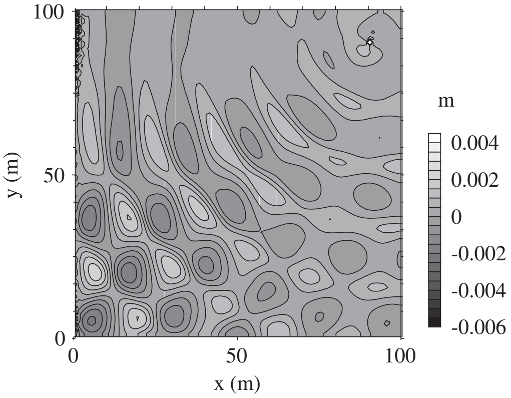

Figure 18. Difference between the water depth fields computed at $t = 1500$ s using one and two boundary conditions.

western boundary of the model (point ($x = 1$m, $y = 99$m), bottom of Figure 19). Although the computed amplitudes are approximately the same in both computations, the average computed depth is approximately 3×10^{-3}m larger with only one than with two boundary conditions. This is surprising at first sight because from symmetry considerations, the velocity v in the direction parallel to the boundary should be expected to be close to zero in this part of the model even when it is not prescribed as a boundary condition. Therefore, the two computational results should be expected to be very similar. The contrast between the two simulations shows that it is not the case and that the number (and nature) of boundary conditions may strongly influence the computational results even in very simple cases. Obviously, further research is needed to characterize the influence of the number and nature of boundary conditions on the accuracy (and possibly the stability) of the computed solutions.

6. CONCLUSIONS

A method has been presented for the solution of the two-dimensional shallow water equations. It is based on the conversion of two-dimensional Riemann problems at the corners of the cells to one-dimensional equivalent Riemann problems (ERPs). The one-dimensional ERP at one cell corner is determined from the two-dimensional Riemann problem by integrating the differential relationships along the bicharacteristics over the entire domain of dependence of the solution. The states of the ERPs are averaged along the edges of the cells assuming a linear distribution. The resulting averaged Riemann problem is solved using any conventional exact or approximate solver. Besides allowing for an increase in the computational time step, the proposed technique allows the isotropy of solutions to be preserved better. The proposed approach, that uses the information available over the entire domain of dependence of the solution, may contribute to improve substantially the quality of numerical solutions on Cartesian and curvilinear grids. This is expected to be true in particular for strongly curved grids, where classical one-dimensional approaches may fail to characterize correctly the domain of dependence of the solution.

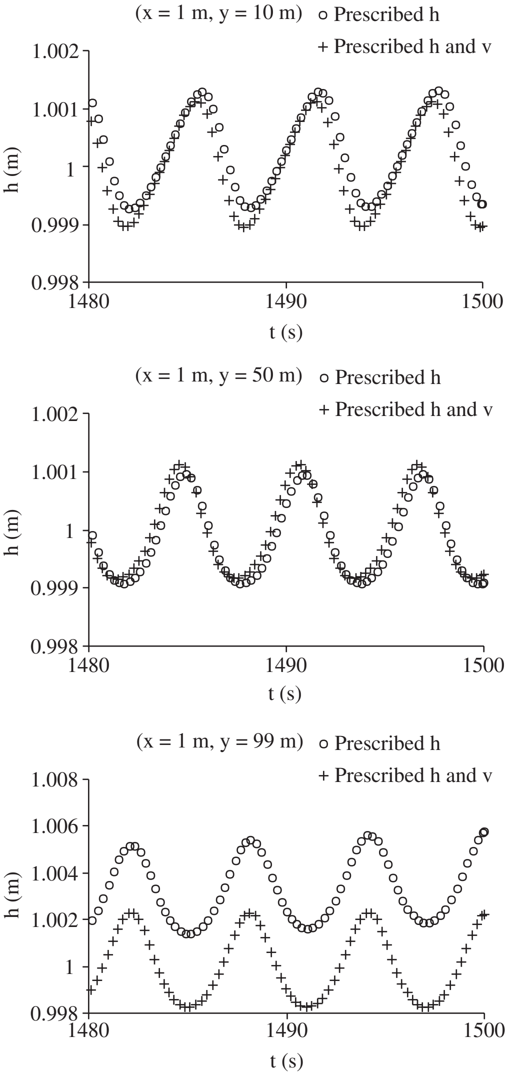

Figure 19. Computed water depths at various points of the model using one and two boundary conditions.

The present paper also addresses the treatment of open boundary conditions. Classical approaches based on local one-dimensional splitting of the equations require that two boundary conditions should be prescribed for a subcritical inflow and that one boundary condition should be imposed for a subcritical outflow. These conditions are shown not to be necessary for the existence of the solutions in the case of genuinely two-dimensional flow. Provided that the flow is supercritical, it is possible to prescribe as many boundary conditions as desired. This has no consequence on the existence and uniqueness of the solution, but it influences its accuracy. Optimal accuracy is achieved when the number of boundary conditions reflects the proportion of the domain of dependence of the solution that is located outside the computational domain. For this reason, the best option is to prescribe two boundary conditions for a subcritical inflow and one boundary condition for a subcritical outflow. The number of boundary conditions to be prescribed influences their nature. When only one boundary condition is to be prescribed, it should not be the component of the velocity in the direction parallel to the boundary, otherwise leading to non-existence of the solution for a number of flow configurations. When two boundary conditions are to be prescribed, at least one of them should include the component of the velocity parallel to the boundary, otherwise leading to non-unique solutions in some cases.

ACKNOWLEDGEMENTS

This work was carried out while the author was working as a lecturer at International Institute for Infrastructural Hydraulics and Environmental Engineering—IHE, Westvest 7, 2601 DA Delft, the Netherlands. The author is much indebted to Dr J.A. Cunge and Professor M.J. Hall from IHE for their remarks and suggestions on the present contribution.

REFERENCES

1. Strang G. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis* 1968; **5**:506–517.
2. Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics* (2nd edn.). Springer: Berlin, 1998.
3. Boris JP, Book DL. Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics* 1973; **11**:38–69.
4. Boris JP, Book DL. *Journal of Computational Physics* 1976; **20**:397.
5. Bell JB, Dawson CN, Shubin GR. An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions. *Journal of Computational Physics* 1988; **74**:1–24.
6. LeVeque RJ. High resolution finite volume methods on arbitrary grids via wave propagation. *Journal of Computational Physics* 1988; **78**:36–63.
7. Davis SF. A rotationally biased upwind difference scheme for the Euler equations. *Journal of Computational Physics* 1984; **56**:65–92.
8. LeVeque RJ, Shyue K-M. Two dimensional front tracking based on high resolution wave propagation methods. *Journal of Computational Physics* 1996; **123**:354–368.
9. Brio M, Zakharian AR, Webb GM. Two-dimensional Riemann solver for Euler equations of gas dynamics. *Journal of Computational Physics* 2001; **167**:177–195.
10. Daubert A, Graffe O. Quelques aspects des écoulements presque horizontaux à deux dimensions en plan et non permanents. Application aux estuaires. *La Houille Blanche* 1967; **22**(8):847–860 (in French).
11. Garabedian PR. *Partial Differential Equations*. Wiley: New York, 1964.
12. Katopodes N, Strelkoff T. Two-dimensional shallow water-wave models. *Journal of the Engineering Mechanics Division* (*ASCE*) 1979; **105**(2):317–334.
13. Gerritsen H. Accurate boundary treatment in shallow water flow computations. *Ph.D. Thesis*, University of Twente, Netherlands, 1982.
14. Guinot V. The discontinuous profile method for simulating two-phase flow in pipes using the single component approximation. *International Journal for Numerical Methods in Fluids* 2001; **37**:341–359.
15. Colella P, Woodward PR. The piecewise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics* 1984; **154**:274–201.