

Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: Application to the Malpasset dam-break flood (France, 1959)[‡]

D. L. George^{*,†}

Cascades Volcano Observatory, U.S. Geological Survey, 1300 SE Cardinal Ct., Bldg. 10,
Vancouver, WA 98683, U.S.A.

SUMMARY

The simulation of advancing flood waves over rugged topography, by solving the shallow-water equations with well-balanced high-resolution finite volume methods and block-structured dynamic adaptive mesh refinement (AMR), is described and validated in this paper. The efficiency of block-structured AMR makes large-scale problems tractable, and allows the use of accurate and stable methods developed for solving general hyperbolic problems on quadrilateral grids. Features indicative of flooding in rugged terrain, such as advancing wet-dry fronts and non-stationary steady states due to balanced source terms from variable topography, present unique challenges and require modifications such as special Riemann solvers. A well-balanced Riemann solver for inundation and general (non-stationary) flow over topography is tested in this context. The difficulties of modeling floods in rugged terrain, and the rationale for and efficacy of using AMR and well-balanced methods, are presented. The algorithms are validated by simulating the Malpasset dam-break flood (France, 1959), which has served as a benchmark problem previously. Historical field data, laboratory model data and other numerical simulation results (computed on static fitted meshes) are shown for comparison. The methods are implemented in GEOCLAW, a subset of the open-source CLAWPACK software. All the software is freely available at www.clawpack.org. Published in 2010 by John Wiley & Sons, Ltd.

Received 11 September 2009; Revised 7 January 2010; Accepted 9 January 2010

KEY WORDS: shallow water; finite volume methods; mesh adaptation; hyperbolic PDE; hydrodynamics; validation

1. INTRODUCTION

Advancing flood waves that result from dam breaches, volcanic eruptions or flash floods in mountainous canyons typically move over steep and rugged topography and are therefore highly dynamic and unpredictable in their inundation extent. The numerical simulation of these types of rugged-terrain floods is valuable for risk mitigation or for the design of infrastructure. However, these floods present great challenges for numerical modeling due to the difficult physics and geometry. For instance, the flow often contains discontinuities due to nonlinearities in the governing equations as well as to geometric irregularities in the topography. Additionally, the advancing flood front is a wet-dry boundary that moves rapidly over non-smooth topography. The numerical method used in a simulation must be robust and accurate given these features: methods designed and tested

^{*}Correspondence to: D. L. George, Cascades Volcano Observatory, U.S. Geological Survey, 1300 SE Cardinal Ct., Bldg. 10, Vancouver, WA 98683, U.S.A.

[†]E-mail: dgeorge@usgs.gov

[‡]This article is a U.S. Government work and is in the public domain in the U.S.A.

for tsunami modeling or more common flooding due to rising waters in low-lying areas are not necessarily adequate.

Aside from the difficult physics of the flow, even the geometry of the flood domain poses challenges with respect to using suitable computational grids. Consider a flood advancing down a ravine that winds through mountainous terrain: the flow itself might represent a small fraction of the smallest rectangular domain that encompasses the flood, making uniformly spaced quadrilateral grids unusable because a reasonably fine grid is prohibitively inefficient. Because topography has an important influence on flood-routing, static unstructured meshes that are tailored to local topography (e.g. with finer cells where the flood is likely to inundate) are traditionally used for flood modeling. Even given these sophisticated meshes (which can be time-consuming and cumbersome to produce), using static meshes of any kind has an inherent inefficiency due to the dynamic geometry of a flood. For instance, it is not always apparent *a priori* where and how far the flood will inundate, and moreover, the necessary grid resolution is finer near the moving flow front than near the source of the flood or areas where the flood has yet to inundate. In summary, the optimal grid resolution at a given location is temporally dependent and unknown *a priori*, suggesting the use of adaptive mesh refinement (AMR). This is particularly relevant when speed and efficiency are a priority for modelers, as might be the case in risk mitigation studies or when large-scale computers are not available.

Dynamic block-structured AMR, developed for general hyperbolic conservation laws (e.g. [1, 2]), enables the efficient resolution of moving waves by providing multiple levels of evolving patches of rectangular sub-grids, or *blocks*, which appear around and move with waves or other dynamic features in the solution. For flood modeling, this means that a single coarse grid can bound the potential flood domain, whereas finer sub-grids form a patchwork that surrounds the flow, and possibly finer sub-grids resolve the flood front as it moves unpredictably through winding or irregular topography. Additionally, block-structured AMR allows the use of robust, stable high-resolution methods, discussed below, which are designed for logically rectangular grids. These methods have properties that can be difficult to achieve on unstructured grids. Although, to the author's knowledge, this is a novel application for block-structured AMR, the shallow-water equations have been solved using adaptive quadtree grids in the context of dam-break flooding by Borthwick and others [3, 4].

Because Godunov methods (i.e. methods that make use of solutions to Riemann problems) can capture propagating discontinuities or irregularities without spurious numerical oscillations, this class of numerical schemes has been used frequently and successfully for solving the shallow-water equations or more general hyperbolic problems exhibiting wave propagation. This includes LeVeque's wave-propagation algorithms [5], designed for logically rectangular quadrilateral grids, which have been extended to block-structured AMR grids for general hyperbolic problems [6]. These methods are *high resolution* in that they satisfy TVD properties while being formally second-order accurate for smooth solutions (see e.g. [7–9] for an overview of these topics). A brief description of these methods in the context of flood modeling is given in Section 3.1.

Recently, considerable effort has been devoted to devising schemes (e.g. [10–15]) that accurately treat hyperbolic equations containing source terms, such as the shallow-water equations over variable topography, because these systems often have non-trivial steady states that are difficult to maintain numerically. *Well-balanced* schemes are methods that numerically preserve and resolve perturbations to such steady states, usually by the use of a specially designed well-balanced Riemann solver that treats steady-state balances carefully. For the shallow-water equations, well-balanced schemes have typically been developed for applications in which the dominant steady-state balance occurs with a stationary pool of water overlying variable topography (see e.g. [16, 17]). However, for flooding in rugged terrain, the dominant steady-state balance is more similar to a steady rushing river. Because the contours of terrain play an important role in determining flow path, there is an incentive to treat the source term accurately for these non-stationary or flowing steady states. The Riemann solver described in [18] was developed for general flow over topography, and was developed with the goal of accurately capturing inundation (moving wet-dry fronts) and being well-balanced with regard to stationary as well as non-stationary steady states, such as those prevalent in floods flowing through rugged topography.

The GEOCLAW software package is actively developed for modeling general shallow-flow over topography by combining LeVeque's wave-propagation algorithms [5, 6], dynamic block-structured AMR [1, 2], and employing well-balanced Riemann solvers such as that described in [18]. Although we originally developed GEOCLAW for tsunami modeling (e.g. [19–21]), the goal of this present work is to test the efficacy of the numerical methods implemented in GEOCLAW for modeling flood waves in rugged terrain. GEOCLAW is included with the open-source software package CLAWPACK, developed more generally for hyperbolic conservation laws, which can be freely obtained at www.clawpack.org.

For validation, a GEOCLAW simulation of the flood resulting from the Malpasset dam failure, which occurred in southern France in 1959, is presented in Section 4. This >60-m high dam failed suddenly and catastrophically, sending a massive flood wave down a winding river bed into the Reyran River Valley and eventually inundating the town of Fréjus on the Mediterranean coast with tragic consequences [22]. This disaster serves as a valuable validation tool because a considerable amount of field data were collected indicating the maximum water levels reached at various locations. Because of the available data, this event was chosen as the benchmark for a European research consortium—Concerted Action on Dam-break Modelling (CADAM), organized by Electricité de France (EDF) and Hydraulic Research Wallingford (HR) in the 1990s to promote the development of models and numerical schemes for dam-break flooding. Researchers as part of the CADAM effort have therefore produced other simulation results that are valuable for comparison and cross-validation.

2. MATHEMATICAL MODEL

Although floods in rugged terrain certainly have three-dimensional (3D) flow variation, many researchers have found depth-averaged two-dimensional (2D) governing equations to be a suitable approximation (e.g. [23–25]), and reducing the dimension of the problem makes large-scale problems tractable. Depth-averaged 2D equations can be derived by integrating 3D governing equations in the vertical z -direction from the solid bed $b(x, y)$ to the free surface $\eta(x, y, t)$ of the flow, and applying boundary conditions at those surfaces (Figure 1(a)). This gives new governing equations for the depth $h(x, y, t) = \eta(x, y, t) - b(x, y)$, and the depth-averaged velocities $u(x, y)$ and $v(x, y)$ in the x - and y -directions, respectively. The most commonly used depth-averaged equations are the shallow-water equations

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0, \quad (1a)$$

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}(huv) = -gh\frac{\partial b}{\partial x} + S_{fx}, \quad (1b)$$

$$\frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) = -gh\frac{\partial b}{\partial y} + S_{fy}, \quad (1c)$$

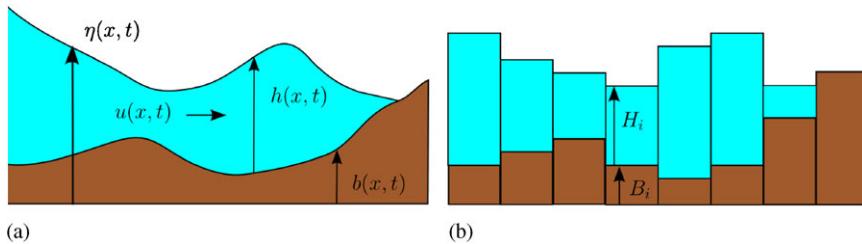


Figure 1. A vertical cross-section of the stationary topography (darker shade) and free-surface flow (lighter shade) for a depth-averaged flow: (a) Diagram of the topography ($b(x, t)$) and dependent variables and (b) a (1D) finite volume discretization for a depth-averaged flow.

where S_{fx} and S_{fy} are friction terms. GEOCLAW employs the commonly used empirical Manning formulae

$$S_{fx} = n^2 u \sqrt{u^2 + v^2} h^{-4/3}, \quad (2a)$$

$$S_{fy} = n^2 v \sqrt{u^2 + v^2} h^{-4/3}, \quad (2b)$$

where n is the Manning coefficient—an estimated parameter reflecting the roughness of the bed. The shallow-water equations also assume incompressible flow with hydrostatic pressure as the stress. More complicated depth-averaged flow equations can be derived for fluids with more complicated stress tensors (non-hydrostatic pressure as well as frictional shear stress). See [26, 27] for water flooding with non-hydrostatic pressure and [25, 28, 29], for debris flows and landslides. Eventually we hope to extend the numerical methods implemented in GEOCLAW to such equations for modeling landslides and various debris flows in rugged terrain.

Equations (1) and more general depth-averaged equations can be written compactly as

$$\frac{\partial}{\partial t} q + \frac{\partial}{\partial x} f^1(q) + \frac{\partial}{\partial y} f^2(q) = \psi(q, x, y), \quad (3)$$

where

$$q = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad (4)$$

$f^1(q)$ and $f^2(q)$ are the fluxes of q in the x - and y -direction, respectively, and $\psi(q, x, y)$ is a source term. The system (3) belongs to the more general class of hyperbolic conservation laws (with a source term) for m conserved quantities, for which $q, f^1, f^2, \psi \in \mathbb{R}^m$. Because of common difficulties exhibited by hyperbolic conservation laws, a considerable amount of research has led to the development of specialized numerical schemes for such systems (see [8] for an overview).

3. NUMERICAL METHOD

3.1. Wave-propagation finite volume method

Finite volume methods provide a natural framework for ensuring numerical conservation—an important property when solving hyperbolic systems generally [8], and particularly for accurately resolving the location of a flow front in flooding problems [19]. Figure 1(b) shows a vertical one-dimensional (1D), or cross-section of a finite volume discretization of a depth-averaged flow, such as that shown in Figure 1(a). For full 2D depth-averaged flows, the numerical solution is a vector-valued piecewise constant function on a uniformly spaced logically rectangular grid with rectangular grid cells $\mathcal{C}_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ (Figure 2). At the beginning of each time step, t^n , the numerical solution Q_{ij}^n is constant in \mathcal{C}_{ij} , and is an approximation to the average value of the true solution $q(x, y, t^n)$ in the cell:

$$Q_{ij}^n := \begin{bmatrix} H_{ij}^n \\ HU_{ij}^n \\ HV_{ij}^n \end{bmatrix} \approx \frac{1}{\lambda(\mathcal{C}_{ij})} \int_{\mathcal{C}_{ij}} q(x, y, t^n) dx dy, \quad (5)$$

where $\lambda(\mathcal{C}_{ij})$ is the area of \mathcal{C}_{ij} .

GEOCLAW employs LeVeque's high-resolution wave-propagation algorithm [5], which is a Godunov-type scheme [30] employing the solution to local Riemann problems. The numerical solution is updated over a time step from Q_{ij}^n to Q_{ij}^{n+1} , by accounting for the effect of the waves (propagating gradients) present in the solutions to the Riemann problems at the four cell interfaces

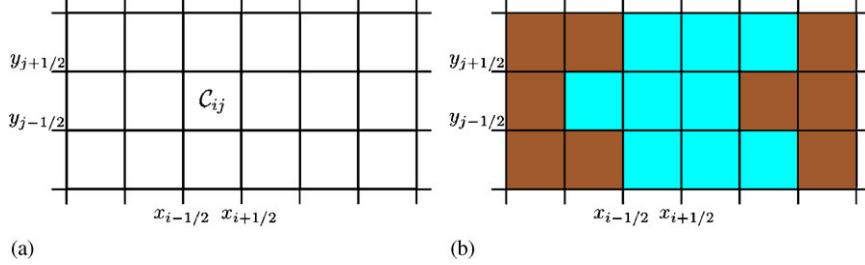


Figure 2. (a) Top view of part of a logically rectangular 2D finite volume grid. (b) Wet and dry cells in the computational domain, as would occur on a 2D finite volume grid for depth-averaged flows. Cells with water $H > 0$ are shown in a lighter shade.

surrounding cell \mathcal{C}_{ij} . For instance, at the cell interface between \mathcal{C}_{ij} and $\mathcal{C}_{i-1,j}$, at time t^n , the Riemann problem consists of the original PDEs (3) with initial conditions:

$$q(x, y, t^n) = \begin{cases} Q_{i-1,j}^n & \text{if } x \leq x_{i-1/2}, \\ Q_{ij}^n & \text{if } x > x_{i-1/2}. \end{cases} \quad (6)$$

The solution to (6) is a similarity solution, here denoted $Q_{i-1/2,j}^{n^+}(\xi)$. Using analogous notation for the solutions to the Riemann problems at each interface, the flux-differences across the waves moving into cell \mathcal{C}_{ij} are denoted by:

$$\mathcal{A}^+ \Delta Q_{i-1/2,j}^{n^+} = f^1(Q_{ij}^n) - f^1(Q_{i-1/2,j}^{n^+}(0^+)), \quad (7a)$$

$$\mathcal{A}^- \Delta Q_{i+1/2,j}^{n^+} = f^1(Q_{i+1/2,j}^{n^+}(0^-)) - f^1(Q_{ij}^n), \quad (7b)$$

$$\mathcal{B}^+ \Delta Q_{i,j-1/2}^{n^+} = f^2(Q_{ij}^n) - f^2(Q_{i,j-1/2}^{n^+}(0^+)), \quad (7c)$$

$$\mathcal{B}^- \Delta Q_{i,j+1/2}^{n^+} = f^2(Q_{i,j+1/2}^{n^+}(0^-)) - f^2(Q_{ij}^n). \quad (7d)$$

Using (7), the high-resolution wave-propagation update is

$$\begin{aligned} Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{\Delta x} (\mathcal{A}^- \Delta Q_{i+1/2,j}^{n^+} + \mathcal{A}^+ \Delta Q_{i-1/2,j}^{n^+}) - \frac{\Delta t}{\Delta y} (\mathcal{B}^- \Delta Q_{i,j-1/2}^{n^+} + \mathcal{B}^+ \Delta Q_{i,j+1/2}^{n^+}) \\ - \frac{\Delta t}{\Delta x} (F_{i+1/2,j}^{1,n^+} - F_{i-1/2,j}^{1,n^+}) - \frac{\Delta t}{\Delta y} (F_{i,j+1/2}^{2,n^+} - F_{i,j-1/2}^{2,n^+}), \end{aligned} \quad (8)$$

where the terms $F_{i\pm 1/2,j\pm 1/2}^{(1,2),n^+}$ are TVD-limited second-order correction fluxes, determined entirely by the propagating waves arising from the Riemann problem at the corresponding interface. See [5, 8] for an overview and a full description of the algorithm: here I discuss only the main features relevant to flooding with the shallow-water equations (1). The most relevant aspect of 8 for the present discussion is that the update to Q_{ij}^n is identically zero, if no waves arise from the four surrounding Riemann problems.

Note that in flood-routing problems, a rectangular computational domain inevitably contains wet ($H_{ij}^n > 0$) and dry ($H_{ij}^n = 0$) cells (Figures 1(b) and 2(b)). No explicit reference to the moving wet–dry front of the flood is used in the numerical method: the edge of the flood is not a numerical boundary, but simply where wetted cells are adjacent to dry cells at any given time step. Correctly resolving the moving wet–dry front requires careful consideration of Riemann problems between wet and dry cells, as discussed briefly in Section 3.2.

3.2. Approximate Riemann solvers for flooding

Because it can be cumbersome to solve Riemann problems exactly, Godunov schemes often make use of approximate Riemann solvers that are developed for a specific application. Although

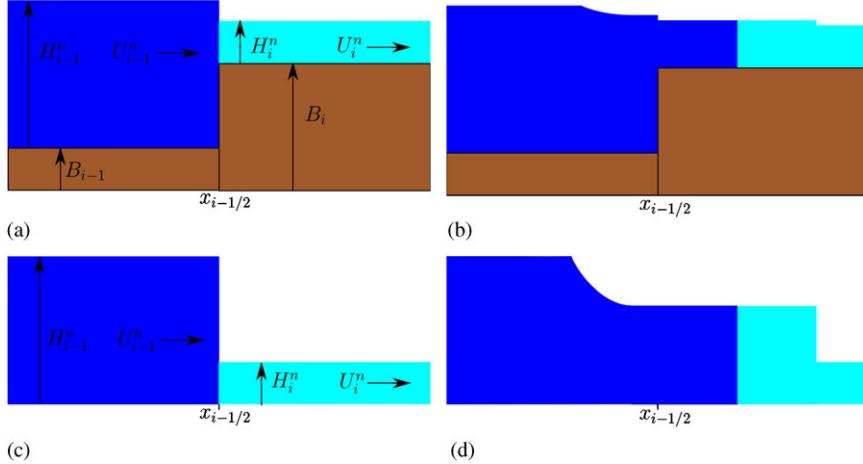


Figure 3. A Riemann problem for the shallow-water equations with topography, illustrating the error that occurs when the role of topography is treated independently from the Riemann problem: (a) An arbitrary Riemann problem with initial data at time t^n ; (b) The actual Riemann solution at $t > t^n$ has propagating waves (a small shock moving to the far right, a contact discontinuity separating the original fluids moving to the right and a rarefaction moving to the left), which represent only the deviation to the steady state; (c) The same Riemann problem, yet neglecting the source term from topography, as would be done with a fractional-step method; and (d) The solution to the Riemann problem without topography at $t > t^n$ has large propagating waves (the large shock moving to the right and large rarefaction moving to the left) producing a large error in the mass-flow that cannot precisely be undone with a separate integration of the source term, which can only add momentum in a single time step in an attempt to redirect the erroneous mass-flow. It should be apparent why this could cause spurious oscillations in the presence of large source terms.

traditional Riemann solvers for general hyperbolic equations (e.g. HLL, Roe) have frequently been used for the shallow-water equations, modeling flooding over highly variable topography introduces several complications, primarily due to the source term (the right-hand side of (1)) resulting from variable topography, as well as the moving wet-dry boundary.

3.2.1. Source terms and well-balanced schemes. It may not be immediately clear how the source term is accounted for in the numerical update (8). If the Riemann problems (6) are solved using the original governing equations (1), including the source term, then the update (8) includes the effect of the source term. When considered part of a Riemann problem, the source term can be viewed as arising from piecewise constant topography that has a step,

$$[[B]] = B_{ij} - B_{i-1,j}, \quad (9)$$

at the same location as the jump in conserved variables (Figure 3). This results in a jump in the fluxes in the similarity solution at the step:

$$f^1(Q_{i-1/2,j}^{n+}(0^-)) \neq f^1(Q_{i-1/2,j}^{n+}(0^+)), \quad (10)$$

which contributes the effect of the source term (see e.g. [15, 18] for details). Because source terms complicate the solution of Riemann problems, the traditional approach is to solve the Riemann problems using the homogeneous governing equations, and then contribute the effect of the source term in a separate integration step. While this works reasonably well for many, if not most applications, it is known to produce unacceptable errors in several well-documented situations (see, e.g. [11, 12, 15, 31]). These cases involve prevalent *balanced steady states*—where a nontrivial flux-divergence and source term offset one another:

$$\frac{\partial}{\partial x} f^1(q) + \frac{\partial}{\partial y} f^2(q) = \psi(q, x, y). \quad (11)$$

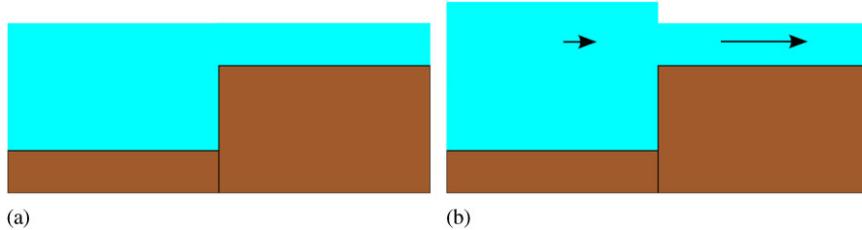


Figure 4. Steady-state Riemann problems: (a) The Riemann problem that arises from a motionless body of water over variable topography and (b) The flowing steady-state Riemann problem that arises with river-like flow, prevalent in flooding in rugged terrain. There is a jump discontinuity in the water surface at the step.

The most intuitive and commonly considered balanced steady state is the case of motionless water over variable topography governed by the shallow-water equations. For problems with highly variable topography, and flow dynamics that are small perturbations to that steady-state, typical fractional step methods produce errors that are unacceptably large, since they do not correctly preserve the balance between the source term and flux gradient [12, 14, 15, 17, 32]. Figure 3 provides an intuitive visual explanation of this shortcoming. These difficulties have led to the notion and development of *well-balanced schemes*, which are designed to preserve certain balanced steady states. (Technically, a well-balanced scheme is one that perfectly preserves an exact discontinuous weak solution for some prevalent steady state.) Most well-balanced schemes have been developed to preserve the motionless water over topography problem (e.g. [13, 16, 33]). Schemes that perfectly preserve that steady state are said to satisfy the \mathcal{C} -property [10].

For modeling floods in rugged terrain, the large source terms that appear in the presence of rapidly varying or discontinuous topography, and the importance of terrain contours in routing the flood, suggests that source terms should be treated as precisely as possible, and steady-state balances should be preserved as closely as possible. Therefore, well-balanced schemes, with respect to motionless and flowing steady states, might be important for flooding in rugged terrain. Figure 4 shows the distinct types of exact steady-state weak solutions to the shallow-water equations. For rapid flow over topography, flow dynamics are more likely to be deviations from the steady state that is depicted in panel (b) of Figure 4. The flowing steady state has a jump discontinuity in the surface elevation at the cell interface. Riemann solvers that neglect this will produce errors for near steady-state flow, and might violate energy dissipation requirements in some cases (analogous to entropy requirements in gas dynamics) [14]. It is not clear how important these errors are for most applications, however, the larger the source term is in relation to deviations from prevalent steady states, the larger the errors will be at finite grid resolutions. Furthermore, with AMR it is desirable to have steady states on relatively coarse grids, implying that well-balancing might be important. While these caveats are largely heuristic, it is hoped that more precise tests for flowing steady states over highly variable topography will be developed in the future. However, since weak solutions for such steady states are unknown in general, developing such tests will continue to be challenging.

Preserving weak solutions that represent flowing steady states is complicated by the fact that a generalized definition of the nonconservative product appearing in the source term, $-h\partial_x b$, is not easily defined at a step in topography b , because h is discontinuous there. One approach is to use a predetermined value for h in the source term, such as the arithmetic average of the two initial states in the Riemann problem, and consider the source term to be a delta function due to the standard generalized definition of $\partial_x b$ (e.g. [15, 34]). This happens to preserve the motionless steady state, but not flowing steady states [18]. The Riemann solver developed for GEOCLAW preserves jump conditions in the flux across a step in topography, for a steady-state solution, which can be shown to satisfy the following:

$$f^1(Q_{i-1/2,j}^{n+}(0^+)) - f^1(Q_{i-1/2,j}^{n+}(0^-)) = [[B]]r_0(Q_{i-1/2,j}^{n+}(0^-), Q_{i-1/2,j}^{n+}(0^+)), \quad (12)$$

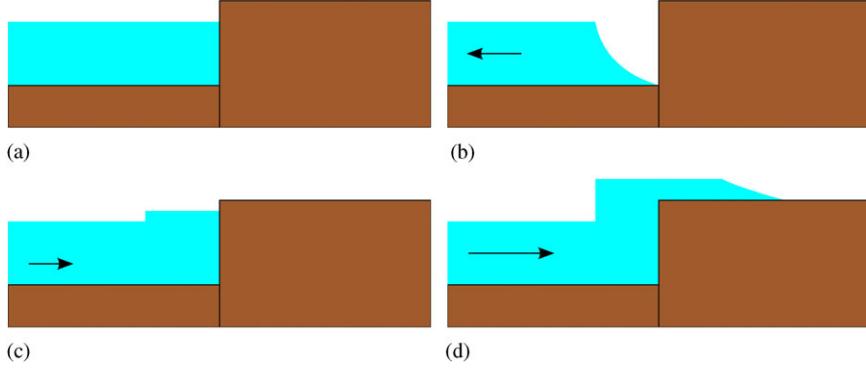


Figure 5. Riemann problems at the wet–dry boundary: (a) Stationary flow at the shoreline; (b) Retreating flow. (c) Flow with velocity that is not sufficient to inundate the adjacent dry grid cell. (d) Flow with larger velocity, inundating the dry grid cell.

where the vector $r_0(Q_{i-1/2,j}^{n+}(0^-), Q_{i-1/2,j}^{n+}(0^+)) \in \mathbb{R}^3$ is a function of the solution to the right and left of the jump in topography (see [18] for a definition and derivation of the function r_0). (This is in fact equivalent to determining the physically relevant value of the source term at an overlapping discontinuity of h and b .) The Riemann solver is ultimately equivalent to the decomposition of the initial flux difference,

$$f^1(Q_{ij}^n) - f^1(Q_{i-1,j}^n) - [[B]]r_0(Q_{ij}^n, Q_{i,j-1}^n) = \sum_{p=1}^3 \alpha_p r_p(Q_{ij}^n, Q_{i,j-1}^n), \quad (13)$$

where the predefined functions $r_p : \mathbb{R}^6 \rightarrow \mathbb{R}^3$, $p = 1, 3$, form a set of linearly independent vectors and $\alpha_p \in \mathbb{R}$ are the weights in the unique decomposition. The vectors r_p represent the propagating discontinuities in the Riemann solution, and they move at an associated wave speed s_p , $p = 1, 3$ (see [18] for details). Note that this allows the fluctuations (7a) to be found by considering the similarity solution defined by (13). If initial Riemann data satisfy (12), the *steady-state jump conditions*, then

$$f^1(Q_{ij}^n) - f^1(Q_{i-1,j}^n) - [[B]]r_0(Q_{ij}^n, Q_{i,j-1}^n) = 0, \quad (14)$$

and no propagating waves arise from 13, the update 8 is null, and the steady state is maintained as a jump discontinuity at the cell interface. Otherwise, propagating waves represent deviations to an approximate steady-state balance. See Figures 3 and 4 for visual explanations.

3.2.2. Capturing wet–dry fronts correctly. In addition to balanced steady states, flooding in rugged topography presents another difficult challenge for Riemann solvers: moving wet–dry fronts that must be computed accurately over variable topography. Again this difficulty is related to careful treatment of the source term due to topography. Figure 5 shows solutions to Riemann problems that occur at the margins of the flow between wet and dry cells. In all of the cases shown in Figure 5, use of a fractional step method would produce large mass-flow errors into the right cell, which would not be precisely undone by the integration of the source term. Furthermore, it is easy to show (e.g. [19]) that a standard discretization of the right-hand side of (1) at the shoreline will produce a non-physical source term that will not correctly resolve the waves shown in panels (a)–(c) of Figure 5. It should not be surprising that inaccurate treatment of the wet–dry boundary can cause a spurious *sloshing* effect near the margins of the flow and could lead to errors in predicting the inundation limits or flow path.

The Riemann solver used in GEOCLAW implements the following: for Riemann problems at the wet–dry interface such as those depicted in Figure 5, the physically relevant value of the source term is determined prior to solving the Riemann problem, by first solving a symmetric Riemann problem that simulates a wall boundary condition at the shoreline (i.e. if the shoreline is at $x_{i-1/2}$,

then initial data $H_i^n = H_{i-1}^n$, $U_i^n = -U_{i-1}^n$ and $B_i = B_{i-1}$ are used in a test problem). The wall boundary condition provides a means for determining whether the flow has sufficient velocity to exceed the height of the topography in the dry cell. If so, the Riemann problem is then solved like any other—the topography is taken into account and a standard treatment of the source term produces the solution depicted in Figure 3(d). Otherwise, the reflections from the wall boundary are used to update the wet cell, as depicted in Figure 3(b)–(c). Proper treatment of Riemann problems at the wet–dry interface can be seen as a generalization of well-balancing, in that true steady states (motionless shorelines) are maintained, and if there are propagating waves in a Riemann solution (retreating or advancing flow), those waves are deviations to the underlying steady-state balance. Another important property for proper treatment of the wet–dry front is the preservation of depth non-negativity [32]. Without this property, it is difficult to maintain conservation of mass at the flow front, while simultaneously determining the inundation extent. This is accomplished by the Riemann solver used in GEOCLAW, by using special wave-speed estimates s_p , $p=1, 3$, as well as bounding jump-conditions for the stationary wave—the jump discontinuity at the cell interface. For a more complete mathematical description of these issues, beyond the scope of this paper, see [18].

3.3. Adaptive mesh refinement

Although the above methods are designed for uniform rectangular grids, AMR can be used to accommodate topographically dependent flow features requiring different grid resolutions. GEOCLAW uses block-structured AMR algorithms [1, 2, 6] developed for hyperbolic systems generally, with some modifications in interpolation strategies in order to handle free-surface flows over topography (see [19] for details). Block-structured AMR allows multiple levels of rectangular sub-grids, refined from coarser grids by user-chosen pre-specified integer refinement ratios between each level. A single level-1 coarse grid represents the entire computational domain, while multiple level-2 sub-grids form rectangular patches, each of which might contain one or more level-3 sub-grids and so on. A grid cell on a level- l grid, if refined, contains $r_x^l \times r_y^l$ level- $(l+1)$ grid cells, where r_x^l and r_y^l are the refinement ratios in the x - and y -directions, respectively, between level- l and $(l+1)$ grids. See Figure 6, depicting the refinement of a single level- l grid cell, $\mathcal{C}_{i,j}^l$, with $r_x^l = r_y^l = 4$, to 16 level- $(l+1)$ grid cells, $\mathcal{C}_{k,s}^{l+1}$ etc. Typically a level- $(l+1)$ grid is not just one, but the union of many refined level- l grid cells in a rectangular arrangement.

The refinement process occurs dynamically every few time steps. Refinement is based on either error estimation techniques (see [2]) or user-selected flow criteria. For flooding problems, I have used the latter since error estimation can be expensive and physical interpretation of the solution usually makes it clear where refinement is needed (e.g. such as where flow gradients or velocities are high). (For instance, the flow criterion for a level-1 grid might simply be the existence of water. For higher level grids, the flow criteria might be threshold values of fluid velocities, or gradients in the water surface elevation.) During the refinement process, level- l grid cells that exceed flow

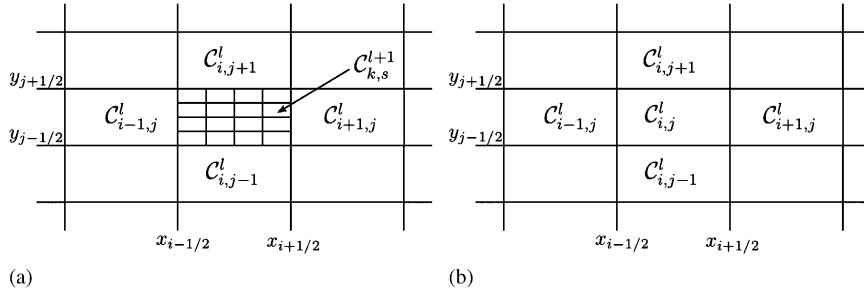


Figure 6. Refinement on a 2D grid. A level- l grid cell $\mathcal{C}_{i,j}^l$ (a) is refined to 16 level- $(l+1)$ grid cells, such as $\mathcal{C}_{k,s}^{l+1}$ (b). In this example $r_x^l = r_y^l = 4$.

criteria for that level are flagged to be refined. Flagged level- l cells and preexisting level- $(l+1)$ cells are then grouped into new level- $(l+1)$ rectangular grids, using a clustering algorithm (see [1, 2]). The solution values in new level- $(l+1)$ cells are determined by spatial interpolation from the level- l grid. For instance, the solution in $\mathcal{C}_{k,s}^{l+1}$ is determined by TVD-limited linear interpolation from the four surrounding coarse cells: $\mathcal{C}_{i\pm 1,j}^l$ and $\mathcal{C}_{i,j\pm 1}^l$ (Figure 6). Preexisting level- $(l+1)$ cells simply retain their values. A few additional cells are included around the perimeter of the new grid, which gives a buffer zone ensuring that solution features are maintained at a given level before the next refinement process. If a coarse level- l grid cell has previously been refined to a set of level- $(l+1)$ grid cells, and it no longer exceeds flow criteria for its level, then it will not be flagged again for refinement. Then, unless it is grouped into a level- $(l+1)$ grid by the clustering algorithm in order to complete a rectangle of cells, it will be de-refined. Upon de-refinement, the solution in the coarse cell \mathcal{C}_{ij}^l inherits the average value of the overlying refined level- $(l+1)$ cells.

Once the refinement process is complete the individual grids begin taking time steps. First, the coarse level-1 grid takes a single time step of size $\Delta t^{(1)}$. Next, the level-2 grids take a time step, of size

$$\Delta t^{(2)} = \Delta t^{(1)} / \max(r_x^1, r_y^1). \quad (15)$$

The level-2 grids will of course take $\max(r_x^1, r_y^1)$ steps, catching up in time to the level-1 grid. After each time step taken by the level-2 grids, the level-3 grids take $\max(r_x^2, r_y^2)$ time steps, of size,

$$\Delta t^{(3)} = \Delta t^{(2)} / \max(r_x^2, r_y^2), \quad (16)$$

and so on. That is, the time stepping is recursive—all level- l grids precede all level- $(l+1)$ grids, which must catch-up to the level- l grids before they take another time step. Once fine grids are advanced to the same point as an underlying coarser grid, a refined coarser grid cell inherits the average value of the solutions in the overlying finer cells. The coarsest grid takes several time steps (a user-selected integer) until the next grid refinement check occurs, and the entire process begins again. A detailed description of the AMR algorithms is far beyond the scope of this paper, see [1, 2] for more details.

A typical set-up for a flooding problem in rugged terrain might be to use a very coarse level-1 grid that comprises an entire region of interest, even if the flood extent will most likely represent a small fraction of the total area of the rectangular domain, such as a narrow winding river basin. If the level-1 grid is extremely coarse with respect to the grid resolution needed to model the flood, the domain can be chosen to be very large without a significant increase in computational expense. The flood itself is then resolved on higher levels of fine grids, perhaps with the highest levels in regions of the flow with steep gradients, such as the flow front. This strategy allows a patchwork of rectangular grids to follow the flow down a winding ravine or river basin with complicated geometry, with no *a priori* knowledge of where the flood will go, and never wastefully integrating fine grid cells before the flood has arrived. If the flood waves are always on higher levels of grids, which appear just before the arrival of the flood, there is no loss of accuracy due to the coarseness of the level-1 grid. Figure 7 is a cartoon depicting how refinement can be chosen to follow a flood, while dry regions (shown in a darker shade) are kept on very coarse grids. Note that the flood waters (shown in a lighter shade) are always on level-3 grids because of the buffer cells. Refinement ratios used in actual simulations are typically much higher than those shown in the cartoon.

In an AMR simulation the computational grid resolution at a given location changes with time, yet, topography or other auxiliary spatial data are typically available at some fixed preexisting set of points, such as a gridded digital elevation model (DEM). In GEOCLAW, the auxiliary topography is read into memory at the start of a simulation and the actual topographic value used in a computational grid cell is determined when the cell is initialized, either at the start of a computation or upon refinement. This value is determined by integrating the unique piecewise bilinear surface, $z = S(x, y)$, which interpolates the gridded auxiliary topographic data (see [19],

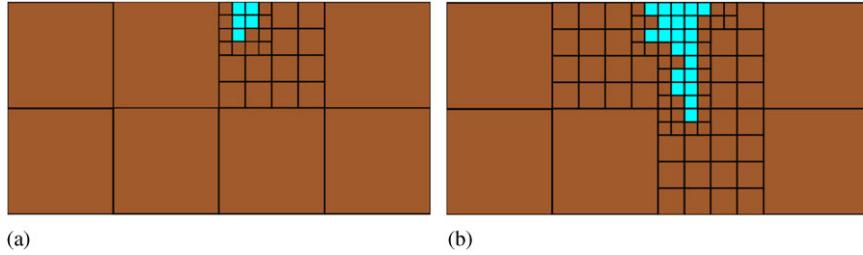


Figure 7. AMR in 2D for flooding problems (top plan view). A coarse level-1 grid exists prior to the arrival of a flood which is resolved on level-2 and level-3 grids. Panels (a) and (b) depict the flood at earlier and later times, respectively, as the water (shown in a lighter shade) advances downstream from top to bottom. Several level-3 time steps (at least 7) and at least 1 refinement process would have occurred from the time shown in panel (a) to that in (b). By the use of buffer cells, the water can be prevented from leaving level-3 grids.

(Berger *et al.*, 2009; in preparation) for more details). The computational topography value is then given by:

$$B_{ij} = \frac{1}{\lambda(\mathcal{C}_{ij})} \int_{\mathcal{C}_{ij}} S(x, y) dx dy. \quad (17)$$

This ensures that the finite volume representation of the auxiliary topography data is conservative (i.e. a discrete integral of the piecewise constant computational topography does not vary with the grid resolution or alignment). This ensures that conservation of mass is maintained during interpolation when refining. Additionally, the alignment and resolution of auxiliary topographic data are not required to be related to the computational grids.[§]

4. FLOOD SIMULATION CASE STUDY

Since it is impossible to determine exact analytical solutions to the shallow-water equations in the presence of general two-dimensional topography, validation of numerical methods for floods in rugged terrain can only be done by comparison with empirical data from real floods, or comparison to other previously validated codes.[¶] The Malpasset flood was therefore chosen as the benchmark problem for the present work due to the abundance of data, as well as the fact that the flood occurred in rugged topography where the contours of the terrain played an important role in flood routing. The suitability of the shallow-water approximation for determining flood inundation levels for the Malpasset flood has previously been supported by the conclusions of the CADAM workshop participants [23, 35]. The intended goal of this section is to test the algorithms in GEOCLAW by comparison with the results produced by other codes using topography-fit static meshes.

4.1. Historical Background of the Malpasset Dam Break

The Malpasset Dam was located in a narrow gorge above the steeply banked Reyran River Valley about 12 km north of the town of Fréjus on the French Riviera. The double arch dam was 66.5-m high and approximately 223-m wide along the top arch. For the 5 years following the completion of the dam in 1954, the reservoir behind the dam was slowly filled, nearing capacity in November 1959. During a period of intense rainfall from November 19 to December 2, the reservoir reached the crest of the dam, prompting the opening of the outlet gates at 6 P.M. on December 2. The dam failed catastrophically at 9:14 P.M., sending a massive wave 40–60-m high into the winding

[§]Of course, for modeling real problems it is desirable to have topographic data as densely spaced as the computational grid in a given region.

[¶]Validation of GEOCLAW on some simple 1D problems with analytical results can be found in [18] or [34].

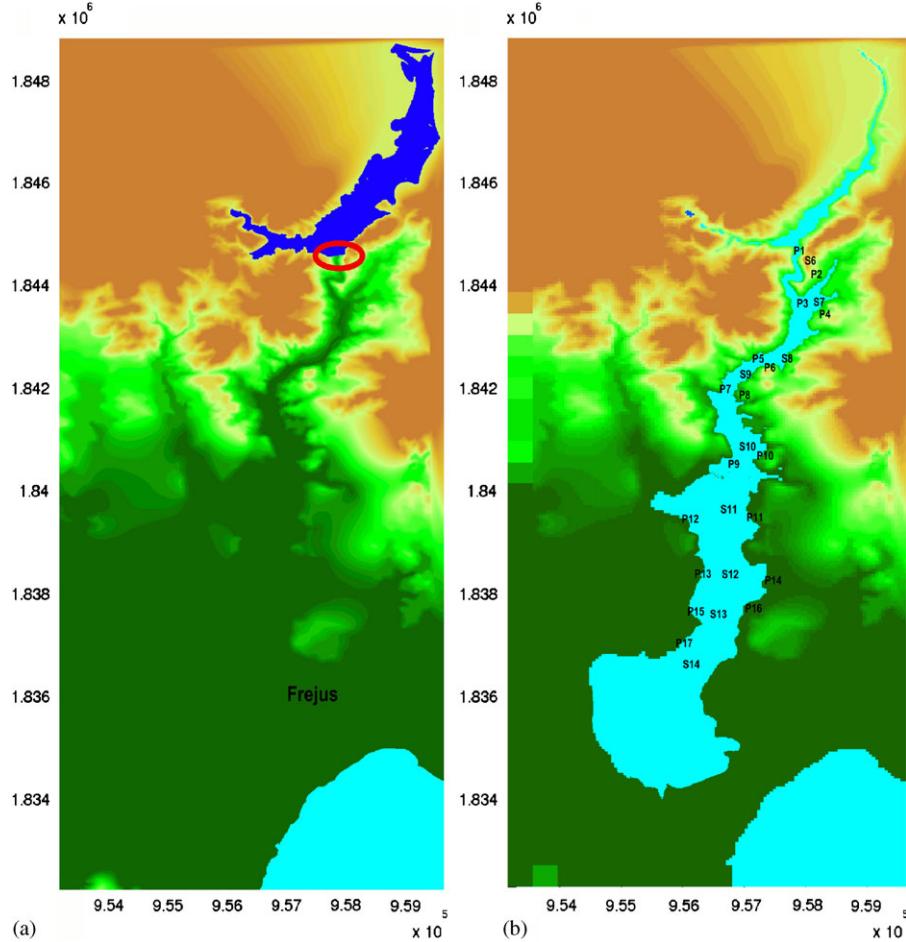


Figure 8. (a) A figure generated from the topography data, showing the domain before the initial dam break. The domain is 6.47 km east–west (horizontal axis) by 16.58 km north–south (vertical axis). The topography ranges from sealevel to over 100 m (lightest shade). The reservoir (shown in the darkest shade) has 100 m surface elevation. The circle indicates the location of the dam at the south end of the reservoir. Fréjus sits in the low-lying area just northwest of the Mediterranean Sea (the southeast corner) and (b) A figure generated from a GEOCLAW simulation, depicting the flood advancing and spreading through the Reyran River Valley and inundating parts of Fréjus. Note that the reservoir has drained considerably. The labels (P1–P17) and (S6–S14) indicate the approximate locations of the 17 police survey points and 9 physical model gauge points, respectively.

gorge below. Although there were no surviving eyewitnesses, the dam failure was believed to have been sudden and explosive [22]. The resulting violent flood wave altered the landscape and destroyed bridges, roads and several buildings along its path down the river gorge before flooding the low-lying Reyran River Valley surrounding the town of Fréjus. The flood waters finally reached the Fréjus Gulf 21 min later. There were 421 casualties.

4.2. Topography

The location of the Malpasset Dam is circled in panel (a) of Figure 8, which was produced from the topography used in the GEOCLAW simulations. The approximate free surface elevation of the reservoir behind the dam is 100 m. The winding ravine below the dam rapidly descends into the Reyran River Valley surrounding Fréjus, just above sealevel to the northwest of the Gulf of Fréjus on the Mediterranean. Figure 8(b) was produced from a GEOCLAW simulation of the flood. The flood in the simulation has inundated a significant portion of Fréjus, yet has spared other parts of the low-lying river plain.

The topography used for the GEOCLAW simulations was provided to CADAM workshop participants by the Laboratoire National d'Hydraulique (LNH), which digitized an Institut Géographique National (IGN) 1:20 000 map of Saint-Tropez n°3 developed prior to the flood (dated 1931) [35]. The topography provided was scattered data (13 541 irregularly spaced points with known elevations), which was interpolated to generate multiple overlapping regular grids prior to the GEOCLAW simulations. The exact locations and grid spacings of these rectangular grids were chosen somewhat arbitrarily, yet they were loosely based on the density of the underlying irregularly spaced data. The grid spacings chosen for the interpolated rectangular gridded data were 20 m for the entire domain, 5 m surrounding the reservoir and 2 m surrounding the ravine below the dam. These spacings are not directly related to the computational AMR grids; recall that GeoClaw reads gridded auxiliary topography into memory at the start of a computation, and the actual values used for computational topography (b in grid cells) are computed dynamically by GeoClaw as needed. Since topographic data sets are not required to match the computational grids in terms of spacing or alignment, using gridded topography is not a strict requirement of the numerical methods used in GEOCLAW, but rather is merely a convention required by the software. Little if any preprocessing of topography is required as long as the data are provided in correctly formatted DEM files where the union of the areas represented by those files comprises the entire domain.^{||}

Presumably, because topography plays an important role in flood routing, the quality of topographic data can have a significant effect on flood modeling results. Since the Reyran River Valley morphology has changed dramatically since 1959 [24], high quality modern topographic data could not be used to model the Malpasset flood. As high quality and high resolution topographic data become more widely available, from LIDAR and other technologies, the error due to poor or coarse topographic data should hopefully become less relevant with future flood modeling.

4.3. Simulation Refinement Scenarios

Various grid refinement parameters were tested in the GEOCLAW simulations of the Malpasset flood. Here the results are described for two of those simulations, referred to as the *12-m* and *3-m* simulations. For both cases the computational domain was 6.47 km east–west (*x*-direction) by 16.58 km north–south (*y*-direction), which includes the city of Fréjus and the surrounding Reyran River Valley, the Fréjus Gulf to the south, the Malpasset Dam and the entire reservoir behind the dam (Figure 8). The computational domain was similar in size to that used by the CADAM workshop participants, which might be similar to a domain chosen by someone modeling for hazard mitigation with no *a priori* knowledge of the historical extent of the flood. However, given uncertainty, a much larger computational domain could be used when using adaptive grid refinement with little additional expense due to the extreme coarseness of the level-1 grid. Both simulations presented had a level-1 grid that was 16×40 grid cells in the *x*- and *y*-directions, respectively, yielding 640 ($\approx 404.4 \text{ m} \times 414.5 \text{ m}$) level-1 grid cells. Because of its coarseness and lack of flowing water, the level-1 grid is essentially a placeholder for the potential extent of the flood with a relatively small computational expense associated with it. The level-2 grids in both simulations were refined by a factor of 8, meaning that a level-1 grid cell, if refined, contained 64 ($\approx 50 \text{ m} \times 50 \text{ m}$) grid cells. The level-2 grids bounded all of the flowing water as well as the reservoir, yet were not fine enough to resolve the actual flood waves. The flood waves were resolved on multiple evolving higher-level grids. Level-3 grids were refined by a factor of 4 yielding 16 ($\approx 12.6 \text{ m} \times 12.9 \text{ m}$) grid cells per level-2 grid cell. This was the highest level of refinement for the 12-m simulation. For the 3-m simulation, an additional 4th level was added with a refinement ratio of 4 to better refine the flowing water. This yielded 16 ($\approx 3.2 \text{ m} \times 3.2 \text{ m}$) level-4 grid cells per level-3 grid cell. Figures 9 and 10 show snapshots from the 3-m simulation.

GEOCLAW allows the user to specify various parameters controlling AMR refinement criteria prior to a simulation. For instance, one can enforce refinement by associating minimum flow criteria with a given level, such as the magnitude or the gradient of a solution variable. To accomplish the

^{||}When multiple sets (overlapping grids) of auxiliary topographic data are available at a given point, GEOCLAW automatically selects the set with the finest resolution.

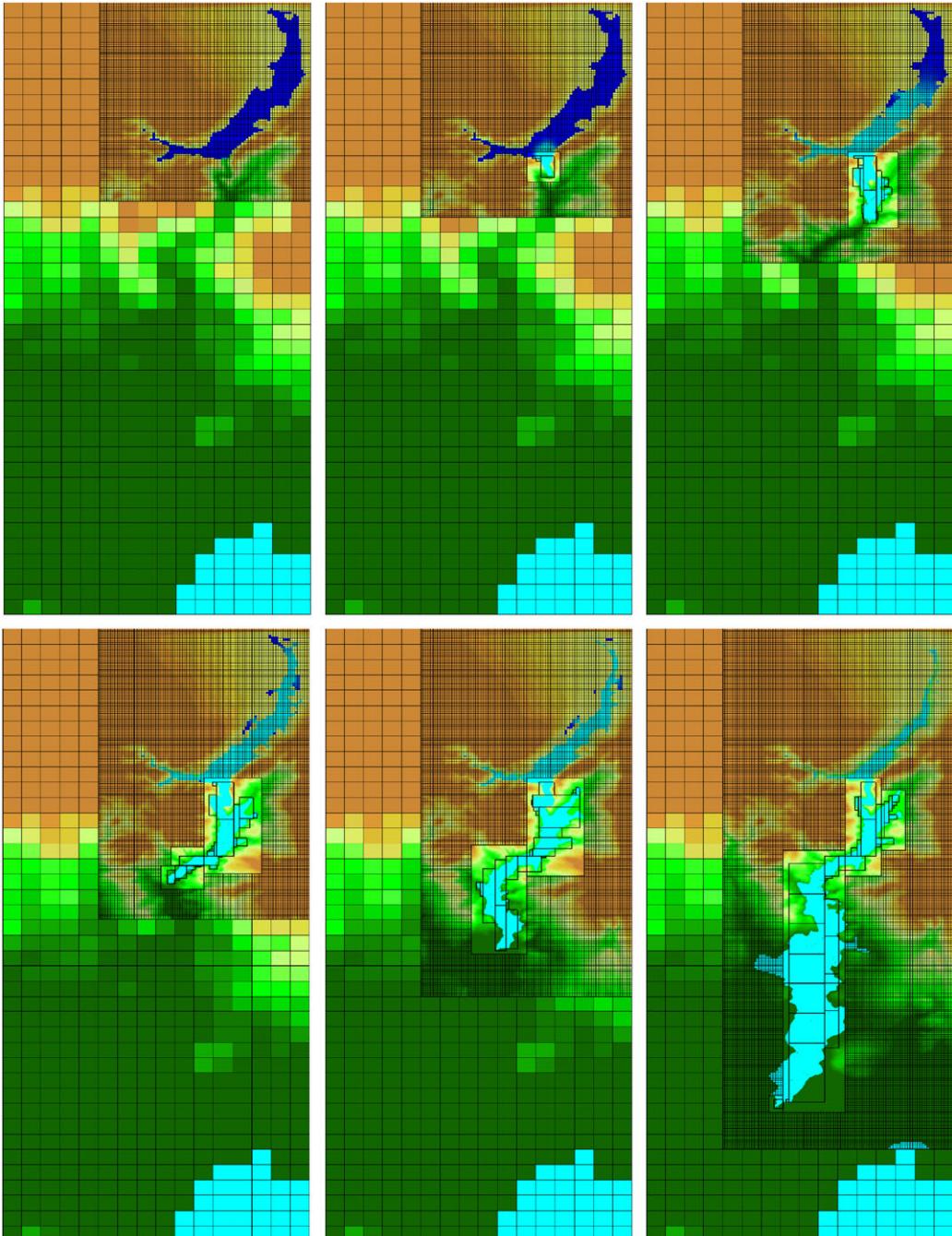


Figure 9. Snapshots of the 3-m simulation. Initially the reservoir is resolved on a level-2 grid, whereas the remainder of the domain is resolved on the coarse level-1 grid. After the initial dam break, level-3 and level-4 grids follow the flood waves as they advance down the river valley. Grid lines are omitted from grid levels 3–4 for clarity. Level-4 grids are outlined.

above refinement scenarios for the Malpasset flood, it was only necessary to establish two such criteria: wet regions with a water surface elevation greater than sealevel were refined to at least level-2, and regions with rapidly flowing water ($\sqrt{u^2 + v^2} > 0.1 \text{ m/s}$) were refined to the highest level available. This ensured that the reservoir was refined to at least level-2, providing a reasonable resolution for the mass flux leaving the reservoir, and caused refinement of the moving flood front to the highest level available (level-3 or level-4 for the 12-m or 3-m simulation, respectively). For

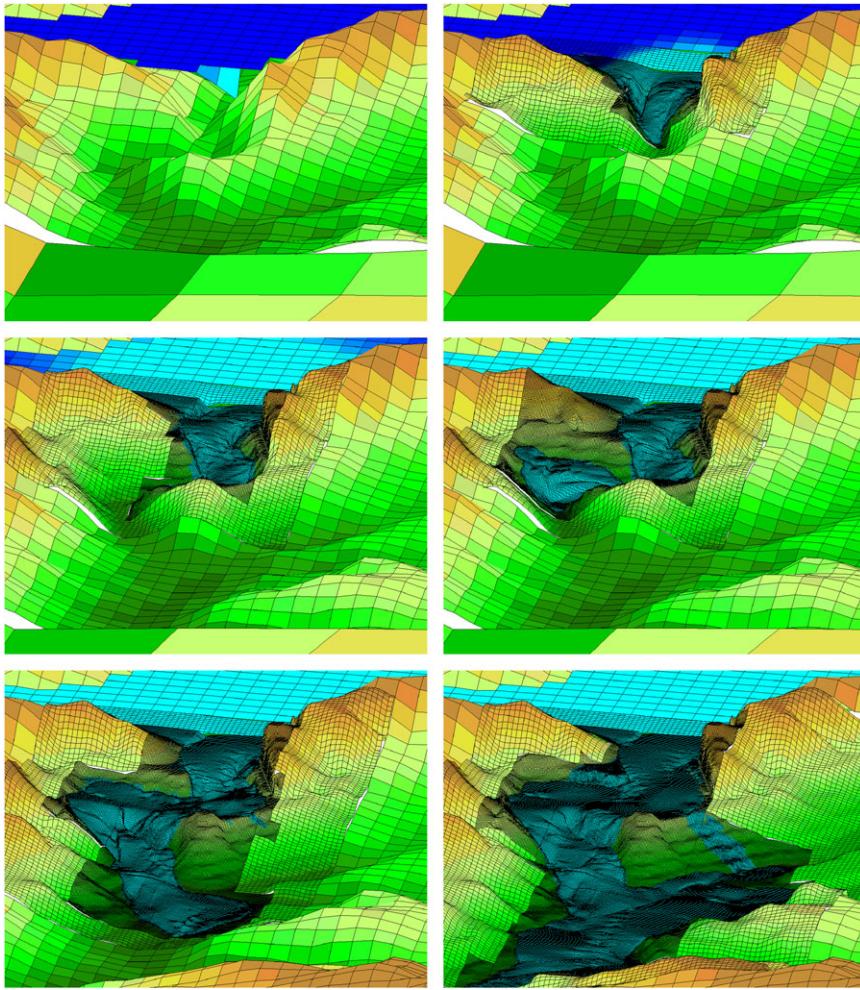


Figure 10. Close-up view of the dam break during the first 200 s of the 3-m simulation. Initially the dam and reservoir are resolved on level-2 grids and coarse level-1 grid cells can be seen in the foreground. After the initial dam break, level-3 and level-4 grids resolve the flood waves as they advance down the winding ravine.

floods where it is desired to capture specific flow features at a higher resolution, such as large gradients near the flow front, more levels and flow criteria could be used, although, this has not been fully tested at this point.

4.4. Results

The results of the GEOCLAW simulations were compared against historical survey data, physical model data and results from other numerical codes as described below. All of the simulation data compare well with the empirical data, supporting the conclusion reached at the CADAM workshops—the shallow-water equations are a reasonable model for flood simulation and risk assessment [35]. The only tunable physical parameter in the simulations was the Manning coefficient, which was chosen to be 0.033 following [24] and as suggested by CADAM participants (e.g. [35]).

4.4.1. Field survey data. After the Malpasset disaster, the high water marks downstream from the dam were measured by the Fréjus police at 17 locations. Following the researchers involved in the CADAM workshops, these points are denoted P1–P17, the approximate locations of which

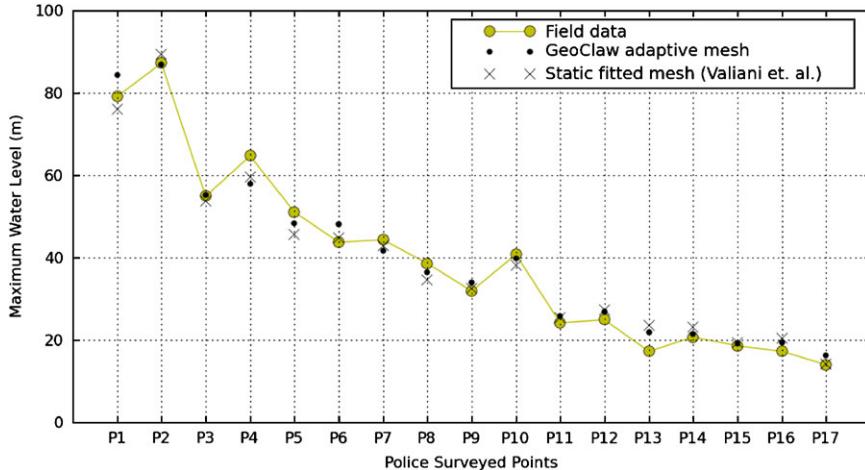


Figure 11. Comparison of simulation results with field data for the maximum water elevation at the 17 police surveyed points.

are shown in panel (b) of Figure 8. In Figure 11, the maximum water elevation at these 17 locations from the 12-m GeoClaw simulation is compared with the field data as well as with the simulation results reported by Valiani *et al.* in [24]. Valiani *et al.* tested several fixed non-uniform meshes generated specifically for the topographic data set, using the commercial software SMS™ (Surface–water Modelling System). The mesh used for the results shown in Figure 11 had 10 696 elements with linear spacing ranging from 10 to 350 m [24]. The GEOCLAW computation took just over 2 h to simulate 2800 s of the flood on a 2400 MHz Intel Core 2 Quad. The deviation of the numerical results from the field survey data is not necessarily a measure of numerical error, because there are several other unquantified sources of error—the field measurements themselves, the topographic data, and the shallow-water model. The discrepancy between the two numerical codes might be due to the different numerical schemes, or to the different grids and resulting topography values. However, the similarity between the results of the different numerical schemes suggests that AMR and rectangular grids might be a viable alternative to topographically fit meshes, which can be difficult to generate for large and irregular domains [24].

4.4.2. Physical model data. A 1:400 scale model of the region inundated by the Malpasset flood was built by Laboratoire National d’Hydraulique (LNH) of EDF in 1964. The maximum water elevations during laboratory experiments were recorded at 14 wave gauge locations. Following the notation used by the CADAM workshop participants, these gauges are designated S1–S14, the locations of which are shown in Figure 8(b). Figure 12 shows a comparison for the maximum water level between several simulations and laboratory data at nine gauges, S6–S14 (data from S1–S5, which were in the model reservoir, were not available). The results from GeoClaw’s 12-m simulation are again compared with the results reported by Valiani *et al.* [24], as well as the results reported by Hervouet and Petitjean [36], who used the commercial software package TELEMAC-2D. All three numerical simulations produced nearly identical results, which differed far more from the laboratory data than from one another. This may not be surprising since all of the numerical simulations are solving the same equations, yet the scale model experiments are, of course, not governed exactly by the shallow-water equations. It might be noted that the numerical simulation results from the different codes are more similar to one another at the gauge locations S6–S14 (Figure 12) than at the police survey points P1–P17 (Figure 11). This might be due to the fact that the police survey points were at high-water marks, and, therefore, near the margins of the inundation extent, while the gauge locations tended to be in the middle of the flood (Figure 8(b)). As suggested above, the margins of the flow are more prone to error due to the difficult task of resolving a moving wet–dry boundary.

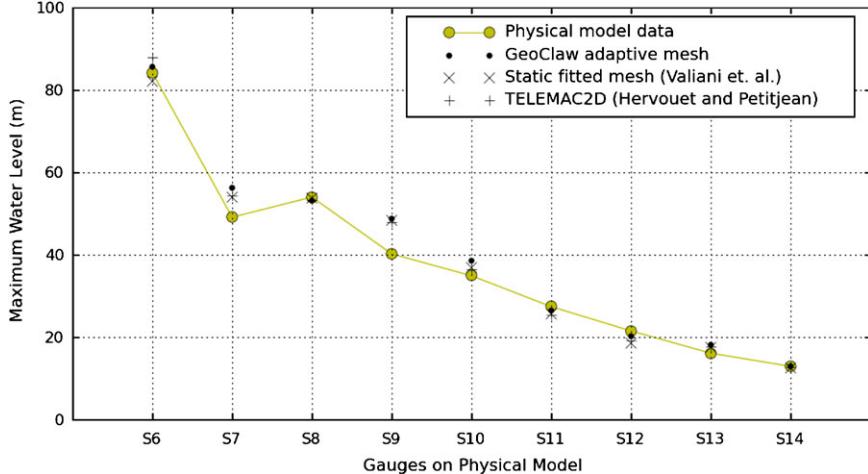


Figure 12. Comparison of simulation results with laboratory model data for the maximum water elevation at nine gauge locations.

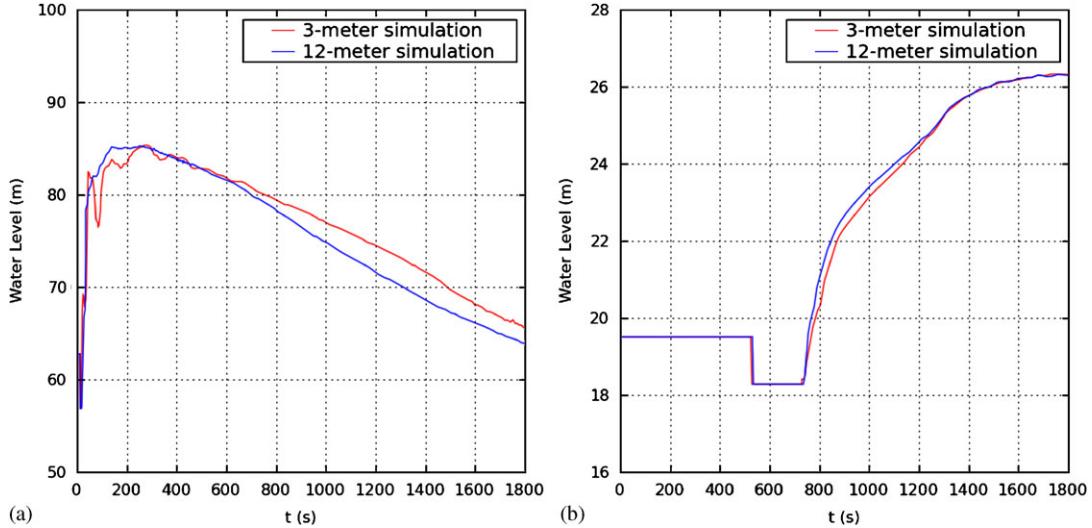


Figure 13. Time series of the water elevation for the 3-m and 12-m simulations. Left: survey point P2. Right: model gauge S11. The abrupt discontinuity at gauge S11 before the arrival of the flood is due to the changing grid resolution, which changes the average topography value over the cell. More discrepancy is seen at P2, which is near the dam and in more rugged terrain than S11.

4.4.3. Comparison between the 3-m and 12-m GEOCLAW simulations. As a final test, the results between the 3-m and 12-m GEOCLAW simulations are compared in order to roughly assess convergence. Following [24], the time series of the water level at field survey point P2 and laboratory model gauge S11 are shown in Figure 13. The differences between the 12-m simulation and 3-m simulation seem relatively small and suggest that the solution error is relatively unaffected in reducing the grid size from 12 m. The discrepancy is larger at the police survey point (left panel of 13) than at the laboratory gauge location (right panel of 13). Again, this may be due to the fact that the police survey point is near the inundation limit, and is more affected by grid resolution and prone to error. Additionally, the police survey point P2 is much closer to the dam and in much more rugged terrain than the gauge S11, and it experiences some high frequency signals as the rushing flood waves arrive that are only captured on the 3-m grid.

5. CONCLUSION

Modeling flooding problems in rugged topography introduces new complications not encountered in traditional applications in which the shallow-water equations have long been used (e.g. tsunami modeling and other long-wave-propagation problems). These added complications are primarily due to the rapid movement of wet-dry fronts over variable topography, balanced steady states with nonzero flow velocities and the difficult geometry of the flow domain.

GEOCLAW employs high-resolution well-balanced methods developed for general shallow-water flow over topography, using AMR. Because these methods employ Riemann solvers that are well-balanced with respect to steady-state balances other than the motionless steady state, it is believed that they can capture non-stationary flow in rugged terrain more accurately than methods developed for traditional shallow-water applications (e.g. tsunami modeling). However, the software has not previously been tested for modeling floods in rugged terrain. Because of the important role of topography, it was not clear if simple rectangular grids could produce solutions as accurately as topography-fit meshes.

In this paper, the use of GEOCLAW for flood modeling was tested by simulating the flood from the Malpasset Dam break. The results of the GEOCLAW simulations are compared with field-data, laboratory model data, and numerical data produced by other codes on topography-fit meshes. Although this is the first validation of this code for dam-break flooding problems, these preliminary results suggest that these methods are a viable alternative to using specially developed meshes with an enormous efficiency advantage in terms of computational cost without a known loss in accuracy. This approach also prevents the need for mesh generation or knowledge of the flood geometry *a priori*, which might also confer an advantage in the event of rapid risk mitigation modeling.

In addition to modeling floods in rugged terrain with the shallow-water equations, the methods employed in GEOCLAW might be applicable to more general flooding in rugged terrain by solving other sets of depth-averaged equations, such as the depth-averaged equations for landslides and debris flows described in [37]. These applications present many of the same difficulties, such as the geometry of the flow domain as well as steady-state balances due to source terms (see, e.g. [25]).

ACKNOWLEDGEMENTS

The author thanks the other developers of GEOCLAW, in particular Randall LeVeque at the University of Washington and Marsha Berger at the Courant Institute NYU. Additional gratitude is due to Willie Scott, Larry Mastin, Roger Denlinger and Richard Iverson at the U.S. Geological Survey, for comments and suggestions, as well as Nicole Goutal of the National d'Hydraulique, EDF, France, for providing assistance and data.

REFERENCES

1. Berger MJ, Colella P. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics* 1989; **82**:64–84.
2. Berger MJ, Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics* 1984; **53**:484–512.
3. Borthwick A, Cruz-León S, Józsa J. The shallow flow equations solved on adaptive quadtree grids. *International Journal for Numerical Methods in Fluids* 2001; **37**:691–719.
4. Liang Q, Borthwick A, Stelling G. Simulation of dam- and dyke-break hydrodynamics on dynamically adaptive quadtree grids. *International Journal for Numerical Methods in Fluids* 2004; **46**:127–162.
5. LeVeque RJ. Wave propagation algorithms for multi-dimensional hyperbolic systems. *Journal of Computational Physics* 1997; **131**:327–335.
6. Berger MJ, LeVeque RJ. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM Journal on Numerical Analysis* 1998; **35**:346–365.
7. Godlewski E, Raviart P. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer: New York, 1996.
8. LeVeque RJ. *Finite Volume Methods for Hyperbolic Problems*. Texts in Applied Mathematics. Cambridge University Press: Cambridge, 2002.
9. Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer: Berlin, 1997.

10. Bermúdez A, Vásquez M. Upwind methods for hyperbolic conservation laws with source terms. *Computers and Fluids* 1994; **23**(8):1049–1071.
11. Greenberg JM, LeRoux AY. A well-balanced scheme for numerical processing of source terms in hyperbolic equations. *SIAM Journal on Numerical Analysis* 1996; **33**:1–16.
12. LeVeque RJ. Balancing source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm. *Journal of Computational Physics* 1998; **146**:346–365.
13. Zhou JG, Causon DM, Mingham CG, Ingram DM. The surface gradient method for the treatment of source terms in the shallow-water equations. *Journal of Computational Physics* 2001; **168**:1–25.
14. Bouchut F. *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well-balanced Schemes for Sources*. Birkhäuser: Basel, 2004.
15. Bale DS, LeVeque RJ, Mitran S, Rossmanith JA. A wave propagation method for conservation laws and balance laws with spatially varying flux functions. *SIAM Journal on Scientific Computing* 2002; **24**:955–978.
16. Audeusse E, Bouchut F, Bristeau MO, Klein R, Perthame B. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing* 2004; **25**(6):2050–2065.
17. García-Navarro P, Vázquez-Cédon ME. On numerical treatment of the source terms in the shallow water equations. *Computers and Fluids* 2000; **29**:17–45.
18. George DL. Augmented Riemann solvers for the shallow water equations over variable topography with steady states and inundation. *Journal of Computational Physics* 2008; **227**(6):3089–3113. DOI: 10.1016/j.jcp.2007.10.027.
19. George DL. Finite volume methods and adaptive refinement for tsunami propagation and inundation. *Ph.D. Thesis*, University of Washington, 2006.
20. George DL, LeVeque RJ. Finite volume methods and adaptive refinement for global tsunami propagation and local inundation. *Science of Tsunami Hazards* 2006; **24**(5):319–328.
21. George DL, LeVeque RJ. High-resolution methods and adaptive refinement for tsunami propagation and inundation. In *Hyperbolic Problems: Theory, Numerics, Applications*, Benzoni-Gavage S, Serre D (eds); *Proceedings of the 11th International Conference on Hyperbolic Problems*, Lyon, France, July 2006. Springer: Berlin, 2008; 541–549.
22. Johnstone WM, Assaf H, Sakamoto D, Hartford D. Analysis of the Malpasset Dam failure using GIS and engineering models. *GeoTec 2003 Symposium*, Vancouver, BC, 16–29 March 2003.
23. Morris M (ed.). *Proceedings, 1st CADAM Meeting*. HR Wallingford: U.K., 1998.
24. Valiani A, Caleffi V, Zanni A. Case study: Malpasset dam-break simulation using a two-dimensional finite volume method. *Journal of Hydraulic Engineering* 2002; **128**(5):460–472.
25. Denlinger RP, Iverson RM. Granular avalanches across irregular three-dimensional terrain: 1. Theory and computation. *Journal of Geophysical Research* 2004; **109**:F01 014. DOI: 10.1029/2003JF000085.
26. Sainte-Marie J, Bristeau M. Derivation of a non-hydrostatic shallow water model; comparison with Saint-Venant and Boussinesq systems. *Discrete and Continuous Dynamical Systems (B)* 2008; **10**(4):733–759.
27. Yamazaki Y, Kowalik Z, Cheung K. Depth-integrated, non-hydrostatic model for wave breaking and run-up. *International Journal for Numerical Methods in Fluids* 2008; **61**:473–497.
28. Savage SB, Hutter K. The motion of a finite mass of granular material down a rough incline. *Journal of Fluid Mechanics* 1989; **199**:177–215.
29. Iverson RM. The physics of debris flows. *Reviews of Geophysics* 1997; **35**(3):245–296.
30. Godunov SK. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskiĭ Sbornik* 1959; **47**:271–306.
31. Murillo J, García-Navarro PJ, Burguete J, Brufau P. The influence of source terms on stability, accuracy and conservation in two-dimensional shallow flow simulation using triangular finite volumes. *International Journal for Numerical Methods in Fluids* 2007; **54**:543–590. DOI: 10.1002/fld.1417.
32. Toro EF. *Shock Capturing Methods for Free Surface Shallow Flows*. Wiley: Chichester, U.K., 2001.
33. Burguete J, García-Navarro P. Efficient construction of high-resolution tvd conservative schemes for equations with source terms: application to shallow water flows. *International Journal for Numerical Methods in Fluids* 2001; **37**:209–248. DOI: 10.1002/fld.175.
34. LeVeque RJ, George DL. High-resolution finite volume methods for the shallow water equations with bathymetry and dry states. In *Advanced Numerical Models for Simulating Tsunami Waves and Runup*, Liu PL, Synolakis C, Yeh H (eds). Advances in Coastal and Ocean Engineering, vol. 10. World Scientific: Singapore, 2008; 43–73. URL <http://www.amath.washington.edu/rjl/pubs/catalina04/>.
35. Alcrudo F (ed.). *Proceedings of the 4th CADAM Meeting*. CADAM: Zaragoza, Spain, 1999.
36. Hervouet JM, Petitjean A. Malpasset dam break revisited with two-dimensional computations. *Journal of Hydraulic Research* 1999; **37**(6):777–788.
37. Iverson R. *Elements of an Improved Model of Debris-flow Motion*, American Physical Society, 2009.