

Article

Not peer-reviewed version

GeoFlood: Computational Model for Overland Flooding

[Brian Kyanjo](#)*, [Donna Calhoun](#), David L. George

Posted Date: 29 April 2025

doi: 10.20944/preprints202403.1029.v2

Keywords: Adaptive Mesh Refinement; GeoFlood; p4est; Flood modeling



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

GeoFlood: Computational Model for Overland Flooding

Brian Kyanjo ^{1,*}, Donna Calhoun ¹ and David L. George ²

¹ Department of Mathematics, Boise State University, Boise, ID, 83706, USA

² U.S. Geological Survey, Cascades Volcano Observatory, Vancouver, WA, USA

* Correspondence: briankyanjo@u.boisestate.edu

Abstract: This paper presents GeoFlood, a new open-source software package for solving the shallow-water equations (SWE) on a quadtree hierarchy of mapped, logically Cartesian grids managed by the parallel, adaptive library ForestClaw [1]. The GeoFlood model is validated using standard benchmark tests from [2] as well as the historical Malpasset dam failure. The benchmark test results are compared against those we obtained from GeoClaw [3] and the software package HEC-RAS (Hydraulic Engineering Center - River Analysis System, Army Corp of Engineers) [4]. The Malpasset outburst flood results are compared with those presented in [5] (obtained from the GeoClaw software), model results from Hervouet and Petitjean [6], and empirical data. The comparisons validate GeoFlood's capabilities for idealized benchmarks compared to other commonly used models as well as its ability to efficiently simulate highly dynamic floods in complex terrain, consistent with historical field data. Because it is massively parallel and scalable, GeoFlood may be a valuable tool for efficiently computing large-scale flooding problems at very high resolutions.

Keywords: adaptive mesh refinement; GeoFlood; p4est; flood modeling

1. Introduction

Overland flooding simulation is critical to society for hazard mitigation because, among natural hazards, flooding is a leading cause of casualties and property damage. Examples include the 1959 Malpasset flood, caused by the Malpasset dam failure, which killed at least 423 people, injured 83 and caused nearly 103.5 million francs in damages [7]; the 1993 Mississippi River floods resulting from extreme weather and hydrologic conditions killed at least 47 people and caused nearly 20 billion dollars in damages [8]; the 2013 Colorado floods triggered by heavy rains resulting from a slow cold front colliding with warm humid monsoonal air killed at least 9 people and cost an estimated 4 billion dollars in damages [9]. The potential for future devastating floods caused by the failure of still-operational dams remains. For instance, a failure of the Mosul dam in Iraq, considered the most dangerous dam in the world, could cause a catastrophic flood affecting millions of people and costing billions of dollars in damages [10]. Simulating the inundation extent and timing of these potential events aids the engineering of mitigation strategies in emergency planning and infrastructure development.

Numerical simulation of advancing water over topography is a powerful tool for understanding and predicting the behavior of overland flooding in complex environments. However, floods occur on large spatial domains over many hours and must be represented with suitable yet tractable mathematical models. Historically, researchers have utilized one-dimensional (1D) channel-flow models and, more recently, the two-dimensional (2D) shallow-water equations (SWE)—a system of hyperbolic partial differential equations (PDEs) for depth-averaged conservation of mass and momentum. While solving a full three dimensional model of overland flooding might capture more flow detail, computationally efficient and robust models for the SWE are capable of handling overland flows in complex terrain. Nevertheless, developing numerical models based on SWE remains a challenging problem with an active research community. In the last few decades, shock-capturing finite-volume methods

have become the dominant class of numerical schemes for this problem, due to their accurate and robust nature for hyperbolic problems. In addition to the codes described in this paper, researchers have developed a variety of such schemes solved on a variety of mesh structures; for example, [11] designed a classical Godunov scheme to solve the SWE on a static fitted mesh, [12] presented an open-source Saint-Venant model (a 2D finite volume solver) for solving SWE on adaptively refined meshes. In addition, [13] created a 2D shallow-water equation model using finite volume techniques to model overland flows in rural and urban areas. More recently, [14], [15], and [16] developed similar SWE-based models for modeling overland flow.

In this paper, we present GeoFlood, a new computational model that employs the wave-propagation algorithms (WPA) utilized in Clawpack [3], the augmented Riemann solvers available in the software GeoClaw [17], and the parallel, adaptive library ForestClaw [1], to solve the shallow-water equations on mapped, logically Cartesian adaptive meshes (??).

GeoFlood demonstrates the ability and benefits of simulating overland flows using a parallel tree-based, adaptive mesh refinement (AMR) structure, verified through comparisons with GeoClaw and HEC-RAS (Hydraulic Engineering Center - River Analysis System) results for some standard benchmark problems [2]. The model is also validated against [5] GeoClaw results for the Malpasset dam break, which provides an ideal test case for the model because it involves nearly instantaneous dam-break initial conditions (the dam collapsed suddenly and catastrophically) followed by downstream flow through complex irregular terrain (??). This instantaneous, catastrophic event is a well-known benchmark problem in the field of overland flooding due in part to the existence of extensive downstream field data, including timing information.

2. Software Overview

In this section we summarize the software packages and libraries that provide components from which GeoFlood is built. We also briefly describe other overland flood modeling packages (GeoClaw and HEC-RAS) that we used for comparison and validation.

2.1. GeoFlood's Fundamental Building Libraries

GeoFlood incorporates components from other open-source software projects along with new code that integrates these diverse libraries. This section provides an overview of these previously developed codes. Section 2.2 provides a broad overview of the standalone package, GeoFlood, including novel features that facilitate simulating and analyzing results for overland flooding applications.

2.1.1. Clawpack and GeoClaw

The GeoClaw software is a submodule of Clawpack [3], an open-source software package for solving general hyperbolic systems of PDEs. GeoClaw was initially developed as an extension to Clawpack for tsunami modeling [18], but has since been extended to overland flooding problems [5] and hurricane-generated storm surges [19]. GeoClaw combines finite-volume wave-propagation algorithms in Clawpack [20], Riemann solvers for shallow-water wave equations [17], patch-based adaptive mesh refinement (AMR) schemes [21,22] with interpolation schemes developed for free-surface flows over topography, and methods for integrating general sets (overlapping, misaligned, etc.) of digital elevation models (DEMs) in a manner that preserves volume conservation in the adaptive setting. See George [18], Leveque et al. [23] for details. Clawpack and GeoClaw have undergone extensive development over the past several decades [24] and are actively maintained by the Clawpack development team.

2.1.2. p4est

The p4est code is a robust library for the parallel computation of adaptive, hierarchical tree meshes, including efficient parallel algorithms for creating, refining, and distributing those meshes. The p4est mesh-management library distributes a quadtree or octree mesh across multiple processors using a Message-Passing Interface (MPI), providing a scalable and fault-tolerant framework for large-scale

simulations. It is designed to be compatible with various parallel-computing architectures and can scale to millions of processor cores [25].

2.1.3. ForestClaw

The ForestClaw library is built as a PDE layer on top of the p4est library for parallel tree-mesh management. While it can be used with any Cartesian-based patch solver, ForestClaw makes extensive use of the wave-propagation algorithms in Clawpack for solving a variety of hyperbolic problems. The resulting ForestClaw library is an adaptive, parallel, multi-block structured finite volume code that parallelizes the solution of hyperbolic PDEs on mapped, logically Cartesian meshes [1]. The GeoClaw extension of ForestClaw incorporates the SWE solvers and AMR interpolation schemes from GeoClaw and is expected to produce numerically consistent results, but on tree-based meshes. This GeoClaw extension serves as the basis for the new standalone code called GeoFlood, the software package that is the focus of this paper.

2.2. GeoFlood Overview

GeoFlood is a standalone package that uses the ForestClaw library, the Riemann solvers in GeoClaw, and other features tailored toward modeling problems in overland flooding. GeoFlood offers several advantages over GeoClaw and introduces enhancements to ForestClaw targeted toward improved overland flood modeling. Like GeoClaw, GeoFlood can restrict and optimize grid refinement to user-specified spatial and temporal regions of interest. A key advantage of GeoFlood over GeoClaw, however, is that it can be run efficiently on large distributed parallel platforms. The tree-based communication patterns inherited from ForestClaw and p4est allow for simplified load balancing and a decentralized, distributed regridding algorithm. The multi-resolution grid hierarchy in a typical GeoFlood mesh is composed of composite structures of non-overlapping fixed-sized grids, each stored as a block in a quadtree multi-block forest. This allows for easy data storage on each processor and for fast neighbor searches. The Cartesian grid layout of each patch in a quadrant simplifies communication among patches.

GeoFlood is written in C, C++ and Fortran and makes use of Python scripts available in Clawpack and GeoClaw for providing input parameters. The build system is managed using CMake, which facilitates the set-up of general overland flooding problems. This design offers the flexibility to integrate alternative Riemann solvers, other numerical methods, or custom scripts for a specific application. Python scripts are provided that can download and handle topography files, specific problem parameters, retrieve initial conditions, and generate a GeoFlood configuration file from user-defined settings and inputs. The configuration file is then read by GeoFlood via the command line while running in serial or parallel mode. Documentation on the installation and operation of GeoFlood is provided on the GeoFlood Wiki [26].

The GeoFlood model can generate a series of frames of the simulation domain in Cartesian or latitude and longitude coordinates. A Python script reads the output frames and generates Keyhole Markup Language [27] files that can be read in the Google Earth browser. In cases where the computational domain coordinates are not latitude and longitude, an automated Python routine has been designed to read user-specified ground control points within the domain and georeference them to a latitude and longitude coordinate system, even when the coordinate projection is unknown (??). This allows the GeoFlood simulation frames to be visualized on Google Earth.

2.3. Software for Comparison with GeoFlood: GeoClaw and HEC-RAS

We used GeoClaw and HEC-RAS for validation of GeoFlood on several idealized benchmark problems and a field-scale flood.

Although most commonly used for tsunami simulations, GeoClaw has been used for outburst-flood simulations that utilize AMR to dynamically resolve the evolving flood domain. For example, [5] simulated the outburst flood resulting from the 1959 Malpasset dam failure. The results for high-water marks and flood arrival times at nine gauge locations were validated against field data and

experimental results from a scaled laboratory model [28,29]. [30] further tested GeoClaw's overland flooding capabilities by extending it to the simulation of the 1976 Teton dam rupture. The results were in agreement with historical observations as well as HEC-RAS simulations of the same event.

HEC-RAS consists of 1D and 2D hydraulic modeling software developed by the U.S. Army Corps of Engineers [31]. It utilizes a variety of numerical schemes for different applications. The model we test against uses an implicit finite-volume method to solve the shallow-water wave equations on uniformly structured grids and is capable of modeling steady or unsteady flows, and sediment transport in complex terrain. HEC-RAS is widely used in government and industry for levee breach analysis and is considered the industry standard for floodplain modeling [4]. It has been used extensively to simulate dam-break floods, including the 1976 Teton dam failure [30], the 2006 Ukai dam failure [32], and the Temenggor dam failure [33]. In this paper, we compare GeoFlood results with those computed using the Eulerian-Lagrangian SWE solver (SWE-ELM) available in HEC-RAS 6.3.1.

3. Governing Models And Numerical Algorithms

3.1. The Shallow-Water Equations

The dynamics of floods in rugged terrain varies in three dimensions (3D); however, the depth-averaged (2D) shallow water equations are widely considered to be a suitable and tractable approximation for determining the extent and timing of a flood scenario for the assessment of hazards [e.g., 5,34–36]. These equations are derived by integrating the 3D Euler (inviscid Navier-Stokes) equations over the vertical z -direction from the solid bed to the free surface of the flow and applying kinematic boundary conditions at these surfaces. When combined with the simplifying assumption of vertically hydrostatic pressure (by neglecting higher-order terms in the asymptotic expansion of the vertical momentum equations under the long-wave or shallowness assumption [e.g., 37]), this leads to the nonlinear shallow water equations (SWE) for conservation of mass and momentum.

The governing equations used in GeoFlood are a SWE system of hyperbolic PDEs given by

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

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

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

where $h(x, y, t)$ is the water depth, $u(x, y, t)$ and $v(x, y, t)$ are the depth-averaged horizontal velocities in the x and y directions respectively, g is the gravitational acceleration, S_{fx} and S_{fy} are the friction slopes in the x and y directions, respectively, and $b(x, y)$ is the bed elevation. The friction slopes are commonly obtained from the empirical resistance relationships in the Manning equations [e.g., 38] which are given by

$$\begin{aligned} S_{fx} &= n^2 g u h^{-4/3} \sqrt{u^2 + v^2}, \\ S_{fy} &= n^2 g v h^{-4/3} \sqrt{u^2 + v^2}, \end{aligned} \quad (2)$$

where n is Manning's roughness coefficient depicting the roughness of the bed surface.

3.2. Finite-Volume Discretizations

Finite-volume discretizations are widely used for hyperbolic systems generally [e.g., 20] and overland flood modeling in particular because they provide a framework that is robust in the presence of drying regions, can capture discontinuities (weak solutions to equations (1)) such as hydraulic bores or non-smooth topography, can be made well-balanced with respect to near-steady flows, and can resolve the inundating shoreline and run-up features; see for instance [5,17], [39], [40,41], [42], and [43].

In one space dimension, the SWE can be expressed in compact conservative form as

$$q_t + f(q)_x = \Psi(x, q), \quad (3)$$

where the vector q represents conserved quantities $q = [h, hu]^T$, $f(q)$ represents the corresponding fluxes,

$$f(q) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix}, \quad (4)$$

and $\Psi(x, q)$ includes bathymetry and friction source terms. The system (3) can be expressed in quasi-linear form as,

$$q_t + A(q)q_x = \Psi(x, q), \quad (5)$$

where $A(q)$ is the flux-Jacobian given by

$$A(q) := f'(q) = \begin{bmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{bmatrix}. \quad (6)$$

Hyperbolicity of the 1D shallow-water equations is conferred by the eigenstructure of $A(q)$, as its eigenvalues (*i.e.*, wave speeds, $u \pm \sqrt{gh}$), are real and distinct for $h > 0$.

Consider $C_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ to be the i^{th} grid cell, the average value over the i^{th} cell at time t_n is given by Equation (7).

$$Q_i^n \approx \frac{1}{\Delta x} \int_{C_i} q(x, t_n) dx, \quad (7)$$

where Δx is the cell size. The vector q represents an exact spatially-varying cell solution at time t_n . The wave-propagation algorithm [20,44] updates the numerical solution from Q_i^n to Q_i^{n+1} by solving Riemann problems at the boundaries of cell C_i and directly re-averaging the resulting waves onto adjacent grid cells.

The solution update, neglecting the source terms, can be accomplished using the first-order method of the form

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left(\mathcal{A}^- \Delta Q_{i+\frac{1}{2}}^n + \mathcal{A}^+ \Delta Q_{i-\frac{1}{2}}^n \right), \quad (8)$$

where $\Delta Q_{i+\frac{1}{2}}^n = Q_i^n - Q_{i-1}^n$, and the fluctuations (flux-differences) $\mathcal{A}^- \Delta Q_{i+\frac{1}{2}}^n$ and $\mathcal{A}^+ \Delta Q_{i-\frac{1}{2}}^n$ represent propagating waves (jump-discontinuities) carrying units of flux. These waves contribute the net effect of all left- and right-going waves propagating into the cell C_i from its right and left boundaries respectively. Additional higher-order correction terms can be added to the wave-propagation method (8) to achieve second-order accuracy in smooth regions:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left(\mathcal{A}^- \Delta Q_{i+\frac{1}{2}}^n + \mathcal{A}^+ \Delta Q_{i-\frac{1}{2}}^n \right) - \frac{\Delta t}{\Delta x} \left(\tilde{F}_{i+\frac{1}{2}}^n - \tilde{F}_{i-\frac{1}{2}}^n \right). \quad (9)$$

The second-order correction fluxes $\tilde{F}_{i\pm\frac{1}{2}}^n$ are obtained from the Riemann solutions and are adjusted using limiters that ensure a total-variation diminishing (TVD) update, preserving the first-order method's convergence to discontinuous weak solutions satisfying the integral form of the system (3) (*i.e.*, shock-capturing). The effect of the source term, $\Psi(x, q)$, is traditionally integrated separately by using a splitting or fractional-step method, in conjunction with the homogeneous update (9), or is incorporated directly into Riemann solutions in various ways to achieve well-balancing [e.g., 17,20,45]. For details on the extension of (9) to two dimensions, see [20].

3.3. Augmented Riemann Solver

Modeling flooding extent in highly variable and irregular topography is challenging due to the need to numerically balance large flux gradients (from variable depth) and source terms resulting from variable topography. The problem is further complicated by the time-varying solution domain

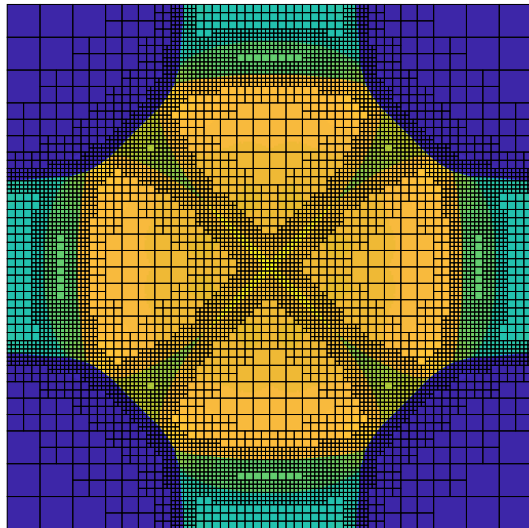
caused by moving wet-dry boundaries. The GeoFlood code employs an approximate Riemann solver developed by George [17] that solves an augmented SWE system that includes the momentum flux and topographic bed (b) as state variables in order to determine stationary and propagating waves in the approximate Riemann solution. It provides a numerical update similar to the f-wave formulation of the wave-propagation algorithm [20,45], where the effect of the topographic source term (the right side of the shallow-water equations (1)) is incorporated into the Riemann solution and, consequently, into (9). This approach eliminates the need for a poorly balanced fractional-step treatment of the source term. The Riemann solver provides well-balanced resolution of pools or steady-flow—a numerically challenging but common flow condition for overland flooding over topography. The augmented solver employs a wave-speed estimate formulation that reduces to the HLLE (Harten, Lax, van Leer, and Einfeldt) wave-speed estimates [46,47] in interior (wet) regions, ensuring entropy-satisfying depth positivity and accurate shock capturing based on Roe averages [48]. The wave-speeds reduce to those of dam-break front speeds for moving wet-dry boundaries. Additionally, the solver handles flows against structures or highly irregular topography by solving artificial Riemann test problems (*i.e.*, problems in which the data in one cell are manipulated to determine the physically relevant solution in a neighboring cell) (see George [5,17], LeVeque [20]).

Coupling these capabilities with block-structured AMR capabilities provided by the ForestClaw library gives GeoFlood the ability to robustly handle situations with complex topography and abruptly moving wet-dry fronts in simulating overland flows.

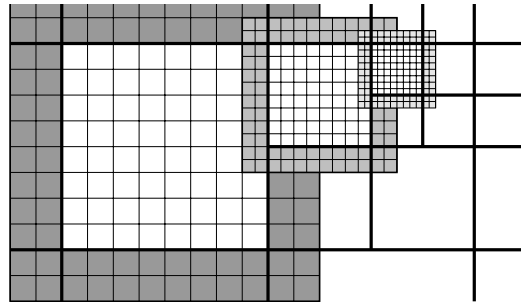
4. Adaptive Mesh Refinement Using Quadtree Meshing

The multi-resolution grid hierarchy in ForestClaw is a composite structure of non-overlapping fixed-sized grids (e.g., 32×32), each stored as a leaf in a quad or octree forest. The multi-block structure available in the forest-of-trees paradigm inherited from p4est makes it possible to solve problems on, for example, the cubed-sphere.

The refinement strategy uses a constant refinement scale factor of 2, ensuring that progression through levels is sequential (e.g., Level 0 to Level 1 to Level 2). Skipping levels is not permitted. For a grid block of fixed size $m_x \times m_y$ (dimensions in the x and y directions) within a $m_i \times m_j$ arrangement of blocks (e.g., trees) the total grid size at the coarsest level (Level 0) is $(m_i \cdot m_x) \times (m_j \cdot m_y)$. The effective resolution (e.g., total number of cells in a uniformly refined grid) at refinement level l is $(m_i \cdot m_x \cdot 2^l) \times (m_j \cdot m_y \cdot 2^l)$. At the finest level of refinement ($l_{\max}$), the effective resolution (E) is $(m_i \cdot m_x \cdot 2^{l_{\max}}) \times (m_j \cdot m_y \cdot 2^{l_{\max}})$. Ideally, the block arrangement is chosen so that for non-square domains of dimensions $L_x \times L_y$, mesh cells are approximately square.



(a) Shallow-water wave equation simulation in a box. Wave structure shows reflections off of solid wall boundaries.



(b) Quadtree of patches [1]

Figure 1. The left figure depicts an adaptively refined ForestClaw solution to the shallow-water equations on a Cartesian grid in the quadtree layout on a single block. The right figure depicts three adjacent adaptive levels, each with an 8×8 simulation grid (with thick borders) and a layer of ghost cells (shaded gray). Quadrant boundaries are indicated by thick lines.

4.1. GeoFlood refinement criteria

A new AMR strategy based on flags imposed by different refinement criteria has been developed in the GeoFlood code. These include: 1) Water depth criteria, where refinement is permitted only in wet cell regions by imposing a flag on cells with water depths greater than a certain threshold value and the refinement level is determined by the water depth. 2) Velocity-depth product flag, used to force refinement in shallow regions where the flow changes rapidly such as near river banks or shorelines. 3) Velocity criteria, which assume that the magnitude of the water velocity in both x - and y - directions is greater than a certain threshold value. 4) Flood source flags, used to force refinement in regions containing the flood source, i.e., the dam in the case of a dam break. This allows the code to refine the flood source at high resolution to capture the flood details along the floodplain and allows the specification of regions to be refined to a given desired resolution by user-specified coordinates and minimum and maximum refinement levels. 5) Flow-grades flag, where refinement is enforced to given levels for depths or velocities greater than user-defined thresholds. This enables the code to refine regions containing lakes, seas, or rivers in the floodplain at varying intermediate levels compared to the flowing material.

5. Benchmark Test Cases

We selected a series of benchmark scenarios, designed by the United Kingdom Environment Agency [2], to evaluate the capabilities of GeoFlood within the context of flood risk management. This evaluation focused on the model's precision in replicating flood progression and the extent of inundation across diverse physical landscapes and configurations. The performance of GeoFlood in these benchmark tests was evaluated by comparing results with those obtained from both HEC-RAS and GeoClaw under identical test conditions. These test cases have also been employed previously in the benchmarking of various other models, including HEC-RAS. For a more comprehensive overview of these benchmarking studies, see, for example, Brunner [4], Cea et al. [49], and Neelz and Pender [2].

5.1. Test Case 1: Speed of Flood Propagation over an Extended Floodplain

The first benchmark was designed to test a model's ability to resolve an advancing flood front over a floodplain in two-dimensions. The test itself consists of a relatively large, flat (uniform elevation)

bed with inflow along a relatively short inlet on one boundary. Although the test involves highly idealized geometry and inflow, it is designed to accentuate model differences or error because the flow is very thin (< 1 m in depth) and it advances for several hours over a relatively large domain (1000 m by 2000 m). Errors in the propagation of a thin wet-dry front are enhanced over large domains and long computation times. Furthermore, the leading edge of an advancing front is prone to spurious numerical oscillations in depth and speed [e.g., 4,17]. Lastly, the problem is nearly radially symmetric which facilitates assessments of 2D algorithms at arbitrary angles on non-radially symmetric meshes.

5.1.1. Problem Setup

The computational domain is a 1000×2000 m rectangular region. An inflow hydrograph specifying discharge is imposed along a 20-m length of the left boundary at its midline (Figure 2). The total inflow discharge (volume flux) rises linearly from 0 to a peak value of $20 \text{ m}^3/\text{s}$ during the first 60 minutes, remains constant for the next 180 minutes, then drops linearly to zero for the final 60 minutes (Figure 3). Dividing the total inflow discharge by the channel length provides the inflow momentum density, $\overline{hu}(t)$. Before each numerical timestep we set the momentum in ghost cells (exterior cells along the the channel left of the physical boundary) with $hu^n = \overline{hu}(t^n)$.

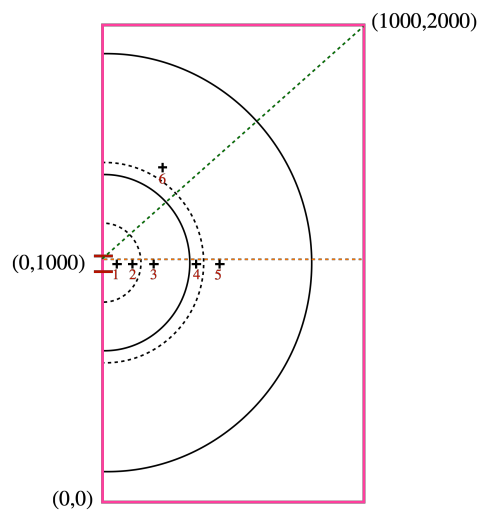


Figure 2. Spatial domain displaying the inflow location (boundary condition) along 20 m of the left boundary ($x = 0$ and $990 \leq y \leq 1010$ m). Level-set contours are shown for the depths $h = 10$ and $h = 20$ cm at $t = 1$ hour (black dashed curves) and $t = 3$ hours (black solid curves), indicating the nearly radially symmetric solution. The diagonal (green dashed) and horizontal (orange dashed) lines depict the two transects considered in Figure 4. The six control points (+ symbols) indicate the location for time series depicted in Figure 4a and Figure 4b.

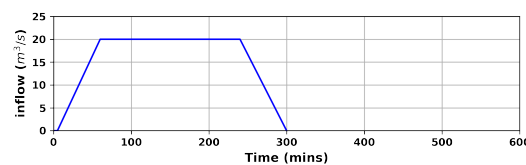


Figure 3. Inflow hydrograph imposed at the inlet boundary.

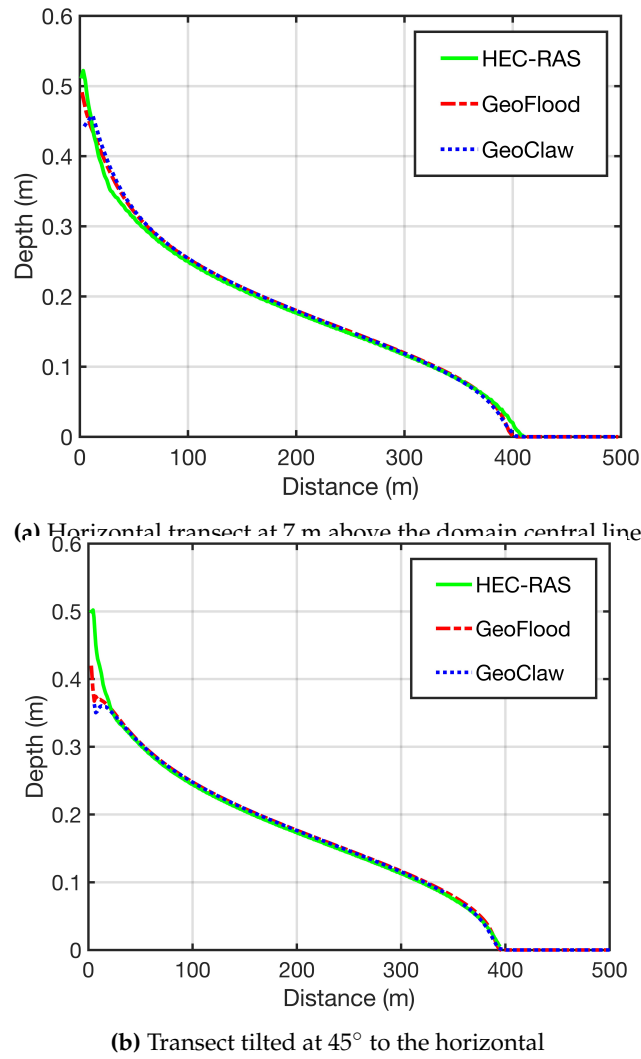


Figure 4. Cross-section of depths along (a) a horizontal line 7 m above the horizontal central line through the domain and (b) tilted at 45° to the horizontal at time $t = 1$ hour for both HEC-RAS, GeoFlood, and GeoClaw. Refer to Figure 2 for a depiction of the horizontal and tilted lines.

To determine the depth $\bar{h}^n$ for the ghost cells at the inflow location, we solve

$$\frac{\bar{h}u^n}{\bar{h}^n} = 2\sqrt{g\bar{h}^n} + u_{int}^n - 2\sqrt{gh_{int}^n}, \quad (10)$$

where $u_{int}^n = (hu)_{int}^n / h_{int}^n$, and h_{int}^n and $(hu)_{int}^n$ refer to the numerical solution determined in the interior values adjacent to the ghost cells. To avoid numerical difficulties, if h_{int}^n is non-zero but less than a minimum tolerance τ , $0 < \tau \ll 1$, we set

$$h_{int}^n = \max\left(\left(\frac{\bar{h}u^n}{\sqrt{g}}\right)^{2/3}, \tau\right), \quad (11)$$

before solving for $\bar{h}^n$. If the solution to (10) led to the condition, $\bar{h}^n > h_{int}^n$, we recomputed $\bar{h}^n$ by solving

$$\frac{\bar{h}u^n}{\bar{h}^n} = u_{int}^n + (\bar{h}^n - h_{int}^n) \sqrt{\frac{g}{2} \left(\frac{1}{\bar{h}^n} + \frac{1}{h_{int}^n} \right)}, \quad (12)$$

utilizing Hugoniot-loci for a single incoming shock rather than the Riemann invariants implied by (10) [e.g., 20]. Equations (10) and (12) were solved using the Newton Raphson method to determine the depth $\bar{h}^n$ in ghost cells. The Newton solver was initialized using an initial estimate $(\bar{h}^n)^0 = \left[\frac{\bar{h}u^n}{\sqrt{g}} F \right]^{2/3}$,

where the constant F is an initial rough estimate of the Froude number, which we set to 0.5 for the Riemann invariant problem and to 1 for the single shock problem.

Following the cessation of inflow after 300 minutes ($\overline{hu}(t^n) = 0$), we filled ghost cell values at the channel edge by applying a wall boundary condition using interior values h_{int}^n and $(hu)_{int}^n$. All other boundaries are closed, and the initial condition considered is a dry bed.

The spatial domain was discretized into 200×400 cells with a uniform grid spacing of 5 m in both the x - and y -directions. The following numerical configurations were used: a wet dry threshold of $\tau = 0.0001$ m, an adaptive time step with a maximum CFL (Courant-Friedrichs-Lewy) of 0.9 to ensure numerical stability, a final time of 6 hours, and a Manning coefficient of 0.03, following [2].

5.1.2. Test Case 1: Simulation Results

GeoFlood produced results that are numerically consistent with the previously benchmarked models, GeoClaw and HEC-RAS. In Figure 4a and Figure 4b, the three modeled results for the depth h along two transects (see Figure 2) are shown for comparison. The grid-aligned (horizontal) transect shown in Figure 4a and the grid-diagonal transect shown in Figure 4b reveal that none of the models exhibit significant spurious grid-alignment effects or oscillations at the flow-front. The only significant discrepancy between the models occurs in the depth near the inflow boundary. There are two possible explanations for the discrepancy that do not relate to the interior solution. First, we plotted the GeoFlood and GeoClaw solutions on rays (horizontal (Figure 4a) and diagonal (Figure 4b) originating at (0, 1007), due to the spatial arrangement of the block-structured AMR which we chose to resolve the inflow boundary segment. Second, and probably more significantly, an inflow discharge hydrograph does not uniquely determine boundary conditions necessary to solve the shallow-water equations, even given equivalent numerical discharge values, hu . Physical or mathematical constraints are required to determine the inflow stage and velocity [e.g., 4,17]. The boundary conditions employed in GeoClaw and GeoFlood (??) are distinct from those employed in HEC-RAS, which may be based on approximate steady channel flow assumptions commonly used in flood modeling, but are not described in numerical detail in HEC-RAS documentation. The boundary discrepancy in stage is a result of how the value for discharge is enforced in different models, not a difference in model accuracy because the exact solution is ambiguous. Nevertheless, equivalent inflow discharges in the three models lead to consistent results downstream of the boundary.

In another test, higher refinement (minimum cell size of 0.6 m) of the propagating flood was employed in the GeoFlood simulation (Figure 5). Refinement through AMR allows a more optimal spatio-temporal allocation of computational resources by neglecting the resolution of dry regions in the domain. An equivalent grid resolution in HEC-RAS with the same time-step stability criterion would require approximately 580 times $((5/0.6)^3)$ more computational expense than the 5 m simulation shown in Figure 5.

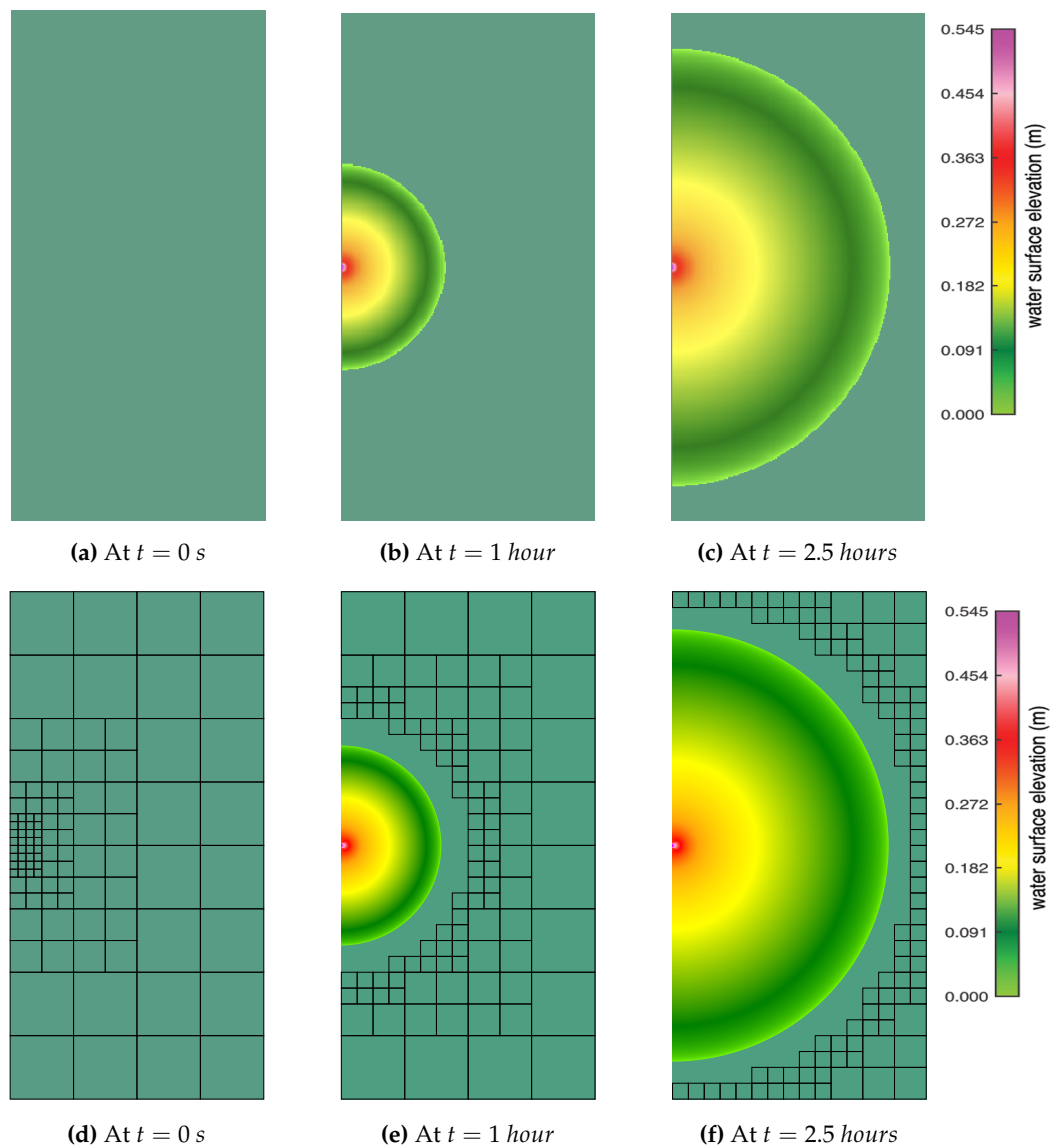


Figure 5. Overhead map perspectives of the water surface elevation at various times for both the HEC-RAS (a-c) and GeoFlood (d-f) simulations. The HEC-RAS simulation was performed on a 5 m uniformly structured grid with 200×400 grid cells while GeoFlood was simulated on an adaptively refined grid with max-level = 4, min-level = 1, starting on the coarsest mesh of 50×50 level 0 grid blocks in a 2×4 block arrangement to finest mesh of 1600×3200 grid cells at 0.6 m grid cell resolution. At $t = 1$ and $t = 2.5$ hours, grid lines for HEC-RAS and the maximum refinement level in GeoFlood are omitted from the plots.

The temporal evolution of flow depth and velocity are compared at several control points (Figure 6 with locations shown in Figure 2) demonstrating that GeoFlood produces results consistent with the previously validated codes, GeoClaw and HEC-RAS.

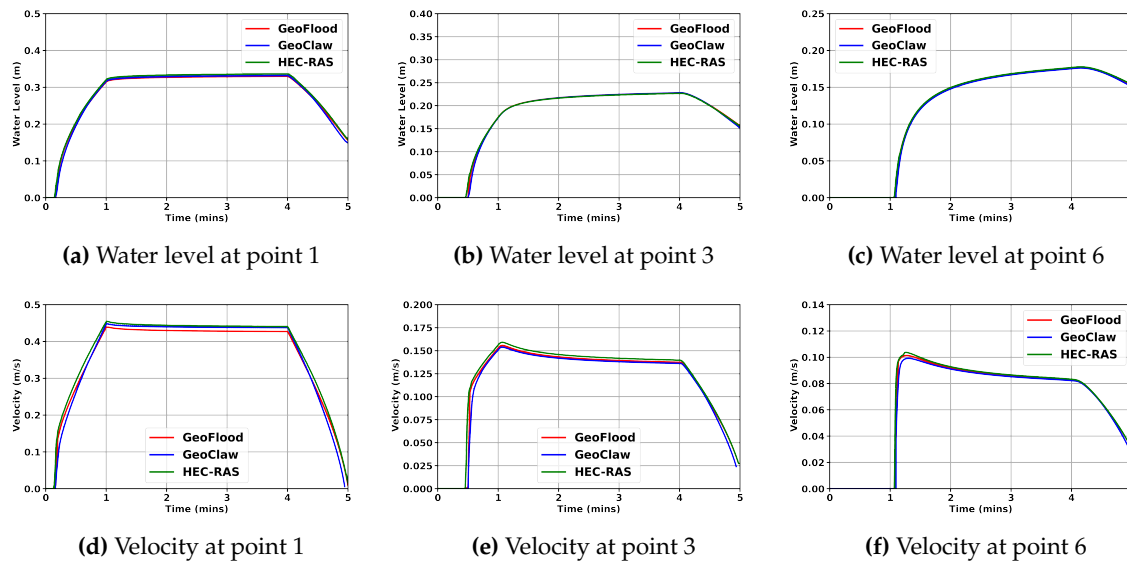


Figure 6. Temporal evolution of the water surface elevation (a-c) and velocity (d-f) at various control points (Figure 2) for GeoFlood compared to GeoClaw and HEC-RAS.

5.2. Test Case 2: Filling of Floodplain Depressions

The second benchmark problem we consider is similar to Test Case 1, in that inflow discharge is specified leading to inundating flow into a highly idealized floodplain (topographic surface). However, unlike Test Case 1, the floodplain contains isolated depressions, some of which fill slowly over several hours. This test is designed to evaluate a model's predictive accuracy in determining the inundation extent and final depth of flooding over a long period of time under conditions of low-momentum flow across intricate topographical landscapes. The primary focus of this assessment is on the final depths in individual depressions rather than their peak levels. Similar to Test Case 1, the problem accentuates model differences and sensitivities.

5.2.1. Problem Setup

The computational domain of this test is a square with a side length of 2000 m. Figure 7a depicts a 4×4 matrix of 0.5 m deep depressions with smooth topographic transitions obtained by multiplying sinusoids in the north-to-south and west-to-east directions. The underlying average slope is 1 : 1500 in the north-south direction and 1 : 3000 in the west-east direction, with an elevation drop of about 2 m along the northwest-to-southeast diagonal. At the upstream boundary (northwest corner of the domain), an inlet hydrograph with a peak discharge of $20 \text{ m}^3/\text{s}$ and a time base of approximately 85 minutes is imposed along a 100-m-long line (blue thick line) that runs north to south Figure 7a. All other boundaries were treated as solid-wall boundaries, and the initial condition assumes a dry bed. The inflow hydrograph boundary condition was implemented as described in Section 5.1.

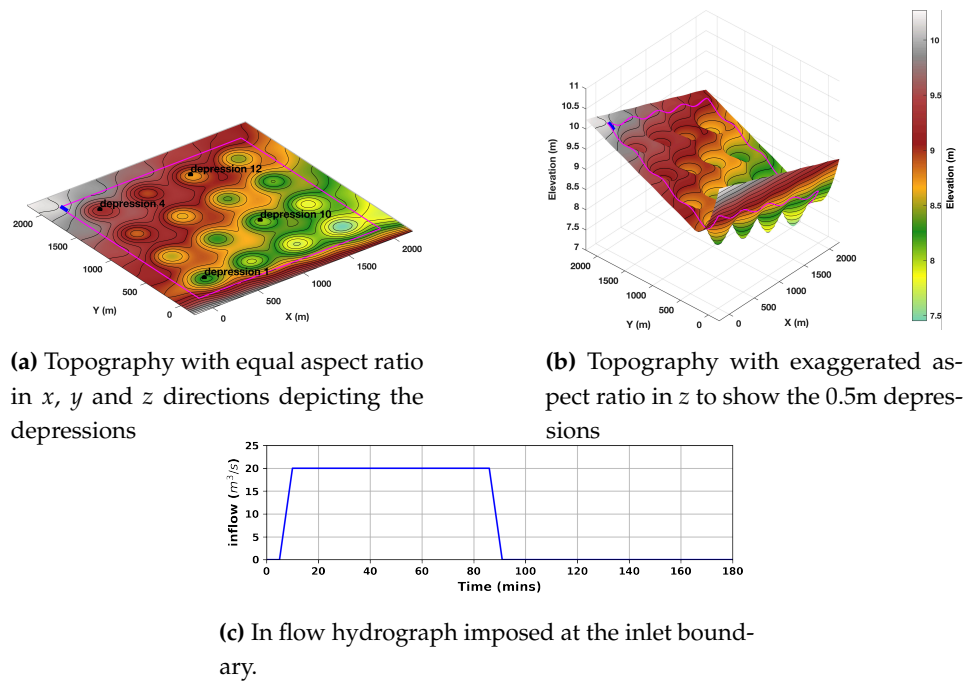


Figure 7. Topography (a-b) showing the inflow location (blue thick line at the northwest corner) and ground elevation with contour lines at 0.5 m intervals. The depressions are numbered sequentially 1 – 16, increasing in the positive Y -direction followed by the positive X -direction. Labeled depressions 1, 4, 10, and 12, indicate where numerical time series are compared in Figure 8. (c) inflow hydrograph for total discharge through the inflow boundary segment.

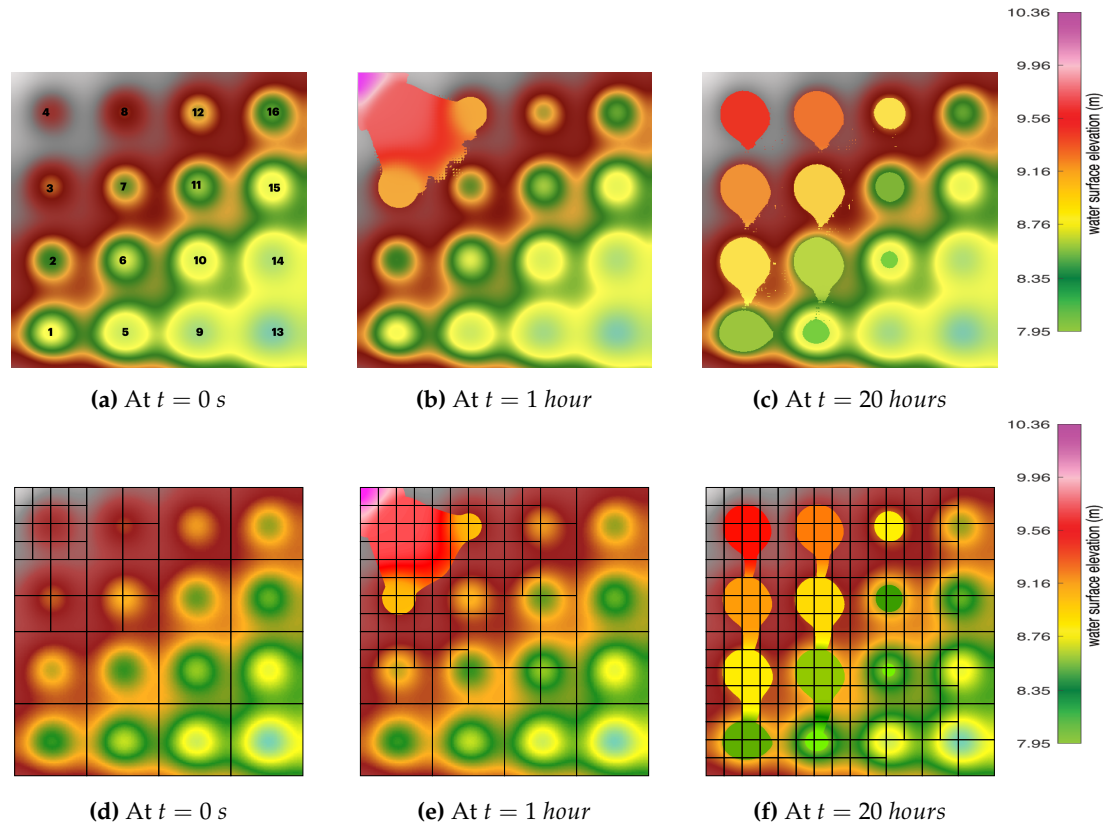


Figure 8. Overhead maps showing the surface elevation (water plus topography, $h + b$) at various times for both HEC-RAS (a-c) and GeoFlood (d-f).

The model simulates two days (48 hours) to achieve a final hydrostatic steady state. The following numerical configurations were used: a wet-dry threshold of $\tau = 0.0001m$, an adaptive time step with a maximum CFL of 0.9, and a Manning coefficient of 0.03 for comparison with Brunner [4].

5.2.2. Test Case 2: Simulation Results

GeoFlood generated results that were comparable to those of HEC-RAS and GeoClaw and predicted the flood extent at all depressions. However, different flood arrival times and initial depths were observed at depressions 5 and 10 for all models, indicating a sensitivity that we attribute to extremely shallow flow between the depressions over the thresholds and relatively slight differences in wave dynamics. We attribute the slight difference between the water elevations observed in the depressions 10 and 12 to the shallow depths and the bowl-shaped topography at these depressions. Figures 8d - 8f show the adaptively refined solution simulated by GeoFlood at times $t = 0s$, $t = 2$ hours, and $t = 2.5$ hours, respectively, at mesh refinement levels, 0, 1, and 2, on a 50×50 grid of 4 level 0 blocks in each of the x- and y-directions, yielding a 10 m grid resolution at level 0 and 2.5m at level 2. We compare the results with those from HEC-RAS on a uniform 200×200 grid of 10 m resolution. Figure 8e shows that during the influx, a transient water level peak was observed close to the inflow. Following the cessation of the inflow, each depression's water level steadily dropped until it eventually reached the level of the lowest "sill" separating it from adjacent depressions, as shown in Figure 8f after 20 hours. Figure 9 depicts the water-surface elevation time series at six depressions.

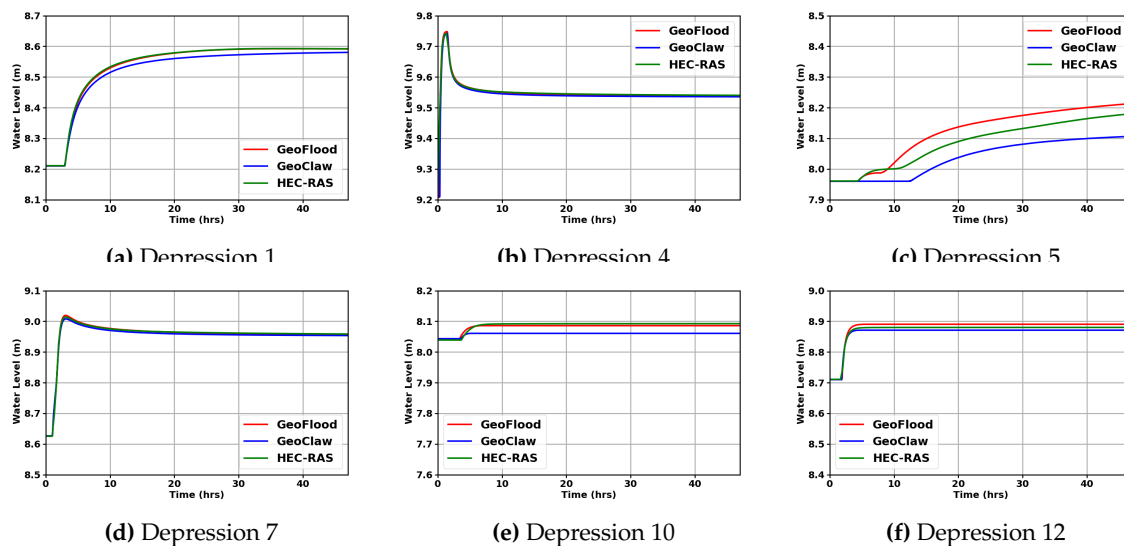


Figure 9. Temporal evolution of the water level at various depressions for GeoFlood compared to GeoClaw and the HEC-RAS model. See figure Figure 8 for locations of depressions on the simulation grid.

5.3. Test Case 3: Dam Break

The third benchmark test is designed to assess a model's ability to accurately simulate transcritical flows, hydraulic jumps, and wakes behind obstacles. This test case involves the rupture of a dam in a laboratory scale model and is presented in Frazão and Zech [50]. Physical parameters have been scaled by 20 times relative to the experimental set-up in order to replicate the scale of a real-world dam break scenario.

5.3.1. Problem Setup

In the experiment, a dam-break wave was generated by the nearly instantaneous opening of the gate at the end of laboratory-scale reservoir. The wave then collided with an oblique rectangular obstacle positioned downstream of the dam, producing a hydraulic jump just upstream of the obstacle and a wake zone downstream, as shown in Figures 10 and 11.

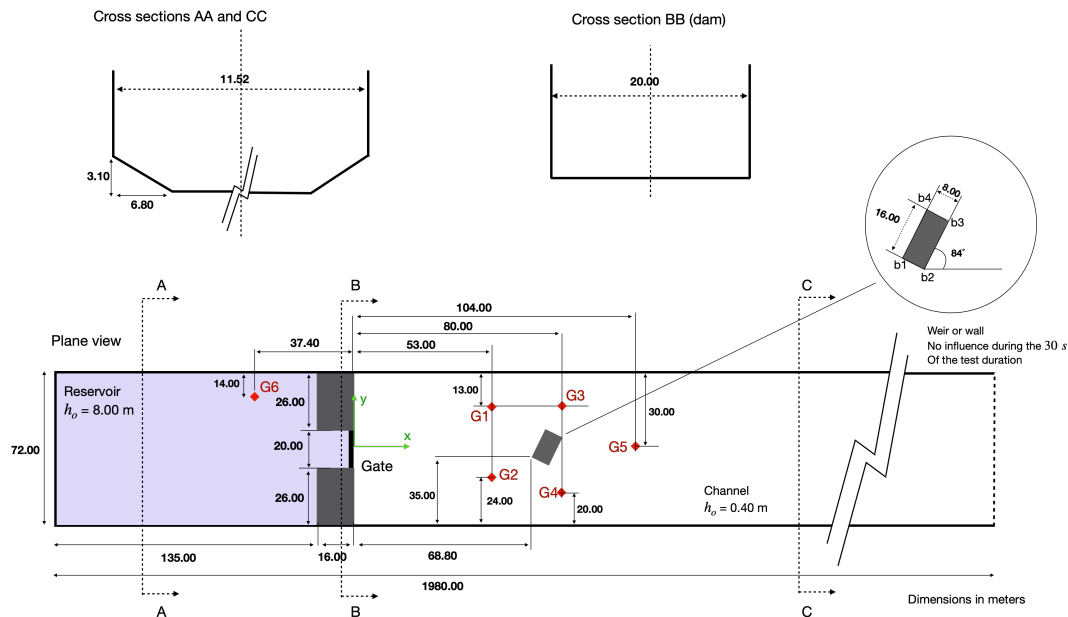


Figure 10. Geometry and dimensions for the experimental dam break test case, scaled-up 20 times. Adapted from Neelz and Pender [2].

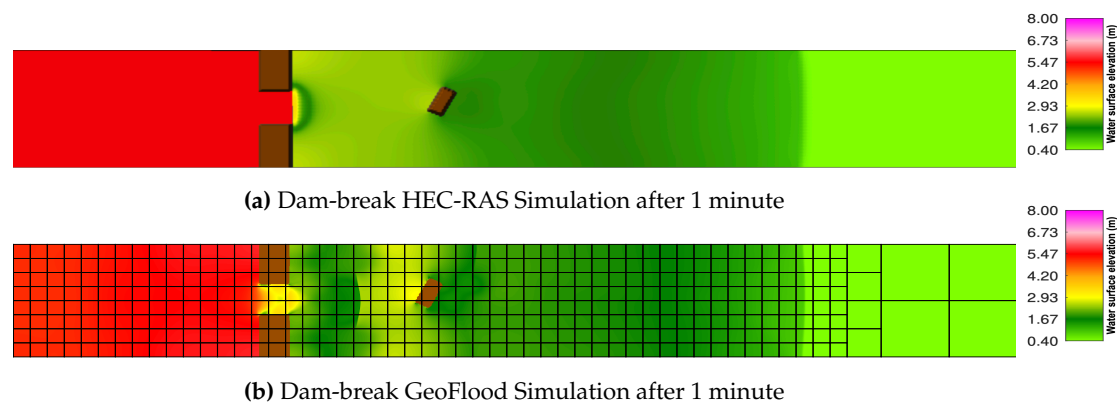


Figure 11. Flow depth and wave structure 1 minute after the dam break (Test Case 3) for the (a) HEC-RAS and (b) GeoFlood simulations, on uniform and adaptively refined grids, respectively. Color scale indicates the water surface elevation.

The scaled-up computational domain was 1980 m long and 72 m wide (Figure 10). The simulation was run for 30 minutes. The following numerical configurations were used: a wet-dry threshold of $\tau = 0.0001$, an adaptive time step with a maximum CFL of 0.9, and a Manning coefficient of 0.05.

5.3.2. Test Case 3: Simulation Results

The advancing waves downstream of the obstacle are qualitatively similar in the GeoFlood and HEC-RAS simulations, but the GeoFlood simulation produces a more pronounced upstream-directed stationary bore above the obstacle (where flow transitions from super- to sub-critical). Nevertheless, comparisons of the temporal variation of the water-surface elevation and velocity for GeoFlood, GeoClaw, and HEC-RAS demonstrate generally consistent predictions at gauge points (Figure 12). The HEC-RAS simulation produces a sharp discontinuity remaining at the location of the dam and relatively smooth flow near the obstacle. The HEC-RAS solution was computed on a uniform 990×36 grid, while the GeoFlood solution used an adaptive grid starting with coarse 36×36 level 0 grid blocks in a 27×1 block arrangement at 2 m grid resolution refining to a 0.3 m per grid cell resolution.

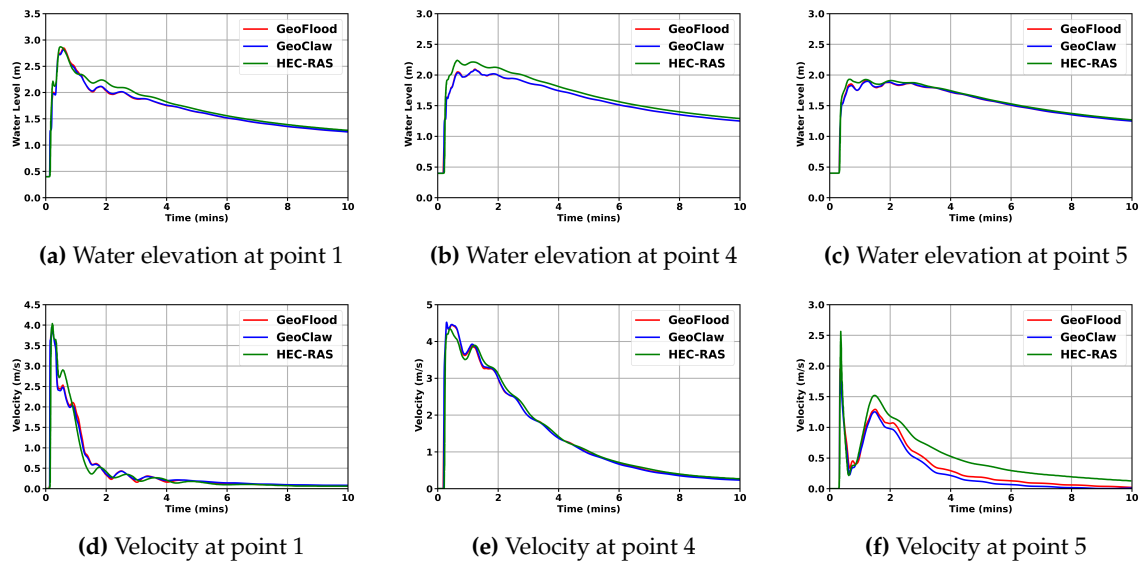


Figure 12. Temporal evolution of the water level (a-c) and velocity (d-f) at various control points for GeoFlood, GeoClaw, and HEC-RAS. The locations for the time series shown, point 1 (a and d), point 4 (b and e), and point 5 (c and f) are shown and labeled G1, G4 and G5, respectively in Figure 10

6. Malpasset Outburst-Flood Simulations

For our final test case, we assess GeoFlood's ability to simulate a historical dam break and overland flood event. The availability of field data, laboratory-scale model data, and previous numerical results provide a means for verification and validation of GeoFlood for a real-world problem that is numerically challenging due to highly-energetic flows in steep and irregular terrain. We also use this test case to benchmark GeoFlood's parallel computational efficiency, and to demonstrate the usage of GeoFlood's AMR schemes for a large-scale, real-world problem.

6.1. Historical Background

The Malpasset Dam is situated approximately 12 km upstream from the town of Fréjus, France (Figure 13). This thin-arch dam was constructed in a narrow gorge above the Reyran River valley in order to impound a reservoir with a storage capacity of 55106 m^3 . Upon reaching reservoir capacity, the dam failed suddenly and catastrophically on December 2, 1959 at 21:14 hours (generating an acoustic shock wave witnessed by residents in Fréjus, suggesting a nearly instantaneous failure). The dam had a maximum height of 66.5 m and a crest span of 223 m. Only remnants of the dam's arch remained after the failure, with significant erosion of the adjacent rock bank. Subsequent investigations suggest the arch dislodged from its base leading to a rapid sequential collapse [11].

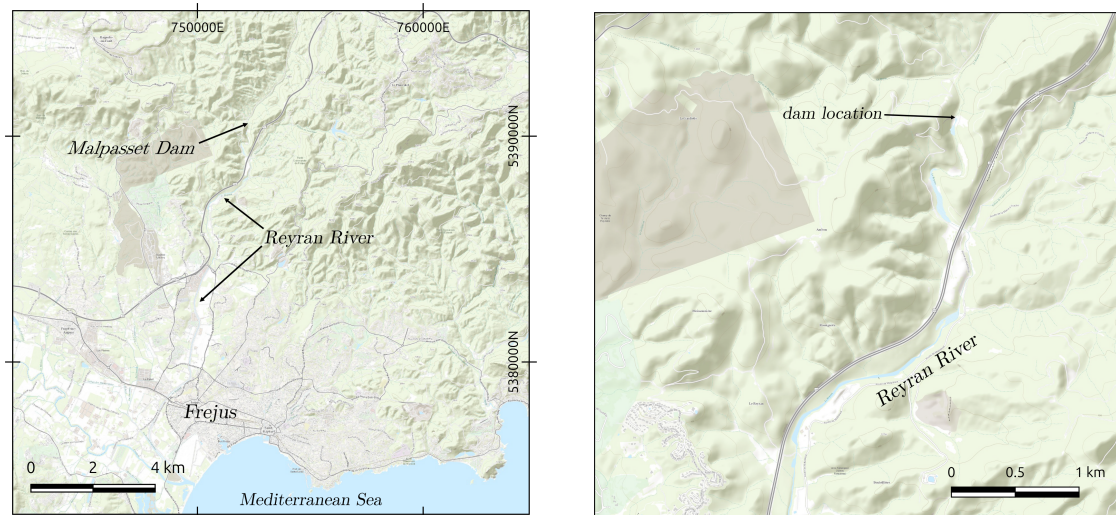


Figure 13. Overhead map of the area surrounding Fréjus, France (a). The reservoir impounded by the Malpasset Dam lay in the river valley to the northeast of the dam. Close-up (b) shows the Reyran River gorge features below the dam (b). Base images: ESRI World Topo.

The catastrophic breach led to a rapid flood wave that descended through a channelized ravine, eventually inundating the wide floodplain adjacent to the Mediterranean and surrounding Fréjus. The event resulted in 433 fatalities and significant infrastructure damages, including the obliteration of a 1.5-km section of highway and an adjoining bridge, and extensive flooding of Fréjus. The downstream displacement of massive concrete blocks indicated the power of the flood. The flood waves rose to ≈ 20 m above the original riverbed. A complete description of this event can be found in [7].

6.2. Topographic Features and Model Domain

Downstream of the dam, the Reyran River passes through a pronounced depression, then winds through a series of steep, fast-flowing serpentine bends before descending through a ≈ 4 km gorge, from which it ultimately emerges onto the broad, low-lying alluvial fan in the Reyran River valley surrounding Fréjus (Figure 13). Highly energetic flows through these features pose numerical challenges due to large flow gradients and topographic source terms, making them a valuable test for model robustness and stability [e.g., 5,6].

Our simulation dimensions are 17500×9000 m. The digital elevation model (DEM) for the GeoFlood simulations originated from a benchmarking exercise in 1999, sponsored by CADAM (Concerted Action on Dam-break Modelling) project, a European research group [28] which included a set of 13541 irregularly spaced elevation points covering the study domain. The points were interpolated to overlapping sets of regularly-spaced Cartesian grids with 2, 5, and 20 m resolutions by George [5] and used in this study.

GeoFlood incorporates GeoClaw's topographic processing schemes, which allow users to list multiple overlapping or nested DEMs of different resolutions without requiring preprocessing. The AMR interpolation schemes are designed to preserve the integral of a unique topographic interpolant defined by the DEMs, which is necessary for volume conservation upon dynamic refinement of a submerged area [23]. Nevertheless, it is generally recommended to maintain the same resolution for rapid flows in steep terrain [5].

6.3. Initial and Boundary Conditions

The sea level and the initial reservoir level are assumed to be constant and are set at 0 and 100 m above sea level, respectively. Although the outlet gate near the bottom of the dam was open during the event, we neglected pre-event streamflow in the channel—the bottom was considered dry. The actual pre-event streamflow discharge is unknown, but we assumed that it was negligible relative to the massive outburst flood discharge. Similarly, we assumed that the inlet discharge upstream of the

reservoir was negligible. The sea level remained constant and uniform with an elevation of zero. We assumed an instantaneous dam failure.

For comparison with George [5] and the results produced by the CADAM project, we used a Manning coefficient of 0.033 for our simulations.

6.4. Simulation Refinement Flags

GeoFlood allows users to control AMR refinement, setting limits or enforcing minimum and maximum refinement levels based on multiple criteria. These criteria include flow features (depth, velocity, depth gradients) or geographic regions. For real-world problems, this has been found to provide more reliable behavior with better computational efficiency than error estimation algorithms (e.g., Richardson extrapolation) [23]. Additionally, users may have interest in particular locations or times.

For our simulation of the Malpasset flood, we enforced the highest level of refinement (level 5) for rapidly flowing water. We also enforced refinement of the reservoir based on its geographic location, by specifying its bounding box as input parameters. (Note that enforcing refinement to the highest level everywhere that water is present would result in unnecessary refinement of the Mediterranean Sea (Figure 14) where large depths would lead to severe time-step restrictions over the duration of the simulation.) We chose to enforce the highest refinement level for the reservoir in order to represent the total outflow discharge at the dam as accurately as possible (the flow dynamics in the lake would not otherwise need a high spatial resolution). More elaborate schemes, such as refining flow at a variety of levels based on flow criteria are possible.

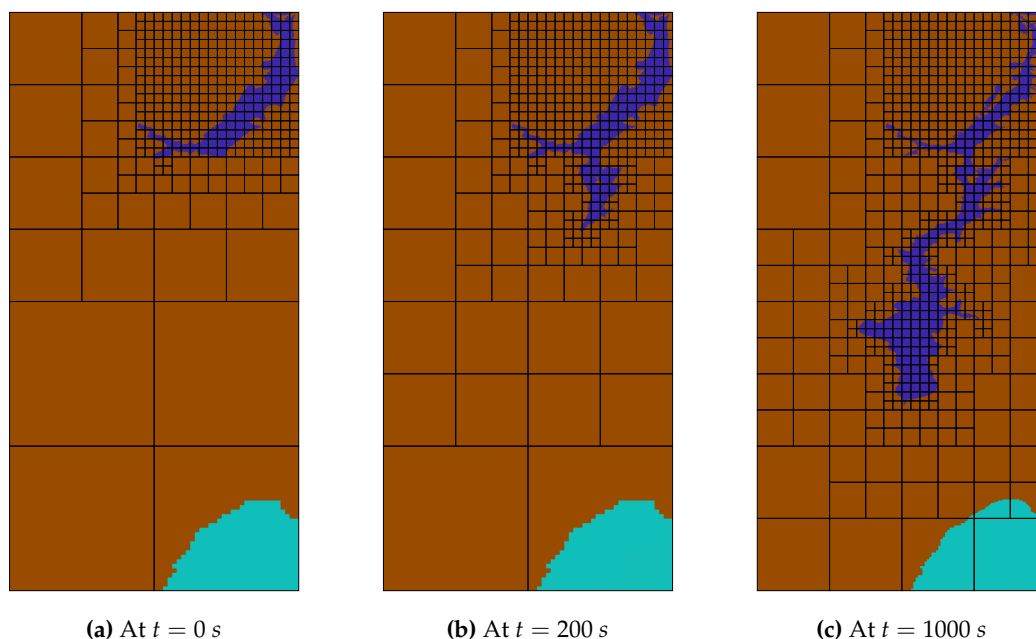


Figure 14. Adaptive mesh refinement (AMR) applied in the simulation of the Malpasset dam failure is depicted at times of $t = 0, 200$, and 1000 seconds. Lines indicate the edges of blocks, each containing 32×32 individual cells. The reservoir was refined to level $l = 5$ throughout the computation. As time progressed, the expanding flood was refined to $l = 5$, while the unaffected areas remained at the coarsest refinement level $l = 1$. The AMR procedures are designed to allow optimizing the balance of feature resolution with computational efficiency.

6.5. Comparison and Validation of Model Results

The Malpasset dam break and flood serves as an excellent real-world test case for model verification and validation, thanks to the availability of field survey data on high-water marks and laboratory-scale model data. These datasets enable verification of the shallow-water equations and facilitate model comparison and validation. We compare GeoFlood results against empirical data and previous results from other models, all of which are based on the shallow-water equations. Despite

the steep terrain, where shallow-water assumptions are most uncertain, GeoFlood performs well, as do the other models. Figure 15 presents a comparison of GeoFlood, GeoClaw [5], and two additional models from the 1999 CADAM workshop [28] against observations from 17 field-surveyed locations and 9 model gauge stations.

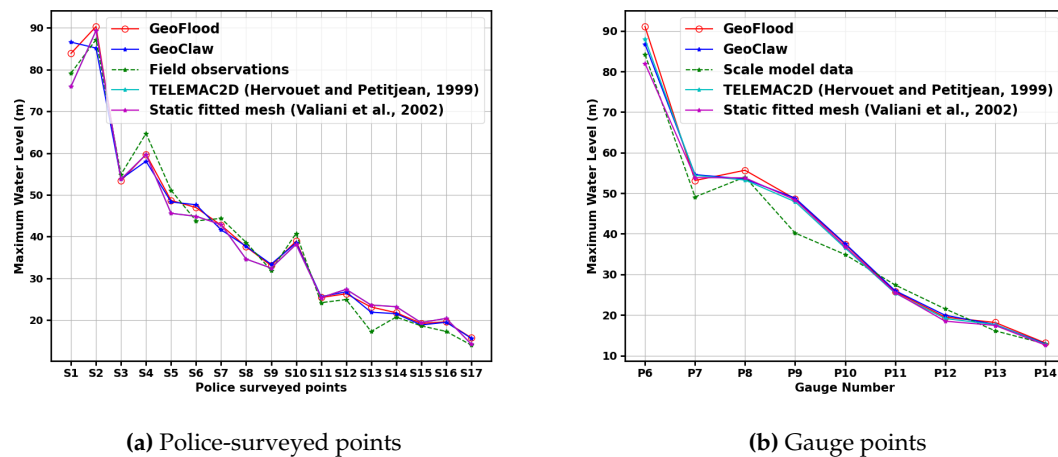


Figure 15. Maximum water levels simulated by GeoFlood, GeoClaw, and other model simulations compared to observations from (a) 17 field-surveyed points (referred to as “police-surveyed points” in related literature) and (b) 9 gauge locations from a laboratory scale model [28].

6.6. Computational Efficiency Benchmarks

GeoFlood simulations performed on a laptop (2.3GHz Quad-Core Intel Core *i7* processor with 16 GB of RAM), required approximately 1500 seconds of wall clock time to complete the entire scenario, a total simulated time of 4000 seconds. Our simulation used lowest level computational grids measuring 64×128 m, with the levels of grid refinement ranging from 1 to 5, yielding an effective resolution of 1024×2048 m and a cell resolution of 6 m per grid cell. The model operated with an initial time step of 1 second, followed by adaptive time-stepping adhering to a maximum CFL number set at 0.75.

We ran multiple GeoFlood and GeoClaw simulations in order to compare computational efficiency, particularly with respect to parallel efficiency and scalability. Experiments involving various numbers of nodes and cores per node were conducted to assess the scalability and efficiency of GeoFlood. GeoClaw runs in shared memory on single node and uses OpenMP for parallelization (one motivation for the development of GeoFlood). Our model efficiency comparisons were therefore derived from tests conducted on a single node of the Linux-based supercomputer (Borah, Boise State University), equipped with dual Intel Xeon Gold 6252 processors, each running at 2.1 GHz with 24 cores, providing a total of 48 cores on the node. Compared to GeoClaw, GeoFlood’s parallel performance and computational speed show improved scaling with even moderately higher processor counts (8, 16, and 32), and GeoFlood outperformed GeoClaw in efficiency across all tested runs (Figure 16). Even on a single node, GeoFlood’s dynamic regridding process is fully parallelized via the mesh management libraries ForestClaw and p4est, whereas GeoClaw performs regridding in serial. This distinction likely contributes to the observed results.

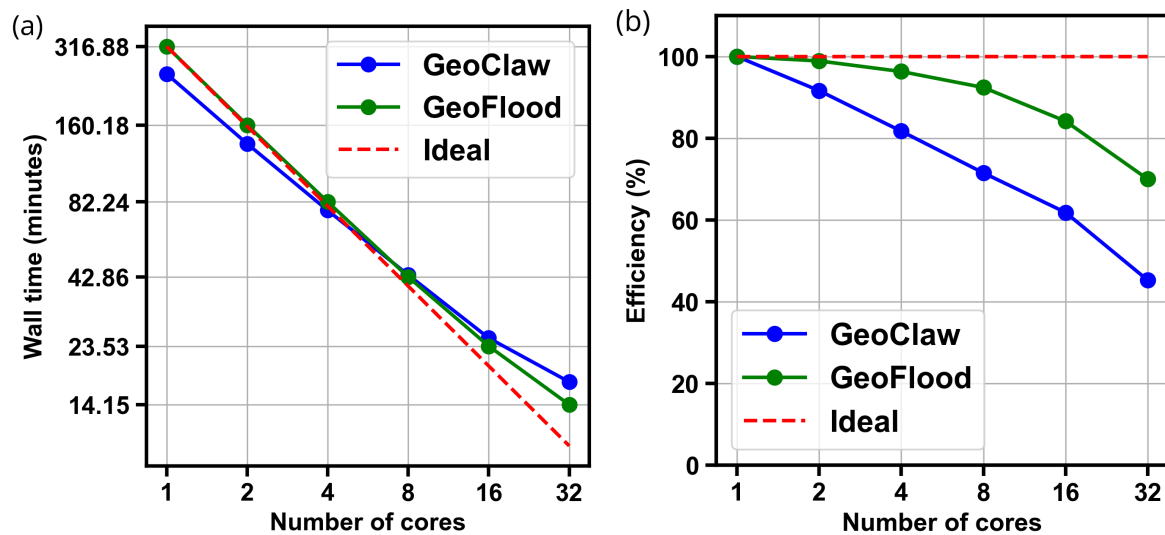


Figure 16. Comparisons of the wall time (a) and parallel efficiency (b) of GeoClaw and GeoFlood. For few processors (1–4), GeoClaw has a reduced wall time compared to GeoFlood. However, even beyond 8 cores, GeoFlood surpasses GeoClaw in parallel performance (a). GeoFlood outperformed GeoClaw in efficiency across all tested runs (b).

6.7. GeoFlood's Google Earth graphical toolbox

Geoflood contains a graphical toolbox that allows rendering simulation frames from an arbitrarily projected Cartesian AMR mesh into Google Earth, allowing the utilization of geo-referenced layers. This GeoFlood toolbox is designed to facilitate a user's ability to dynamically and rapidly assess a flood scenario across detailed geo-referenced topography as a simulation proceeds. Figure 17 shows some example Google Earth screen-capture snapshots of the Malpasset flood at various times.

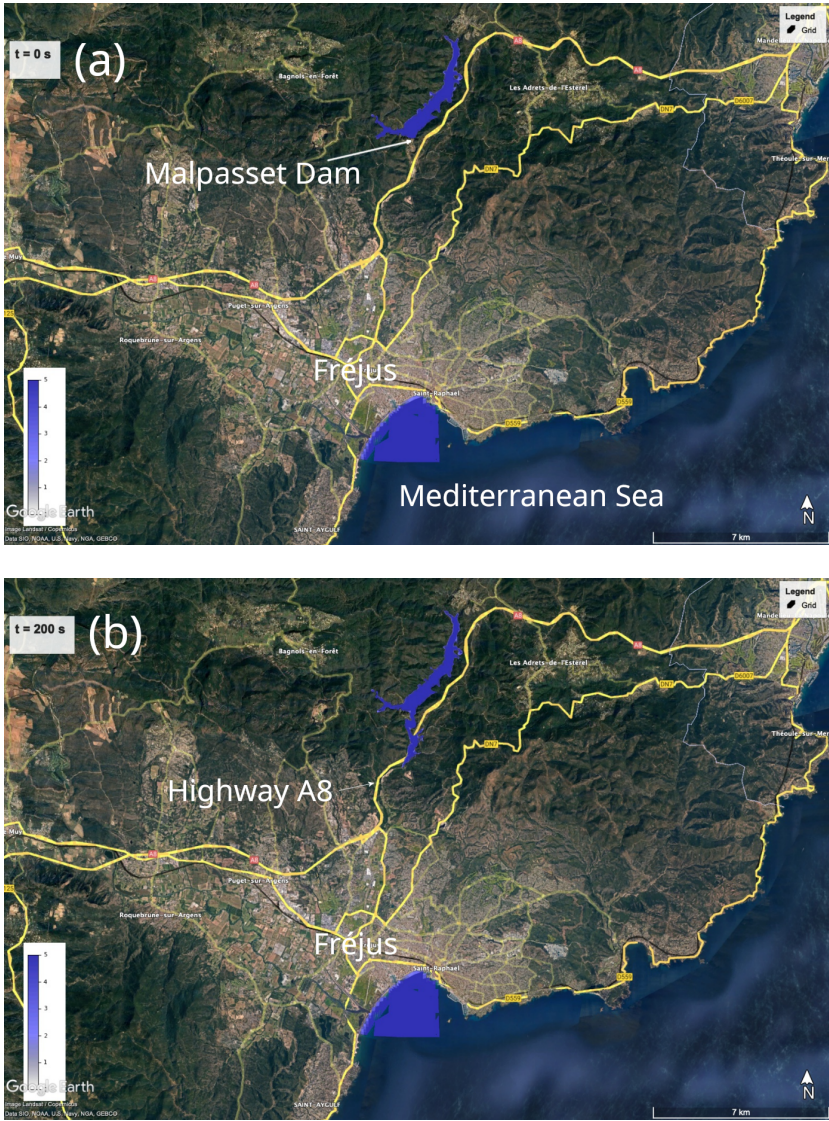


Figure 17. The sequence of the flood resulting from the Malpasset Dam failure is illustrated using Google Earth imagery at various times. At $t = 0$ seconds (panel a), the full reservoir with the dam intact can be seen in blue. By $t = 200$ seconds after dam failure (panel b), the floodwaters overtop the A8 highway, where the initial casualties occurred.(©Google Earth imagery).

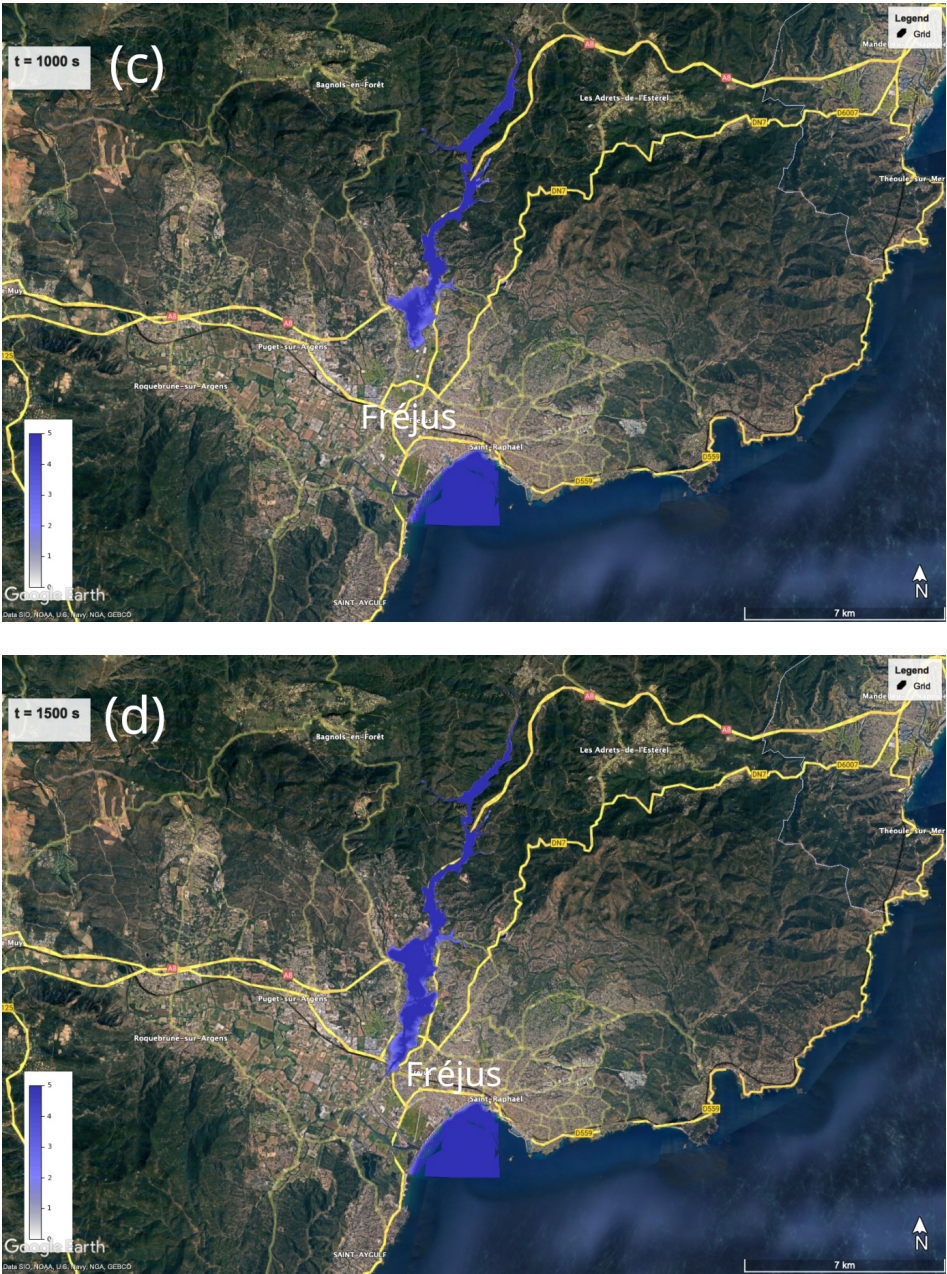


Figure 17. (cont.) The sequence of the flood resulting from the Malpasset Dam failure is illustrated using Google Earth imagery at various times. Between $t = 1000$ (panel c) and $t = 1500$ seconds (panel d), the flood wave progressed through the valley, reaching Fréjus approximately 21 minutes after the dam’s collapse.(©Google Earth imagery)

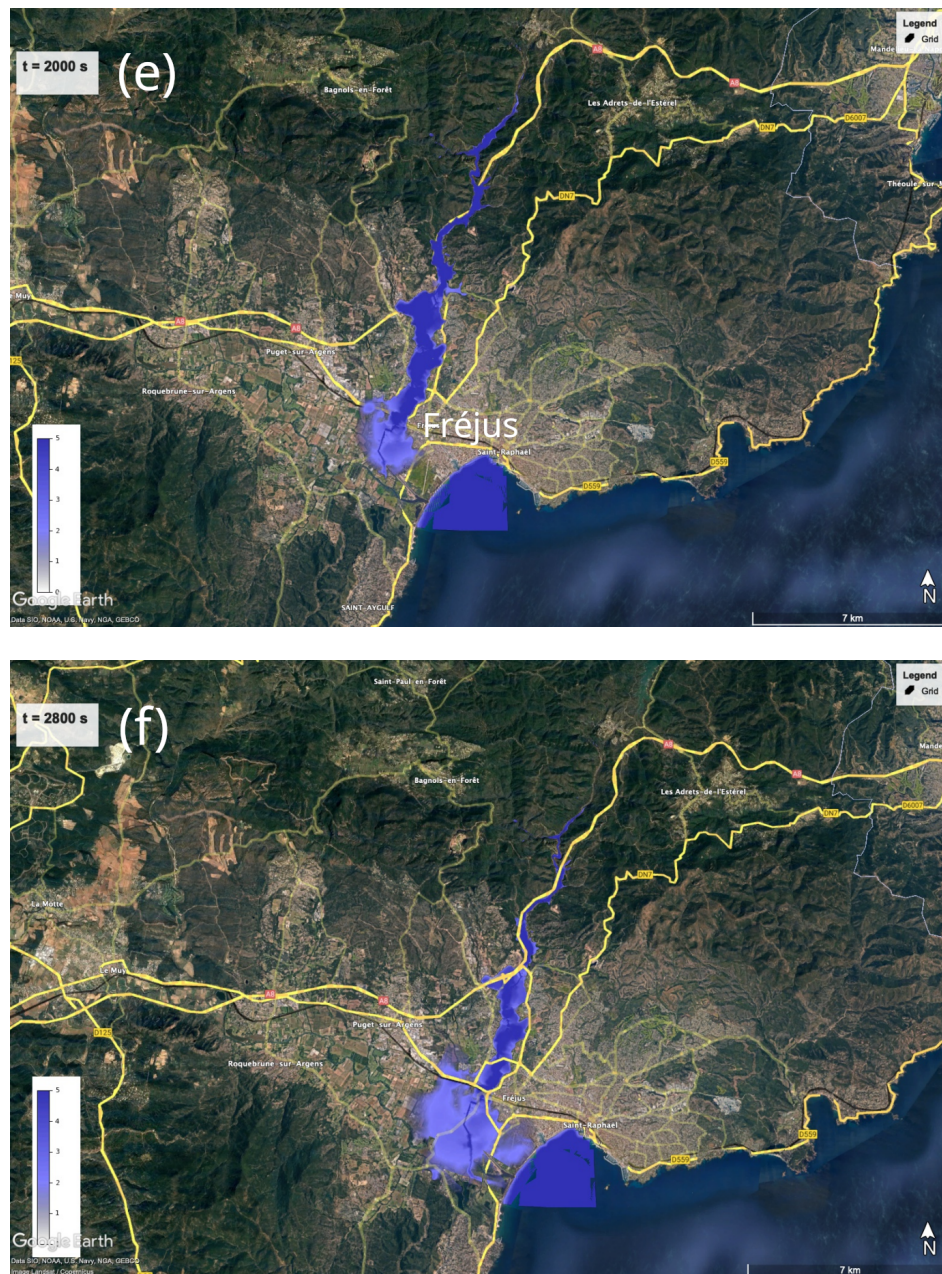


Figure 17. (cont.) The sequence of the flood resulting from the Malpasset Dam failure is illustrated using Google Earth imagery at various times. The imagery at $t = 2000$ seconds (panel e) shows the flooded area surrounding Fréjus. At $t = 2800$ seconds (panel f), the flood waves enter the Mediterranean Sea. (©Google Earth imagery).

7. Conclusions

A key motivation behind GeoFlood's development was to integrate the strengths of multiple specialized codes into a single package that is massively parallel and scalable, while still being robust and accurate for highly dynamic flood modeling applications in intricate or irregular topography. This was achieved through multi-block-based adaptive mesh refinement and PDE solvers from Clawpack and GeoClaw via ForestClaw, the creation of novel routines for data and input handling, and innovative functional linkages between code libraries. GeoFlood also includes features that facilitate model set-up and analysis (topography and other data input, graphical toolbox, etc.). Because the model is novel, the goal behind this paper is to validate GeoFlood's ability to solve the shallow-water equations for flooding applications, as well as provide a preliminary demonstration of its scalability compared to another flood modeling package, GeoClaw. For very large-scale problems spanning vast regions, we

anticipate that GeoFlood's ability to scale across many more cores (millions) will significantly reduce computational time compared to traditional flood modeling software.

GeoFlood's performance was rigorously tested and confirmed through three distinct benchmark challenges. Benchmark simulation results were compared with those from the HEC-RAS and GeoClaw models. GeoFlood results on the Malpasset dam break incident were corroborated using field data, laboratory scale-model data, and numerical results from prior studies. These evaluations revealed GeoFlood's capability to effectively forecast the trajectory of flood waves validated against actual recorded events. Its validation and scalability indicate that GeoFlood may be a valuable tool for leveraging computational resources for large-scale flood risk management strategies.

Author Contributions: Brian Kyanjo drafted the manuscript, set up and carried out the simulations, and designed the code under the supervision of Donna Calhoun and David L George. The authors read and approved the final manuscript.

Data Availability Statement: The current version of GeoFlood is available on Github: <https://github.com/KYANJO/GeoFlood> under the BSD 2-Clause Licence. The exact version of the model used to generate results used in this paper is archived on Zenodo [51], as are input data and scripts to run the model and produce the plots for all the simulations presented in this paper [52].

Acknowledgments: The authors extend their gratitude to all contributors and developers of ForestClaw, GeoClaw, and p4est (Carsten Burstedde) for making their codes publicly available. Yu-hsuan (Melody) Shih for the initial development of the GeoClaw solver in ForestClaw, and Boise State University for providing computational resources for this study. The authors acknowledge the financial support of the Department of Defense (DARPA AtmosSense program), NASA (ROSES Earth Surface and Interior Program), and the National Science Foundation (NSF-DMS award #1819257). Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Conflicts of Interest: The authors declare no competing interests present.

References

1. Calhoun, D.; Burstedde, C. ForestClaw: A parallel algorithm for patch-based adaptive mesh refinement on a forest of quadrees. *arXiv preprint arXiv:1703.03116* **2017**.
2. Neelz, S.; Pender, G. Benchmarking the latest generation of 2D Hydraulic Modelling Packages. Technical report, Environment Agency: Bristol, UK, 2013.
3. Clawpack Development Team. Clawpack software, 2020. Version 5.7.1, <https://doi.org/10.5281/zenodo.4025432>.
4. Brunner, G. Benchmarking of the HEC-RAS two-dimensional hydraulic modeling capabilities. Technical report, US Army Corps of Engineers: Davis, CA, USA, 2018.
5. George, D. Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: Application to the Malpasset dam-break flood (France, 1959). *Int. J. Numer. Methods Fluids* **2011**, *66*, 1000–1018. <https://doi.org/10.1002/fld.2298>.
6. Hervouet, J.M.; Petitjean, A. Malpasset dam-break revisited with two-dimensional computations. *J. Hydraul. Res.* **1999**, *37*, 777–788. <https://doi.org/10.1080/00221689909498511>.
7. Boudou, M.; Moatty, A.; Lang, M. 1 - Analysis of Major Flood Events: Collapse of the Malpasset Dam, December 1959. In *Floods*; Vinet, F., Ed.; Elsevier, 2017; pp. 3–19. <https://doi.org/10.1016/B978-1-78548-268-7.50001-8>.
8. Johnson, G.P.; Holmes, R.R.; Waite, L.A. The Great Flood of 1993 on the Upper Mississippi River—10 years later, 2004.
9. Blumhardt, M. Colorado's devastating 2013 Flood: A look back 9 years later, 2022.
10. Filkins, D. A bigger problem than ISIS?, 2016.
11. Valiani, A.; Caleffi, V.; Zanni, A. Case study: Malpasset dam-break simulation using a two-dimensional finite volume method. *J. Hydraul. Eng.* **2002**, *128*, 460–472. [https://doi.org/10.1061/\(ASCE\)0733-9429\(2002\)128:5\(460\)](https://doi.org/10.1061/(ASCE)0733-9429(2002)128:5(460)).

12. Kirstetter, G.; Delestre, O.; Lagrée, P.Y.; Popinet, S.; Josserand, C. B-flood 1.0: an open-source Saint-Venant model for flash-flood simulation using adaptive refinement. *Geosci. Model Dev.* **2021**, *14*, 7117–7132. <https://doi.org/10.5194/gmd-14-7117-2021>.
13. Yu, H.L.; Chang, T.J. A hybrid shallow water solver for overland flow modelling in rural and urban areas. *J. Hydrol.* **2021**, *598*, 126262. <https://doi.org/10.1016/j.jhydrol.2021.126262>.
14. Coulibaly, G.; Leye, B.; Tazen, F.; Mounirou, L.A.; Karambiri, H. Urban Flood Modeling Using 2D Shallow-Water Equations in Ouagadougou, Burkina Faso. *Water* **2020**, *12*. <https://doi.org/10.3390/w12082120>.
15. Shamkhalchian, A.; de Almeida, G.A.M. Effects of reconstruction of variables on the accuracy and computational performance of upscaling solutions of the shallow water equations. *J. Hydraul. Res.* **2023**, *61*, 409–421, [<https://doi.org/10.1080/00221686.2023.2201210>]. <https://doi.org/10.1080/00221686.2023.2201210>.
16. Rousseau, M.; Cerdan, O.; Delestre, O.; Dupros, F.; James, F.; Cordier, S. Overland Flow Modeling with the Shallow Water Equations Using a Well-Balanced Numerical Scheme: Better Predictions or Just More Complexity. *J. Hydrol. Eng.* **2015**, *20*, 04015012, [<https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29HE.1943-5584.0001171>]. [https://doi.org/10.1061/\(ASCE\)HE.1943-5584.0001171](https://doi.org/10.1061/(ASCE)HE.1943-5584.0001171).
17. George, D.L. Augmented Riemann solvers for the shallow water equations over variable topography with steady states and inundation. *J. Comput. Phys.* **2008**, *227*, 3089–3113. <https://doi.org/10.1016/j.jcp.2007.10.027>.
18. George, D.L. *Finite volume methods and adaptive refinement for tsunami propagation and inundation*; University of Washington, 2006.
19. Mandli, K.T. A numerical method for the two layer shallow water equations with dry states. *Ocean Model.* **2013**, *72*, 80–91. <https://doi.org/10.1016/j.ocemod.2013.08.001>.
20. LeVeque, R.J., *Finite Volume Methods*. In *Finite Volume Methods for Hyperbolic Problems*; Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002; p. 64–86. <https://doi.org/10.1017/CBO9780511791253.005>.
21. Berger, M.J.; Olinger, J. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* **1984**, *53*, 484–512. [https://doi.org/10.1016/0021-9991\(84\)90073-1](https://doi.org/10.1016/0021-9991(84)90073-1).
22. Berger, M.J.; Colella, P. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.* **1989**, *82*, 64–84. [https://doi.org/10.1016/0021-9991\(89\)90035-1](https://doi.org/10.1016/0021-9991(89)90035-1).
23. Leveque, R.J.; George, D.L.; Berger, M.J. Tsunami modelling with adaptively refined finite volume methods. *Acta Numer.* **2011**, *20*, 211–289. <https://doi.org/10.1017/S0962492911000043>.
24. Mandli, K.T.; Ahmadi, A.J.; Berger, M.; Calhoun, D.; George, D.L.; Hadjimichael, Y.; Ketcheson, D.I.; Lemoine, G.I.; Leveque, R.J. Clawpack: building an open source ecosystem for solving hyperbolic PDEs. *Peer J Computer Science* **2016**, *2*, e68. <https://doi.org/10.7717/peerj-cs.68>.
25. Burstedde, C.; Wilcox, L.C.; Ghattas, O. p4est: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees. *SIAM J. Sci. Comput.* **2011**, *33*, 1103–1133. <https://doi.org/10.1137/100791634>.
26. Kyanjo, B. GeoFlood wiki. Git repository, 2023.
27. KML. OGC KML 2.3. <https://www.ogc.org/standards/kml/>, 2015.
28. Soares-Frazão, S.; Alcrudo, F.; Goutal, N. Dam-break test cases summary: 4th CADAM meeting. In *Proceedings of the Proceedings from the Zaragoza Meeting, 18–19 November 1999*. CADAM Concerted Action on Dam-Break Modelling, European Commission, 1999, pp. 9–25. Zaragoza, Spain.
29. Morris, M. CADAM: Concerted Action on Dambreak Modelling, 2000.
30. Spero, H.; Calhoun, D.; Shubert, M. Simulating the 1976 Teton Dam Failure using Geoclaw and HEC-RAS and comparing with Historical Observations. *arXiv preprint arXiv:2206.00766* **2022**.
31. Brunner, G.W. HEC-RAS (river analysis system). In *Proceedings of the North American water and environment congress & destructive water*. ASCE, 2002, pp. 3782–3787.
32. Patel, D.P.; Ramirez, J.A.; Srivastava, P.K.; Bray, M.; Han, D. Assessment of flood inundation mapping of Surat city by coupled 1D/2D hydrodynamic modeling: a case application of the new HEC-RAS 5. *Nat. Hazards* **2017**, *89*, 93–130. <https://doi.org/10.1007/s11069-017-2956-6>.
33. Shahrim, M.; Ros, F. Dam break analysis of Temenggor dam using HEC-RAS. In *Proceedings of the IOP Conference Series: Earth and Environmental Science*. IOP Publishing, 2020, Vol. 479, p. 012041. <https://doi.org/10.1088/1755-1315/479/1/012041>.
34. Bai, F.; Yang, Z.; Huai, W.; Zheng, C. A depth-averaged two dimensional shallow water model to simulate flow-rigid vegetation interactions. *Procedia Eng.* **2016**, *154*, 482–489. <https://doi.org/10.1016/j.proeng.2016.07.542>.

35. Altaie, H.; Dreyfuss, P. Numerical solutions for 2D depth-averaged shallow water equations. In Proceedings of the Int. Math. Forum, 2018, Vol. 13, pp. 79–90. <https://doi.org/10.1298/imf.2018.712102>.
36. Qin, X.; Motley, M.; Leveque, R.; Gonzalez, F.; Mueller, K. A comparison of a two-dimensional depth averaged flow model and a three-dimensional RANS model for predicting tsunami inundation and fluid forces. *Nat. Hazards Earth Syst. Sci.* **2018**, *18*, 2489–2506. <https://doi.org/10.5194/nhess-18-2489-2018>.
37. Vreugdenhil, C.B., Shallow-water flows. In *Numerical Methods for Shallow-Water Flow*; Springer Netherlands: Dordrecht, 1994; pp. 1–14. https://doi.org/10.1007/978-94-015-8354-1_1.
38. Molls, T.; Zhao, G.; Molls, F. Friction slope in depth-averaged flow. *J. Hydraul. Eng.* **1998**, *124*, 81–85. [https://doi.org/10.1061/\(ASCE\)0733-9429\(1999\)125:5\(549\)](https://doi.org/10.1061/(ASCE)0733-9429(1999)125:5(549)).
39. Zhao, J.; Liang, Q. Novel variable reconstruction and friction term discretisation schemes for hydrodynamic modelling of overland flow and surface water flooding. *Adv. Water Resour.* **2022**, *163*, 104187. <https://doi.org/10.1016/j.advwatres.2022.104187>.
40. Song, L.; Zhou, J.; Li, Q.; Yang, X.; Zhang, Y. An unstructured finite volume model for dam-break floods with wet/dry fronts over complex topography. *Int. J. Numer. Methods Fluids* **2011**, *67*, 960–980. <https://doi.org/10.1002/fld.2397>.
41. Song, L.; Zhou, J.; Liu, Y.; Bi, S. A finite volume method for modeling shallow flows with wet-dry fronts on adaptive Cartesian grids. *Math. Probl. Eng.* **2012**, *2014*, 1–16. <https://doi.org/10.1155/2014/209562>.
42. Caleffi, V.; Valiani, A.; Zanni, A. Finite volume method for simulating extreme flood events in natural channels. *J. Hydraul. Res.* **2003**, *41*, 167–177. <https://doi.org/10.1080/00221680309499959>.
43. Yoshioka, H.; Unami, K.; Fujihara, M. A simple finite volume model for dam break problems in multiply connected open channel networks with general cross-sections. *Theor. Appl. Mech. Japan* **2014**, *62*, 131–140. <https://doi.org/10.11345/nctam.62.131>.
44. Leveque, R.J. Wave propagation algorithms for multidimensional hyperbolic systems. *J. Comput. Phys.* **1997**, *131*, 327–353. <https://doi.org/10.1006/jcph.1996.5603>.
45. Bale, D.S.; Leveque, R.J.; Mitran, S.; Rossmannith, J.A. A wave propagation method for conservation laws and balance laws with spatially varying flux functions. *SIAM J. Sci. Comput.* **2003**, *24*, 955–978. <https://doi.org/10.1137/S106482750139738X>.
46. Einfeldt, B. On Godunov-type methods for gas dynamics. *SIAM J. Num. Anal.* **1988**, *25*, 294–318. <https://doi.org/10.1137/0725021>.
47. Einfeldt, B.; Munz, C.D.; Roe, P.L.; Sjögren, B. On Godunov-type methods near low densities. *J. Comput. Phys.* **1991**, *92*, 273–295. [https://doi.org/10.1016/0021-9991\(91\)90211-3](https://doi.org/10.1016/0021-9991(91)90211-3).
48. Roe, P.L. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* **1981**, *43*, 357 – 372. [https://doi.org/10.1016/0021-9991\(81\)90128-5](https://doi.org/10.1016/0021-9991(81)90128-5).
49. Cea, L.; Bladé, E.; Sanz-Ramos, M.; Fraga, I.; Sañudo, E.; García-Feal, O.; Gómez-Gesteira, M.; González-Cao, J. Benchmarking of the Iber capabilities for 2D free surface flow modelling, 2020. <https://doi.org/10.17979/spudc.9788497497640>.
50. Frazão, S.S.; Zech, Y. Dam break in channels with 90 bend. *J. Hydraul. Eng.* **2002**, *128*, 956–968. [https://doi.org/10.1061/\(ASCE\)0733-9429\(2002\)128:11\(956\)](https://doi.org/10.1061/(ASCE)0733-9429(2002)128:11(956)).
51. Kyanjo, B.; Calhoun, D.; George, D.L. GeoFlood model, 2024. <https://doi.org/10.5281/zenodo.10929142>.
52. Kyanjo, B. Datasets used in GeoFlood comparison to other models, 2024. <https://doi.org/10.5281/zenodo.10897305>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.