

Lawrence Berkeley National Laboratory

LBL Publications

Title

Development of a River Dynamical Core for E3SM to simulate compound flooding on Exascale-class heterogeneous supercomputers

Permalink

<https://escholarship.org/uc/item/8gb489hm>

Journal

Environmental Modelling & Software, 197

ISSN

1364-8152

Authors

Bisht, Gautam

Xu, Donghui

Johnson, Jeffrey

et al.

Publication Date

2026-02-01

DOI

10.1016/j.envsoft.2025.106804

Copyright Information

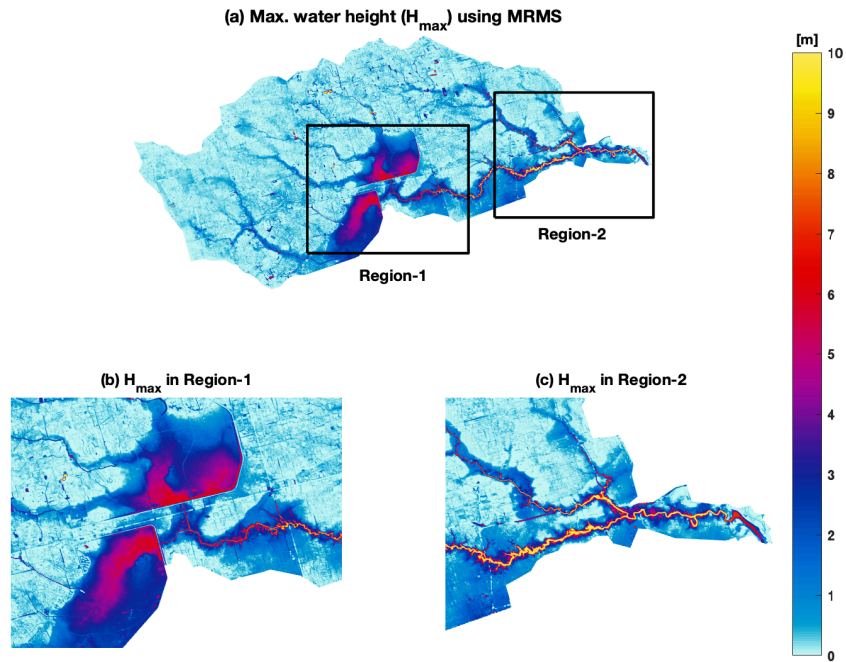
This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

Graphical Abstract

Development of a River Dynamical Core for E3SM to Simulate Compound Flooding on Exascale-class Heterogeneous Supercomputers

Gautam Bisht, Donghui Xu, Jeffrey Johnson, Jed Brown, Matthew Knepley, Mark Adams, Dongyu Feng, Dalei Hao, Darren Engwirda, Mukesh Kumar, Zeli Tan



Highlights

Development of a River Dynamical Core for E3SM to Simulate Compound Flooding on Exascale-class Heterogeneous Supercomputers

Gautam Bisht, Donghui Xu, Jeffrey Johnson, Jed Brown, Matthew Knepley, Mark Adams, Dongyu Feng, Dalei Hao, Darren Engwirda, Mukesh Kumar, Zeli Tan

- An open-source library, named RDycore, is developed for solving 2D shallow water equations
- The joint use of PETSc and libCEED allows RDycore to run efficiently on CPUs and GPUs
- The scalability of RDycore is demonstrated for a problem with 471 million grid cells
- A one-way coupling of RDycore within E3SM is applied to simulate flooding during Hurricane Harvey

Development of a River Dynamical Core for E3SM to Simulate Compound Flooding on Exascale-class Heterogeneous Supercomputers

Gautam Bisht^a, Donghui Xu^a, Jeffrey Johnson^b, Jed Brown^c, Matthew Knepley^d, Mark Adams^e, Dongyu Feng^a, Dalei Hao^a, Darren Engwirda^{f,g}, Mukesh Kumar^f, Zeli Tan^a

^a*Atmospheric, Climate, & Earth Sciences Division, Pacific Northwest National Laboratory, Richland, 99352, WA, USA*

^b*Cohere Consulting, LLC, Seattle, 98105, WA, USA*

^c*Department of Computer Science, University of Colorado Boulder, Boulder, 80309, CO, USA*

^d*Computer Science and Engineering, University of Buffalo, Buffalo, 14260, NY, USA*

^e*Applied Mathematics & Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, 94720, CA, USA*

^f*Fluid Dynamics and Solid Mechanics Group (T-3), Los Alamos National Laboratory, Los Alamos, 87545, NM, USA*

^g*Environment, Commonwealth Scientific and Industrial Research Organisation, Hobart, 7004, TAS, Australia*

^h*Civil, Construction and Environmental Engineering, University of Alabama, Tuscaloosa, 35487, AL, USA*

Abstract

Physically-consistent quantification of flood risks in global models requires kilometer scale flood simulations using 2D physics schemes, both of which are unavailable in the current generation Earth System Models. In this work, we have developed the River Dynamical Core (RDycore), which is an open-source, 2D shallow water equation library for the Energy Exascale Earth System Model (E3SM). RDycore uses PETSc and libCEED libraries to run efficiently on CPUs and GPUs. RDycore was validated for analytical, manufactured, and a well-studied dam break problem. For a problem with 471

million grid cells, RDycore achieves a speedup of 6.6x and 7.6x on GPUs compared to CPUs on Perlmutter and Frontier supercomputers, respectively. The one-way E3SM-RDycore coupling is demonstrated by performing multiple 5-day flooding simulations during Hurricane Harvey driven by five precipitation datasets. The work presented here is the foundational step in providing hardware and algorithmic portability for simulating kilometer-scale river dynamics within E3SM.

Keywords: Shallow water equations, Heterogeneous computing architectures, River modeling, Flood simulations

1. Introduction

Flooding accounted for the largest fraction (44%) of weather-related disaster events from 2000-2019 and also affected the most number of people (1.6 billion) worldwide [1]. The number of global extreme flood events has steadily increased between 1985 and 2019 [2]. As of 2024, nine of the ten most expensive billion-dollar weather and climate disasters in the United States involved flooding, with an estimated cost between \$46.3Bn and \$200Bn [3]. The drivers of flooding include extreme precipitation, river flow, and coastal storm surges that lead to pluvial, fluvial, and coastal flooding, respectively [4]. The flood risk can be compounded by the joint occurrences of flood drivers, such as the interaction between storm surge and river discharge at the land-ocean boundary [5].

The flooding risk is expected to increase in many regions of the world [6]. More moisture can be held by the atmosphere at higher temperatures, which has been shown to increase, intensify, and prolong extreme precipitation

events in future climate projections [7]. The flood risk due to rain-on-snow events increases at high elevations and latitudes in a warmer climate as more precipitation falls as rain rather than snow on existing snowpack, causing it to melt [8]. The coastal flooding risk increases exponentially due to sea level rise (SLR) [9], and even the Tropics, with smaller wave heights, are expected to experience a greater increase in flooding frequency due to SLR [10]. The probability of compound flooding due to tropical storms and storm surges in urban coastal areas has also increased [11].

However, the current-generation Earth System Models (ESMs) cannot accurately quantify the local risk of flooding because of the use of simplified physics schemes and coarse spatial resolution in their river model. ESMs, including the Energy Exascale Earth System Model (E3SM), routinely use a one-dimensional (1D) river routing scheme that simulates the one-way water transport through a river network from the land to the ocean [12, 13]. Such models only account for the transport within the main river channel that has a representative channel geometry and the spatial extent of the fluvial flooding when the river overflows its bank is computed using a floodplain elevation profile derived from a digital elevation model (DEM) at finer resolution [14]. Recently, the E3SM model components were extended to allow for the two-way exchange of water between the land-river [15] and river-ocean [16] components, though the physics within the river model remained 1D. Furthermore, the typical spatial resolution of the land and river components of ESMs participating in the Coupled Model Intercomparison Project Phase 6 (CMIP6) was 100 km, and the High Resolution Model Intercomparison Project (HighResMIP) for CMIP6 had a target resolution of 25-50km.

However, policymakers need information about flood risk at or finer than kilometer scale (k-scale) [17]. With the recent advances in high-performance computing, the spatial resolution of ESMs is increasing to k-scale. A set of 40-day simulations was performed at 1 and 3.25 km in the atmosphere component of E3SM [18], a 30-day simulation with 5.5 km was performed using the LASG/IAP climate system ocean model version 3 [19], and an 8 month coupled simulation with resolution ranging from 2.8km (atmosphere) to 4km (ocean) was performed using ECMWF’s IFS [20]. These global simulations are the first steps toward century-long climate simulations at the k-scale.

Uncoupled to ESMs, high-fidelity hydrodynamic models have been applied at k-scale or finer spatial resolution to provide the risk of future flooding. The application of the Two-Dimensional Runoff Inundation Toolkit for Operational Needs (TRITON) [21], a 2D structured grid SWE model, showed that future warming would lead to a larger maximum inundation extent for a Harvey-like hurricane [22]. A modified version of LISFLOOD-FP [23] that accounted for storm surge and rainfall intensity was applied to quantify the compound flooding risk in New York City due to storm surge and rainfall during tropical and extratropical cyclones [11]. LISFLOOD-FP has also been applied to predict fluvial, pluvial, and coastal flood risk in the US [4] and UK [24] under current and future climate. Recently, global flood inundation maps were developed using LISFLOOD-FP at an unprecedented 30 m spatial resolution and simulated future change in fluvial, pluvial, and coastal flooding [25]. Instead of performing a single global LISFLOOD-FP simulation at 30 m, the global river network and coastline were broken into overlapping independent domains, and simulations were performed for ten annual flood

return periods. [25] reported that these simulations took “several months to compute on a cluster consisting of 2500 computational cores.”

The increase in the computational cost associated with using a high-fidelity hydrodynamic model at k-scale within an ESM can be offset by using computational libraries that efficiently utilize heterogeneous computing architectures of Exascale class supercomputers. Homogeneous CPU-based supercomputers are transitioning to heterogeneous computing architectures that include CPUs and GPU accelerators; thus, the new model developments must include support for GPUs for longevity. Presently, the U.S. Department of Energy’s supercomputers at the National Energy Research Scientific Computing Center (NERSC), the Oak Ridge Leadership Computing Facility (OLCF), and the Argonne Leadership Computing Facility (ALCF) have GPU accelerators from different vendors. The Perlmutter supercomputer at NERSC has NVIDIA GPUs, the Frontier Exascale supercomputer at OLCF has AMD GPUs, and the Aurora Exascale supercomputer at ALCF has Intel GPUs. Currently, there are many GPU programming models, such as Kokkos [26], RAJA [27], CUDA [28], ROCm [29], and OpenCL [30], and a few 2D SWE models have used these programming languages to add support for GPUs. For example, HiPMS [31], TRITON, and LISFLOOD-FP 8.1 [32] use CUDA to include support for NVIDIA GPUs, while SERGHEI-SWE [33] uses Kokkos to support GPUs from multiple vendors. While some previously mentioned programming languages can provide hardware portability, higher level abstractions are required for applications to achieve algorithmic portability. Thus, hydrodynamic models must natively implement spatial and temporal discretization approaches for solving SWE. Numerical libraries such as

the Portable Extensible Toolkit for Scientific computation (PETSc) [34] and the library for Code for Efficient Extensible Discretization (libCEED) [35] can provide hardware and algorithmic portability. PETSc includes support for NVIDIA, AMD, and Intel GPUs [36] and the time-stepping object, `TS`, in PETSc provides multiple algorithms for ordinary differential equations and differential algebraic equations. libCEED provides fast element-based algebraic computation for CPUs and GPUs, and uses just-in-time compilation to choose the appropriate GPU back-end. The joint use of PETSc and libCEED has been previously used in a solid mechanics application for solving problems with up to 1.9 billion unknowns and scalable performance is achieved by ensuring all data remains on GPU during a solve [37].

This work aims to develop a SWE-based river model for E3SM to accurately model pluvial, fluvial, and coastal flooding while efficiently using heterogeneous computing architectures. In this study, we developed a standalone open-source, 2D SWE-based river dynamical core (RDycore) library for E3SM that supports structured and unstructured grids. RDycore uses first-order finite volume spatial discretization and explicit time integration to solve the discretized equations. PETSc and libCEED are used to provide RDycore with hardware and algorithmic portability. The numerical implementation of RDycore was verified against both analytical and manufactured solutions. Additionally, RDycore was benchmarked against existing SWE models for the well-studied Malpasset dam break problem. The computational performance of RDycore was studied for a problem with 1.4 billion unknowns on Perlmutter and Frontier. The initial integration of the RDycore library within E3SM was completed, and the flooding simulation during

Hurricane Harvey was used to benchmark the model integration against a previously published study [38]. The model developed in this study provides the foundation for E3SM to use a scalable and hardware-agnostic approach to accurately resolve multiple types of flooding globally at k-scale.

2. Numerical Formulations and Implementation of RDycore

In this section, we describe the numerical discretization of the 2D SWE implemented in RDycore, the software framework used for developing the RDycore library, how numerical libraries provide hardware and algorithmic portability, and the initial coupling of RDycore within E3SM. The 2D SWE numerical discretization implemented in RDycore is based on the approach used in the Overland Flow Model (OFM) [39] with one exception. The correction terms for the momentum equations for well-balancing [40] that is included in OFM, is presently not included in RDycore. The parallel version of OFM uses PETSc to run on multiple CPUs [2], however, the joint use of PETSc and libCEED allows RDycore to run on CPUs and GPUs.

2.1. Governing Equations and Numerical Discretizations

The conservative form of the 2D SWEs that are solved in RDycore are given by

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S}_r + \mathbf{S}_b + \mathbf{S}_f \quad (1)$$

where x and y represents the Cartesian space, t represents the time domain, $\mathbf{U}$ is the conservative flow vector, $\mathbf{E}$ and $\mathbf{G}$ are the flux vectors in x - and y -direction, respectively, $\mathbf{S}_r$ is the source term for water associated with the runoff production, $\mathbf{S}_b$ is the source term for momentum due to bed slope, and

$\mathbf{S}_f$ is the source term for momentum due to bed friction. The conservative flow vector is

$$\mathbf{U}^\top = \begin{bmatrix} h & hu & hv \end{bmatrix}^\top \quad (2)$$

where h is the water height, u is the velocity in x -direction, and v is the velocity in y -direction. The flux vectors in x - and y -direction are given as

$$\mathbf{E}^\top = \begin{bmatrix} hu & hu^2 + \frac{1}{2}gh^2 & huv \end{bmatrix}^\top \quad (3)$$

$$\mathbf{G}^\top = \begin{bmatrix} hv & huv & hv^2 + \frac{1}{2}gh^2 \end{bmatrix}^\top \quad (4)$$

where g is the gravity acceleration. The three source terms in equation 1 are given as

$$\mathbf{S}_r^\top = \begin{bmatrix} R & 0 & 0 \end{bmatrix}^\top \quad (5)$$

$$\mathbf{S}_b^\top = \begin{bmatrix} 0 & -gh\frac{\partial z}{\partial x} & -gh\frac{\partial z}{\partial y} \end{bmatrix}^\top \quad (6)$$

$$\mathbf{S}_f^\top = \begin{bmatrix} 0 & -C_D u \sqrt{u^2 + v^2} & -C_D v \sqrt{u^2 + v^2} \end{bmatrix}^\top \quad (7)$$

where R is the total runoff rate, z is the bed elevation, $C_D = gn^2/h^{\frac{1}{3}}$ is the drag coefficient, and n is the Manning's coefficient.

RDycore uses the finite volume scheme for spatially integrating equation 1 over an arbitrary two-dimensional control volume Ω_i with a boundary $d\Omega_i$:

$$\begin{aligned} \int_{\Omega_i} \frac{\partial \mathbf{U}}{\partial t} d\Omega + \int_{\Omega_i} \frac{\partial \mathbf{E}}{\partial x} d\Omega + \int_{\Omega_i} \frac{\partial \mathbf{G}}{\partial y} d\Omega + &= \int_{\Omega_i} (\mathbf{S}_r + \mathbf{S}_b + \mathbf{S}_f) d\Omega \\ \frac{d}{dt} \int_{\Omega_i} \mathbf{U} d\Omega + \oint_{d\Omega_i} (\mathbf{E} dy - \mathbf{G} dx) &= \int_{\Omega_i} (\mathbf{S}_r + \mathbf{S}_b + \mathbf{S}_f) d\Omega \\ \frac{d}{dt} \int_{\Omega_i} \mathbf{U} d\Omega + \oint_{d\Omega_i} (\vec{\mathbf{F}} \cdot \vec{n}) dl &= \int_{\Omega_i} (\mathbf{S}_r + \mathbf{S}_b + \mathbf{S}_f) d\Omega \\ A_i \frac{d\mathbf{U}_i}{dt} + \sum_j (\vec{\mathbf{F}}_{ij} \cdot \vec{n}_{ij}) L_{ij} &= \mathbf{S}_{r,i} A_i + \mathbf{S}_{b,i} A_i + \mathbf{S}_{f,i} A_i \quad (8) \end{aligned}$$

where A_i is the area of the i -th control volume, L_{ij} is the length of the edge shared by i -th and j -th control volumes, $\vec{\mathbf{F}}_{ij}$ is the flux vector between i -th and j -th control volumes, and $\vec{n}_{ij}$ is the outward pointing unit vector between i -th and j -th control volumes. RDycore adopts the Roe’s Riemann solver to compute the flux vector (Appendix A), and spatial discretization of the source terms and boundary conditions can be found in Appendix B and Appendix C, respectively. The spatially discretized 2D SWE given by equation 8 is temporally integrated using forward Euler time integration, except the bed friction term, which uses semi-implicitly discretization as denoted by t^* for numerical stability, and can be written as

$$\frac{\mathbf{U}_i^{t+1} - \mathbf{U}_i^t}{\Delta t} = - \sum_j \left(\vec{\mathbf{F}}_{i,j}^t \cdot \vec{n}_{i,j} \right) \left(\frac{L_{i,j}}{A_i} \right) + \mathbf{S}_{\mathbf{r}_i}^t + \mathbf{S}_{\mathbf{b}_i}^t + \mathbf{S}_{\mathbf{f}_i}^{t*} \quad (9)$$

The temporal discretization scheme used in RDycore is presented in Appendix D.

2.2. Software Development Framework of RDycore

RDycore is developed as a numerical library rather than a model to facilitate easy integration into existing numerical models. Like many DOE-funded model development projects, RDycore follows an open-source, open-development policy that allows public access to release and development versions of the code. RDycore is available at <https://github.com/rdycore/rdycore> under a 2-clause BSD license. The library is written in C and includes Fortran bindings for coupling with ESMs that are typically written in Fortran. A CMake-based build system allows RDycore to run on machines ranging from personal laptops to DOE’s leadership-class computing facilities. Automated unit testing, integration testing, and code coverage

reporting provide quality control on new model developments. Simulation input is structured using YAML, which can be easily generated and parsed with readily available tools. Structured and unstructured meshes are supported. RDycore’s unstructured mesh can include any mix of triangles and quadrilaterals. RDycore uses PETSc’s `DMplex` object for mesh management [41, 42] ParMETIS for parallel domain decomposition [43]. RDycore can read meshes from files in the Exodus II [44] format, as well as PETSc’s `DMplex` HDF5 [45, 46] format. `DMplex`’s HDF5 format is more appropriate for larger meshes, since this format allows a mesh to be read in parallel, resulting in faster simulation startup times. Checkpoint and restart capabilities are included for long-running simulations. RDycore provides a simple but effective mechanism for running ensembles within a single job without requiring external workflow tools. RDycore supports 32-bit and 64-bit integer indices, allowing for simulations with billions of unknowns. User, technical, and developer guides are available at <https://rdycore.github.io/RDycore/>.

2.3. Hardware and Algorithmic Portability in RDycore

The combined use of PETSc and libCEED enables RDycore to run efficiently at scale on CPUs and GPUs, including those from NVIDIA, AMD, and Intel. PETSc’s `TS` provides RDycore with multiple explicit time integration methods such as forward Euler and Runge-Kutta (RK) schemes, including those with strong stability-preserving methods. An RDycore simulation is first spatially decomposed across multiple CPUs using the Message Passing Interface (MPI). In simulations using GPUs, a single GPU is mapped to a single MPI rank.

An example of an idealized mesh containing 3×2 ($= 6$) grid cells with

natural cell IDs of grid cells N0 – N5 is shown in Figure 1(a). The example mesh is decomposed across two MPI ranks and the grid cells present locally on MPI ranks 0 and 1 are shown in gray and yellow respectively. In addition to locally owned grid cells, each MPI rank also has halo grid cells, e.g. N2 and N5 on MPI rank 0 and N1 and N4 on MPI rank 1. On each MPI rank, the grid cell IDs are numbered locally, starting with L0 (Figure 1(b-c)). RDycore’s mesh data structure stores information about internal edge connections between two cells (shown in red) and boundary edge connections between a cell and a boundary (shown in blue). The internal edges between locally owned cells and a halo cell are present only on one of the MPI ranks. For example, internal edges between N1–N2 and N4–N5 are only present on MPI rank 0.

In order to understand the use of PETSc and libCEED, RDycore’s spatially discretized SWE given by the equation 9 can be written as:

$$\frac{d\mathbf{U}}{dt} = G(t, \mathbf{U}) \equiv \text{RHSFunction} \quad (10)$$

where G , a nonlinear function of $\mathbf{U}$, includes fluxes through internal and boundary edges, and the three source terms. One provides a C function to compute G , `RHSFunction`, to PETSc’s TS operator. This function is evaluated multiple times per timestep depending on the selected method of temporal integration. While PETSc includes native support for multiple GPU back-ends within the time integration algorithms, the RDycore assumes responsibility for performing the function evaluation on the GPU itself, to avoid the cost of transferring data between the CPU and GPU.

We use libCEED in RDycore to evaluate `RHSFunction` in a hardware-agnostic manner. In this case, the evaluation of the function is performed

by three `CeedOperators`: one to compute fluxes through internal edges, another to compute fluxes through boundary edges, and a third to compute source terms. These calculations are indicated by blue, green, and pink boxes in Figure 1(d-f). Each `CeedOperator` consists of (1) an `ElemRestriction` that describes the local grid cell IDs for each algebraic expression, and (2) a `QFunction` that defines the algebraic expression. For example, the `CeedOperator` on MPI rank 1 associated with the flux through an internal edge has a single `ElemRestriction` that corresponds to grid cells L0 and L1 (Figure 1(f)). The associated `QFunction` named `SWEFlux_Roe` solves an approximate Roe Riemann problem using grid cell values left (L0) and right (L1) of the internal edge. `RDycore` supports critical outflow and reflective wall boundary conditions, which can be applied to different boundary edges of the mesh (as shown in the green box of Figure 1(e-f)).

A key feature of PETSc and libCEED is the ability to select an algorithm and the hardware (CPU or GPU) at runtime without requiring code to be recompiled. Appending `-ceed /gpu/cuda -dm_vec_type cuda` to the `RDycore`'s run command instructs PETSc and libCEED to use the CUDA back-ends for NVIDIA GPUs, while HIP for AMD GPUs can be selected via `-ceed /gpu/hip -dm_vec_type hip`. Similarly, time integration algorithms such as forward Euler, RK, or strong stability preserving RK can be selected by setting the runtime option `-ts_type` to be `euler`, `rk`, and `ssp`, respectively.

2.4. One-way Coupling of *RDycore* with *E3SMv2*

E3SM is a state-of-science ESM that aims to efficiently use DOE's scientific technology for providing a robust actionable prediction of Earth system

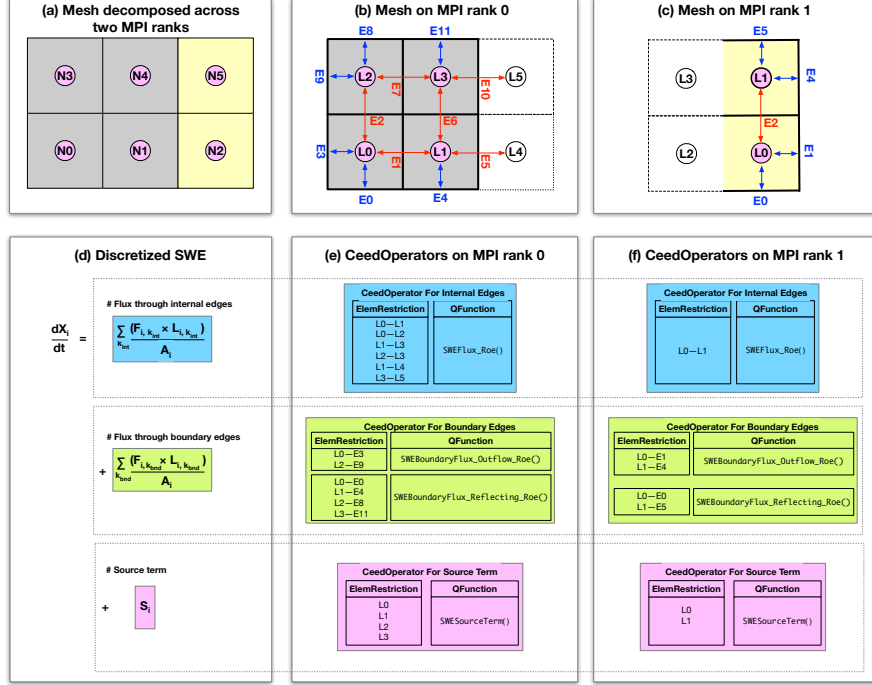


Figure 1: An overview of the joint use of PETSc and libCEED to solve RDycore's discretized SWE. (a) A schematic representation of a mesh consisting of six grid cells with cell IDs in “natural” order from N0 to N5. The mesh is decomposed across two MPI ranks, with cells owned by MPI rank 0 and 1 shown in gray and yellow, respectively. (b-c) The portion of the mesh present on each MPI ranks with owned cells (shown with gray or yellow boxes) and halo cells (shown with unfilled boxes). The cell IDs are in the “local” order starting with L0. The mesh contains internal edges between cells (shown in red) and boundary edges (shown in blue). The edges present on each MPI rank are shown with bold black lines. (d) RDycore's spatially discretized SWE can be split into three terms representing fluxes through internal edges, fluxes through boundary edges, and source terms. The flux through an edge is computed using the Roe approximate Riemann solver. (e-f) CeedOperators with element restrictions (ElemRestriction) and QFunctions on each MPI rank corresponding to the three terms of the spatial discretization. An outflow boundary condition is applied on the east and west boundary of the mesh. A reflective boundary condition is applied on the north and south boundary. An additional description of CeedOperators is presented in Section 2.3.

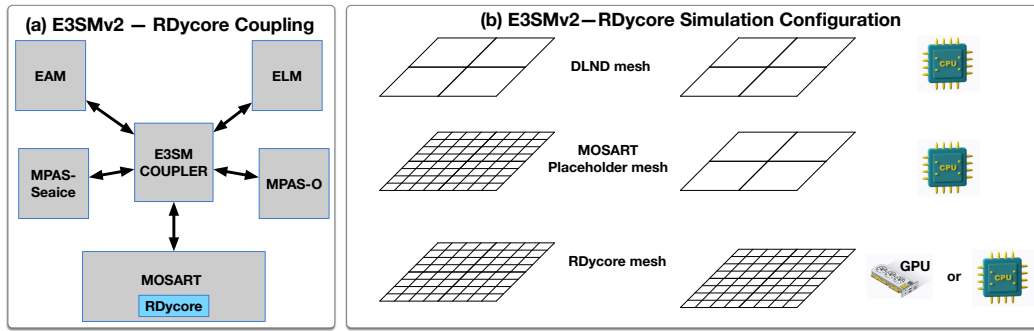


Figure 2: An overview of E3SM–RDycore coupling. (a) The components of E3SMv2 include models for the atmosphere (EAM), land (ELM), river (MOSART), ocean (MPAS-O), and sea ice (MPAS-Seaice). E3SM coupler exchanges fluxes and states between the components. RDycore is coupled within E3SMv2 as part of MOSART and can be turned on/off for a simulation. (b) The E3SMv2–RDycore coupling is tested in the data land (DLND) configuration. In a simulation with RDycore, a placeholder MOSART mesh must be used at the same or coarser resolution as RDycore’s mesh. DLND and MOSART run on CPUs, while RDycore can run on CPUs or GPUs.

variability and change [47]. E3SM version 2 (E3SMv2) includes models for the atmosphere, land, river, ocean, and sea ice (Figure 2(a)). The typical horizontal resolution in E3SMv2 components are 110 km in the atmosphere, 165 km in the land, 50 km in the river, and 30 – 60 km in the ocean and sea ice. The E3SM coupler exchanges state variables and fluxes between different components. The land component of E3SM (E3SM Land Model, or ELM), was developed based on the Community Land Model 4.5 [48], while the river component of E3SM uses the Model for Scale Adaptive River Transport (MOSART) [13]. MOSART first routes the surface runoff from ELM across the hillslope to tributaries. Then the discharged surface runoff and ELM’s subsurface runoff are transported via a sub-network of tributaries to the main river channel. Lastly, the water in the main channel is transported to the ocean. The default configuration of MOSART in E3SMv2 uses a 1D kinematic wave routing method. Other relevant E3SM modeling capabilities that are supported in specific E3SM configurations include a 1D diffusion wave routing scheme for backwater effects and floodplain inundation in MOSART [14], two-way interaction between MOSART and ELM to allow for the infiltration of floodplain inundated water within ELM’s soil column [15], and the two-way interaction between MOSART and ocean component (MPAS-O) for simulating compound flooding [16].

Our long-term goal is to add RDycore as an independent component within E3SM that communicates directly with the E3SM coupler. However, in this initial work, we have developed a one-way coupling between RDycore and E3SMv2 in which RDycore becomes part of MOSART. A placeholder MOSART mesh must be used (Figure 2(b)). The meshes of MOSART and

RDycore can be decomposed differently across CPUs. The MOSART placeholder mesh is registered with the E3SM Coupler to receive total runoff from the land model. Unlike MOSART, RDycore does not differentiate between surface and subsurface runoff, so the total runoff rate is used in the calculation of $\mathbf{S}_r$ in Equation 5. In addition to active components, E3SM supports a “data mode” for various components in which the model states and/or fluxes are read from files and sent to the E3SM coupler. In this work, we have used the data land (DLND) configuration to test the RDycore coupling with E3SMv2 for flooding during Hurricane Harvey. A detailed description of this configuration is described later in Section 3.4. Lastly, while the DLND and MOSART can only use CPUs, RDycore can use CPUs or GPUs via runtime options.

3. Verification and Validation

3.1. Problems With Analytical Solutions

Two idealized 1D dam break problems over a frictionless bed with analytical solutions described in the Shallow Water Analytic Solutions for Hydraulic and Environmental Studies (SWASHES) [49] were used to verify RDycore. A 1D bed of length L with a flat bottom has an initial water discontinuity located at x_0 that is given by

$$h(x) = \begin{cases} h_l & \text{for } 0 \leq x \leq x_0 \\ h_r & \text{for } x_0 < x \leq L \end{cases} \quad (11)$$

with h_l and h_r are initial water height [m] and $h_l \geq h_r$. The water on either side of the dam is initially at rest (i.e., $u(x) = 0 \text{ ms}^{-1}$). The two verification

cases correspond to a dry and wet domain right of the dam, respectively. The parameters common to both cases are $h_l = 0.005$ m, $x_0 = 5$ m, and $L = 10$ m. The values of h_r for the dry and wet domain cases are 0.0 and 0.001 m, respectively. For each case, RDycore simulations were performed for four successively refined grids containing 100, 1000, 10000, and 100000 quadrilateral elements. The analytical solutions of the water height and water velocity for the two cases are provided in Appendix E. The L_1 and L_2 errors for water height and momentum between the simulated and analytical solutions were computed using the following equations:

$$L_1 = \frac{1}{\Omega} \sum_{n=1}^N |u_{sim,n} - u_{ana,n}| \omega_n \quad (12)$$

$$L_2 = \left(\frac{1}{\Omega} \sum_{n=1}^N |u_{sim,n} - u_{ana,n}|^2 \omega_n \right)^{1/2} \quad (13)$$

where $u_{sim,n}$ and $u_{ana,n}$ are the simulated and analytical water height or momentum for the n -th cell, ω_n is the area of the n -th cell, and Ω is the area of the whole domain. Additionally, a convergence rate, R , is computed based on L_1 error norm and grid spacing, Δx , at two mesh resolutions, $k1$ and $k2$, using the following equation:

$$R = \log \left(\frac{(L_1)_{k1}}{(L_1)_{k2}} \right) \bigg/ \log \left(\frac{\Delta x_{k1}}{\Delta x_{k2}} \right) \quad (14)$$

As the number of grid points is increased, the agreement between simulated and analytical solutions improves for the two problem types (Figure 3). The L_1 and L_2 error norms decrease for water height and momentum for the two problem types as the mesh is refined (Table 1). The value of R is

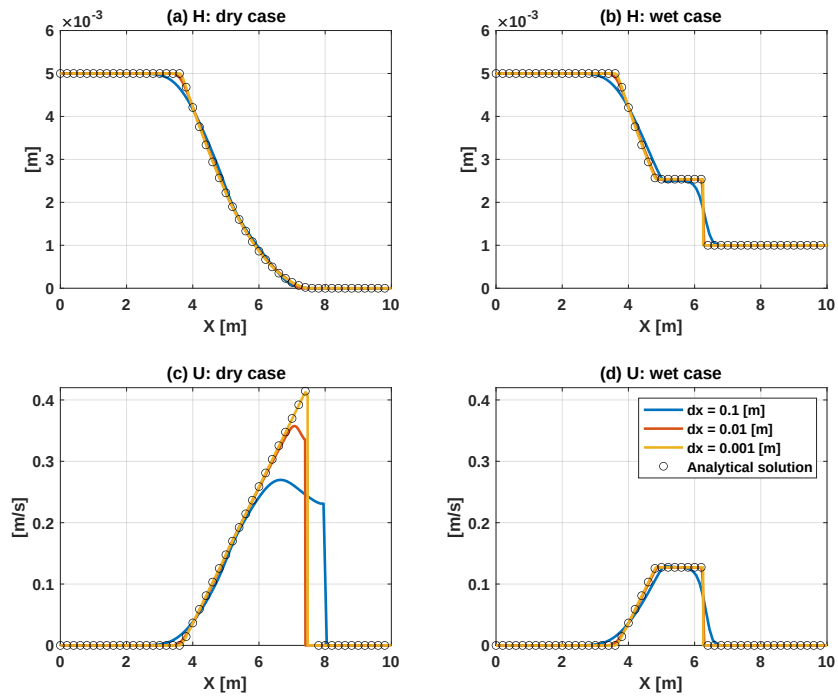


Figure 3: The comparison between analytical solution and RDycore simulation of (a-b) water height and (c-d) velocity in x -direction for the 1D dam break problem for the dry and wet bed case as described in Section 3.1.

Table 1: The error norms, L_1 and L_2 , and the estimated convergence rate, R , for RDycore-simulated water height, h , and momentum in x -direction, hu , for the 1D dry and wet frictionless dam break problems from SWASHES.

Problem type	N	$L_1(h)$	$L_2(h)$	$R(h)$	$L_1(hu)$	$L_2(hu)$	$R(hu)$
Dry dam break	100	4.54e-04	2.81e-04	-	8.05e-05	5.29e-05	-
	1,000	1.15e-04	8.03e-05	0.60	2.19e-05	1.73e-05	0.57
	10,000	2.01e-05	1.73e-05	0.76	3.93e-06	3.94e-06	0.75
	100,000	2.62e-06	2.85e-06	0.88	5.95e-07	6.50e-07	0.82
Wet dam break	100	6.47e-04	4.38e-04	-	1.11e-04	8.63e-05	-
	1,000	1.09e-04	1.29e-04	0.77	1.85e-05	2.68e-05	0.78
	10,000	1.67e-05	4.29e-05	0.81	2.99e-06	1.03e-05	0.79
	100,000	2.58e-06	2.73e-05	0.81	5.76e-07	6.08e-06	0.72

slightly lower than the theoretically expected value of 1.0 for first-order spatial discretization method; however, the R value reported here agrees with those obtained by SERGHEI-SWE, which similarly uses first-order spatial discretization. The average R values for height obtained by SERGHEI-SWE for the wet dam break problem was 0.81 [33]. The errors are expected to be larger for the dry dam break case than the wet dam break due to the sharp wet-dry transition.

3.2. Method of Manufactured Solutions

The Method of Manufactured Solutions (MMS) is a code verification technique that does not require a problem to have an analytical solution, but rather “manufactures” a solution for the verification problem [50]. Symbolic manipulation of the governing equations is performed for the given

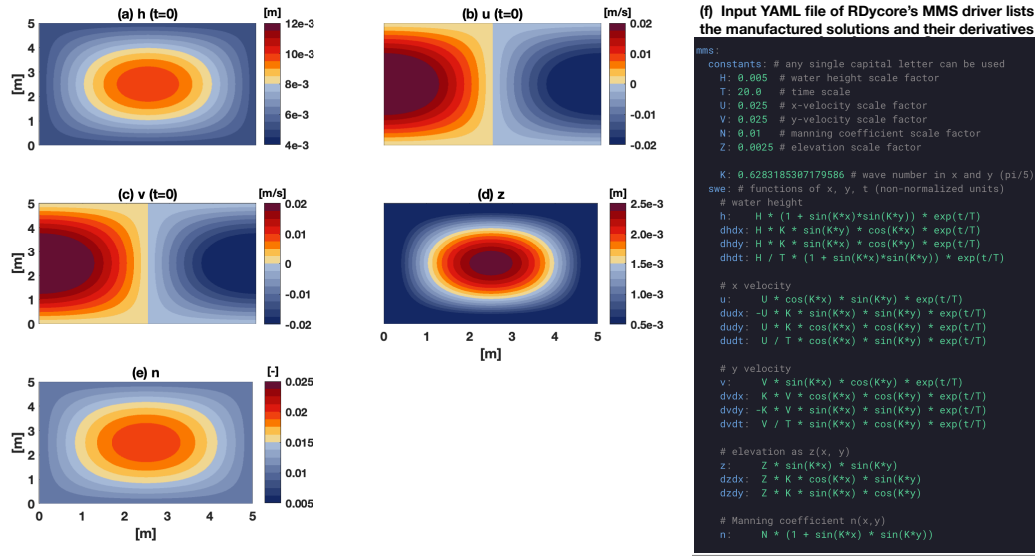


Figure 4: An overview of MMS framework used for verifying RDycore. (a-e) The manufactured solutions used for verifying RDycore as given by equations 15–19. (f) A snippet of the input YAML file for RDycore’s MMS driver showing the manufactured solutions and their derivatives that is parsed using `muparser`.

manufactured solution to obtain analytical expressions for the corresponding boundary conditions and source terms. A numerical simulation using those boundary conditions and source terms, along with an initial condition corresponding to the manufactured solution, is expected to produce results that match the manufactured solution. The difference between the simulation result and the manufactured solution is then used to verify the model’s numerical implementation. An advantage of MMS is that it not only allows for variables of the governing equations to vary in space and time, but it can also accommodate variations in model parameters. The MMS requires manufactured solutions to vary smoothly in space and time, so the derivatives of such solutions are continuous in space and time. MMS also requires the numerical model to support spatially varying initial conditions and spatiotemporally varying boundary conditions and source terms. The MMS technique has been previously used to verify the physics scheme of a standalone library for ELM for the transport of water through the soil–root–xylem continuum and water and energy transport in the subsurface [51].

The manufactured solutions used for verifying RDycore has spatiotemporally varying water height and water velocities in x - and y -direction, along with spatially varying bed elevation and Manning’s roughness coefficient that are given as:

$$h = H \left[1 + \sin \left(\frac{\pi x}{L_x} \right) \sin \left(\frac{\pi y}{L_y} \right) \right] \exp \left(\frac{t}{T} \right) \quad (15)$$

$$u = U \cos \left(\frac{\pi x}{L_x} \right) \sin \left(\frac{\pi y}{L_y} \right) \exp \left(\frac{t}{T} \right) \quad (16)$$

$$v = V \sin \left(\frac{\pi x}{L_x} \right) \cos \left(\frac{\pi y}{L_y} \right) \exp \left(\frac{t}{T} \right) \quad (17)$$

$$z = Z \sin \left(\frac{\pi x}{L_x} \right) \sin \left(\frac{\pi y}{L_y} \right) \quad (18)$$

$$n = N \left[1 + \sin \left(\frac{\pi x}{L_x} \right) \sin \left(\frac{\pi y}{L_y} \right) \right] \quad (19)$$

where $H = 0.005$ m, $U = 0.025$ ms⁻¹, $V = 0.025$ ms⁻¹, $T = 20.0$ s, $Z = 0.0025$ m, $N = 0.01$ [-], and $L_x = L_y = 5$ m.

The manufactured solutions at $t = 0$ are shown in Figure 4(a-e). The source terms for SWE corresponding to the abovementioned manufactured solution requires computing the derivatives of those five variables with respect to x , y , and t . The input YAML file for the RDycore's MMS driver lists the manufactured solutions and their derivatives. and the **muparser** (<https://github.com/beltoforion/muparser>) is to parse those mathematical expressions into code Figure 4(f). The source terms for SWE are presented in Appendix F.

RDycore was verified by computing L_1 , L_2 , and L_∞ error norms for water height and momentum in two directions. The simulations were performed at four successively refined grids with equal grid spacing in both directions, which were 0.5, 0.25, 0.125, and 0.0625 m. Error convergence was estimated by fitting a regression line between the observed error and grid spacing. All three error norms decrease as the mesh is refined (Figure 5a-c). The

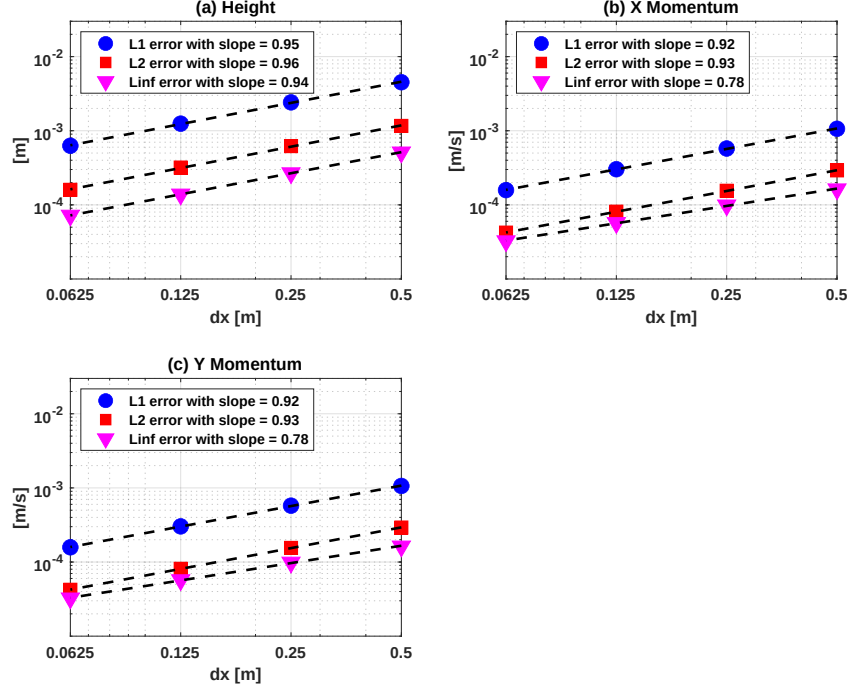


Figure 5: The L_1 , L_2 , and L_∞ norm for RDycore's simulated (a) water height, (b) momentum in x -direction, and (c) momentum in y -direction for the 2D manufactured problem described in Section 3.2.

error convergence rate of L_1 and L_2 norms is close to the expected value of 1.0 for the three prognostic variables, while the error convergence of the L_∞ norm is slightly lower. These expected error convergence rates provide confidence that the numerical discretizations are correctly implemented in RDycore library.

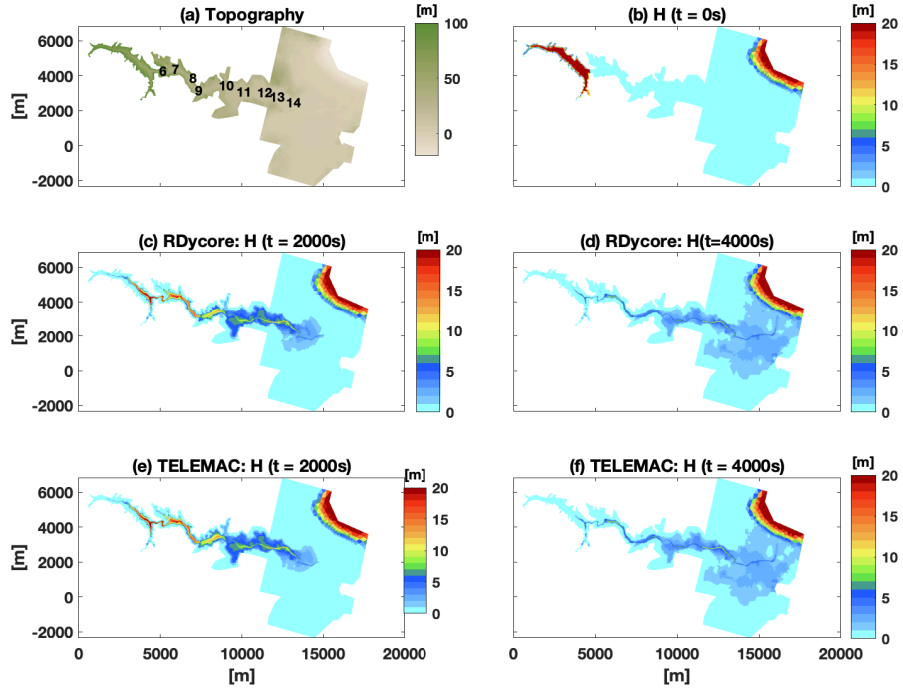


Figure 6: An overview and results for the Malpasset dam break problem. (a) Topography and the location of gauges within study domain. (b) Initial water height for the simulations. RDycore simulated water height using the TELEMAC mesh at (c) time = 2000 sec and (d) time = 4000 sec. TELEMAC simulated water height at (e) time = 2000 sec and (f) time = 4000 sec.

3.3. Malpasset Dam Break Problem

The Malpasset dam break problem is a widely used test case for benchmarking 2D SWE models [52, 33]. The Malpasset dam was located on the Reyran River, about 12 km upstream of Frejus in southern France, and it failed in December 1959, causing more than 400 casualties. The topography of the computation domain and the location of nine gauges are shown in Figure 6(a), while the initial water height is shown in Figure 6(b). The RDycore simulation for this problem was compared with previously published results using TELEMAC [53] and PIHM_Hydro [52]. The two published studies used triangular meshes containing 26,000 (TELEMAC) and 38,136 (PIHM_Hydro) triangular elements. RDycore simulations were compared with each model using the mesh used in its simulation. The time step and simulation duration for RDycore simulations were 0.01 sec and 4000 sec, respectively. The simulated maximum inundated depth for all three models was evaluated against observations at the multiple gauge sites using statistical metrics such as bias, root mean square error (RMSE), and coefficient of determination (R^2).

After the dam breaks, the water floods the domain from left to right. The spatial distribution of water height at 2000 and 4000 sec simulated by RDycore (Figure 6(c-d)) agrees well with TELEMAC (Figure 6(e-f)). Compared to observations at the nine gauges, the RDycore performance is comparable for TELEMAC and PIHM_Hydro for the three statistical metrics (Table 2). The bias and RMSE of RDycore's results is smaller than that of TELEMAC. PIHM_Hydro, which uses the second-order spatial discretization method, has a smaller bias but a larger RMSE as compared to RDycore. The R^2 of all three models against observations is close to 1.0. These results give con-

Table 2: Bias, RMSE, and R^2 of the maximum water surface elevation simulated by RDycore, TELEMAC, and PHIM_Hydro as compared to observations for the Malpasset dam break problem.

	TELEMAC	RDycore using TELEMAC grid	PHIM_Hydro	RDycore using PHIM_Hydro grid
Bias [m]	-1.25	-1.12	-1.31	-1.55
RMSE [m]	3.80	3.32	3.55	3.24
$R^2[-]$	0.99	0.99	0.99	0.99

fidence in RDycore’s ability to accurately perform flooding simulations for real-world settings.

3.4. Flooding During Hurricane Harvey

Hurricane Harvey made landfall near Houston, Texas, on late August 25, 2017 as a category 4 hurricane. It was one of the most destructive storms that caused extensive flooding with dozen of US Geological Survey (USGS) river gauges recording return period flows greater than 1 in 100 years [54]. We apply the one-way E3SMv2 – RDycore coupling to simulate flooding in Houston during Hurricane Harvey using the finest resolution simulation configuration described in [38]. The RDycore mesh contains ≈ 2.9 million triangular grid cells that was generated by splitting the uniform 30 m quadrilateral cells into two along the diagonal, resulting in an effective resolution of ≈ 21 m ($= \sqrt{(30 * 30)/2}$ m). The elevation of the study domain is shown in Figure 7a. A set of five E3SM – RDycore simulations were performed using five precipitation datasets that included (i) Multi-Radar Multi-Sensor System (MRMS), (ii) North American Land Data Assimilation System (NLDAS),

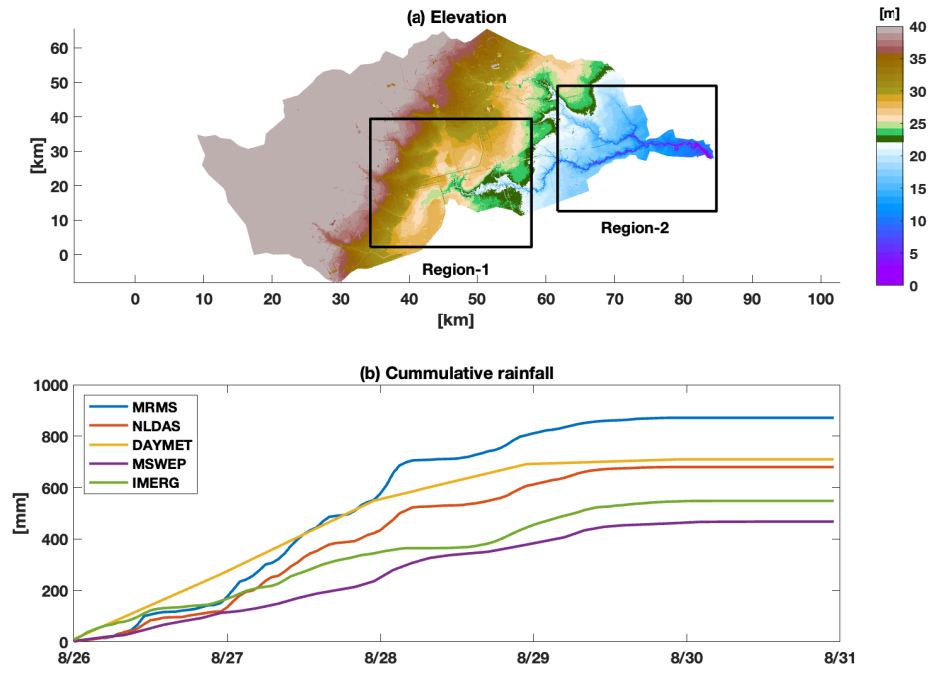


Figure 7: (a) Elevation for the study domain based on 30 m DEM with two regions outlined with black lines. Region-1 includes two flood control dams, and Region-2 encompasses the main river channel closer to the watershed outlet. (b) Cumulative rainfall over the study period for the five precipitation datasets.

(iii) DAYMET, (iv) Integrated multi-satellite retrievals for global precipitation measurement (IMERG), and (v) Multi-Source Weighted-Ensemble Precipitation (MSWEP). The precipitation datasets were converted from native spatio-temporal scales to hourly, 1 km datasets as described in [38]. The MRMS and MSWEP datasets had the largest and smallest cumulative rainfall during the study period (Figure 7b). The E3SM – RDycore simulations were performed for 5 days starting 26 August and used the DLND configuration. We assumed all precipitation during the event resulted in runoff. The water height from the USGS gauge 08074710, a tidal stream gauge, was used for prescribing a boundary condition at the watershed outlet. Additional details about the precipitation datasets and boundary conditions are presented in [38]. The outflow from the watershed and the maximum inundation height during the event were compared against the results obtained using the OFM used in [38].

The simulated maximum water height ($H_{\max}$) during Hurricane Harvey shows a spatially varying pattern (Figure 8). Large portions of the domain are inundated with water behind the two flood control dams (Figure 8b), while the largest value of $H_{\max}$ is in the river’s main channel (Figure 8c). The spatial pattern of simulated $H_{\max}$ is similar for all five precipitation datasets (Figure 9). Highest $H_{\max}$ values are obtained for the simulation with MRMS, and this result is consistent with the MRMS having the highest cumulative rainfall over the study period. The comparison of E3SM–RDycore simulated $H_{\max}$ against OFM results [38] reveals high agreement between the two models with low error (bias and RMSE) and high R^2 for all five precipitation datasets (Figure 10). The small differences between the two models can be

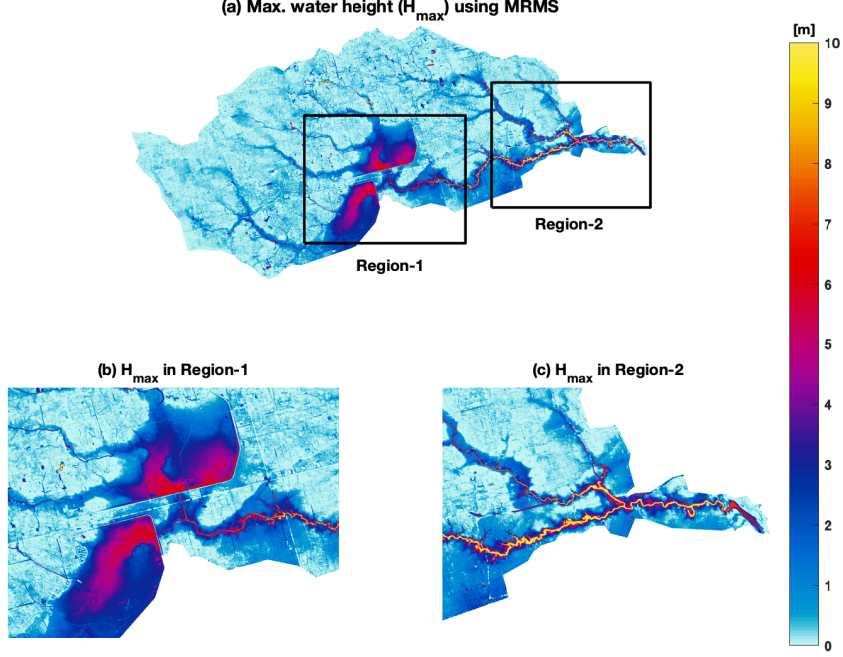


Figure 8: RDycore simulated maximum water height ($H_{\max}$) during Hurricane Harvey using the precipitation from MRMS. (a) $H_{\max}$ for the entire domain, while (b) and (c) show $H_{\max}$ for the two regions shown by the black lines in (a).

attributed to the following two factors. First, while both models used hourly precipitation datasets, E3SM–RDycore internally performed a linear temporal interpolation scheme. Second and more importantly, OFM includes the flux correction scheme of [40] that efficiently ensures a hydrostatic balance for the lake at rest case without requiring second-order spatial discretization.

We additionally performed short 2-hour long E3SM–RDycore simulations using CPUs and GPUs on Perlmutter and Frontier, and a detailed description

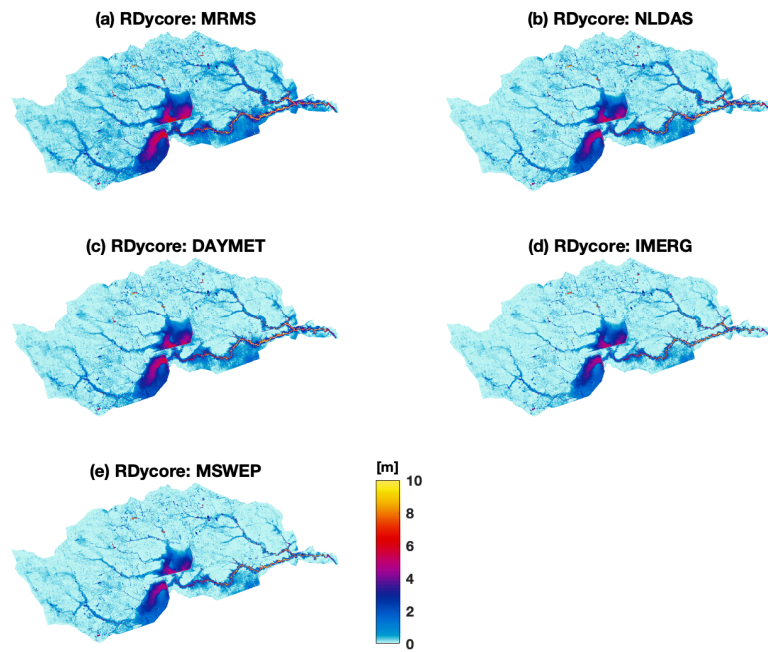


Figure 9: RDycore simulated maximum water height during Hurricane Harvey using the precipitation from (a) MRMS, (b) NLDAS, (c) DAYMET, (d) IMERG, and (e) MSWEP.

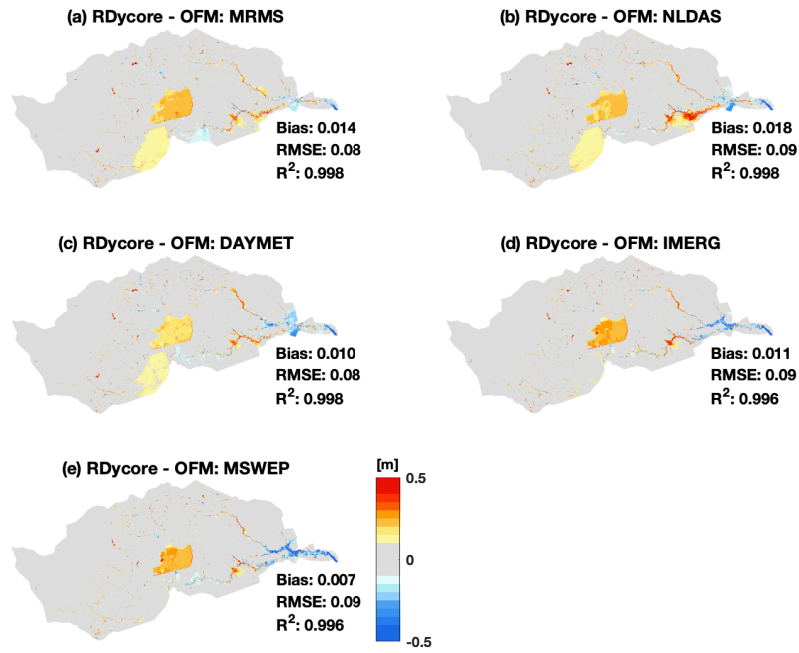


Figure 10: Same as Figure 9, except difference in the maximum water height simulated by RDycore and OFM.

of those systems is presented in Section 4. These simulations used a single compute node on each machine, which meant using 128 CPUs vs 4 GPUs on Perlmutter and 56 CPUs vs 8 GPUs on Frontier. A routine performance metric to measure the cost of E3SM simulations is the Simulated Year Per Day (SYPD), which corresponds to the number of years that will be simulated if the model runs uninterrupted for one day. The SYPD of simulations using CPUs were 0.06 (Perlmutter) and 0.03 (Frontier). The SYPD increased to 0.90 (Perlmutter) and 0.63 (Frontier) when GPUs were used, resulting in an overall speedup of 15x (Perlmutter) and 21x (Frontier).

4. Computational Performance of RDycore on Heterogeneous Computing Architectures

The effective use of hardware is critical in the efficient use of DOE hardware acquisitions and strong and weak scaling are common metrics for performance analysis. We performed a strong scaling performance analysis of RDycore for the flooding during Hurricane Harvey by using a fixed problem size and measuring the time spent in time integration as the number of CPUs or GPUs was increased. Unlike the simulation setup described in Section 3.4 that used 30 m Digital Elevation Model (DEM), a 3 m DEM from the National Elevation Dataset [55] was used here. Each quadrilateral DEM cell was split into four triangles, resulting in a 2D mesh with an effective resolution of 1.5 m ($= \sqrt{(3 * 3)/4}$ m) that contained 471 million grid cells and 1.4 billion unknowns.

Model performance using CPUs and GPUs was evaluated on Frontier and Perlmutter machines, which are the second fastest and nineteenth fastest

supercomputers in the world as of November 2024, respectively <https://top500.org/lists/top500/list/2024/11/>. The Frontier machine has compute nodes that contain one 64-core AMD EPYC CPU and four AMD MI250X GPUs, each with two Graphics Compute Dies (GCDs) for a total of eight GCDs per node. On the other hand, the Perlmutter machine has two types of compute nodes: (1) CPU nodes with two 64-core AMD EPYC 7763 CPUs per node, and (2) GPU nodes with one 64-core AMD EPYC 7763 CPU and four NVIDIA A100 Tensor Core GPUs. These studies use the optimal number of CPU cores per node, 56 and 128 on Frontier and Perlmutter, respectively. RDycore simulations on GPUs were configured to associate one MPI rank with one GPU. Thus, 8 and 4 MPI ranks per node were used on Frontier and Perlmutter, respectively. After reading metadata such as the mesh and material parameters, 8000 time steps of an explicit time integrator were processed and measured with PETSc `TSSStep` timers. No model output was written out during the 8000 time steps. The model was run on both machines using 5, 20, 80, and 320 compute nodes. The performance was computed as the number of grid cells advanced per second. The multiple GPU backends are supported in libCEED. The “shared” memory backed uses optimized element operations (using GPU shared memory, which is explicitly-managed memory accessible from all threads in a thread block) and a work array that is accumulated into cells using a separate kernel. In contrast, the “gen” backend fuses all operations into a single kernel and uses atomic “fetch and add” operations to accumulate into cells. The order of accumulation is non-deterministic with “gen”, and since floating point addition is not associative, the results are not deterministic. Using the shared

memory for libCEED’s operation ensures the results are deterministic, but the performance would be lower. For GPUs, we performed simulations without and with shared memory for libCEED to quantify the decrease in model performance to ensure the results are deterministic. The computational performance of RDycore simulations for the three set of hardware configurations on the two machines are summarized in Table 3 and presented in Figure 11.

First, RDycore’s CPU performance on both machines continues to increase as the number of compute nodes is increased (blue lines in Figure 11), and the speedup was $\approx 4x$ as the number of compute nodes was quadrupled (Table 3 fourth column). Second, the GPU performance of RDycore while using libCEED without shared memory was higher than using shared memory (red lines compared yellow lines in Figure 11), and the average performance reduction when using shared memory was 0.69x and 0.85x on Perlmutter and Frontier, respectively. Third, RDycore’s GPU performance also increases as the number of compute nodes increases; however, at 320 compute nodes, the performance increased only by an average of 1.9x and 1.3x when compute nodes were quadrupled from 80 as indicated by significant deviation of red and yellow lines from dashed black lines in Figure 11. Fourth, based on performance at 80 nodes, about a minimum of 6 million cells per node ($= 471 \text{ M cells}/80 \text{ nodes}$) is needed on GPUs to remain efficient. Fifth, RDycore is faster on GPUs than CPUs on both machines, with the highest (lowest) speedup of 32.9x (6.6x) and 41.9x (7.6x) across all compute nodes on Perlmutter and Frontier, respectively.

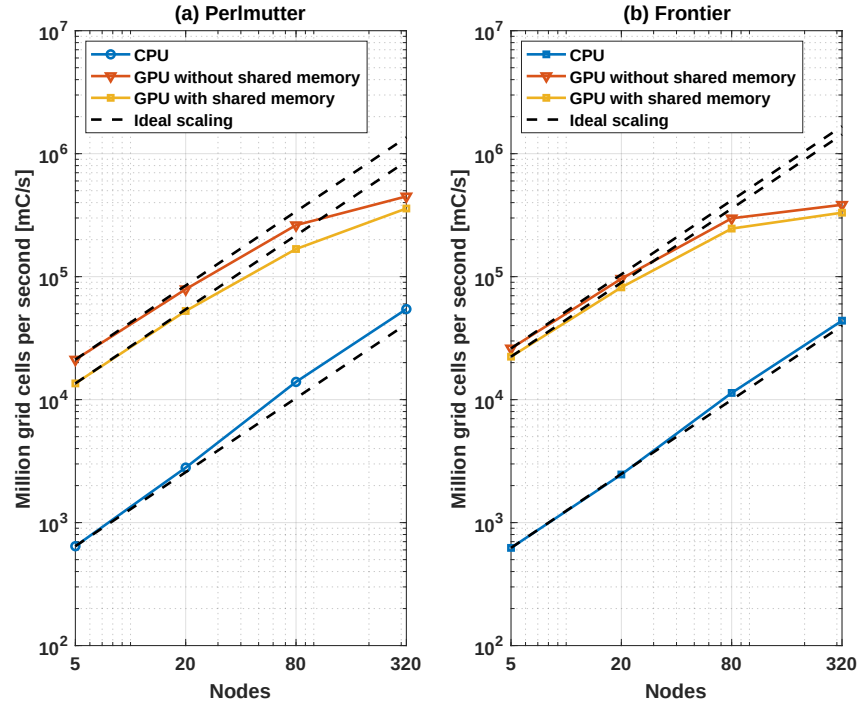


Figure 11: Strong scaling performance of RDycore for the 471 million grid cells Hurricane Harvey problem on (a) Perlmutter and (b) Frontier. The three solid lines represents model performance on CPU, GPU without shared memory, and GPU with shared memory, while the dashed lines represent ideal scaling. The results presented here are summarized in Table 3.

Table 3: Performance for RDycore’s time integration in millions of grid cells processed per sec (mC/s) for the flooding during the Hurricane Harvey simulation using a mesh containing 471 million grid cells on Perlmutter and Frontier systems. Speedup for a given number of nodes on CPUs (or GPUs) is estimated as the gain in performance relative to CPUs (or GPUs) with five compute nodes. Speedup on GPUs with respect to CPUs is estimated as the performance ratio on GPUs and CPUs for a given number of nodes. Performance on GPUs is collected when libCEED without and with shared memory such that the results are non-deterministic or deterministic, respectively.

Machine	Nodes	CPU		GPU without shared memory			GPU with shared memory		
		mC/s	Speedup on CPUs	mC/s	Speedup on GPUs	Speedup w.r.t. CPUs	mC/s	Speedup on GPUs	Speedup w.r.t CPUs
Perlmutter	5	643	-	21,179	-	32.9x	13,548	-	21.1x
	20	2,800	4.4x	79,047	3.7x	28.2x	52,589	3.9x	18.8x
	80	13,934	21.7x	262,249	12.4x	18.8x	167,795	12.4x	12.0x
	320	54,494	84.7x	449,449	21.2x	8.2x	358,277	26.4x	6.6x
Frontier	5	623	-	26,096	-	41.9x	22,349	-	35.9x
	20	2,462	4.0x	96,208	3.7x	39.0x	81,819	3.7x	33.2x
	80	11,338	18.2x	298,078	11.4x	26.3x	245,969	11.0x	21.7x
	320	43,853	70.4x	383,996	14.7x	8.8x	331,195	14.8x	7.6x

5. Conclusions and Outlook

In this work, we present the development of RDycore, a 2D SWE model with support for heterogeneous computing architecture, for E3SM to model pluvial, fluvial, and compound flooding. The combined use of PETSc and libCEED provides RDycore hardware and algorithmic portability. The new model is verified for problems with analytical and manufactured solutions and benchmarked against other SWE models for the Malpasset dam break problem and flooding during Hurricane Harvey. The computational performance of RDycore was assessed on Perlmutter and Frontier for a problem with 471 million grid cells on up to 320 compute nodes. The initial one-way coupling of RDycore into E3SM was achieved by introducing RDycore as a part of E3SM’s existing river component.

We next summarize potential future developments to RDycore’s existing physics scheme, numerical implementation, and coupling with other components within E3SM. Appropriate well-balancing should be added in RDycore’s physics scheme, and such an extension could be based on one of many existing approaches [40, 56, 57]. Specifically, an algorithm to track wet/dry fronts on arbitrary topography should be included in the next version of RDycore, and such an algorithm can be based on the volume-free surface relationship method [58, 59] or piecewise linear reconstruction of water surface in partially wet cells [60, 61]. At smaller water depths, the bed friction source term becomes large as it includes water height in the denominator, causing explicit time integration methods to take a small time step to ensure numerical stability. Various approaches have been previously developed to deal with the stiffness of the bed friction term [52, 62] and the use of such approaches

in improving RDycore’s simulations must be explored. The numerical implementation of RDycore should be enhanced by allowing the shared edges between local and halo cells to be present on both MPI ranks, allowing the streamlining of per-timestep MPI communication and potentially increasing computational performance.

The current coupling of RDycore within E3SM could be improved in the following ways. RDycore should be added as a independent component within E3SM to significantly improve its adoption within the E3SM research community. A standalone RDycore simulation can simultaneously apply source and boundary conditions that are spatially and temporally varying. Thus, a RDycore standalone simulation can simulate various flooding types due to pluvial, fluvial, and costal processes, as well as compound flooding arising due to any combination of the three types of floodings. However, the current coupling of RDycore in E3SM is limited to a single component, which is DLND, and only includes one-way interaction. The coupling of RDycore should be extended to support more components such as the data ocean (DOCN) and active component models (i.e, ELM and MPAS-O). Even with one-way coupling of RDycore with DLND and DOCN, the model will be enabled to simulate compound flooding due to the interaction among all three types of aforementioned floodings types. For the case of active land and ocean components, the RDycore coupling should be extended to include two-way interactions. Our previous works that developed two-way coupling between land-river [15] and river-ocean [16] components of E3SM will serve as building blocks for developing the two-way coupled ELM–RDycore–MPAS-O configuration within E3SM. Typically, the coupling between the river and

ocean components in ESMs is done only at the single grid cell representing the mouth of the river [63, 64, 16, 65, 66], and such an approach is acceptable for coarser scale models. RDycore, which is applicable at sub-kilometer scales, can capture the dynamic interaction between river and ocean along the whole coastline. In order to support the coupling between RDycore and MPAS-O along the coastline, we will leverage our previously developed ELM and DOCN coupling along the coastline [67].

The work presented here provides a foundation for studying compound flooding within E3SM using a higher fidelity river model. Furthermore, the use of scalable and hardware-agnostic approaches allows for the possibility of performing global river simulations at k-scale.

Software and data availability

Software availability: River Dynamical Core

Developers: Gautam Bisht, Donghui Xu, Jeffrey Johnson, Jed Brown, Mathew Knepley, Mark Adams, Dongyu Feng, Dalei Hao, Darren Engwirda, Mukesh Kumar, and Zeli Tan

Contact: gautam.bisht@pnnl.gov

Year first available: 2025

Software required: PETSc

Program Language: C, Fortran

Source code at: <https://github.com/rdycore/rdycore>

Documentation: <https://rdycore.github.io/RDycore/>

CRedit authorship contribution statement

Gautam Bisht: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Funding acquisition. **Donghui Xu:** Conceptualization, Methodology, Validation, Investigation, Data Curation, Writing - Review & Editing. **Jeffrey Johnson :** Software, Writing - Review & Editing, Funding acquisition. **Jed Brown :** Software, Writing - Review & Editing, Funding acquisition. **Matthew Knepley:** Software, Writing - Review & Editing, Funding acquisition. **Mark Adams:** Formal analysis, Investigation, Writing - Review & Editing, Funding acquisition. **Dongyu Feng:** Validation, Data Curation, Writing - Review & Editing. **Dalei Hao:** Validation, Writing - Review & Editing. **Darren Engwirda:** Writing - Review & Editing, Funding acquisition. **Mukesh Kumar:** Writing - Review & Editing, Funding acquisition. **Zeli Tan:** Writing - Review & Editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no conflicts of interest to this work.

Acknowledgments

This work was supported by Scientific Discovery through Advanced Computing 5 (Capturing the Dynamics of Compound Flooding in E3SM), funded by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research. The Pacific Northwest National Laboratory is operated by Battelle for the U.S. Department of Energy under Contract

DE-AC05-76RLO1830. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725, and the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility using NERSC award BER-ERCAPm4267. We sincerely thank the two anonymous reviewers for their constructive comments and suggestions, which helped in improving the quality of this manuscript.

Data availability

The data for the simulations are published at <https://zenodo.org/records/14915444>.

Appendix A. Spatial discretization of the flux vector

The normal flux across an edge, $F_\perp$, is computed using an approximate Roe's Riemann solver and is given as

$$\vec{\mathbf{F}} \cdot \vec{n} \approx F_\perp = \frac{1}{2} \left(\mathbf{F}_{\perp,L} + \mathbf{F}_{\perp,R} - \hat{\mathbf{R}} |\hat{\mathbf{\Lambda}}| \Delta \hat{\mathbf{V}} \right) \quad (\text{A.1})$$

where L and R denote the cells on the left and right of an edge, $\mathbf{F}_\perp$ is scalar flux, $\hat{\mathbf{R}}$ are the right eigenvectors and $\hat{\mathbf{\Lambda}}$ are the absolute eigenvalue of the Jacobian of $\mathbf{F}_\perp$. The $\Delta \hat{\mathbf{V}}$ is defined as $\hat{\mathbf{L}} (\mathbf{U}_R - \mathbf{U}_L)$ where $\hat{\mathbf{L}}$ are the left eigenvectors and $(\mathbf{U}_R - \mathbf{U}_L)$ denote the finite difference of $\mathbf{U}$ across the edge. The various terms of the approximate Roe's Riemann solver in Equation A.1 are given by

$$\mathbf{F}_\perp = \begin{bmatrix} hu_\perp \\ hu u_\perp + \frac{1}{2}gh^2 \cos \phi \\ hvu_\perp + \frac{1}{2}gh^2 \sin \phi \end{bmatrix} \quad (\text{A.2})$$

$$\hat{\mathbf{R}} = \begin{bmatrix} 1 & 0 & 1 \\ \hat{u} - \hat{a} \cos \phi & -\sin \phi & \hat{u} + \hat{a} \cos \phi \\ \hat{v} - \hat{a} \sin \phi & \cos \phi & \hat{v} + \hat{a} \sin \phi \end{bmatrix} \quad (\text{A.3})$$

$$|\hat{\mathbf{\Lambda}}| = \begin{bmatrix} |\hat{u}_\perp - \hat{a}|^* & 0 & 0 \\ 0 & |\hat{u}_\perp| & 0 \\ 0 & & |\hat{u}_\perp + \hat{a}|^* \end{bmatrix} \quad (\text{A.4})$$

$$\Delta \hat{\mathbf{V}} = \begin{bmatrix} \frac{1}{2} \left(\Delta h - \frac{\hat{h} \Delta u_\perp}{\hat{a}} \right) \\ \hat{h} u_\parallel \\ \frac{1}{2} \left(\Delta h + \frac{\hat{h} \Delta u_\perp}{\hat{a}} \right) \end{bmatrix} \quad (\text{A.5})$$

where $u_\perp (= u \cos \phi + v \sin \phi)$ is the velocity perpendicular to the edge, $u_\parallel (= -u \sin \phi + v \cos \phi)$ is the velocity parallel to the edge, ϕ is the angle between the face normal and the x axis, and a is the celerity of a simple internal wave. The quantities denoted with a hat symbol in Equation A.3-A.5 are

Roe averages that are computed as

$$\hat{h} = \sqrt{h_L h_R} \quad (\text{A.6})$$

$$\hat{u} = \frac{\sqrt{h_L} u_L + \sqrt{h_R} u_R}{\sqrt{h_L} + \sqrt{h_R}} \quad (\text{A.7})$$

$$\hat{v} = \frac{\sqrt{h_L} v_L + \sqrt{h_R} v_R}{\sqrt{h_L} + \sqrt{h_R}} \quad (\text{A.8})$$

$$\hat{a} = \sqrt{\frac{g}{2} (h_L + h_R)} \quad (\text{A.9})$$

The asterisks in Equation A.4 denote that the eigenvalues $\hat{\lambda}_1 = \hat{u}_\perp - \hat{a}$ and $\hat{\lambda}_3 = \hat{u}_\perp + \hat{a}$ are adjusted since Roe's method does not provide correct flux for critical flow. The adjusted eigenvalues are given by

$$\hat{\lambda}_1^* = \frac{(\hat{\lambda}_1)^2}{\Delta \lambda_1} + \frac{\Delta \lambda_1}{4} \quad \text{if} \quad -\Delta \lambda_1/2 < \hat{\lambda}_1 < \Delta \lambda_1/2 \quad (\text{A.10})$$

$$\hat{\lambda}_3^* = \frac{(\hat{\lambda}_3)^2}{\Delta \lambda_3} + \frac{\Delta \lambda_3}{4} \quad \text{if} \quad -\Delta \lambda_3/2 < \hat{\lambda}_3 < \Delta \lambda_3/2 \quad (\text{A.11})$$

where $\Delta \lambda = 4(\lambda_R - \lambda_L)$

Appendix B. Spatial discretization of the bed and friction source terms

The spatial discretization of the bed slope source term for the momentum in x -direction given in Equation 6 is computed via

$$\int_{\Omega_i} -gh \left(\frac{\partial z}{\partial x} \right) d\Omega = -gh_i \left(\frac{\partial z}{\partial x} \right)_i A_i \quad (\text{B.1})$$

where $\left(\overline{\frac{\partial z}{\partial x}}\right)_i$ is the average bed slope for i -th triangular cell in x -direction and that is computed using Green's theorem as

$$\left(\overline{\frac{\partial z}{\partial x}}\right)_i = \frac{(y_{i,2} - y_{i,0})(z_{i,1} - z_{i,0}) - (y_{i,1} - y_{i,0})(z_{i,2} - z_{i,0})}{(y_{i,2} - y_{i,0})(x_{i,1} - x_{i,0}) - (y_{i,1} - y_{i,0})(x_{i,2} - x_{i,0})} \quad (\text{B.2})$$

where subscript 0 denote cell centroid, while subscripts 1, 2, and 3 denotes cell vertices in counter-clockwise direction. The spatial discretization of the bed slope source term for the momentum in y -direction and the average bed slope in the y -direction can be computed similarly. For a quadrilateral control volume in RDycore, the average slope in each direction is computed by splitting a quadrilateral into four triangles and computing the mean of the average slope for the four triangles.

The spatial discretization of the bed friction source term for the momentum in x -direction given in Equation 7 is

$$\int_{\Omega_i} C_D u \sqrt{u^2 + v^2} d\Omega = C_D u_i \sqrt{u_i^2 + v_i^2} A_i = \left(\frac{gn_i^2}{h_i^{1/3}}\right) \frac{(hu)_i}{h_i} \sqrt{u_i^2 + v_i^2} A_i \quad (\text{B.3})$$

Appendix C. Supported boundary conditions

RDycore supports reflective wall and critical flow boundary conditions (Figure C.12). For both these two types of boundary conditions, the cell on the left of a boundary edge is an internal cell and the cell on the right is a boundary cell for which the water height and velocities in x - and y -direction must be estimated. The implementation details for the two conditions is presented below.

Appendix C.1. Reflective wall boundary condition

The water height of right of the edge is set to the same as water height left of the edge. The velocity perpendicular to the edge on the right hand side

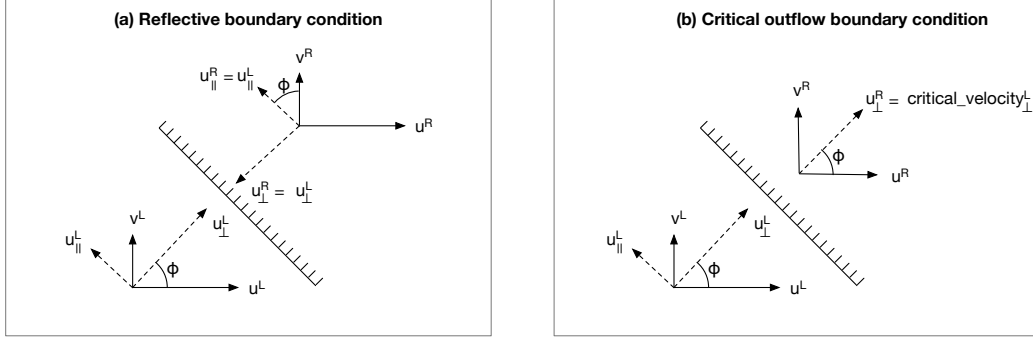


Figure C.12: Treatment of boundary conditions on RDycore: (a) reflective boundary condition and (b) critical outflow boundary condition. The cell on the left of the edge is within the domain, while the cell on the right corresponds to the boundary cell.

is equal in magnitude, but in the opposite direction to the velocity perpendicular on the left hand side of the edge, while the right hand side velocity parallel to the edge has the same magnitude and direction as the velocity left of the edge (Figure C.12a). Given these assumptions, the quantities on the right hand side of the edge are given as

$$h_R = h_L \quad (\text{C.1})$$

$$\begin{aligned} u_R &= -u_{\parallel}^L \sin \phi - u_{\perp}^L \cos \phi \\ &= -(-u^L \sin \phi + v^L \cos \phi) \sin \phi - (u^L \cos \phi + v^L \sin \phi) \cos \phi \\ &= u^L \sin^2 \phi - v^L \sin \phi \cos \phi - u^L \cos^2 \phi - v^L \sin \phi \cos \phi \\ &= (\sin^2 \phi - \cos^2 \phi) u^L - (2 \sin \phi \cos \phi) v^L \end{aligned} \quad (\text{C.2})$$

$$\begin{aligned} v_R &= u_{\parallel}^L \cos \phi - u_{\perp}^L \sin \phi \\ &= (-u^L \sin \phi + v^L \cos \phi) \cos \phi - (u^L \cos \phi + v^L \sin \phi) \sin \phi \\ &= -u^L \sin \phi \cos \phi + v^L \cos^2 \phi - u^L \sin \phi \cos \phi - v^L \sin^2 \phi \\ &= -(2 \sin \phi \cos \phi) u^L - (\sin^2 \phi - \cos^2 \phi) v^L \end{aligned} \quad (\text{C.3})$$

Appendix C.2. Critical outflow boundary condition

The outflow from the left cell, q^L , is given as

$$q^L = h^L u_{\perp}^L \quad (\text{C.4})$$

For critical flow, the Froude number is equal to one leading to

$$\begin{aligned} \frac{u_{\perp}^R}{\sqrt{gh^R}} &= 1 \\ \frac{q^R/h^R}{\sqrt{gh^R}} &= 1 \\ (h^R)^{3/2} &= \frac{q^R}{g^{1/2}} \\ h^R &= \left(\frac{(q^R)^2}{g} \right)^{1/3} \end{aligned} \quad (\text{C.5})$$

The flux across the edge is conserved, i.e., $q^R = q^L$. So, using equation C.4, the equation C.5 can be rewritten as

$$h^R = \left(\frac{(h^L u_{\perp}^L)^2}{g} \right)^{1/3} \quad (\text{C.6})$$

Again using the Froude number, the velocity perpendicular to the edge on the right hand side is then computed as

$$u_{\perp}^R = \sqrt{gh^R} \quad (\text{C.7})$$

The velocity parallel to edge on the right hand side is assumed to be zero. Thus, the velocity in the x- and y-direction on the right hand side of the edge is computed by projecting $u_{\perp}^R$ as shown in Figure C.12b.

$$u_R = u_\perp^R \cos \phi \quad (\text{C.8})$$

$$v_R = u_\perp^R \sin \phi \quad (\text{C.9})$$

Appendix D. Time integration of momentum equations

The temporal discretization of the friction terms uses a semi-implicit discretization in which equation B.1 is written as

$$\left(\frac{gn_i^2}{(h_i^t)^{1/3}} \right) \frac{(hu)_i^{t+1}}{h_i^t} \sqrt{(u_i^t)^2 + (v_i^t)^2} A_i \quad (\text{D.1})$$

where the superscript denotes quantities computed at time level t and $t + 1$. The explicit temporal discretization is used for all other terms in the momentum equation in x -direction given in equation 9. Thus, the spatio-temporal discretized momentum equation in x -direction is given by

$$\begin{aligned} \frac{hu_i^{t+1} - hu_i^t}{\Delta t} = & - \sum_j (\mathbf{F}_{i,j}^t \cdot \mathbf{n}_{i,j}) \left(\frac{L_{i,j}}{\Omega_i} \right) - gh_i^t \left(\frac{\partial z}{\partial x} \right)_i A_i \\ & - \left(\frac{gn_i^2}{(h_i^t)^{1/3}} \right) \frac{(hu)_i^{t+1}}{h_i^t} \sqrt{(u_i^t)^2 + (v_i^t)^2} A_i \end{aligned} \quad (\text{D.2})$$

Using algebraic manipulation, the equation D.2 can be rewritten as

$$\begin{aligned} \frac{(hu)_i^{t+1} - (hu)_i^t}{\Delta t} &= \alpha_i^t - \beta_i^t + \gamma_i^t (hu)_i^{t+1} \\ [1 + \gamma_i^t \Delta t] (hu)_i^{t+1} &= (hu)_i^t + \Delta t \alpha_i^t - \Delta t \beta_i^t \\ [1 + \gamma_i^t \Delta t] (hu)_i^{t+1} &= [1 + \gamma_i^t \Delta t] (hu)_i^t - \gamma_i^t \Delta t (hu)_i^t + (\Delta t) \alpha_i^t - (\Delta t) \beta_i^t \\ [1 + \gamma_i^t \Delta t] \left\{ \frac{(hu)_i^{t+1} - (hu)_i^t}{\Delta t} \right\} &= -\gamma_i^t (hu)_i^t + \alpha_i^t - \beta_i^t \\ \frac{(hu)_i^{t+1} - (hu)_i^t}{\Delta t} &= \frac{-\gamma_i^t (hu)_i^t + \alpha_i^t - \beta_i^t}{[1 + \gamma_i^t \Delta t]} \end{aligned} \quad (\text{D.3})$$

where

$$\alpha_i^t = -\frac{1}{A_i} \sum_j (\mathbf{F}_{i,j}^t \cdot \mathbf{n}_{i,j}) \left(\frac{L_{i,j}}{\Omega_i} \right) \quad (\text{D.4})$$

$$\beta_i^t = gh_i^t \left(\frac{\partial z}{\partial x} \right)_i \quad (\text{D.5})$$

$$\gamma_i^t = \frac{gn_i^2 \sqrt{(u_i^t)^2 + (v_i^t)^2}}{(h_i^t)^{4/3}} \quad (\text{D.6})$$

Similarly, the temporal discretization of the momentum in y -direction leads to

$$\frac{(hv)_i^{t+1} - (hv)_i^t}{\Delta t} = \frac{-\gamma_i^t (hu)_i^t - \alpha_i^t - \delta^t}{[1 + \gamma_i^t \Delta t]} \quad (\text{D.7})$$

where

$$\delta_i^t = gh_i^t \left(\frac{\partial z}{\partial y} \right)_i \quad (\text{D.8})$$

Appendix E. Analytical Solution for Dry and Wet Domains

The analytical solutions for the 1D dam break problem with dry and wet initial conditions are given by equation 11 (Section 3.1) are presented here for completeness. For a detailed description of these problems, the reader is referred to SWASHES [49].

Appendix E.1. Dry Domain

$$h(t, x) = \begin{cases} h_l & \text{if } 0 \leq x_A(t) \\ \frac{4}{9g} \left(\sqrt{gh_l} - \frac{x}{2t} \right)^2 & \text{if } x_A(t) \leq x \leq x_B(t) \\ 0 & \text{if } x_B(t) \leq x \end{cases} \quad (\text{E.1})$$

$$u(t, x) = \begin{cases} 0 & \text{if } 0 \leq x_A(t) \\ \frac{2}{3} \left(\frac{x}{t} + \sqrt{gh_l} \right) & \text{if } x_A(t) \leq x \leq x_B(t) \\ 0 & \text{if } x_B(t) \leq x \end{cases} \quad (\text{E.2})$$

where $x_A(t) = x_0 - t\sqrt{gh_l}$ and $x_B(t) = x_0 + 2t\sqrt{gh_l}$.

Appendix E.2. Wet Domain

$$h(t, x) = \begin{cases} h_l & \text{if } 0 \leq x_A(t) \\ \frac{4}{9g} \left(\sqrt{gh_l} - \frac{x}{2t} \right)^2 & \text{if } x_A(t) \leq x \leq x_B(t) \\ \frac{c_m^2}{g} & \text{if } x_B(t) \leq x \leq x_C(t) \\ h_r & \text{if } x_C(t) \leq x \end{cases} \quad (\text{E.3})$$

$$u(t, x) = \begin{cases} 0 & \text{if } 0 \leq x_A(t) \\ \frac{2}{3} \left(\frac{x}{t} + \sqrt{gh_l} \right) & \text{if } x_A(t) \leq x \leq x_B(t) \\ 2 \left(\sqrt{gh_l} - c_m \right) & \text{if } x_B(t) \leq x \leq x_C(t) \\ 0 & \text{if } x_C(t) \leq x \end{cases} \quad (\text{E.4})$$

where $x_A(t) = x_0 - t\sqrt{gh_l}$, $x_B(t) = x_0 + t(2\sqrt{gh_l} - 3c_m)$, and $x_C(t) = x_0 + t \left(\frac{2c_m^2(\sqrt{gh_l} - c_m)}{c_m^2 - gh_r} \right)$. The c_m is given as the solution of $-8gh_rc_m^2(gh_l - c_m^2)^2 + (c_m^2 - gh_r)^2(c_m^2 + gh_r) = 0$.

Appendix F. Method of Manufactured Solutions

The derivatives of manufactured solutions given by equation 15-19 with respect to x , y , and t are given below.

$$\frac{\partial h}{\partial x} = \left(\frac{\pi h_0}{L_x} \right) \cos \left(\frac{\pi x}{L_x} \right) \sin \left(\frac{\pi y}{L_y} \right) \exp \left(\frac{t}{t_0} \right) \quad (\text{F.1})$$

$$\frac{\partial h}{\partial y} = \left(\frac{\pi h_0}{L_y} \right) \sin \left(\frac{\pi x}{L_x} \right) \cos \left(\frac{\pi y}{L_y} \right) \exp \left(\frac{t}{t_0} \right) \quad (\text{F.2})$$

$$\frac{\partial h}{\partial t} = \left(\frac{h_0}{t_0} \right) \left[1 + \sin \left(\frac{\pi x}{L_x} \right) \sin \left(\frac{\pi y}{L_y} \right) \right] \exp \left(\frac{t}{t_0} \right) = \frac{h}{t_0} \quad (\text{F.3})$$

$$\frac{\partial u}{\partial x} = - \left(\frac{\pi u_0}{L_x} \right) \sin \left(\frac{\pi x}{L_x} \right) \sin \left(\frac{\pi y}{L_y} \right) \exp \left(\frac{t}{t_0} \right) \quad (\text{F.4})$$

$$\frac{\partial u}{\partial y} = \left(\frac{\pi u_0}{L_y} \right) \cos \left(\frac{\pi x}{L_x} \right) \cos \left(\frac{\pi y}{L_y} \right) \exp \left(\frac{t}{t_0} \right) \quad (\text{F.5})$$

$$\frac{\partial u}{\partial t} = \frac{u_0}{t_0} \cos \left(\frac{\pi x}{L_x} \right) \sin \left(\frac{\pi y}{L_y} \right) \exp \left(\frac{t}{t_0} \right) = \frac{u}{t_0} \quad (\text{F.6})$$

$$\frac{\partial v}{\partial x} = \left(\frac{\pi v_0}{L_x} \right) \cos \left(\frac{\pi x}{L_x} \right) \cos \left(\frac{\pi y}{L_y} \right) \exp \left(\frac{t}{t_0} \right) \quad (\text{F.7})$$

$$\frac{\partial v}{\partial y} = - \left(\frac{\pi v_0}{L_y} \right) \sin \left(\frac{\pi x}{L_x} \right) \sin \left(\frac{\pi y}{L_y} \right) \exp \left(\frac{t}{t_0} \right) \quad (\text{F.8})$$

$$\frac{\partial v}{\partial t} = \frac{v_0}{t_0} \cos \left(\frac{\pi x}{L_x} \right) \sin \left(\frac{\pi y}{L_y} \right) \exp \left(\frac{t}{t_0} \right) = \frac{v}{t_0} \quad (\text{F.9})$$

$$\frac{\partial z}{\partial x} = \frac{z_0 \pi}{L_x} \cos \left(\frac{\pi x}{L_x} \right) \sin \left(\frac{\pi y}{L_y} \right) \quad (\text{F.10})$$

$$\frac{\partial z}{\partial y} = \frac{z_0 \pi}{L_y} \sin \left(\frac{\pi x}{L_x} \right) \cos \left(\frac{\pi y}{L_y} \right) \quad (\text{F.11})$$

Appendix F.1. Source Term for the Height Equation

The source term for the height equation, S_h^{MMS} , is given by

$$\begin{aligned} S_h^{MMS} &= \frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} - R \\ &= \frac{h}{t_0} + u \frac{\partial h}{\partial x} + h \frac{\partial u}{\partial x} + v \frac{\partial h}{\partial y} + h \frac{\partial v}{\partial y} \end{aligned} \quad (F.12)$$

Appendix F.2. Source Term for Momentum Equation in X-direction

The source term for the x-direction momentum equation, S_{hu}^{MMS} , is given by

$$\begin{aligned} S_{hu}^{MMS} &= \frac{\partial(hu)}{\partial t} + \frac{\partial(hu^2 + 1/2gh^2)}{\partial x} + \frac{\partial(huv)}{\partial y} + gh \frac{\partial z}{\partial x} + C_D u \sqrt{u^2 + v^2} \\ &= u \frac{\partial h}{\partial t} + h \frac{\partial u}{\partial t} + u^2 \frac{\partial h}{\partial x} + 2hu \frac{\partial u}{\partial x} + gh \frac{\partial h}{\partial x} + uv \frac{\partial h}{\partial y} + hv \frac{\partial u}{\partial y} \\ &\quad + hu \frac{\partial v}{\partial y} + gh \frac{\partial z}{\partial x} + \left(\frac{gn}{h^{1/3}} \right) u \sqrt{u^2 + v^2} \end{aligned} \quad (F.13)$$

Appendix F.3. Source Term for Momentum Equation in Y-direction

The source term for the y-direction momentum equation, S_{hv}^{MMS} , is given by

$$\begin{aligned} S_{hv}^{MMS} &= \frac{\partial(hv)}{\partial t} + \frac{\partial(hvu)}{\partial x} + \frac{\partial(hv^2 + 1/2gh^2)}{\partial y} + gh \frac{\partial z}{\partial y} + C_D v \sqrt{u^2 + v^2} \\ &= v \frac{\partial h}{\partial t} + h \frac{\partial v}{\partial t} + vu \frac{\partial h}{\partial x} + hu \frac{\partial v}{\partial x} + hv \frac{\partial u}{\partial x} + v^2 \frac{\partial h}{\partial y} + 2hv \frac{\partial v}{\partial y} \\ &\quad + gh \frac{\partial h}{\partial y} + gh \frac{\partial z}{\partial y} + \left(\frac{gn}{h^{1/3}} \right) v \sqrt{u^2 + v^2} \end{aligned} \quad (F.14)$$

References

- [1] N. Yaghmaei, Human Cost of Disasters: An Overview of the Last 20 Years, 2000-2019, UN Office for Disaster Risk Reduction, 2020.

- [2] V. Y. Ivanov, D. Xu, M. C. Dwelle, K. Sargsyan, D. B. Wright, N. Katopodes, J. Kim, V. N. Tran, A. Warnock, S. Fatichi, et al., Breaking down the computational barriers to real-time urban flood forecasting, *Geophysical Research Letters* 48 (20) (2021) e2021GL093585.
- [3] National Centers for Environmental Information, <https://www.ncei.noaa.gov/access/billions/events>, accessed: 2025-01-19.
- [4] P. D. Bates, N. Quinn, C. Sampson, A. Smith, O. Wing, J. Sosa, J. Savage, G. Olcese, J. Neal, G. Schumann, et al., Combined modeling of us fluvial, pluvial, and coastal flood hazard under current and future climates, *Water Resources Research* 57 (2) (2021) e2020WR028673.
- [5] H. Moftakhari, J. E. Schubert, A. AghaKouchak, R. A. Matthew, B. F. Sanders, Linking statistical and hydrodynamic modeling for compound flood hazard assessment in tidal channels and estuaries, *Advances in Water Resources* 128 (2019) 28–38.
- [6] Y. Hirabayashi, R. Mahendran, S. Koirala, L. Konoshima, D. Yamazaki, S. Watanabe, H. Kim, S. Kanae, Global flood risk under climate change, *Nature climate change* 3 (9) (2013) 816–821.
- [7] J. Sillmann, V. V. Kharin, F. W. Zwiers, X. Zhang, D. Bronaugh, Climate extremes indices in the cmip5 multimodel ensemble: Part 2. future climate projections, *Journal of geophysical research: atmospheres* 118 (6) (2013) 2473–2493.
- [8] K. N. Musselman, F. Lehner, K. Ikeda, M. P. Clark, A. F. Prein, C. Liu, M. Barlage, R. Rasmussen, Projected increases and shifts in rain-on-

- snow flood risk over western north america, *Nature Climate Change* 8 (9) (2018) 808–812.
- [9] M. Taherkhani, S. Vitousek, P. L. Barnard, N. Frazer, T. R. Anderson, C. H. Fletcher, Sea-level rise exponentially increases coastal flood frequency, *Scientific reports* 10 (1) (2020) 1–17.
 - [10] S. Vitousek, P. L. Barnard, C. H. Fletcher, N. Frazer, L. Erikson, C. D. Storlazzi, Doubling of coastal flooding frequency within decades due to sea-level rise, *Scientific reports* 7 (1) (2017) 1399.
 - [11] A. Sarhadi, R. Rousseau-Rizzi, K. Mandli, J. Neal, M. P. Wiper, M. Feldmann, K. Emanuel, Climate change contributions to increasing compound flooding risk in new york city, *Bulletin of the American Meteorological Society* 105 (2) (2024) E337–E356.
 - [12] D. Yamazaki, S. Kanae, H. Kim, T. Oki, A physically based description of floodplain inundation dynamics in a global river routing model, *Water Resources Research* 47 (4) (2011).
 - [13] H. Li, M. S. Wigmosta, H. Wu, M. Huang, Y. Ke, A. M. Coleman, L. R. Leung, A physically based runoff routing model for land surface and earth system models, *Journal of Hydrometeorology* 14 (3) (2013) 808–828.
 - [14] X. Luo, H.-Y. Li, L. R. Leung, T. K. Tesfa, A. Getirana, F. Papa, L. L. Hess, Modeling surface water dynamics in the amazon basin using mosart-inundation v1. 0: impacts of geomorphological parameters and

- river flow representation, *Geoscientific Model Development* 10 (3) (2017) 1233–1259.
- [15] D. Xu, G. Bisht, T. Zhou, L. R. Leung, M. Pan, Development of land-river two-way hydrologic coupling for floodplain inundation in the energy exascale earth system model, *Journal of Advances in Modeling Earth Systems* 14 (8) (2022) e2021MS002772.
 - [16] D. Feng, Z. Tan, D. Engwirda, J. D. Wolfe, D. Xu, C. Liao, G. Bisht, J. J. Benedict, T. Zhou, H.-Y. Li, et al., Simulation of compound flooding using river-ocean two-way coupled e3sm ensemble on variable-resolution meshes, *Journal of Advances in Modeling Earth Systems* 16 (6) (2024) e2023MS004054.
 - [17] S. Fox, F. Agyemang, L. Hawker, J. Neal, Integrating social vulnerability into high-resolution global flood risk mapping, *Nature communications* 15 (1) (2024) 3155.
 - [18] L. Bertagna, O. Guba, M. A. Taylor, J. G. Foucar, J. Larkin, A. M. Bradley, S. Rajamanickam, A. G. Salinger, A performance-portable non-hydrostatic atmospheric dycore for the energy exascale earth system model running at cloud-resolving resolutions, in: *Sc20: International conference for high performance computing, networking, storage and analysis*, IEEE, 2020, pp. 1–14.
 - [19] J. Wei, P. Lin, J. Jiang, H. Liu, L. Zhao, Y. Zhang, X. Han, F. Zhang, J. Huang, Y. Wang, et al., Accelerating lasg/iap climate system ocean

model version 3 for performance portability using kokkos, *Future Generation Computer Systems* (2024).

- [20] T. Rackow, X. Pedruzo-Bagazgoitia, T. Becker, S. Milinski, I. Sandu, R. Aguridan, P. Bechtold, S. Beyer, J. Bidlot, S. Boussetta, W. Deconinck, M. Diamantakis, P. Dueben, E. Dutra, R. Forbes, R. Ghosh, H. F. Goessling, I. Hadade, J. Hegewald, T. Jung, S. Keeley, L. Kluft, N. Koldunov, A. Koldunov, T. Kölling, J. Kousal, C. Kühnlein, P. Maciel, K. Mogensen, T. Quintino, I. Polichtchouk, B. Reuter, D. Sármany, P. Scholz, D. Sidorenko, J. Streffing, B. Sützl, D. Takasuka, S. Tietsche, M. Valentini, B. Vannière, N. Wedi, L. Zampieri, F. Ziemen, Multi-year simulations at kilometre scale with the Integrated Forecasting System coupled to FESOM2.5 and NEMOv3.4, *Geoscientific Model Development* 18 (1) (2025) 33–69.

URL <https://gmd.copernicus.org/articles/18/33/2025/>

- [21] M. Morales-Hernández, M. B. Sharif, A. Kalyanapu, S. K. Ghafoor, T. T. Dullo, S. Gangrade, S.-C. Kao, M. R. Norman, K. J. Evans, Triton: A multi-gpu open source 2d hydrodynamic flood model, *Environmental Modelling & Software* 141 (2021) 105034.
- [22] X. Li, D. Fu, J. Nielsen-Gammon, S. Gangrade, S.-C. Kao, P. Chang, M. M. Hernández, N. Voisin, Z. Zhang, H. Gao, Impacts of climate change on future hurricane induced rainfall and flooding in a coastal watershed: A case study on hurricane harvey, *Journal of Hydrology* 616 (2023) 128774.

- [23] J. Neal, G. Schumann, P. Bates, A subgrid channel model for simulating

- river hydraulics and floodplain inundation over large and data sparse areas, *Water Resources Research* 48 (11) (2012).
- [24] P. D. Bates, J. Savage, O. Wing, N. Quinn, C. Sampson, J. Neal, A. Smith, A climate-conditioned catastrophe risk model for uk flooding, *Natural Hazards and Earth System Sciences* 23 (2) (2023) 891–908.
 - [25] O. E. Wing, P. D. Bates, N. D. Quinn, J. T. Savage, P. F. Uhe, A. Cooper, T. P. Collings, N. Addor, N. S. Lord, S. Hatchard, et al., A 30 m global flood inundation model for any climate scenario, *Water Resources Research* 60 (8) (2024) e2023WR036460.
 - [26] H. C. Edwards, C. R. Trott, D. Sunderland, Kokkos: Enabling manycore performance portability through polymorphic memory access patterns, *Journal of parallel and distributed computing* 74 (12) (2014) 3202–3216.
 - [27] D. A. Beckingsale, J. Burmark, R. Hornung, H. Jones, W. Killian, A. J. Kunen, O. Pearce, P. Robinson, B. S. Ryujin, T. R. Scogland, Raja: Portable performance for large-scale scientific applications, in: 2019 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC), IEEE, 2019, pp. 71–81.
 - [28] NVIDIA, P. Vingelmann, F. H. Fitzek, Cuda, release: 10.2.89 (2020).
URL <https://developer.nvidia.com/cuda-toolkit>
 - [29] A. ROCm, a new era in open gpu computing (2019).
 - [30] J. E. Stone, D. Gohara, G. Shi, Opencl: A parallel programming standard for heterogeneous computing systems, *Computing in science & engineering* 12 (3) (2010) 66.

- [31] X. Xia, Q. Liang, X. Ming, A full-scale fluvial flood modelling framework based on a high-performance integrated hydrodynamic modelling system (hipims), *Advances in Water Resources* 132 (2019) 103392.
- [32] M. K. Sharifian, G. Kesserwani, A. A. Chowdhury, J. Neal, P. Bates, Lisflood-fp 8.1: New gpu accelerated solvers for faster fluvial/pluvial flood simulations, *Geoscientific Model Development Discussions* 2022 (2022) 1–28.
- [33] D. Caviedes-Voullième, M. Morales-Hernández, M. R. Norman, I. Özgen-Xian, *Serghei (serghei-swe) v1. 0: a performance-portable high-performance parallel-computing shallow-water solver for hydrology and environmental hydraulics*, *Geoscientific Model Development* 16 (3) (2023) 977–1008.
- [34] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, J. Zhang, *PETSc/TAO users manual*, Tech. Rep. ANL-21/39 - Revision 3.21, Argonne National Laboratory (2024). doi: 10.2172/2205494.
- [35] J. Brown, A. Abdelfattah, V. Barra, N. Beams, J. S. Camier, V. Dobrev, Y. Dudouit, L. Ghaffari, T. Kolev, D. Medina, W. Pazner, T. Ratnayaka,

- J. Thompson, S. Tomov, libCEED: Fast algebra for high-order element-based discretizations, *Journal of Open Source Software* 6 (63) (2021) 2945. doi:10.21105/joss.02945.
- [36] R. T. Mills, M. F. Adams, S. Balay, J. Brown, A. Dener, M. Knepley, S. E. Kruger, H. Morgan, T. Munson, K. Rupp, et al., Toward performance-portable petsc for gpu-based exascale systems, *Parallel Computing* 108 (2021) 102831.
- [37] J. Brown, V. Barra, N. Beams, L. Ghaffari, M. Knepley, W. Moses, R. Shakeri, K. Stengel, J. L. Thompson, J. Zhang, Performance portable solid mechanics via matrix-free p -multigrid, *arXiv preprint arXiv:2204.01722* (2022).
- [38] D. Xu, G. Bisht, D. Engwirda, D. Feng, Z. Tan, V. Y. Ivanov, Uncertainties in simulating flooding during hurricane harvey using 2d shallow water equations, *Water Resources Research* 61 (1) (2025) e2024WR038032, e2024WR038032 2024WR038032. arXiv:<https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2024WR038032>, doi:<https://doi.org/10.1029/2024WR038032>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2024WR038032>
- [39] J. Kim, A. Warnock, V. Y. Ivanov, N. D. Katopodes, Coupled modeling of hydrologic and hydrodynamic processes including overland and channel flow, *Advances in Water Resources* 37 (2012) 104–126.
- [40] S. F. Bradford, B. F. Sanders, Finite-volume model for shallow-water

- flooding of arbitrary topography, *Journal of hydraulic engineering* 128 (3) (2002) 289–298.
- [41] M. G. Knepley, D. A. Karpeev, Mesh algorithms for PDE with Sieve I: Mesh distribution, *Scientific Programming* 17 (3) (2009) 215–230, <http://arxiv.org/abs/0908.4427>. doi:10.3233/SPR-2009-0249.
URL <http://arxiv.org/abs/0908.4427>
 - [42] M. Lange, L. Mitchell, M. G. Knepley, G. J. Gorman, Efficient mesh management in Firedrake using PETSc-DMPlex, *SIAM Journal on Scientific Computing* 38 (5) (2016) S143–S155. arXiv:<http://arxiv.org/abs/1506.07749>, doi:10.1137/15M1026092.
 - [43] G. Karypis, K. Schloegel, V. Kumar, Parmetis: Parallel graph partitioning and sparse matrix ordering library (1997).
 - [44] L. A. Schoof, V. R. Yarberry, Exodus ii: a finite element data model, Tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States) (1994).
 - [45] V. Hapla, M. G. Knepley, M. Afanasiev, C. Boehm, M. van Driel, L. Krischer, A. Fichtner, Fully parallel mesh I/O using PETSc DMPlex with an application to waveform modeling, *SIAM Journal on Scientific Computing* 43 (2) (2021) C127–C153. arXiv:<http://arxiv.org/abs/2004.08729>, doi:10.1137/20M1332748.
 - [46] D. A. Ham, V. Hapla, M. G. Knepley, L. Mitchell, K. Sagiyaama, Efficient n-to-m checkpointing algorithm for finite element simulations, *SIAM Journal on Scientific Computing* 46 (6) (2024). doi:10.1137/

23M161372.

URL <https://arXiv.org/abs/2401.05868>

- [47] J.-C. Golaz, L. P. Van Roekel, X. Zheng, A. F. Roberts, J. D. Wolfe, W. Lin, A. M. Bradley, Q. Tang, M. E. Maltrud, R. M. Forsyth, et al., The doe e3sm model version 2: Overview of the physical model and initial model evaluation, *Journal of Advances in Modeling Earth Systems* 14 (12) (2022) e2022MS003156.
- [48] K. W. Oleson, D. M. Lawrence, G. B. Bonan, B. Drewniak, M. Huang, C. D. Koven, S. Levis, F. Li, W. J. Riley, Z. M. Subin, et al., Technical description of version 4.0 of the community land model (clm), NCAR Tech. Note NCAR/TN-478+ STR 257 (2010) 1–257.
- [49] O. Delestre, C. Lucas, P.-A. Ksinant, F. Darboux, C. Laguerre, T.-N.-T. Vo, F. James, S. Cordier, Swashes: a compilation of shallow water analytic solutions for hydraulic and environmental studies, *International Journal for Numerical Methods in Fluids* 72 (3) (2013) 269–300.
- [50] S. Steinberg, P. J. Roache, Symbolic manipulation and computational fluid dynamics, *Journal of Computational Physics* 57 (2) (1985) 251–284.
- [51] G. Bisht, W. J. Riley, Development and verification of a numerical library for solving global terrestrial multiphysics problems, *Journal of Advances in Modeling Earth Systems* 11 (6) (2019) 1516–1542.
- [52] S. Li, C. J. Duffy, Fully coupled approach to modeling shallow water flow, sediment transport, and bed evolution in rivers, *Water Resources Research* 47 (3) (2011).

- [53] J.-M. Hervouet, A. Petitjean, Malpasset dam-break revisited with two-dimensional computations, *Journal of hydraulic research* 37 (6) (1999) 777–788.
- [54] O. E. Wing, C. C. Sampson, P. D. Bates, N. Quinn, A. M. Smith, J. C. Neal, A flood inundation forecast of hurricane harvey using a continental-scale 2d hydrodynamic model, *Journal of Hydrology* X 4 (2019) 100039.
- [55] D. Gesch, M. Oimoen, S. Greenlee, C. Nelson, M. Steuck, D. Tyler, The national elevation dataset, *Photogrammetric engineering and remote sensing* 68 (1) (2002) 5–32.
- [56] Q. Liang, F. Marche, Numerical resolution of well-balanced shallow water equations with complex source terms, *Advances in water resources* 32 (6) (2009) 873–884.
- [57] J. Murillo, P. García-Navarro, Augmented versions of the hll and hllc riemann solvers including source terms in one and two dimensions for shallow flow applications, *Journal of Computational Physics* 231 (20) (2012) 6861–6906.
- [58] L. Begnudelli, B. F. Sanders, Unstructured grid finite-volume algorithm for shallow-water flow and scalar transport with wetting and drying, *Journal of hydraulic engineering* 132 (4) (2006) 371–384.
- [59] L. Begnudelli, B. F. Sanders, Conservative wetting and drying methodology for quadrilateral grid finite-volume models, *Journal of Hydraulic Engineering* 133 (3) (2007) 312–322.

- [60] X. Liu, J. Albright, Y. Epshteyn, A. Kurganov, Well-balanced positivity preserving central-upwind scheme with a novel wet/dry reconstruction on triangular grids for the saint-venant system, *Journal of Computational Physics* 374 (2018) 213–236.
- [61] X. Liu, A well-balanced and positivity-preserving numerical model for shallow water flows in channels with wet–dry fronts, *Journal of Scientific Computing* 85 (3) (2020) 60.
- [62] X. Xia, Q. Liang, A new efficient implicit scheme for discretising the stiff friction terms in the shallow water equations, *Advances in water resources* 117 (2018) 87–97.
- [63] D. Bao, Z. G. Xue, J. C. Warner, M. Moulton, D. Yin, C. A. Hegermiller, J. B. Zambon, R. He, A numerical investigation of hurricane florence-induced compound flooding in the cape fear estuary using a dynamically coupled hydrological-ocean model, *Journal of Advances in Modeling Earth Systems* 14 (11) (2022) e2022MS003131.
- [64] D. Bao, Z. G. Xue, J. C. Warner, Quantifying compound and non-linear effects of hurricane-induced flooding using a dynamically coupled hydrological-ocean model, *Water Resources Research* 60 (7) (2024) e2023WR036455.
- [65] Y. Shen, Z. Zhu, Q. Zhou, C. Jiang, An improved dynamic bidirectional coupled hydrologic–hydrodynamic model for efficient flood inundation prediction, *Natural Hazards and Earth System Sciences* 24 (7) (2024) 2315–2330.

- [66] H. Zhang, D. Shen, S. Bao, P. Len, Two-way coupling of the national water model (nwm) and semi-implicit cross-scale hydroscience integrated system model (schism) for enhanced coastal discharge predictions, *Hydrology* 11 (9) (2024) 145.
- [67] D. Xu, G. Bisht, D. Feng, Z. Tan, L. Li, H. Qiu, L. R. Leung, Impacts of sea-level rise on coastal groundwater table simulated by an earth system model with a land-ocean coupling scheme, *Earth's Future* 12 (8) (2024) e2024EF004479.