



SERGHEI (SERGHEI-SWE) v1.0: a performance-portable high-performance parallel-computing shallow-water solver for hydrology and environmental hydraulics

Daniel Caviedes-Voullième^{1,2}, Mario Morales-Hernández^{3,4}, Matthew R. Norman⁴, and Ilhan Özgen-Xian^{5,6}

¹Simulation and Data Lab Terrestrial Systems, Jülich Supercomputing Centre, Forschungszentrum Jülich, Jülich, Germany

²Institute of Bio- and Geosciences: Agrosphere (IBG-3), Forschungszentrum Jülich, Germany

³Fluid Mechanics, I3A, Universidad de Zaragoza, Zaragoza, Spain

⁴Oak Ridge National Laboratory, Oak Ridge, USA

⁵Institute of Geoecology, Technische Universität Braunschweig, Braunschweig, Germany

⁶Earth & Environmental Sciences Area, Lawrence Berkeley National Laboratory, Berkeley, USA

Correspondence: Daniel Caviedes-Voullième (d.caviedes.voullieme@fz-juelich.de)

Received: 22 August 2022 – Discussion started: 8 September 2022

Revised: 9 December 2022 – Accepted: 30 December 2022 – Published: 8 February 2023

Abstract. The Simulation EnviRonment for Geomorphology, Hydrodynamics, and Ecohydrology in Integrated form (SERGHEI) is a multi-dimensional, multi-domain, and multi-physics model framework for environmental and landscape simulation, designed with an outlook towards Earth system modelling. At the core of SERGHEI's innovation is its performance-portable high-performance parallel-computing (HPC) implementation, built from scratch on the Kokkos portability layer, allowing SERGHEI to be deployed, in a performance-portable fashion, in graphics processing unit (GPU)-based heterogeneous systems. In this work, we explore combinations of MPI and Kokkos using OpenMP and CUDA backends. In this contribution, we introduce the SERGHEI model framework and present with detail its first operational module for solving shallow-water equations (SERGHEI-SWE) and its HPC implementation. This module is designed to be applicable to hydrological and environmental problems including flooding and runoff generation, with an outlook towards Earth system modelling. Its applicability is demonstrated by testing several well-known benchmarks and large-scale problems, for which SERGHEI-SWE achieves excellent results for the different types of shallow-water problems. Finally, SERGHEI-SWE scalability and performance portability is demonstrated and evaluated on several TOP500 HPC systems, with very good scaling in the

range of over 20 000 CPUs and up to 256 state-of-the art GPUs.

1 Introduction

The upcoming exascale high-performance parallel-computing (HPC) systems will enable physics-based geoscientific modelling with unprecedented detail (Alexander et al., 2020). Although the need for such HPC systems is traditionally driven by climate, ocean, and atmospheric modelling, hydrological models are progressively becoming as physical, sophisticated, and computationally intensive. Physically based, integrated hydrological models such as Parflow (Kuffour et al., 2020), Amanzi/ATS (Coon et al., 2019), and Hydrogeosphere (Brunner and Simmons, 2012) are becoming more prominent in hydrological research and Earth system modelling (ESM) (Fatichi et al., 2016; Paniconi and Putti, 2015), making HPC more and more relevant for computational hydrology (Clark et al., 2017).

Hydrological models, as with many other HPC applications, are currently facing challenges in exploiting available and future HPC systems. These challenges arise, not only because of the intrinsic complexity of maintaining complex codes over large periods of time, but because HPC and its hardware are undergoing a large paradigm change (Leiserson

et al., 2020; Mann, 2020), which is strongly driven by the end of Moore's law (Morales-Hernández et al., 2020). In order to gain higher processing capacity, computers will require heterogeneous and specialised hardware (Leiserson et al., 2020), potentially making high-performing code harder to develop and maintain and demanding that developers adapt and optimise code for an evolving hardware landscape. It has become clear that upcoming exascale systems will have heterogeneous architectures embedded in modular and reconfigurable architectures (Djemame and Carr, 2020; Suarez et al., 2019) that will consist of different types of CPUs and accelerators, possibly from multiple vendors requiring different programming models. This puts pressure on domain scientists to write *portable* code that *performs* efficiently on a range of existing and future HPC architectures (Bauer et al., 2021; Lawrence et al., 2018; Schulthess, 2015) and to ensure the *sustainability* of such code (Gan et al., 2020).

Different strategies are currently being developed to cope with this grand challenge. One strategy is to offload the architecture-dependent parallelisation tasks to the compiler – see, for example, Vanderbauwhede and Takemi (2013); Vanderbauwhede and Davidson (2018); Vanderbauwhede (2021). Another strategy is to use an abstraction layer that provides a unified programming interface to different computational backends – a so-called “performance portability framework” – that allows the same code to be compiled across different HPC architectures. Examples of this strategy include RAJA (Beckingsale et al., 2019) and Kokkos (Edwards et al., 2014; Trott et al., 2021), which are both very similar in their scope and their capability. Both RAJA and Kokkos are C++ libraries that implement a shared-memory programming model to maximise the amount of code that can be compiled across different hardware devices with nearly the same parallel performance. They allow access to several computational backends, in particular multi-graphics processing unit (GPU) and heterogeneous HPC systems.

This paper introduces the Kokkos-based computational (eco)hydrology framework SERGHEI (Simulation EnviRonnement for Geomorphology, Hydrodynamics, and Ecohydrology in Integrated form) and its surface hydrology module SERGHEI-SWE. The primary aim of SERGHEI's implementation is scalability and performance portability. In order to achieve this, SERGHEI is written in C++ and based from scratch on the Kokkos abstraction. Kokkos currently supports CUDA, OpenMP, HIP, SYCL, and Pthreads as backends. We chose Kokkos over other alternatives because it is actively engaged in securing the sustainability of its programming model, fostering its partial inclusion into ISO C++ standards (Trott et al., 2021). Indeed, there is an increasing number of applications in multiple domains leveraging on Kokkos – for example, Bertagna et al. (2019); Demeshko et al. (2018); Grete et al. (2021); Halver et al. (2020); Watkins et al. (2020). Thus, among other similar solutions, Kokkos has been identified as advantageous in terms of performance portability and project sustainability, although it is perhaps somewhat more

invasive and less clear on the resulting code (Artigues et al., 2019). We present the full implementation of the SERGHEI-SWE module, the shallow-water equations (SWEs) solver for free-surface hydrodynamics at the heart of SERGHEI.

SERGHEI-SWE enables the simulation of surface hydrodynamics of overland flow and streamflow seamlessly and across scales. Historically, hydrological models featuring surface flow have relied on kinematic or zero-inertia (diffusive) approximations due to their apparent simplicity (Caviedes-Voullième et al., 2018; Kollet et al., 2017) and because until the last decade, robust SWE solvers were not available (Caviedes-Voullième et al., 2020a; García-Navarro et al., 2019; Simons et al., 2014; Özgen-Xian et al., 2021). However, the current capabilities of SWE solvers, the increase in computational capabilities, and the need to better exploit parallelism – easier to achieve with explicit solvers than with implicit solvers, as usually required by diffusive equations (Caviedes-Voullième et al., 2018; Fernández-Pato and García-Navarro, 2016) – have been pushing to replace simplified surface flow models (for hydrological purposes) with fully dynamic SWE solvers. There is an increasing number of studies using SWE solvers for rainfall runoff and overland flow simulations from hillslope to catchment scales – for example, Bellos and Tsakiris (2016); Bout and Jetten (2018); Caviedes-Voullième et al. (2012, 2020a); Costabile and Costanzo (2021); Costabile et al. (2021); David and Schmalz (2021); Dullo et al. (2021a, b); Fernández-Pato et al. (2020); García-Alén et al. (2022); Simons et al. (2014); Xia and Liang (2018). This trend contributes to the transition from engineering hydrology towards Earth system science (Sivapalan, 2018), a shift that was motivated by necessity and opportunity, as continental (and larger) ESM will progressively require fully dynamic SWE solvers to cope with increased-resolution digital-terrain models and the dynamics that respond to them, improved spatiotemporal rainfall data and simulations, and increasingly more sophisticated process interactions across scales, from patch to hillslope to catchments (Fan et al., 2019).

SERGHEI-SWE distinguishes itself from other HPC SWE solvers through a number of key novelties. Firstly, SERGHEI-SWE is open sourced under a permissive BSD license. While there are indeed many GPU-enabled SWE codes, many of these are research codes that are not openly available – for example, Aureli et al. (2020); Buttinger-Kreuzhuber et al. (2022); Echeverribar et al. (2020); Hou et al. (2020); Lacasta et al. (2014, 2015); Liang et al. (2016); Vacondio et al. (2017) – or they are commercial codes, such as RiverFlow2D, TUFLOW, HydroAS_2D – see Jodhani et al. (2021) for a recent non-comprehensive review. Open source solvers are a fundamental need for the community, ensuring transparency, reproducibility, and providing a base for model (software) sustainability. We note that open source SWE solvers are becoming increasingly more available – see Table 1. However, only a handful of these freely available models are enabled for GPUs, mostly through CUDA. Fewer

Table 1. Overview of openly available SWE solvers.

Model	Reference	GPU	MPI	Availability	Notes
SERGHEI-SWE	This paper	Kokkos	Yes	Open-source (BSD)	Highly scalable
TRITON	Morales-Hernández et al. (2021)	CUDA	Yes	Open-source (BSD)	Highly scalable
PARFLOOD	Vacondio et al. (2014)	CUDA	Yes	–	Highly scalable; source code can be requested; MPI parallelisation by Turchetto et al. (2019)
HiPIMS	Xia et al. (2019)	CUDA	–	Open-source (GPLv3)	Multi-GPU support based on Thrust (on single node)
DRR/FI	Kobayashi et al. (2015)	–	Yes	–	Highly scalable
SW2D-GPU	Carlotto et al. (2021)	CUDA	–	Open-source	–
LisFlood-FP 8.0	Shaw et al. (2021)	CUDA	–	Open-source (BSD)	SWE solver embedded into LisFlood (Bates and Roo, 2000), which originally did not solve SWE.
IBER	García-Feal et al. (2018)	CUDA	–	Freeware	–
SW2D-Lemon	Caldas Steinstraesser et al. (2021)	–	–	Freeware	Source code can be requested
B-flood	Kirstetter et al. (2021)	–	–	Open-source (GPL)	Adaptive mesh refinement
FullSWOF	Delestre et al. (2017)	–	Yes	Open-source (CeCILL)	MPI parallelisation by Wittmann et al. (2017)
TELEMAC	Moulinec et al. (2011)	–	Yes	Open-source (GPLv3/LGPL)	–
GeoClaw	Berger et al. (2011)	–	Yes	Open-source (BSD)	Adaptive mesh refinement
HEC-RAS2D	Brunner (2021)	–	–	Freeware	–
HMS	Simons et al. (2014)	–	Yes	Open-source (GPL)	MPI parallelisation by Steffen et al. (2020)

of them have multi-GPU capabilities and are capable of fully leveraging HPC hardware. All of these multi-GPU-enabled codes are currently dependent on CUDA and are therefore somewhat limited to Nvidia hardware. This leads into the second and most relevant novelty of SERGHEI-SWE: it is a performance-portable, highly scalable, and GPU-enabled solver. SERGHEI-SWE generalises hardware (CPU, GPU, accelerators) support to a performance-portability concept through Kokkos. This gives SERGHEI-SWE the key advantage of having a single code base for the currently fully operational OpenMP and CUDA backends, as well as HIP, which is currently experimental in SERGHEI but, most importantly, keeps this code base relevant for other backends, such as SYCL. This is particularly important, as the current HPC landscape features not only Nvidia GPUs but also a currently increased adoption of AMD GPUs, with the most recent leading TOP 500 systems – Frontier and LUMI – as well as upcoming systems (e.g. El Capitan) relying on AMD GPUs. In this way, SERGHEI is safely avoiding the vendor lock trap.

SERGHEI-SWE has been developed by harnessing the past 15 years' worth of numerical advances in the solution of SWE, ranging from fundamental numerical formulations (Echeverribar et al., 2019; Morales-Hernández et al., 2020) to HPC GPU implementations (Brodtkorb et al., 2012; Hou et al., 2020; Lacasta et al., 2014, 2015; Liang et al., 2016; Vacondio et al., 2017; Sharif et al., 2020). Most of this work was done in the context of developing solvers for flood modelling, with rather engineering-oriented applications, demanding high quantitative accuracy and predictive capability. Most of the established models in Table 1 were devel-

oped within such contexts, although many are currently also adopted for more hydrological applications. Leveraging on this technology, SERGHEI-SWE is designed to cope with the classical shallow-water applications of fluvial and urban flooding, as well as with the emerging rainfall runoff problems in both natural and urban environments (for which coupling to sewer system models is a longer-term objective) and with other flows of broad hydrological and environmental interest that occur on (eco)hydrological timescales, priming it for further uses in ecohydrology and geomorphology. Nevertheless, all shallow-water applications should benefit from the high performance and high scalability of SERGHEI-SWE. With an HPC-ready SWE solver, catchment-scale rainfall runoff applications around the 1 m² resolution are feasible. Similarly, large river and floodplain simulations can be enabled for operational flood forecasting, and flash floods in urban environments can be tackled with extremely high spatial resolution. Moreover, it is noteworthy that SERGHEI-SWE is not confined to HPC environments, and users with workstations can also benefit from improved performance.

1.1 The SERGHEI framework

SERGHEI is envisioned as a modular simulation framework around a physically based hydrodynamic core, which allows a variety of water-driven and water-limited processes to be represented in a flexible manner. In this sense, SERGHEI is based on the idea of water fluxes as a connecting thread among various components and processes within the Earth system (Giardino and Houser, 2015). As illustrated by the conceptual framework in Fig. 1, SERGHEI's hydrodynamic

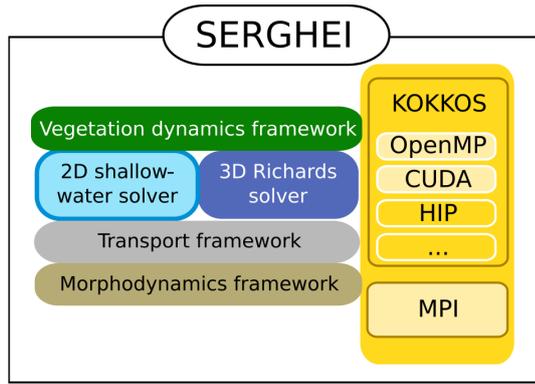


Figure 1. A conceptual framework of SERGHEI.

core will consist of a mechanistic surface (SERGHEI-SWE, the focus of this paper) and subsurface flow solvers (light and dark blue), around which a generalised transport framework for multi-species transport and reaction will be implemented (grey). The transport framework will further enable the implementation of morphodynamics (gold) and vegetation dynamics (green) models. The transport framework will also include a Lagrangian particle-tracking module (currently also under development). At the time of the writing of this paper, the subsurface flow solver – based on the three-dimensional extension of the Richards solver by Li et al. (2021) – is experimentally operative and is underway to being coupled to the surface flow solver, thus making the hydrodynamic core of SERGHEI applicable to integrated surface–subsurface hydrology. The initial infrastructure for the three other transport-based frameworks is currently under development.

2 Mathematical and numerical model of SERGHEI-SWE

In this section we provide an overview of the underlying mathematical model and the numerical schemes implemented in SERGHEI-SWE. The implementation is based on well-established numerical schemes, and consequently, we limit this to a minimal presentation.

SERGHEI-SWE is based on the resolution of the two-dimensional (2D) shallow-water equations that can be expressed in a compact differential conservative form as

$$\left. \begin{aligned} \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} &= \mathbf{S}_r + \mathbf{S}_b + \mathbf{S}_f, \\ \mathbf{U} &= \begin{bmatrix} h \\ q_x \\ q_y \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} q_x \\ \frac{q_x^2}{h} + \frac{1}{2}gh^2 \\ \frac{q_x q_y}{h} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \frac{q_y}{h} \\ \frac{q_x q_y}{h} \\ \frac{q_y^2}{h} + \frac{1}{2}gh^2 \end{bmatrix}, \\ \mathbf{S}_r &= \begin{bmatrix} r_o - r_f \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{S}_b = \begin{bmatrix} 0 \\ -gh \frac{\partial z}{\partial x} \\ -gh \frac{\partial z}{\partial y} \end{bmatrix}, \quad \mathbf{S}_f = \begin{bmatrix} 0 \\ -\sigma_x \\ -\sigma_y \end{bmatrix}. \end{aligned} \right\} \quad (1)$$

Here, t [T] is time, x [L] and y [L] are Cartesian coordinates, \mathbf{U} is the vector of conserved variables (that is to say the unknowns of the system) containing the water depth, h [L], and the unit discharges in x and y directions, called $q_x = hu$ [$L^2 T^{-1}$] and $q_y = hv$ [$L^2 T^{-1}$], respectively. \mathbf{F} and \mathbf{G} are the fluxes of these conserved variables with gravitational acceleration g [$L T^{-2}$]. The mass source terms \mathbf{S}_r account for rainfall, r_o [$L T^{-1}$], and infiltration or exfiltration, r_f [$L T^{-1}$]. The momentum source terms include gravitational bed slope terms, \mathbf{S}_b , expressed according to the gradient of the elevation z [L], and friction terms, \mathbf{S}_f , as a function of the friction slope σ . This friction slope is often modelled by means of Gauckler–Manning’s equation in terms of Manning’s roughness coefficient n [$TL^{-1/3}$] but also frequently with the Chezy and the Darcy–Weisbach formulations (Caviedes-Voullième et al., 2020a). In addition, specialised formulations of the friction slope exist to consider the effect of microtopography and vegetation for small water depths, e.g. variable Manning’s coefficients (Jain and Kothiyari, 2004; Mügler et al., 2011) or generalised friction laws (Özgen et al., 2015b). A recent systematic comparison and in-depth discussion of several friction models with a focus on rainfall runoff simulations is given in Crompton et al. (2020). Implementing additional friction models is of course possible – and relevant, especially to address the multiscale nature of runoff in catchments – but not essential to the points in this paper. The observant reader will note that in Eq. (1), viscous and turbulent fluxes have been neglected. The focus here is on applications (rainfall runoff, dam breaks) where the influence of these can be safely neglected. Turbulent viscosity may become significant for ecohydraulic simulations of river flow, and turbulent fluxes of course play an important role in mixing in transport simulations. We will address these issues in future implementations of the transport solvers in SERGHEI.

SERGHEI-SWE uses a first-order accurate upwind finite-volume scheme with a forward Euler time integration to solve the system of Eq. (1) on uniform Cartesian grids with grid spacing Δx [L]. The numerical scheme, presented in detail in Morales-Hernández et al. (2021), harnesses many solutions that have been reported in the literature in the past decade, ensuring that all desirable properties of the scheme (well-balancing, depth positivity, stability, robustness) are preserved under the complex conditions of realistic envi-

ronmental problems. In particular, we require the numerical scheme to stay robust and accurate in the presence of arbitrary rough topography and shallow-water depths with wetting and drying.

Well-balancing and water depth positivity are ensured by solving numerical fluxes at each cell edge k with augmented Riemann solvers (Murillo and García-Navarro, 2010, 2012) based on the Roe linearisation (Roe, 1981). In fluctuation form, the rule for updating the conserved variables in cell i from time step n to time step $n + 1$ reads as follows:

$$\mathbf{U}_i^* = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x} \sum_{k=1}^4 \sum_{m=1}^3 \frac{\tilde{\lambda}_k^-}{\tilde{\lambda}_k} [(\tilde{\lambda}_k \tilde{\alpha} - \tilde{\beta}) \tilde{\mathbf{e}}]_{m,k}^n, \quad (2)$$

followed by

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^* + (r_o - r_f)_i^n \Delta t, \quad (3)$$

where $\tilde{\lambda}$ and $\tilde{\mathbf{e}}$ are the eigenvalues and eigenvectors of the linearised system of equations, $\tilde{\alpha}$ and $\tilde{\beta}$ are the fluxes and bed slope and friction source term linearisations, respectively, and the minus sign accounts for the upwind discretisation. Note that all the tilde variables are defined at each computational edge. The time step Δt is restricted to ensure stability, following the Courant–Friedrichs–Lewy (CFL) condition:

$$\Delta t = \text{CFL} \min_i \left\{ \frac{\Delta x}{\left| \frac{q_x}{h} \right|_i + \sqrt{gh_i}, \left| \frac{q_y}{h} \right|_i + \sqrt{gh_i}} \right\} \\ \text{CFL} \leq 0.5. \quad (4)$$

Although the wave speed values are formally defined at the interfaces, the corresponding cell values are used instead for the CFL condition. As pointed in Morales-Hernández et al. (2021), this approach does not compromise the stability of the system but accelerates the computations and simplifies the implementation.

It is relevant to acknowledge that second (and higher)-order schemes for SWE are available (e.g. Buttinger-Kreuzhuber et al., 2019; Caviedes-Voullième et al., 2020b; Hou et al., 2015; Navas-Montilla and Murillo, 2018). However, first-order schemes are still a pragmatic choice (Ayog et al., 2021), especially when dealing with very high resolutions (as targeted with SERGHEI), which offsets their higher discretisation error and numerical diffusivity in comparison to higher-order schemes. Similarly, robust schemes for unstructured triangular meshes are well established together with their well-known advantages in reducing cell counts and numerical diffusion (Bomers et al., 2019; Caviedes-Voullième et al., 2012, 2020a). As these advantages are less relevant at very high resolutions, we opt for Cartesian grids to avoid issues with memory mapping, coalescence and cache misses in GPUs (Lacasta et al., 2014), and additional memory footprints while also making domain decomposition simpler. Both higher-order schemes and unstructured (and adaptive) meshes may also be implemented within SERGHEI.

3 HPC implementation of the SERGHEI framework

In this section we describe the key ingredients of the HPC implementation of SERGHEI. Conceptually, this requires, firstly, handling parallelism inside a computational device (multicore CPU or GPU) with shared memory and the related portability and corresponding backends (i.e. OpenMP, CUDA, HIP, etc.). On a higher level of parallelism, distributing computations across many devices requires domain decomposition and a distributed memory problem, implemented via MPI. The complete implementation of SERGHEI encompasses both, distributing parallel computations into many subdomains, each of which is mapped onto a computational device. Here we start the discussion from the higher level of domain decomposition and highlight that the novelty of SERGHEI lies with the multiple levels of parallelism together with the performance-portable shared-memory approach via Kokkos.

3.1 Domain decomposition

The surface domain is a two-dimensional plane, discretised by a Cartesian grid with a total cell number of $N_t = N_x N_y$, where N_x and N_y are the number of cells in x and y directions, respectively. Operations are usually performed per subdomain, each one associated with an MPI rank. During initialisation, each MPI process constructs a local subdomain with n_x cells in x direction and n_y cells in y direction. The user specifies the number of subdomains in each Cartesian direction at runtime, and SERGHEI determines the subdomain size from this information. Subdomains are the same size, except for correction due to non-integer-divisible decompositions. In order to communicate information across subdomains, SERGHEI uses so-called “halo cells”, non-physical cells on the boundaries of the subdomain that overlap with physical cells from neighbouring subdomains. The halo cells augment the number of cells in x and y direction by 1 at each boundary. Thus, the subdomain size is $n_t = (n_x + 2)(n_y + 2)$. The definitions are sketched – without loss of generality – for a square-shaped subdomain in Fig. 2, and the way these subdomains overlap in the global domain is sketched in Fig. 3 (left). Halo cells are not updated as part of the time stepping. Instead, they are updated by receiving data from the neighbouring subdomain, a process which naturally requires MPI communications.

Besides the global cell index that ranges from 0 to N_t , each subdomain uses two sets of local indices to access data stored in its cells. The first set spans over all physical cells inside the subdomain, and the second index spans over both halo cells and physical cells – see Fig. 2. The second set maps into memory position. For example, in order to access the physical cell 14 in Fig. 2, one has to access memory position 27.

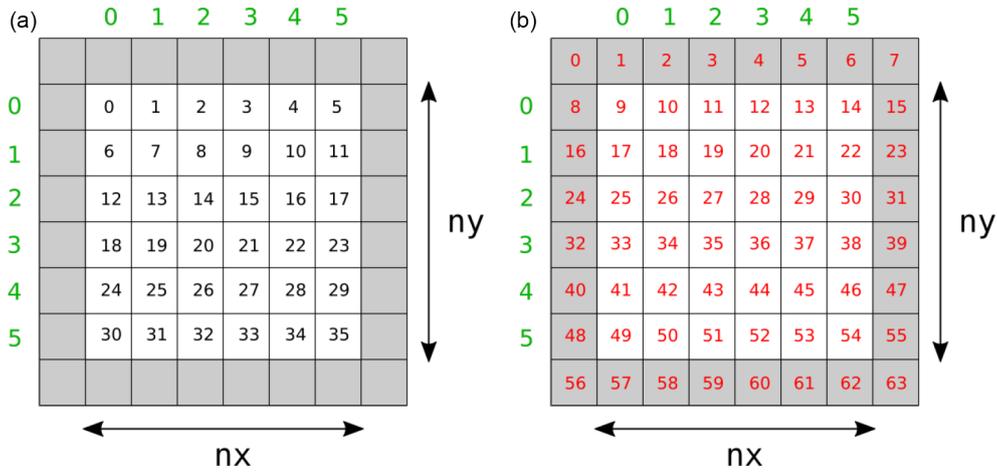


Figure 2. Domain decomposition and indexing in SERGHEI: a subdomain consists of physical cells (white) and halo cells (grey). SERGHEI uses two sets of indices: an index for physical cells (a) and an index for all cells including the halo cells (b).

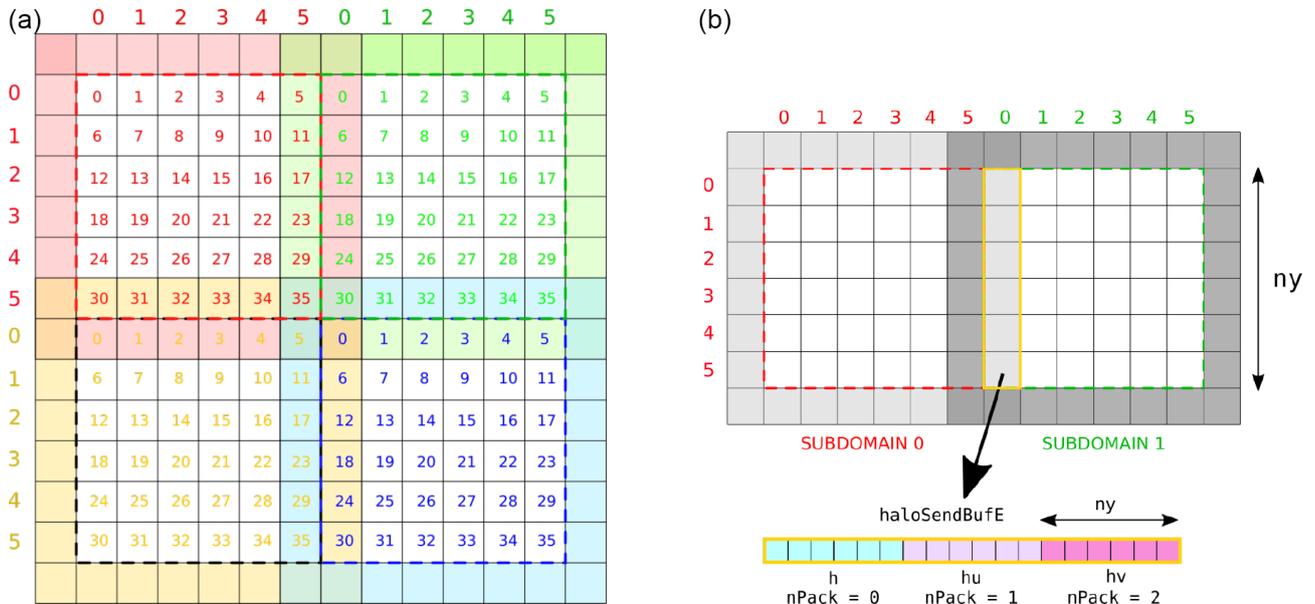


Figure 3. Data exchange between subdomains in SERGHEI: in the global surface domain, subdomains overlap with each other through their halo cells (a). These halo cells are used to exchange data between the subdomains (b).

3.2 Data exchange between subdomains

The underlying methods for data exchange between subdomains are centred on the subdomains rather than on the interfaces. Data are exchanged through MPI-based send-and-recv calls (non-blocking) that aggregate data in the halo cells across the subdomains. Note that, by default, Kokkos implicitly assumes that the MPI library is GPU aware, allowing GPU-to-GPU communication provided that the MPI libraries support this feature. Figure 3 (right) illustrates the concept of sending a halo buffer containing state variables from subdomain 1 to update halo cells of subdomain 0. The halo buffer contains state variables for n_y cells, grouped as

water depth (h), unit discharge in x direction (hu), and unit discharge in y direction (hv).

3.3 Performance-portable implementation

Intra-device parallelism is achieved per subdomain through the Kokkos framework, which allows the user to choose between shared-memory parallelism and GPU backends for further acceleration. SERGHEI's implementation makes use of the Kokkos concept of Views, which are memory-space-aware abstractions. For example, for arrays of real numbers, SERGHEI defines a type `realArr`, based on `View`. This takes the form of Listing 1 for the shared (host) memory

space and Listing 2 for the unified virtual memory (UVM) GPU device CUDA memory space. The UVM significantly facilitates development while avoiding writing explicit host-to-device (and vice versa) memory movements.

For a CUDA backend, the use of unified memory (`CudaUVMspace`) is shown in Listing 2.

Similar definitions can be constructed for integer arrays. These arrays describe spatially distributed fields, such as conserved variables, model parameters, and forcing data. Deriving these arrays from `View` allows us to operate on them via Kokkos to achieve performance portability.

Conceptually, the SERGHEI-SWE solver consists of two computationally intensive kernels: (i) cell-spanning and (ii) edge-spanning kernels. The update of the conserved variables following Eq. (2) results in a kernel around a cell-spanning loop. These cell-spanning loops are the most frequent ones in SERGHEI-SWE and are used for many processes of different computational demand. The standard C++ implementation of such a kernel is illustrated in Listing 3, which spans indices `i` and `j` of a 2D Cartesian grid. Here, the loops may be parallelised using, for example, OpenMP or CUDA. However, such a direct implementation of, for example, an OpenMP parallelisation would not automatically allow leveraging GPUs. That is to say, such an implementation is not portable.

In order to achieve the desired portability, we replace the standard `for` by a `Kokkos::parallel_for`, which enables a lambda function, is minimally intrusive, and reformulates this kernel to the code shown in Listing 4. As a result, this implementation can be compiled for both OpenMP applications and GPUs with Kokkos handling the low-level parallelism on different backends.

Edge-spanning loops are conceptually necessary to compute numerical fluxes (Eq. 2). Although numerical fluxes can be computed in a cell-centred fashion, this would lead to inefficiencies due to duplicated computations. In Listing 5 we illustrate the edge-spanning kernel solving the numerical fluxes in SERGHEI-SWE. Notably, Listing 5 is indexed by cells, and the construction of edge-wise tuples occurs inside of the kernel. This bypasses the need for additional memory structures to hold edge-based information, but only for Cartesian meshes. Generalisation to adaptive or unstructured meshes would require explicitly an edge-based loop with an additional `View` of size equal to the number of edges.

4 Verification and validation

In this section we report evidence supporting the claim that SERGHEI-SWE is an accurate, robust and efficient shallow-water solver. The formal accuracy testing strategy is based on several well-known benchmark cases with well-defined reference solutions. Herein, for brevity, we focus only on the results of these tests, while providing a minimal presentation of the set-ups. We refer the interested reader to the orig-

Table 2. Analytical steady flows: summary of L norms for errors in water depth; L norms for errors in unit discharge are in the range of machine accuracy and omitted here.

Case	L_1 (m)	L_2 (m)	L_∞ (m)
Fig. 4	0.0	0.0	0.0
Fig. 5 (left)	0.68584	0.01909	0.0015
Fig. 5 (right)	1.02096	0.06826	0.0622

inal publications (and to the many instances in which these tests have been used) for further details on the geometries, parametrisations and forcing.

We purposely report an extensive testing exercise to show the wide applicability of SERGHEI across hydraulic and hydrological problems, with a wide range of the available benchmark tests. Analytical, experimental and field-scale tests are included. The first are aimed at showing formal convergence and accuracy. The experimental cases are meant as validation of the capabilities of the model to reach physically meaningful solutions under a variety of conditions. The field-scale tests showcase the applicability of the solver for real problems, and allow for strenuous computational tasks to show performance, efficiency, and parallel scaling. All solutions reported here were computed using double precision arithmetic.

4.1 Analytical steady flows

We test SERGHEI's capability to capture moving equilibria in a number of steady-flow test cases compiled in Delestre et al. (2013). Details of the test cases for reproduction purposes can be retrieved from Delestre et al. (2013) and the accompanying software, SWASHES – in this work, we use SWASHES version 1.03. In the following test cases, the domain is always discretised using 1000 computational cells. A summary of L norms for all test cases is given in Table 2. The definition of the L norms is given in Appendix A.

4.1.1 C property

These tests feature a smooth bump in a one-dimensional, frictionless domain which can be used to validate the C property, well-balancing, and the shock-capturing ability of the numerical solver (Morales-Hernández et al., 2012; Murillo and García-Navarro, 2012). Figure 4 shows that SERGHEI-SWE satisfies the C property by preserving a *lake at rest* in the presence of an emerged bump (an immersed bump test is shown in Sect. A1) and matches the analytical solution provided by SWASHES.

4.1.2 Well-balancing

To show well-balancing under steady flow, we computed two transcritical flows based on the analytical bench-

```
typedef Kokkos::View<real*,Kokkos::LayoutRight> realArr;
```

Listing 1. `realArr` definition based on View for CPU.

```
typedef Kokkos::View<real*,Kokkos::LayoutRight,
    Kokkos::Device<Kokkos::Cuda,Kokkos::CudaUVMSpace>> realArr;
```

Listing 2. `realArr` definition based on View for GPU.

mark of a one-dimensional flume with varying geometry proposed by MacDonald et al. (1995). These tests are well known and widely used as benchmark solutions (e.g. Caviedes-Voullième and Kesserwani, 2015; Delestre et al., 2013; Kesserwani et al., 2019; Morales-Hernández et al., 2012; Murillo and García-Navarro, 2012). Additional well-balancing tests can be found in Sect. A2. At steady state, local acceleration terms and source terms balance each other out such that the free surface water elevation becomes a function of bed slope and friction source terms. Thus, these test cases can be used to validate the implementation of these source terms and the well-balanced nature of the complete numerical scheme. This is particularly important to subcritical fluvial flows and rainfall runoff problems, since both are usually dominated by these two terms.

Figure 5 shows comparisons between SERGHEI-SWE and analytical solutions (obtained through SWASHES) for two transcritical steady flows. Very good agreement is obtained. Note that the unit discharge is captured with machine accuracy in the presence of friction and bottom changes, which is mainly due to the upwind friction discretisation used in the SERGHEI-SWE solver. As reported by Burguete et al. (2008) and Murillo et al. (2009), a centred friction discretisation does not ensure a perfect balance between fluxes and source terms for steady states even if using the improved discretisation by Xia et al. (2017).

4.2 Analytical dam break

We verify SERGHEI-SWE’s capability to capture transient flow based on analytical dam breaks (Delestre et al., 2013). Dam break problems are defined by an initial discontinuity in the water depth in the domain $h(x)$, such that

$$h(x) = \begin{cases} h_L & \text{if } x \leq x_0, \\ h_R & \text{otherwise,} \end{cases} \quad (5)$$

where h_L denotes a specified water depth on the left-hand side of the location of the discontinuity x_0 , and h_R denotes the specified water depth on the right-hand side of x_0 . The domain is 10 m long, the discontinuity is located at $x_0 = 5$ m, and the total run time is 6 s. Initial velocities are nil in the entire domain. In the following, we report empirical evidence of the numerical-schemes mesh convergence property by

comparing model predictions for test cases with 100, 1000, 10 000, and 100 000 elements, respectively.

A classical frictionless dam break over a wet bed is reported in Sect. A3. Here we focus on a frictionless dam break over a dry bed. Flow featuring depth close to dry bed is a special case for the numerical solver because regular wave speed estimations become invalid Toro (2001). Initial conditions are set as $h_L = 0.005$ m and $h_R = 0$ m. Model results are plotted against the analytical solution by Ritter for different grid resolutions in Fig. 6. The model results converge to the analytical solution as the grid is refined. This is also seen in Table 3, where errors and convergence rates for this test case are summarised. Note that the norms definition can be found in Sect. A2. The observed convergence rate is below the theoretical convergence rate of $R = 1$ because of the increased complexity introduced by the discontinuity in the solution and the presence of dry bed.

4.3 Analytical oscillation: parabolic bowl

We present transient two-dimensional test cases with moving wet–dry fronts that consider the periodical movement of water in a parabolic bowl, so-called “oscillations” that have been studied by Thacker (1981). We replicate two cases from the SWASHES compilation (Delestre et al., 2013), using a mesh spacing of $\Delta x = 0.01$ m, one reported here and the other in Sect. A4.

The well-established test case by Thacker (1981) for a periodic oscillation of a planar surface in a frictionless paraboloid has been extensively used for validation of shallow-water solvers (e.g. Aureli et al., 2008; Dazzi et al., 2018; Liang et al., 2015; Murillo and García-Navarro, 2010; Vacondio et al., 2014; Zhao et al., 2019) because of its rather complex 2D nature and the presence of moving wet–dry fronts. The topography is defined as

$$z(r) = -h_0 \left(1 - \frac{r^2}{a^2} \right),$$

$$r = \sqrt{(x - L/2)^2 + (y - L/2)^2}, \quad (6)$$

where r is the radius, h_0 is the water depth at the centre of the paraboloid, a is the distance from the centre to the zero-elevation shoreline, L is the length of the square-shaped domain, and x and y denote coordinates inside the domain.

```

inline void computeNewState(State &state, const Domain &dom, const SourceSinkData &ss){
    for (int j=0; j<dom.ny; j++){
        for (int i=0; i<dom.nx; i++){
            // computationally intensive code to update cells
        }
    }
}

```

Listing 3. Conserved variable update in standard C++.

```

inline void computeNewState(State &state, const Domain &dom, const SourceSinkData &ss){
    Kokkos::parallel_for(dom.nCellDomain, KOKKOS_LAMBDA(int iGlob){
        // computationally intensive code to update cells
    })
}

```

Listing 4. Conserved variable update using Kokkos.

The analytical solution is derived in Thacker (1981). We use the same values as Delestre et al. (2013), that is $h_0 = 0.1$ m, $a = 1$ m, and $L = 4$ m. The simulation is run for three periods ($T = 2.242851$ s), with a spatial resolution of $\delta x = 0.01$ m. The analytical solution can be found in Thacker (1981); Delestre et al. (2013).

Snapshots of the simulation are plotted in Fig. 7 and compared to the analytical solution. The model results agree well with the analytical solution after three periods, with slight growing-phase error, as is commonly observed on this test case.

4.4 Variable rainfall over a sloping plane

Govindaraju et al. (1990) presented an analytical solution to a time-dependent rainfall over a sloping plane, which is commonly used (Caviedes-Voullième et al., 2020a; Gottardi and Venutelli, 2008; Singh et al., 2015). The plane is 21.945 m long, with a slope of 0.04. We select rainfall B from Govindaraju et al. (1990), a piecewise constant rainfall with two periods of alternating low and high intensities (50.8 and 101.6 mm h⁻¹) up until 2400 s. Friction is modelled with Chezy's equation, with a roughness coefficient of 1.767 m^{1/2} s⁻¹. The computational domain was defined by a 200 × 10 grid, with $\delta x = 0.109725$ m.

The simulated discharge hydrograph at the outlet is compared against the analytical solution in Fig. 8. The numerical solutions matches the analytical one very well. The only relevant difference occurs in the magnitude of the second discharge peak, which is slightly underestimated in the simulation.

5 Laboratory-scale experiments

5.1 Experimental steady and dam break flows over complex geometry

Martínez-Aranda et al. (2018) presented experimental results of steady and transient flows over several obstacles while recording transient 3D water surface elevation in the region of interest. We selected the so-called G3 case and simulated both a dam break and steady flow. The experiment took place in a double-sloped plexiglass flume, 6 m long and 24 cm wide. The obstacles in this case are a symmetric contraction and a rectangular obstacle on the centreline, downstream of the contraction.

For both cases the flume (including the upstream wider reservoir) was discretised at a 5 mm resolution, resulting in a computational domain with 106 887 cells. Manning's roughness was set to 0.01 s m^{-1/3}. The steady simulation was run from an initial state with uniform depth $h = 5$ cm up to $t = 300$ s. The dam break simulation duration was 40 s.

The steady-flow case had a discharge of 2.5 L s⁻¹. Steady water surface results in the obstacle region are shown in Fig. 9 for a centreline profile ($y = 0$) and a cross-section at the rectangular obstacle, specifically at $x = 2.40$ m (the coordinate system is set at the centre of the flume inlet gate). The simulation results approximate experimental results well. The mismatches are similar to those analysed by Martínez-Aranda et al. (2018) and can be attributed to turbulent and 3D phenomena near the obstacles.

The dam break case is triggered by a sudden opening of the gate followed by a wave advancing along the dry flume. Results for this case at three gauge points are shown in Fig. 10. Again, the simulations approximate experiments well, capturing both the overall behaviour of the water depths and the

```

inline void computeDeltaFluxXRoe(State &state , Domain const &dom, Parallel &par) {
  Kokkos::parallel_for( dom.ncells , KOKKOS_LAMBDA (int iGlob) {
    int i, j, ncells;
    int id1, id2;
    unpackIndices(iGlob, dom.ny+2*hc, dom.nx+2*hc, j, i);
    if (i>hc-2 && i<dom.nx+hc && j>hc-1 && j<dom.ny+hc){
      ncells=dom.ncells;
      id1=iGlob;
      id2=j*(dom.nx+2*hc)+i+1;
      // computationally intensive code to compute fluxes at the edge between cells id1 and id2
    }
  }
}

```

Listing 5. Flux computations.

Table 3. Analytical dam break: L norms and empirical convergence rates (R) for water depth (h) and velocity (u).

n	$L_1(h)$ (m)	$L_2(h)$ (m)	$R(h)$ (m)	$L_1(u)$ ((ms^{-1}))	$L_2(u)$ ((ms^{-1}))	$R(u)$ ((ms^{-1}))
100	0.01566	0.02343	–	0.23	0.526	–
1000	0.00396	0.00645	0.6	0.138	0.4053	0.22
10 000	0.00068	0.00137	0.76	0.08169	0.34	0.22
100 000	0.0001	0.00026	0.83	0.04193	0.248	0.28

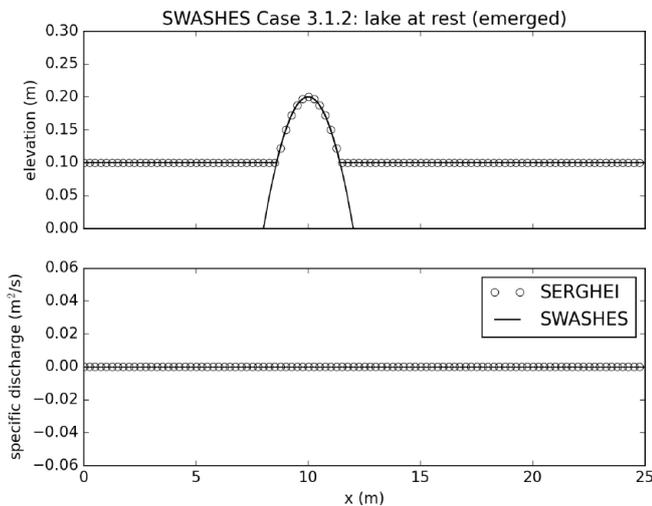


Figure 4. Lake at rest solution for emerged bump. SERGHEI-SWE satisfies the C property.

arrival of the dam break wave, with local errors attributable to the violent dynamics (Martínez-Aranda et al., 2018).

5.2 Experimental unsteady flow over an island

Briggs et al. (1995) presented an experimental test of an unsteady flow over a conical island. This test has been extensively used for benchmarking (Bradford and Sanders, 2002; Choi et al., 2007; García-Navarro et al., 2019; Hou et al., 2013b; Liu et al., 1995; Lynett et al., 2002; Nikolos and Delis, 2009). A truncated cone of base diameter 7.2 m and

top diameter 2.2 m and with a height of 0.625 m was placed at the centre of a 26 m × 27.6 m smooth and flat domain. An initial hydrostatic water level of $h_0 = 0.32$ m was set, and a wave was imposed on the boundary following

$$h_b = h_0 + A \operatorname{sech}^2 \left(\frac{B(t-T)}{C} \right), \tag{7}$$

$$B = \sqrt{gh_0} \left(1 + \frac{A}{2h_0} \right), \tag{8}$$

$$C = h_0 \sqrt{\frac{4h_0 B}{3A \sqrt{gh_0}}}, \tag{9}$$

where $A = 0.032$ m is the wave amplitude, and $T = 2.84$ s is the time at which the peak of the wave enters the domain. Figure 11 shows results for a simulation with a 2.5 cm resolution, resulting in 1.2 million cells. A roughness coefficient of $0.013 \text{ s m}^{-1/3}$ was used for the concrete surface. The results are comparable to previous solutions in the literature, in general reproducing well the water surface, with some delay over experimental measurements.

5.3 Experimental rainfall runoff over an idealised urban area

Cea et al. (2010a) presented experimental and numerical results for a range of laboratory-scale rainfall runoff experiments on an impervious surface with different arrangements of buildings, which have been frequently used for model validation (Caviedes-Voullième et al., 2020a; Cea et al., 2010b; Cea and Bladé, 2015; Fernández-Pato et al., 2016; Su et al., 2017; Xia et al., 2017). This laboratory-scale test includes

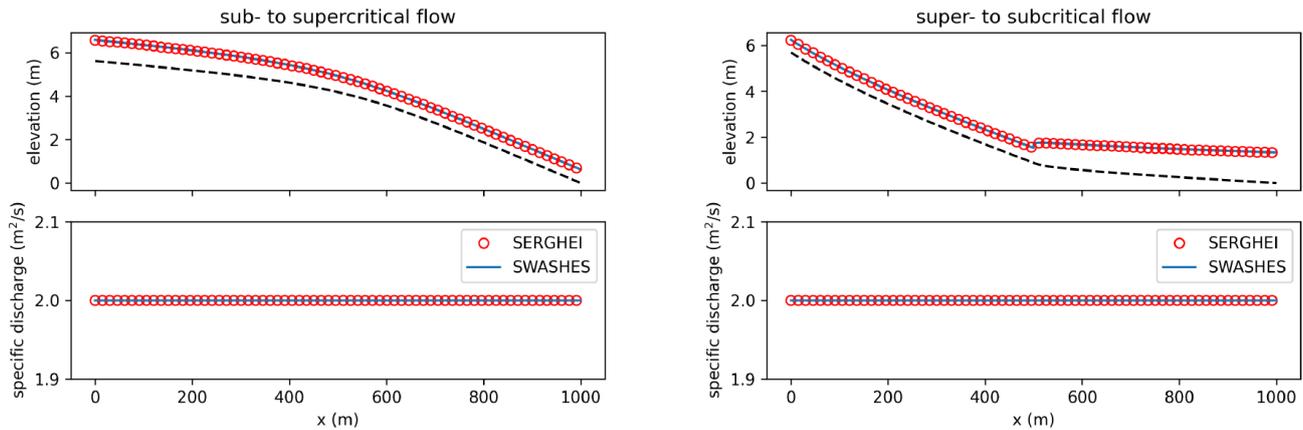


Figure 5. Analytical steady flows: flumes. SERGHEI-SWE captures moving equilibria solutions for two transcritical steady flows. Note that the solution is stable (no oscillations) and well-balanced (discharge remains constant along the flume).

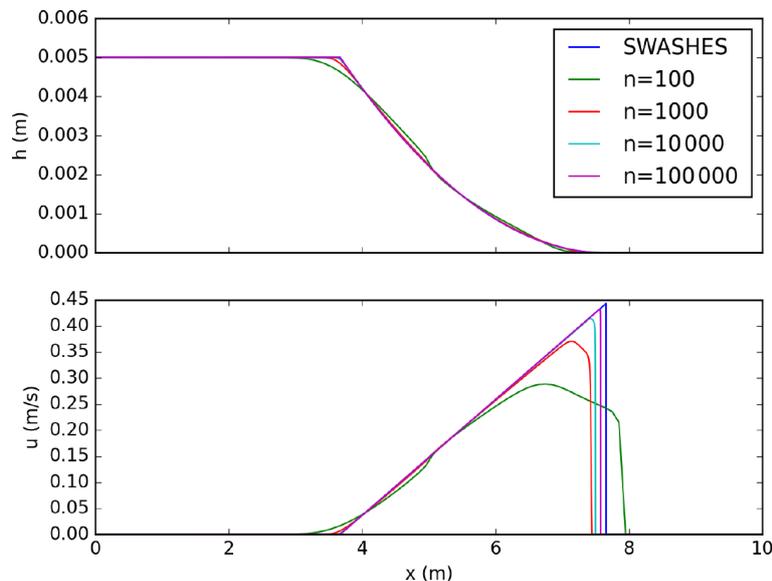


Figure 6. Dam break on dry bed without friction: model predictions for different number of grid cells. SERGHEI-SWE converges to the analytical solution (Ritter's solution) as the grid is refined.

non-trivial topographies, small water layers, and wetting–drying fronts, making it a good benchmark for realistic rainfall runoff conditions.

The dimensions of the experimental flume are $2\text{ m} \times 2.5\text{ m}$. Here, we select one building arrangement named A12 by Cea et al. (2010a). The original digital elevation model (DEM) is available (from Cea et al., 2010a) at a resolution of 1 cm. The buildings are 20 cm high and are represented as topographical features on the domain. All boundaries are closed, except for the free outflow at the outlet. The domain was discretised with a $\delta x = 1\text{ cm}$ resolution, resulting in 54 600 cells. The domain was forced by two constant pulses of rain of 85 and 300 mm h^{-1} (lowest and highest intensities in the experiments) with durations of 60 and 20 s. The simulation was run up to $t = 200\text{ s}$. Friction

was modelled by Manning's equation, with a constant roughness coefficient of $0.010\text{ s m}^{-1/3}$ for steel (Cea et al., 2010a).

Figure 12 shows the experimental and simulated outflow discharge for both rainfall pulses. There is a very good qualitative agreement, and peak flow is quantitatively well reproduced by the simulations. For the 300 mm h^{-1} intensity rainfall, the onset of runoff is earlier than in the experiments, and overall the hydrograph is shifted towards earlier times. Cea et al. (2010a) observed a similar behaviour and pointed out that this is likely caused by surface tension during the early wetting of the surface, and it was most noticeable on the experiments with higher rainfall intensity.

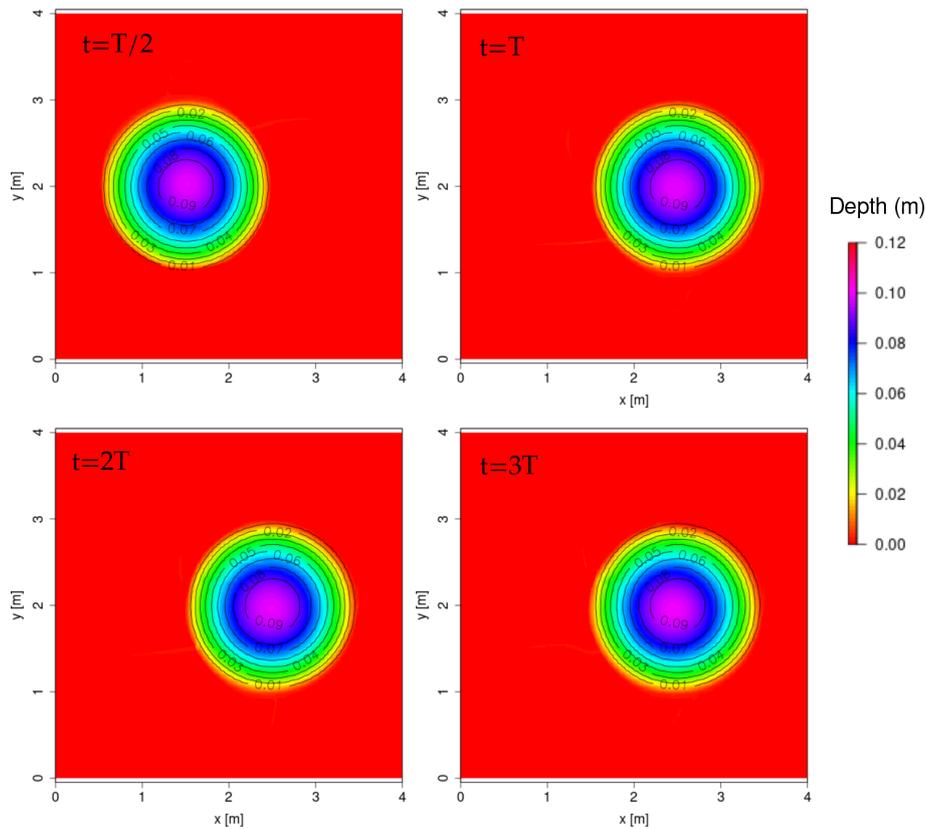


Figure 7. Planar surface in a paraboloid: snapshots of water depth by the model compared to the analytical solution (contour lines). Period $T = 2.242851$ s.

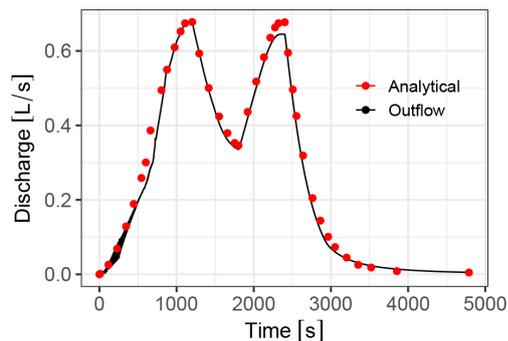


Figure 8. Simulated and analytical discharge for the analytical case of rainfall in a flume.

6 Plot-scale to catchment-scale experiments

6.1 Plot-scale field rainfall runoff experiment

Tatard et al. (2008) presented a rainfall runoff plot-scale experiment performed in Thiès, Senegal. This test has been used often for benchmarking of rainfall runoff models (Caviedes-Voullième et al., 2020a; Chang et al., 2016; Mügler et al., 2011; Özgen-Xian et al., 2020; Park et al., 2019; Simons et al., 2014; Yu and Duan, 2017; Weill, 2007). The

domain is a field plot of $10 \text{ m} \times 4 \text{ m}$, with an average slope of 0.01. A rainfall simulation with an intensity of 70 mm h^{-1} during 180 s was performed. Steady velocity measurements were taken at 62 locations. The Gauckler–Manning roughness coefficient was set to $0.02 \text{ s m}^{-1/3}$, and a constant infiltration rate was set to $0.0041667 \text{ mm s}^{-1}$ (Mügler et al., 2011). The domain was discretised with $\delta x = 0.02666 \text{ m}$, resulting in 56 250 cells, with a single free-outflow boundary downslope.

Simulated velocities are compared to experimental velocities at the 62 gauged locations in Fig. 13. A good agreement between simulated and experimental velocities exists, especially in the lower-velocity range. The agreement is similar to previously reported results (e.g. Caviedes-Voullième et al., 2020a), and the differences between simulated and observed velocities have been shown to be a limitation of a depth-independent roughness and Manning’s model (Mügler et al., 2011).

6.2 Malpasset dam break

The Malpasset dam break event (Hervouet and Petitjean, 1999) is the most commonly used real-scale benchmark test in shallow-water modelling (An et al., 2015; Brodtkorb et al., 2012; Brufau et al., 2004; Caviedes-Voullième et al.,

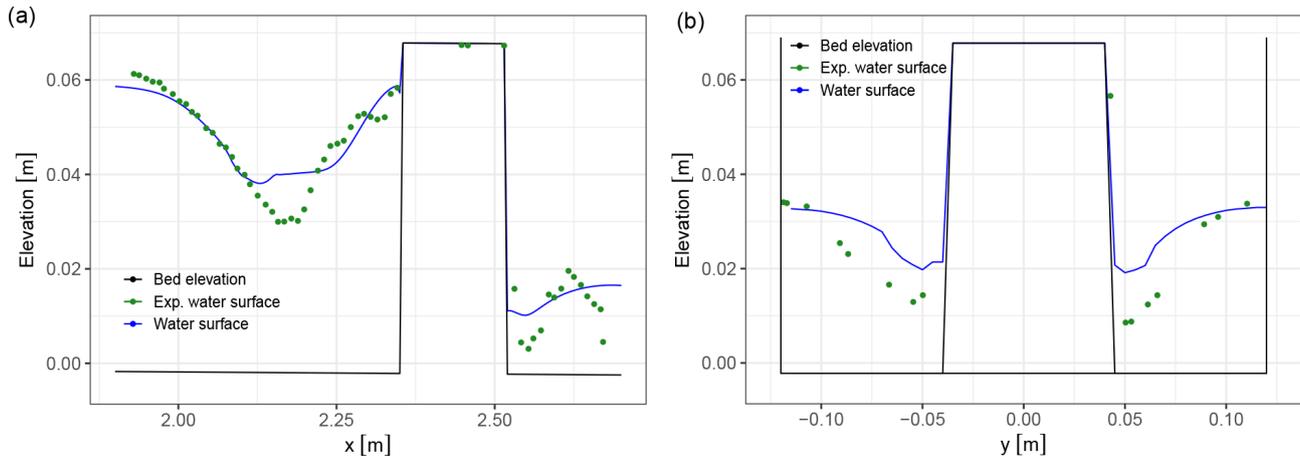


Figure 9. Simulated and experimental steady water surface in the obstacle region of the G3 flume for the centreline profile $y = 0$ m (a) and a cross-section $x = 2.40$ m (b).

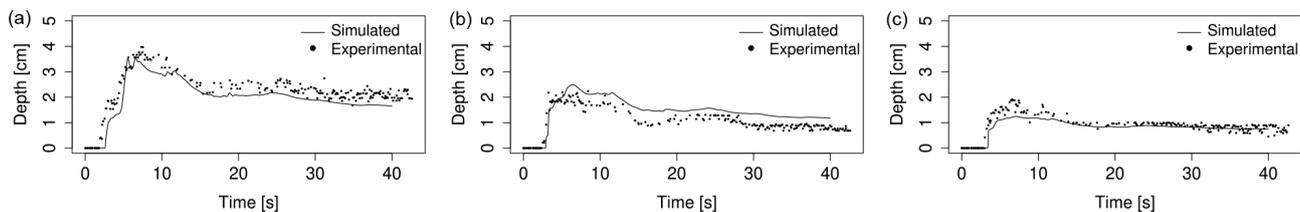


Figure 10. Simulated and experimental transient water depths at three gauge points (a $x = 2.25$ m, $y = 0$ m; b $x = 2.40$ m, $y = 0.08$ m; c $x = 2.60$ m, $y = 0$ m) for the G3 flume dam break over several obstacles.

2020b; Duran et al., 2013; George, 2010; Hervouet and Petitjean, 1999; Hou et al., 2013a; Kesserwani and Liang, 2012; Kesserwani and Sharifian, 2020; Kim et al., 2014; Liang et al., 2007; Sætra et al., 2015; Schwanenberg and Harms, 2004; Smith and Liang, 2013; Valiani et al., 2002; Xia et al., 2011; Yu and Duan, 2012; Wang et al., 2011; Zhou et al., 2013; Zhao et al., 2019). Although it may not be particularly challenging for current solvers, it remains an interesting case due to its scale and the available field and experimental data (Aureli et al., 2021). The computational domain was discretised to $\delta x = 25$ m and $\delta x = 10$ m (resulting in 83 137 and 515 262 cells, respectively). The Gauckler–Manning coefficient was set to a uniform value of $0.033 \text{ s m}^{-1/3}$, which has been shown to be a good approximation in the literature. Figure 14 shows a comparison of simulated water surface elevation (WSE) and arrival time for two resolutions against the reference experimental and field data. Figure 15 shows the geospatial distribution of the relative WSE error and the ratio of the simulated arrival time to the observed time. Overall, WSE shows a good agreement and somewhat smaller scatter for the higher resolution. Arrival time tends to be overestimated, somewhat more for coarser resolutions.

7 Performance and scaling

In this section we report an investigation of the computational performance and parallel scaling of SERGHEI-SWE for selected test cases. To demonstrate performance portability, we show performance metrics for both OpenMP and CUDA backends enabled by Kokkos, computed on CPU and GPU architectures, respectively. For that, hybrid MPI-OpenMP and MPI-CUDA implementations are used, with one MPI task per node for MPI-OpenMP and one MPI task per GPU for MPI-CUDA. Most of the runs were performed on JUWELS at JSC (Jülich Supercomputing Centre). Additional HPC systems were also used for some cases. Properties of all systems are shown in Table 4. Additionally, we provide performance metrics on non-HPC systems, including some consumer-grade GPUs.

It is important to highlight that no performance tuning or optimisation has been carried out for these tests and that no system-specific porting efforts were done. All runs relied entirely on Kokkos for portability. The code was simply compiled with the available software stacks in the HPC systems and executed. All results reported here were computed using double-precision arithmetic.

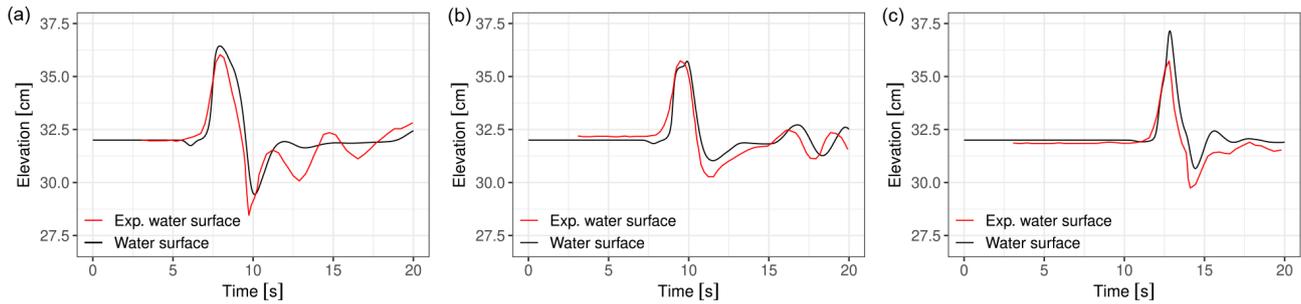


Figure 11. Simulated and experimental results of unsteady flow over an island for gauges G9 (a), G16 (b), and G22 (c).

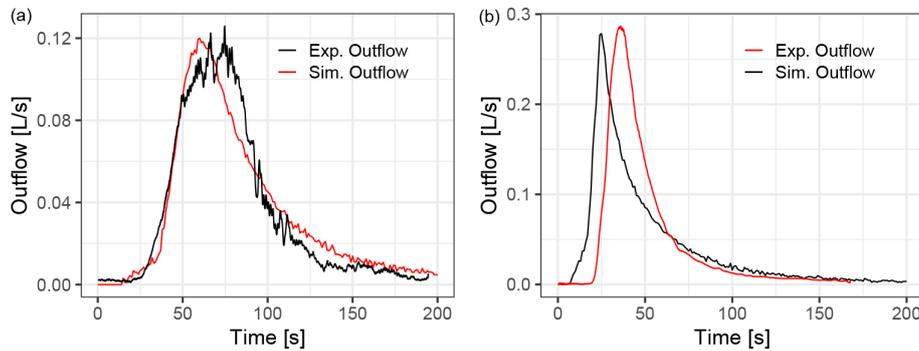


Figure 12. Simulated hydrographs compared to experimental data from Cea et al. (2010a) for two rainfall pulses on the A12 building arrangement. (a) Rain intensity 85 mm h^{-1} , duration 60 s. (b) Rain intensity 300 mm h^{-1} , duration 20 s

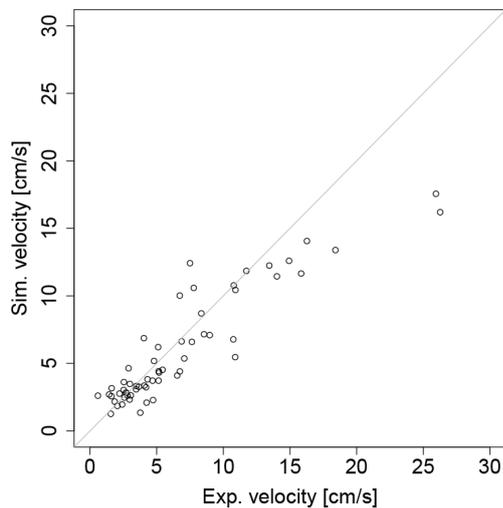


Figure 13. Comparison of simulated (line) and experimental (circles) steady velocities in the Thiès field case.

7.1 Single-node scaling – Malpasset dam break

The commonly used Malpasset dam break test (introduced in Sect. 6.2) was also tested for computational performance at a resolution of $\delta x = 10 \text{ m}$. Results are shown in Fig. 16. The case was computed on CPUs, a single JUWELS node, and a single JURECA-DC node. Three additional runs with

single Nvidia GPUs were carried out: a commercial-grade GeForce RTX 3070, 8 GB GPU (in a desktop computer), and two scientific-grade cards V100 and A100, respectively (in JUWELS). As Fig. 16 shows, CPU runtime quickly approaches an asymptotic behaviour (therefore demonstrating that additional nodes are not useful in this case). Notably, all three GPUs outperform a single CPU node, and the performance gradient among the GPUs is evident. The A100 GPU is roughly 6.5 faster than a full JUWELS CPU node, and even for the consumer-grade RTX 3070, the speed-up compared to a single HPC node is 2.2. Although it is possible to scale up this case with significantly higher resolution and test it with multiple GPUs, it is not a case well suited to such a scaling test. Multiple GPUs (as well as multiple nodes with either CPUs or GPUs) require a domain decomposition. The orientation of the Malpasset domain is roughly NW–SE, which makes both 1D decompositions (along x or y) and 2D decompositions (x and y) inefficient, as many regions have no computational load. Moreover, the dam break nature of the case implies that a large part of the valley is dry for long periods of time; therefore, load balancing among the different nodes and/or GPUs will be poor.

7.2 HPC scaling – 2D circular dam break case

This is a simple analytical verification test in the shallow-water literature, which generalises the 1D dam break solu-

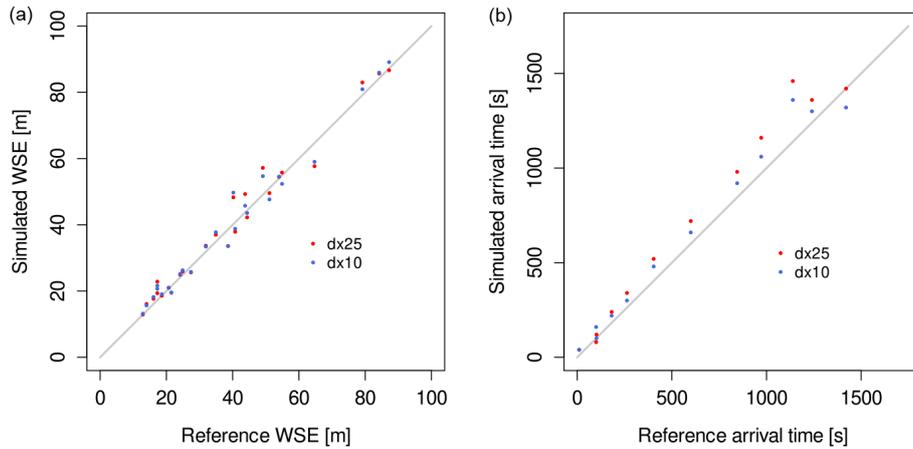


Figure 14. Water surface elevations (a) and arrival time (b) result comparison for two meshes of the Malpasset case.

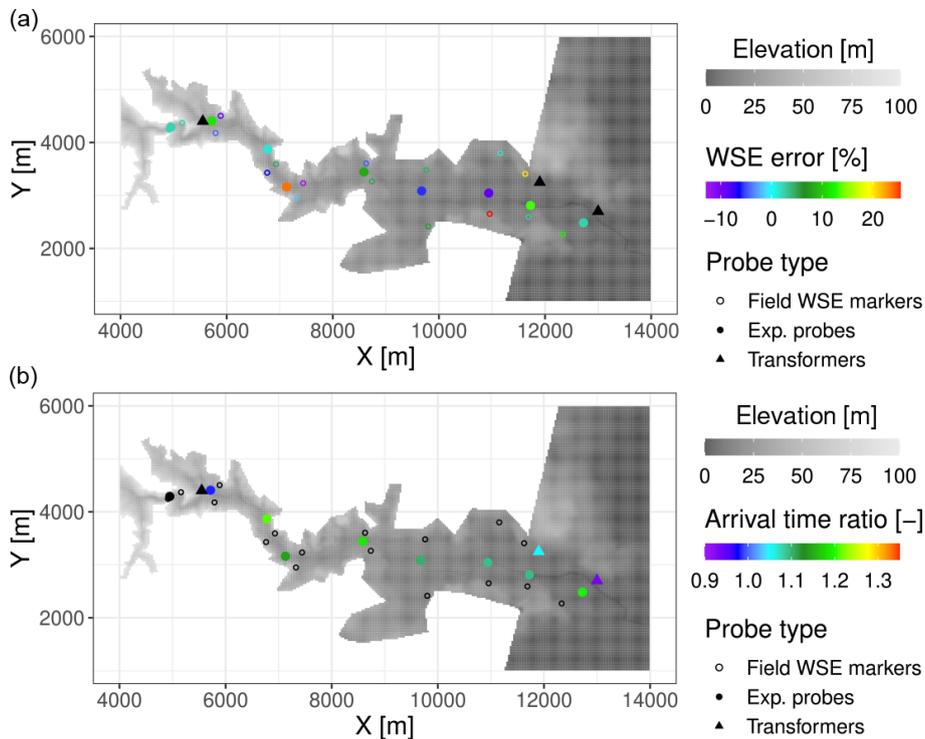


Figure 15. Geolocated relative WSE error (a) and ratio of arrival time (b) for the Malpasset dam break test case with $\delta x = 10$ m.

tion. We purposely select this case (instead of one of the many verification problems) for its convenience for scaling studies. Firstly, resolution can be increased at will. Additionally, the square domain allows for trivial domain decomposition, which together with the fully wet domain and the radially symmetric flow field minimises load-balancing issues. Essentially, it allows for a very clean scalability test with minimal interference from the problem topology, which facilitates scalability and performance analysis (in contrast to the limitations of the Malpasset domain discussed in Sect. 7.1). We take a $400 \text{ m} \times 400 \text{ m}$ flat domain with the cen-

tre at $(0, 0)$ and initial conditions given by

$$h(x, y, t = 0) = \begin{cases} 4 & \text{if } \sqrt{x^2 + y^2} \leq 50 \\ 1 & \text{otherwise.} \end{cases} \quad (10)$$

We generated three computational grids, with $\delta x = 0.05, 0.025, 0.0175$ m, which correspond to 64, 256, and 552 million cells, respectively. Figure 17 shows the strong-scaling results for the 64 and 256 million cells cases, computed in the JUWELS-booster system, on A100 Nvidia GPUs. The 64 million does not scale well beyond 4 GPUs. However, the 256-million-cells problem scales well up to 64 GPUs (and

Table 4. HPC systems in which SERGHEI-SWE has been tested.

Name	Centre	Institution	Country	Devices	Vendor	Device or node	Nodes
JUWELS	JSC	FZJ	Germany	Xeon Platinum 8168 CPU	Intel	$2 \times (2 \times 24)$	2567
				Volta V100 GPU	Nvidia	4	56
				Ampere A100 GPU	Nvidia	4	936
JURECA-DC	JSC	FZJ	Germany	EPYC 7742 2.25 GHz	AMD	$2 \times (2 \times 64)$	480
Summit	OLCF	ORNL	USA	Volta V100 GPU	Nvidia	6	4608
Cori	NERSC	LBNL	USA	Xeon E5-2698 v3 CPU	Intel	32	2388

JSC: Jülich Supercomputing Centre; FZJ: Forschungszentrum Jülich; OLCF: Oak Ridge Leadership Computing Facility; ORNL: Oak Ridge National Laboratory; NERSC: National Energy Research Scientific Computing Center; LBNL: Lawrence Berkeley National Laboratory

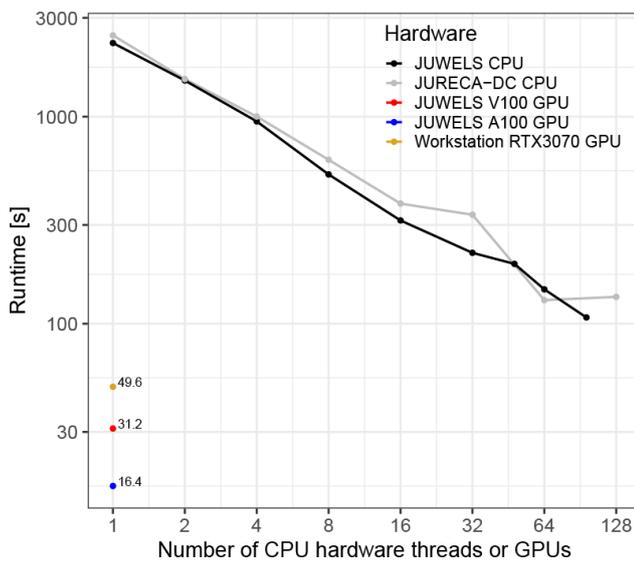


Figure 16. Scaling for the Malpasset case ($\delta x = 10$ m) on a single node and on single GPUs. GPU speed ups relative to a full JUWELS node are 6.5 (A100), 3.4 (V100), and 2.2 (RTX 3070).

efficiency starts to decrease with 128), showing that the first case simply is too small for significant gains.

For the 552-million-cells grid, only two runs were computed with 128 and 160 GPUs (corresponding to 32 and 40 nodes in JUWELS-booster, respectively). Runtime for these was 95.4 and 84.7 s, respectively, implying a very good 89 % scaling efficiency for this large number of GPUs. For this problem and these resources, the time required for inter-GPU communications is comparable to that used by kernels computing fluxes and updating cells, signalling scalability limits for this case on the current implementation.

7.3 HPC scaling of rainfall runoff in a large catchment

To demonstrate scaling under production conditions of real scenarios, we use an idealised rainfall runoff simulation over the Lower Triangle region in the East River Watershed (Colorado, USA) (Carroll et al., 2018; Hubbard et al., 2018; Özgen-Xian et al., 2020). The domain has an area

of 14.82 km² and elevations ranging from 2759–3787 m. The computational problem is defined with a resolution of $\delta x = 0.5$ m (matching the highest-resolution DEM available), resulting in 122×10^6 computational cells. Although this is not a particularly large catchment, the very-high-resolution DEM available makes it an interesting performance benchmark, which is our sole interest in this paper.

For practical purposes, two configurations have been used for this test: a short rainfall of $T = 870$ s, which was computed in Cori and JUWELS to assess CPU performance and scalability (results shown in Fig. 18), and a long rainfall event lasting $T = 12\,000$ s, which was simulated in Summit and JUWELS to assess GPU performance and scalability, with results shown in Fig. 19. CPU results (Fig. 18) show that the strong scaling behaviour in Cori and JUWELS is very similar. Absolute runtimes are longer for Cori, since the scaling study was carried out starting from a single core, whereas in JUWELS it was with a full node (i.e. 48 cores). Most importantly, the GPU strong-scaling behaviour overlaps almost completely between JUWELS and Summit, although computations in Summit were somewhat faster. CPU and GPU scaling are clearly highly efficient, with similar behaviour. These results demonstrate the performance portability delivered via Kokkos to SERGHEI.

8 Conclusions and outlook

In this paper we present the SERGHEI framework and, in particular, the SERGHEI-SWE module. SERGHEI-SWE implements a 2D fully dynamic shallow-water solver, harnessing state-of-the-art numerics and leveraging on Kokkos to facilitate portability across architectures. We show through empirical evidence with a large set of well-established benchmarks that SERGHEI-SWE is accurate, numerically stable, and robust. Importantly, we show that SERGHEI-SWE's parallel scaling is very good for CPU-based HPC systems, consumer-grade GPUs, and GPU-based HPC systems. Consequently, we claim that SERGHEI is indeed performance portable and approaching exascale readiness. These features make SERGHEI-SWE a plausible community code for

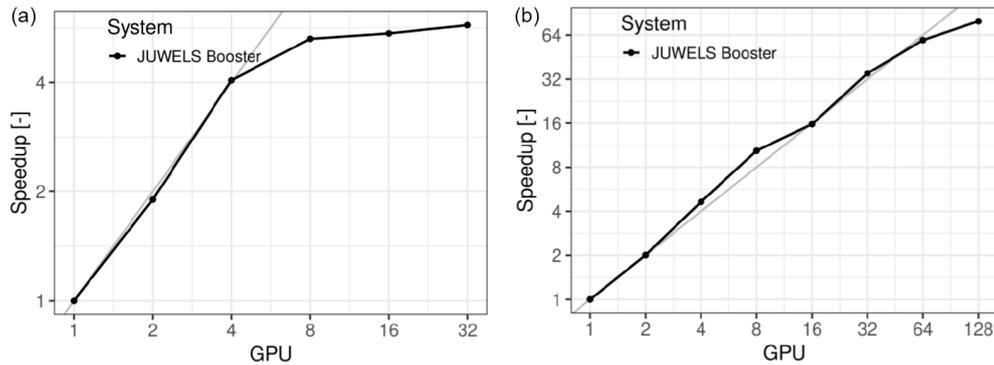


Figure 17. Strong scaling behaviour for a circular dam break test case with two resolutions. (a) $\delta x = 0.05$, 64 million cells. (b) $\delta x = 0.025$, 256 million cells.

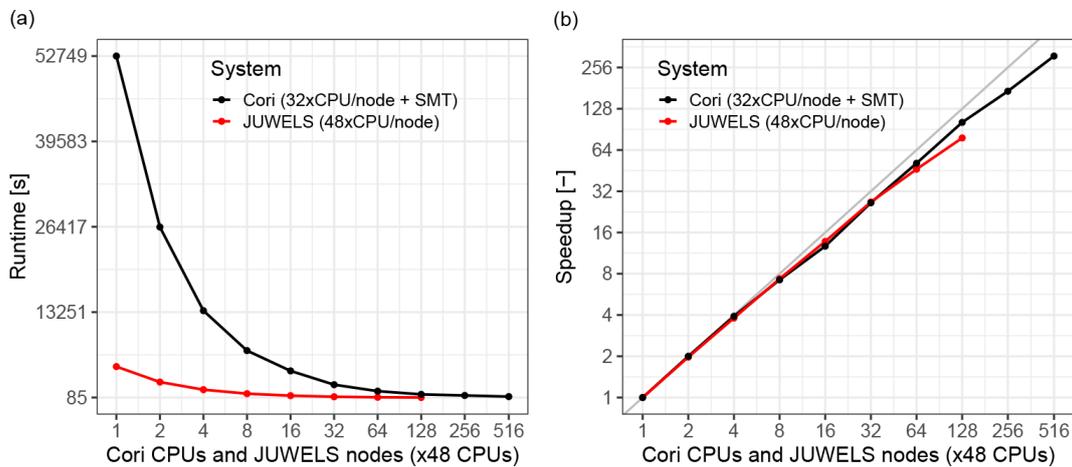


Figure 18. Runtime (a) and speed up (b) for a strong-scaling experiment with SERGHEI-SWE using CPUs on Cori and JUWELS for the short rainfall event. See Table 4 for details on the systems.

shallow-water modelling for a wide range of applications requiring large-scale, very-high-resolution simulations.

Exploiting increasingly better and highly resolved geospatial information (DEMs, land use, vector data of structures) prompts the need for high-resolution solvers. At the same time, the push towards the study of multiscale systems and integrated management warrants increasingly larger domains. Together, these trends result in larger computational problems, motivating the need for exascale-ready shallow-water solvers. Additionally, HPC technology is evermore available, not only via (inter)national research facilities but also through cloud-computing facilities. It is arguably timely to enable such an HPC-ready solver.

The HPC allows for not only large simulations but also large ensembles of simulations, allowing uncertainty issues to be addressed and enabling scenario analysis for engineering problems, parameter space exploration, and hypothesis testing. Furthermore, although the benefits of high resolution may be marginal for runoff hydrograph estimations, they allow the local dynamics to be better resolved in the domain. Flow paths, transit times, wetting–drying dynamics,

and connectivity play important roles in transport and eco-hydrological processes. For these purposes, enabling very-high-resolution simulations will prove to be highly beneficial. We also envision that, provided with sufficient computational resources, SERGHEI-SWE could be used for operational flood forecasting and probabilistic flash-flood modelling. Altogether, this strongly paves the way for the uptake of shallow-water solvers by the broader ESM community and its coupling to Earth system models, as well as their many applications, from process and system understanding to hydrometeorological risk and impact assessment. We also envision that, for users not requiring HPC capabilities, the benefit of SERGHEI-SWE is access to a transparent, open-sourced, performance-portable software that allows workstation GPUs to be exploited efficiently.

As additional SERGHEI modules become operational, the HPC capabilities will further enable simulations that are unfeasible with the current generation of available solvers. For example, with a fully operational transport and morphology module, it will be possible to run decade-long morphological simulations relevant for river management applica-

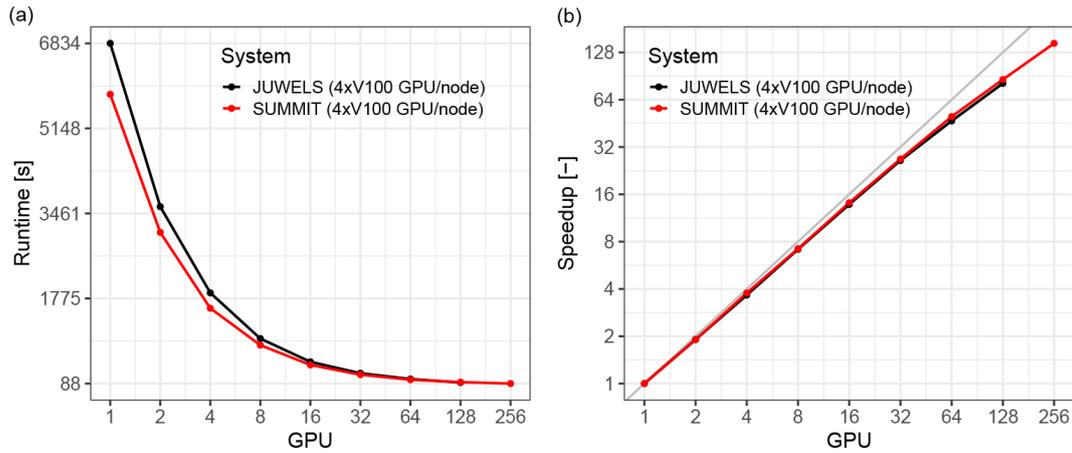


Figure 19. Runtime (a) and speed up (b) for a strong-scaling experiment with SERGHEI-SWE using GPUs on Summit and JUWELS for the long rainfall event. See Table 4 for details on the systems.

tions; to better capture sediment connectivity and sediment cascades across the landscape, a relevant topic for erosion and catchment management; or to perform catchment-scale hydro-biogeochemical simulations with unprecedented high spatial resolutions for better understanding of ecohydrological and biogeochemical processes.

Finally, SERGHEI is conceptualised and designed with extendibility and software interoperability in mind, with design choices made to facilitate foreseeable future developments on a wide range of topics, such as

1. numerics, e.g. the discontinuous Galerkin discretisation strategies (Caviedes-Voullième and Kesserwani, 2015; Shaw et al., 2021) and multiresolution adaptive meshing (Caviedes-Voullième et al., 2020b; Kesserwani and Sharifian, 2020, 2022; Özgen-Xian et al., 2020);
2. interfaces to mature geochemistry engines, e.g. CrunchFlow (Steeffel, 2009) and PFLOTRAN (Lichtner et al., 2015);
3. vegetation models with varying degrees of complexity, for example, Ecosys (e.g. Grant et al., 2007; Grant and the Ecosys development team, 2022) and Ech2O (Maneta and Silverman, 2013).

Appendix A: Additional validation test cases

This appendix contains an extended set of relevant test cases that are commonly used as validation cases in the literature. It complements and extends the verification evidence in Sect. 4.

A1 C property: immersed bump

Using the same set-up as in Sect. 4.1.1, but with a higher water surface elevation, in Fig. A1 we demonstrate how SERGHEI-SWE also conserves the C property for an immersed bump.

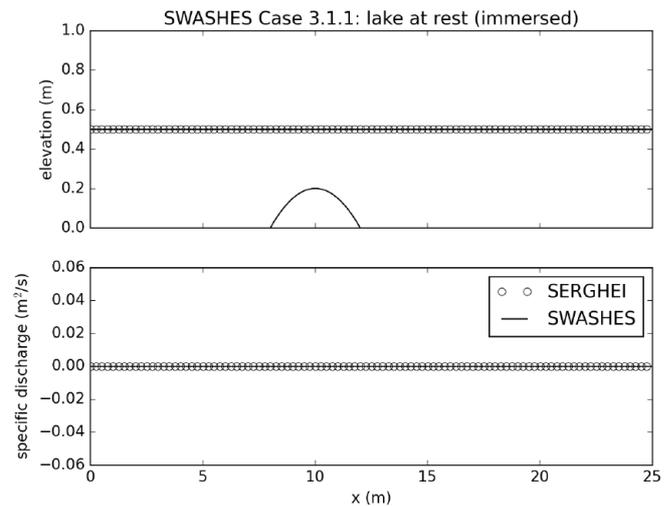


Figure A1. Lake-at-rest solution for an immersed bump.

A2 Well-balancing

To further show that SERGHEI-SWE is well-balanced, we computed three steady flows over a bump. We include a transcritical flow with a shock wave, a fully subcritical flow, and a transcritical flow, as shown in Fig. A2. All of SERGHEI-SWE predictions show excellent agreement with the analytical solution. The constant unit discharge is captured with machine accuracy without oscillations at the shock, which is an inherent feature of the augmented Roe solver (Murillo and García-Navarro, 2010).

We also include two additional cases from MacDonald et al. (1995) for fully supercritical and subcritical flows in Fig. A3. These results and their L norms in Table A1 further confirm well-balancing.

Additionally, MacDonald-type solutions can be constructed for frictionless flumes to study the bed slope source

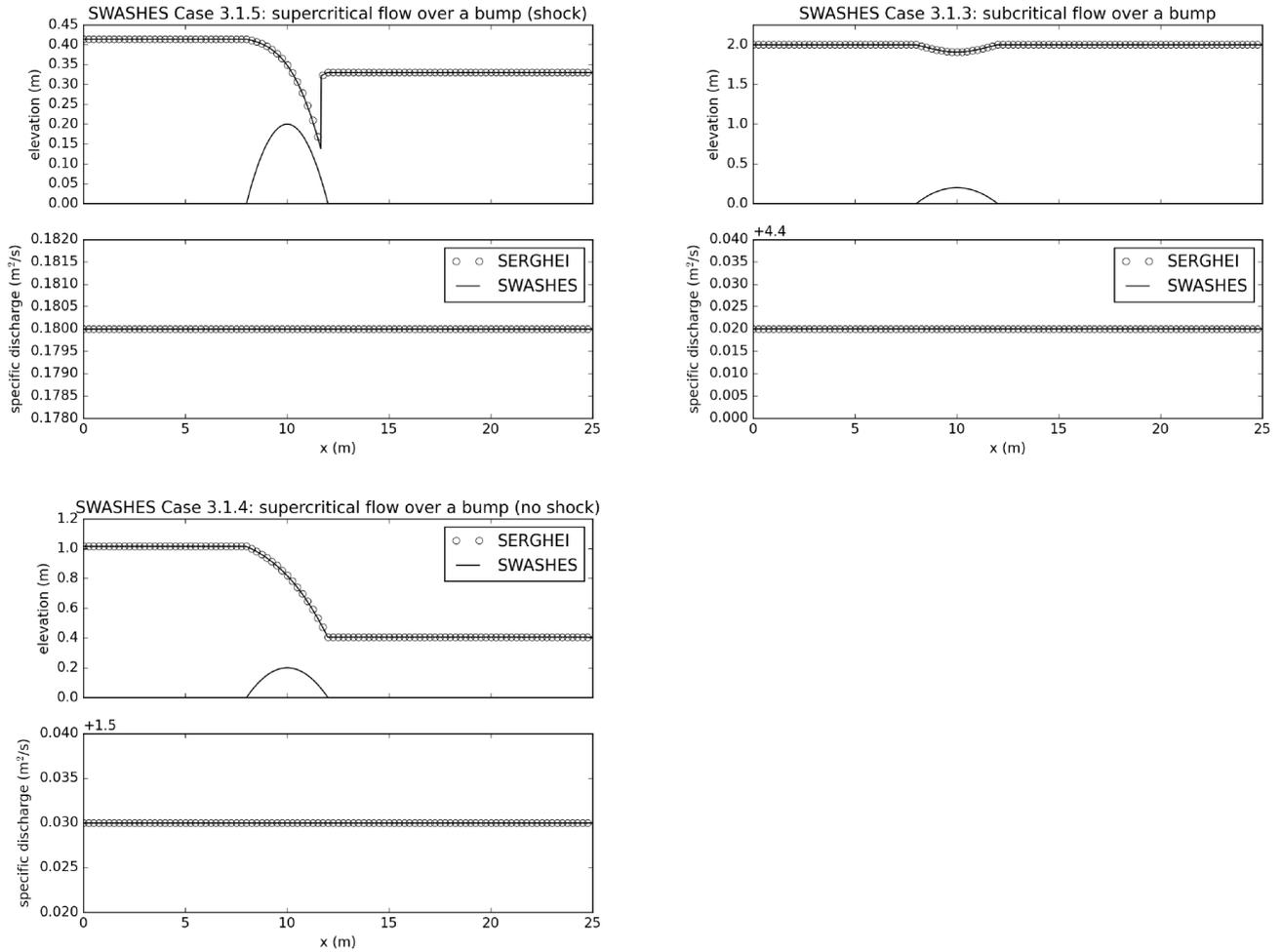


Figure A2. Analytical steady flows over a bump. SERGHEI-SWE captures moving equilibria solutions for transcritical flow with a shock (top left), fully subcritical flow (top right), and transcritical flow without a shock (bottom)

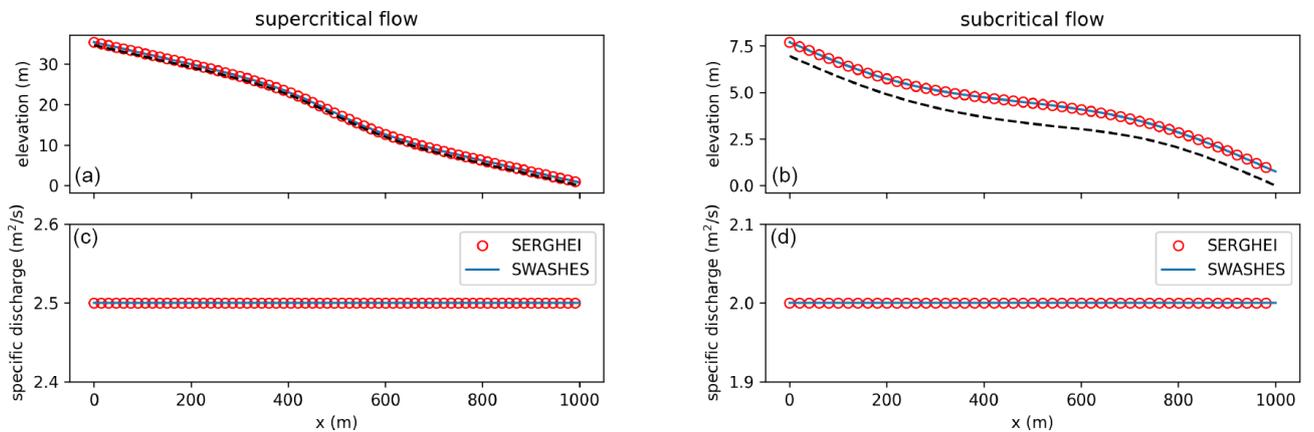


Figure A3. Analytical steady flows: flumes. SERGHEI-SWE captures moving equilibria solutions for a subcritical (a, c) and supercritical (b, d) flow. Note that the solution is stable (no oscillations) and well-balanced (discharge remains constant along the flume).

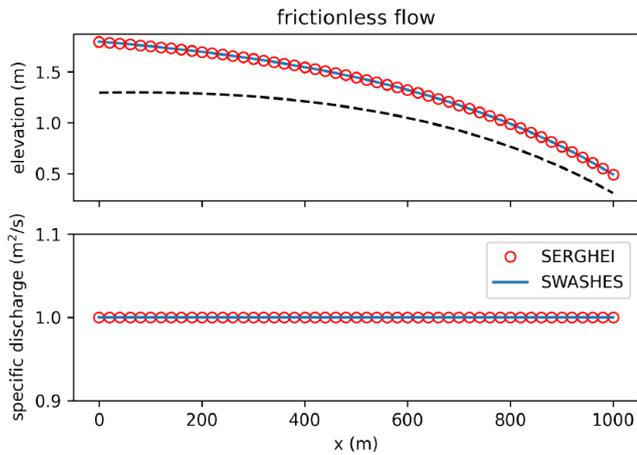


Figure A4. Analytical steady flows: flumes. SERGHEI-SWE captures moving equilibrium solution for frictionless test case, with a stable and well-balanced solution.

term implementation in isolation. We present a frictionless test case with SERGHEI-SWE that is not part of the SWASHES benchmark compilation. We discretise the bed elevation of the flume as

$$z(x) = C_0 - \frac{1}{2} \exp(-0.001x) - \frac{2q_0^2 \exp(0.002x)}{g}, \quad (\text{A1})$$

where C_0 is an arbitrary integration constant and q_0 is a specified unit discharge. The water depth for this topography is

$$h(x) = \frac{1}{2} \exp(-0.001x). \quad (\text{A2})$$

Using $C_0 = 1.0$ m and $q_0 = 1.0$ m² s⁻¹, we obtain the solution plotted in Fig. A4. SERGHEI-SWE’s prediction and the analytical solution show good agreement.

L norms for errors in water depth are summarised in Table A1 for the sake of completeness. L norms of a vector \mathbf{x} with length N and entries x_i , where $i \in [0, N) \subset \mathbb{Z}^+$ is the index of the entries, are calculated as

$$L_{\langle n \rangle} = \left(\sum_i^N |x_i|^{\langle n \rangle} \right)^{\frac{1}{\langle n \rangle}}, \quad (\text{A3})$$

with $\langle n \rangle \in \mathbb{Z}^+$ being the order of the L norm. The L_∞ norm is calculated as

$$L_\infty = \max |x_i|, \quad x_i \in \mathbf{x}. \quad (\text{A4})$$

The L norms for errors in unit discharge are in the range of machine accuracy for all cases and are omitted here.

A3 Dam break over a wet bed without friction

The dam break on a wet-bed-without-friction test case is configured by setting water depths in the domain as

Table A1. Analytical steady flows: summary of L norms for errors in water depth; L norms for errors in unit discharge are in the range of machine accuracy and are omitted here.

Case	L_1 (m)	L_2 (m)	L_∞ (m)
Fig. A1	0.0	0.0	0.0
Fig. A2 (top left)	0.371	0.07285	0.06984
Fig. A2 (top right)	0.293	0.02618	0.00332
Fig. A2 (bottom)	0.693	0.0306	0.00356
Fig. A3 (left)	5.21459	0.12162	0.00435
Fig. A3 (right)	1.0389	0.03805	0.00191
Fig. A4	0.74571	0.02743	0.00178

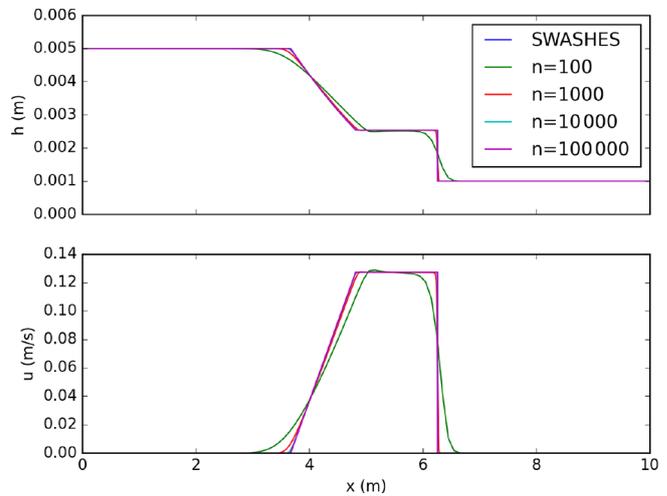


Figure A5. Dam break on wet bed without friction: model predictions for different number of grid cells. SERGHEI-SWE converges to the analytical solution (Stoker’s solution) as the grid is refined.

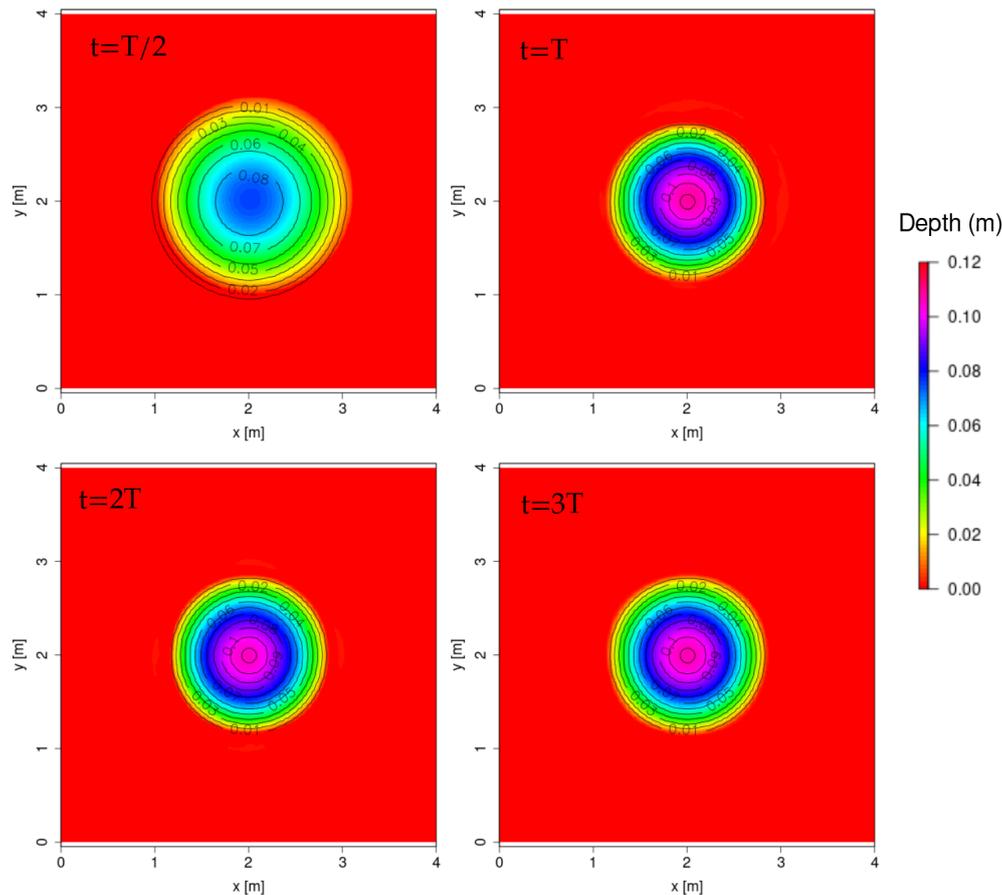
$h_L = 0.005$ m and $h_R = 0.001$ m. The domain is 10 m long, and the discontinuity is located at $x_0 = 5$ m. The total run time is 6 s. Figure A5 shows the model results obtained on successively refined grids compared against the analytical solution by Stoker (1957). Errors for this test case are reported in Table A2. We also report the observed convergence rate qR , calculated on the basis of the L_1 norm. As the grid is refined, the model result converges to the analytical solution. Due to the discontinuities in the solution, the observed convergence rate is below the theoretical convergence rate of $R = 1$.

A4 Radially symmetrical paraboloid

Using the same computational domain and bed topography as the case in Sect. 4.3, results for the radially symmetrical oscillation in a frictionless paraboloid (Thacker, 1981) are presented here. The details about the initial condition and the analytical solution for the water depth and velocities can be found in Delestre et al. (2013). In particular, the analytical solution at $t = 0$ s is set as the initial condition, and three pe-

Table A2. Analytical dam break: L norms and empirical convergence rates (R) for water depth (h) and velocity (u).

n	$L_1(h)$ (m)	$L_2(h)$ (m)	$R(h)$ (m)	$L_1(u)$ ((ms^{-1}))	$L_2(u)$ ((ms^{-1}))	$R(u)$ ((ms^{-1}))
100	0.01623	0.03303	–	0.11194	0.14115	–
1000	0.00265	0.00932	0.79	0.01842	0.0424	0.78
10 000	0.00041	0.00327	0.81	0.00272	0.01458	0.83
100 000	6×10^{-5}	0.00125	0.83	0.00037	0.00581	0.87

**Figure A6.** Radially symmetrical paraboloid: snapshots of water depth by the model compared to the analytical solution (contour lines). Period $T = 2.242851$ s.

riods are simulated using $\delta x = 0.01$ m as the grid resolution. Figure A6 shows the numerical and analytical solution at four different times. Although the analytical solution is periodic without dumping, the numerical results show a diffusive behaviour attributed to the numerical diffusion introduced by the first-order scheme. Other than that, model results show good agreement with the analytical solution.

A5 Experimental laboratory-scale tsunami

A 1 : 400-scale experiment of a tsunami run-up over the Monai valley (Japan) was reported by Matsuyama and Tanaka (2001); The third international workshop on long-wave run-up models (2004), providing experimental data on

the temporal evolution of the water surface at three locations and of the maximum run-up. A laboratory basin of $2.05 \text{ m} \times 3.4 \text{ m}$ was used to create a physical scale model of the Monai coastline. A tsunami was simulated by appropriate forcing of the boundary conditions. This experiment has been extensively used to benchmark SWE solvers (Arpaia and Ricchiuto, 2018; Caviedes-Voullième et al., 2020b; Hou et al., 2015, 2018; Kesserwani and Liang, 2012; Kesserwani and Sharifian, 2020; Morales-Hernández et al., 2014; Murillo et al., 2009; Murillo and García-Navarro, 2012; Nikolos and Delis, 2009; Serrano-Pacheco et al., 2009; Vater et al., 2019). The domain was discretised with a resolution of 1.4 cm, producing 95 892 elements. Simulated water surface elevations

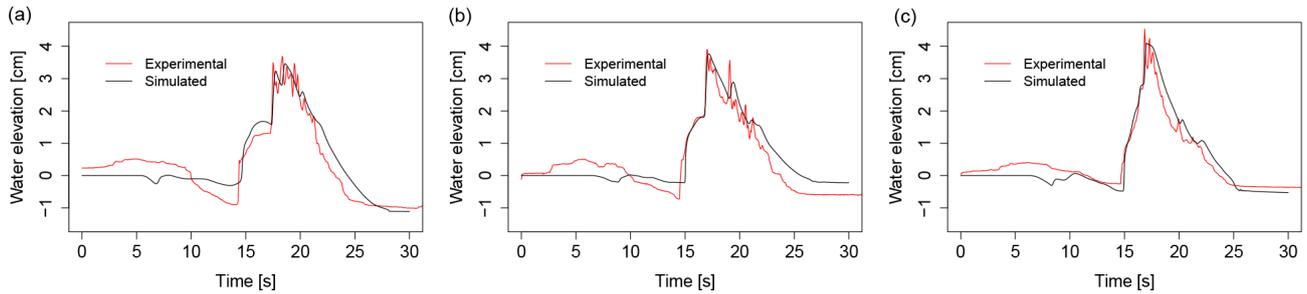


Figure A7. Simulated and experimental results for the laboratory-scale tsunami case at gauges G1 (a), G2 (b), and G3 (ca).

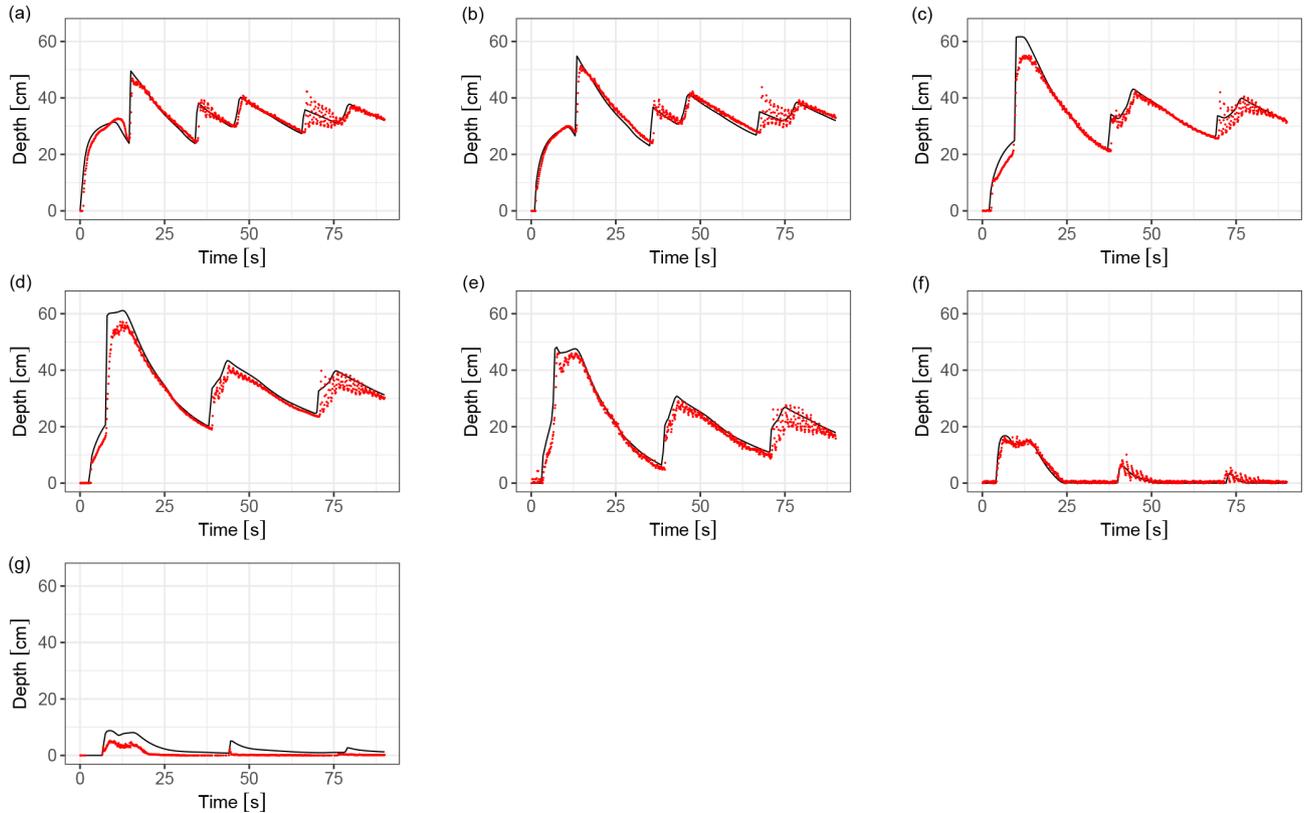


Figure A8. Simulated (black lines) and experimental (red points) transient water depths at seven gauge points ($x = 17.5$, $x = 19.5$, $x = 23.5$, $x = 25.5$, $x = 26.5$, $x = 28.5$, $x = 35.5$, from a to b) for the dam break over a triangular sill.

are shown together with the experimental measurements in Fig. A7 at three gauge locations. The results agree well with experimental measurements, both in the water surface elevations and the arrival times of the waves.

A6 Experimental dam break over a triangular sill

Hiver (2000) presented a large flume experiment of a dam break over a triangular sill, which is a standard benchmark in dam break problems (Caviedes-Voullième and Kesserwani, 2015; Bruwier et al., 2016; Kesserwani and Liang, 2010; Loukili and Soulaïmani, 2007; Murillo and García-Navarro, 2012; Yu and Duan, 2017; Zhou et al., 2013), together with

the reduced-scale version (Soares-Frazão, 2007; Hou et al., 2013a, b; Yu and Duan, 2017).

The computational domain was discretised with a 380×5 grid, with a $\delta x = 0.1$ m resolution. Figure A8 shows simulated and experimental results for the triangular sill case. A very good agreement can be observed, both in terms of peak depths occurring whenever the shock wave passes through a gauge and in the timing of the shock wave movement. The simulations tend to slightly overestimate the peaks of the shock wave, as well as to overestimate the waves downstream of the sill (see plot for gauge at $x = 35.5$ m). Both behaviours are well documented in the literature.

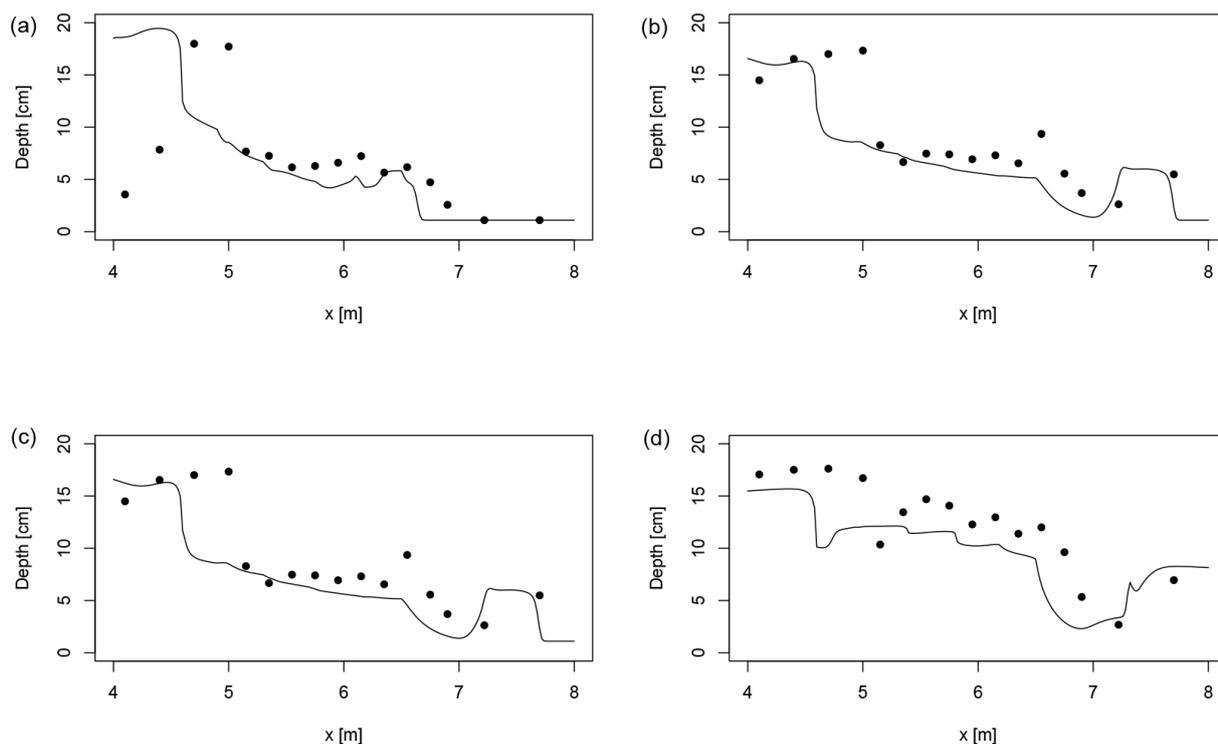


Figure A9. Simulated (lines) and experimental (points) water depth profiles at $y = 0.2$ m, at four times (4, 5, 6 and 10 s, **a** to **d**) for the idealised urban dam break case.

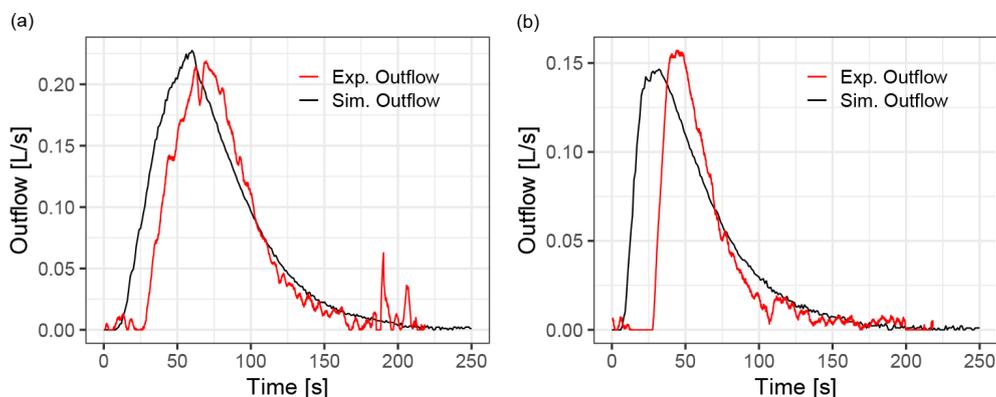


Figure A10. Simulated hydrographs compared to experimental data from Cea et al. (2010b) for two rainfall pulses on the L180 building arrangement. (a) Rainfall intensity 180 mm h^{-1} , duration 60 s. (b) Rainfall intensity 300 mm h^{-1} , duration 20 s.

A7 Experimental idealised urban dam break

A laboratory-scale experiment of a dam break over an idealised urban area was reported by Soares-Frazão and Zech (2008) in a concrete channel including 25 obstacles representing buildings separated by 10 cm. It is widely used in the shallow-water community (Abderrezzak et al., 2008; Caviedes-Voullième et al., 2020b; Ginting, 2019; Hartanto et al., 2011; Jeong et al., 2012; Özgen et al., 2015a; Petaccia et al., 2010; Wang et al., 2017) because of its fundamental phenomenological interest and because it is demand-

ing in terms of numerical stability and model performance. The small buildings and streets in the geometry require sufficiently high resolutions, both to capture the geometry and to capture the complex flow phenomena which are triggered in the streets. Experimental measurements of transient water depth exist at different locations, including in between the buildings. A resolution of 2 cm was used for the simulated results in Fig. A9, together with experimental data. The results agree well with the experimental observations to a similar degree as to what has been reported in the literature.

A8 Experimental rainfall runoff over a dense idealised urban area

Cea et al. (2010b) presented a laboratory-scale experiment in a flume with a dense idealised urban area. The case elaborates on the set-up of Cea et al. (2010a) (Sect. 5.3), including 180 buildings (case L180) in contrast to the 12 buildings in Sect. 5.3, which potentially requires a higher resolution to resolve the building (6.2 cm sides) and street width (~ 2 cm) and the flow in the streets. We keep a 1 cm resolution. Rainfall is a single pulse of constant intensity. Two set-ups were used with intensities 180 and 300 mm h^{-1} and durations of 60 and 20 s, respectively. As Fig. A10 shows, the hydrographs are well captured by the simulation, albeit with a delay. Analogously to Sect. 5.3, this can be attributed to surface tension in the early wetting phase.

Appendix B: Glossary

CFL	Courant–Friedrichs–Lewy
Cori	Cori supercomputer at the National Energy Research Scientific Computing Center (USA)
CPU	Central processing unit
CUDA	Compute Unified Device Architecture, programming interface for Nvidia GPUs
El Capitan	El Capitan supercomputer at the Lawrence Livermore National Laboratory (USA)
ESM	Earth system modelling
Frontier	Frontier supercomputer at the Oak Ridge Leadership Computing Facility (USA)
GPU	Graphics processing unit
HIP	Heterogeneous Interface for Portability, programming interface for AMD GPUs
HPC	High-performance computing
JURECA-DC	Data Centric module of the Jülich Research on Exascale Cluster Architectures supercomputer at the Jülich Supercomputing Centre (Germany)
JUWELS	Jülich Wizard for European Leadership Science, supercomputer at the Jülich Supercomputing Centre (Germany)
JUWELS-booster	Booster module of the JUWELS supercomputer (Germany)
Kokkos	Kokkos, a C++ performance portability layer
LUMI	LUMI supercomputer at CSC (Finland)
OpenMP	Open MultiProcessing, shared-memory programming interface for parallel computing
MPI	Message Passing Interface for parallel computing
SERGHEI	Simulation EnviRonment for Geomorphology, Hydrodynamics, and Ecohydrology in Integrated form
SERGHEI-SWE	SERGHEI's shallow-water equations solver
Summit	Summit supercomputer at the Oak Ridge Leadership Computing Facility (USA)
SWE	Shallow-water equations
SYCL	A programming model for hardware accelerators
UVM	Unified Virtual Memory
WSE	Water surface elevation

Code and data availability. SERGHEI is available through GitLab, at <https://gitlab.com/serghei-model/serghei> (last access: 6 February 2023), under a 3-clause BSD license. SERGHEI v1.0 was tagged as the first release at the time of submission of this paper. A static version of SERGHEI v1.0 is archived in Zenodo, DOI: <https://doi.org/10.5281/zenodo.7041423> (Caviedes Voullième et al., 2022a).

A repository containing test cases is available at https://gitlab.com/serghei-model/serghei_testcases. This repository contains many of the cases reported here, except those for which we cannot publicly release data but which can be obtained from the original authors of the datasets. A static version of this datasets is archived in Zenodo, with DOI: <https://doi.org/10.5281/zenodo.7041392> (Caviedes Voullième et al., 2022b).

Additional convenient pre- and post-processing tools are also available at <https://gitlab.com/serghei-model/sergheir> (last access: 6 February 2023).

Author contributions. DCV contributed to conceptualisation, investigation, software development, model validation, visualisation, and writing. MMH contributed to conceptualisation, methodology design, software development, formal analysis, model validation, and writing. MRN contributed to software development. IÖX contributed to formal analysis, software development, model validation, visualisation, and writing.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgements. The authors gratefully acknowledge the Earth System Modelling Project (ESM) for supporting this work by providing computing time on the ESM partition of the JUWELS supercomputer at the Jülich Supercomputing Centre (JSC) through the compute time project Runoff Generation and Surface Hydrodynamics across Scales with the SERGHEI model (RUGSHAS), project no. 22686. This work used resources of the National Energy Research Scientific Computing Center (NERSC), a US Department of Energy, Office of Science, user facility operated under contract no. DE-AC02-05CH11231. This research was also supported by the US Air Force Numerical Weather Modelling programme and used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is a US Department of Energy (DOE) Office of Science User Facility.

Financial support. The article processing charges for this open-access publication were covered by the Forschungszentrum Jülich.

Review statement. This paper was edited by Charles Onyutha and reviewed by Reinhard Hinkelmann and Kenichiro Kobayashi.

References

- Abderrezzak, K. E. K., Paquier, A., and Mignot, E.: Modelling flash flood propagation in urban areas using a two-dimensional numerical model, *Nat. Hazards*, 50, 433–460, <https://doi.org/10.1007/s11069-008-9300-0>, 2008.
- Alexander, F., Almgren, A., Bell, J., Bhattacharjee, A., Chen, J., Colella, P., Daniel, D., DeSlippe, J., Diachin, L., Draeger, E., Dubey, A., Dunning, T., Evans, T., Foster, I., Francois, M., Germann, T., Gordon, M., Habib, S., Halappanavar, M., Hamilton, S., Hart, W., Huang, Z. H., Hungerford, A., Kasen, D., Kent, P. R. C., Kolev, T., Kothe, D. B., Kronfeld, A., Luo, Y., Mackenzie, P., McCallen, D., Messer, B., Mniszewski, S., Oehmen, C., Perazzo, A., Perez, D., Richards, D., Rider, W. J., Rieben, R., Roche, K., Siegel, A., Sprague, M., Steefel, C., Stevens, R., Syamlal, M., Taylor, M., Turner, J., Vay, J.-L., Voter, A. F., Windus, T. L., and Yelick, K.: Exascale applications: skin in the game, *Philos. T. R. Soc. A*, 378, 20190056, <https://doi.org/10.1098/rsta.2019.0056>, 2020.
- An, H., Yu, S., Lee, G., and Kim, Y.: Analysis of an open source quadtree grid shallow water flow solver for flood simulation, *Quatern. Int.*, 384, 118–128, <https://doi.org/10.1016/j.quaint.2015.01.032>, 2015.
- Arpaia, L. and Ricchiuto, M.: r-adaptation for Shallow Water flows: conservation, well balancedness, efficiency, *Comput. Fluids*, 160, 175–203, <https://doi.org/10.1016/j.compfluid.2017.10.026>, 2018.
- Artigues, V., Kormann, K., Rampp, M., and Reuter, K.: Evaluation of performance portability frameworks for the implementation of a particle-in-cell code, *Concurr. Comput.-Pract. E.*, 32, <https://doi.org/10.1002/cpe.5640>, 2019.
- Aureli, F., Maranzoni, A., Mignosa, P., and Ziveri, C.: A weighted surface-depth gradient method for the numerical integration of the 2D shallow water equations with topography, *Adv. Water Resour.*, 31, 962–974, <https://doi.org/10.1016/j.advwatres.2008.03.005>, 2008.
- Aureli, F., Prost, F., Vacondio, R., Dazzi, S., and Ferrari, A.: A GPU-Accelerated Shallow-Water Scheme for Surface Runoff Simulations, *Water*, 12, 637, <https://doi.org/10.3390/w12030637>, 2020.
- Aureli, F., Maranzoni, A., and Petaccia, G.: Review of Historical Dam-Break Events and Laboratory Tests on Real Topography for the Validation of Numerical Models, *Water*, 13, 1968, <https://doi.org/10.3390/w13141968>, 2021.
- Ayog, J. L., Kesserwani, G., Shaw, J., Sharifian, M. K., and Bau, D.: Second-order discontinuous Galerkin flood model: Comparison with industry-standard finite volume models, *J. Hydrol.*, 594, 125924, <https://doi.org/10.1016/j.jhydrol.2020.125924>, 2021.
- Bates, P. and Roo, A. D.: A simple raster-based model for flood inundation simulation, *J. Hydrol.*, 236, 54–77, [https://doi.org/10.1016/S0022-1694\(00\)00278-X](https://doi.org/10.1016/S0022-1694(00)00278-X), 2000.
- Bauer, P., Dueben, P. D., Hoefler, T., Quintino, T., Schulthess, T. C., and Wedi, N. P.: The digital revolution of Earth-system science, *Nature Computational Science*, 1, 104–113, <https://doi.org/10.1038/s43588-021-00023-0>, 2021.

- Beckingsale, D. A., Burmark, J., Hornung, R., Jones, H., Killian, W., Kunen, A. J., Pearce, O., Robinson, P., Ryu-jin, B. S., and Scogland, T. R.: RAJA: Portable Performance for Large-Scale Scientific Applications, in: 2019 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC), 71–81, <https://doi.org/10.1109/p3hpc49587.2019.00012>, 2019.
- Bellos, V. and Tsakiris, G.: A hybrid method for flood simulation in small catchments combining hydrodynamic and hydrological techniques, *J. Hydrol.*, 540, 331–339, <https://doi.org/10.1016/j.jhydrol.2016.06.040>, 2016.
- Berger, M. J., George, D. L., LeVeque, R. J., and Mandli, K. T.: The GeoClaw software for depth-averaged flows with adaptive refinement, *Adv. Water Resour.*, 34, 1195–1206, <https://doi.org/10.1016/j.advwatres.2011.02.016>, 2011.
- Bertagna, L., Deakin, M., Guba, O., Sunderland, D., Bradley, A. M., Tezaur, I. K., Taylor, M. A., and Salinger, A. G.: HOMMEXX 1.0: a performance-portable atmospheric dynamical core for the Energy Exascale Earth System Model, *Geosci. Model Dev.*, 12, 1423–1441, <https://doi.org/10.5194/gmd-12-1423-2019>, 2019.
- Bomers, A., Schielen, R. M. J., and Hulscher, S. J. M. H.: The influence of grid shape and grid size on hydraulic river modelling performance, *Environ. Fluid Mech.*, 19, 1273–1294, <https://doi.org/10.1007/s10652-019-09670-4>, 2019.
- Bout, B. and Jetten, V.: The validity of flow approximations when simulating catchment-integrated flash floods, *J. Hydrol.*, 556, 674–688, <https://doi.org/10.1016/j.jhydrol.2017.11.033>, 2018.
- Bradford, S. F. and Sanders, B. F.: Finite-Volume Model for Shallow-Water Flooding of Arbitrary Topography, *J. Hydraul. Eng.*, 128, 289–298, [https://doi.org/10.1061/\(asce\)0733-9429\(2002\)128:3\(289\)](https://doi.org/10.1061/(asce)0733-9429(2002)128:3(289)), 2002.
- Briggs, M. J., Synolakis, C. E., Harkins, G. S., and Green, D. R.: Laboratory experiments of tsunami runup on a circular island, *Pure Appl. Geophys.*, 144, 569–593, <https://doi.org/10.1007/bf00874384>, 1995.
- Brodtkorb, A. R., Sætra, M. L., and Altinakar, M.: Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation, *Comput. Fluids*, 55, 1–12, <https://doi.org/10.1016/j.compfluid.2011.10.012>, 2012.
- Brufau, P., García-Navarro, P., and Vázquez-Cendón, M. E.: Zero mass error using unsteady wetting-drying conditions in shallow flows over dry irregular topography, *Int. J. Numer. Meth. Fl.*, 45, 1047–1082, <https://doi.org/10.1002/fld.729>, 2004.
- Brunner, G.: HEC-RAS 2D User's Manual Version 6.0, Hydrologic Engineering Center, Davis, CA, USA, <https://www.hec.usace.army.mil/confluence/rasdocs/r2dum/latest> (last access: 22 August 2022), 2021.
- Brunner, P. and Simmons, C. T.: HydroGeoSphere: A Fully Integrated, Physically Based Hydrological Model, *Ground Water*, 50, 170–176, <https://doi.org/10.1111/j.1745-6584.2011.00882.x>, 2012.
- Bruwier, M., Archambeau, P., Erpicum, S., Piroton, M., and Dewals, B.: Discretization of the divergence formulation of the bed slope term in the shallow-water equations and consequences in terms of energy balance, *Appl. Math. Model.*, 40, 7532–7544, <https://doi.org/10.1016/j.apm.2016.01.041>, 2016.
- Burguete, J., García-Navarro, P., and Murillo, J.: Friction term discretization and limitation to preserve stability and conservation in the 1D shallow-water model: Application to unsteady irrigation and river flow, *Int. J. Numer. Meth. Fl.*, 58, 403–425, <https://doi.org/10.1002/fld.1727>, 2008.
- Buttinger-Kreuzhuber, A., Horváth, Z., Noelle, S., Blöschl, G., and Waser, J.: A fast second-order shallow water scheme on two-dimensional structured grids over abrupt topography, *Adv. Water Resour.*, 127, 89–108, <https://doi.org/10.1016/j.advwatres.2019.03.010>, 2019.
- Buttinger-Kreuzhuber, A., Konev, A., Horváth, Z., Cornel, D., Schwerdtf, I., Blöschl, G., and Waser, J.: An integrated GPU-accelerated modeling framework for high-resolution simulations of rural and urban flash floods, *Environ. Modell. Softw.*, 156, 105480, <https://doi.org/10.1016/j.envsoft.2022.105480>, 2022.
- Caldas Steinstraesser, J. G., Delenne, C., Finaud-Guyot, P., Guinot, V., Kahn Casapia, J. L., and Rousseau, A.: SW2D-LEMON: a new software for upscaled shallow water modeling, in: Simhydro 2021 – 6th International Conference Models for complex and global water issues – Practices and expectations, Sophia Antipolis, France, <https://hal.inria.fr/hal-03224050> (last access: 22 August 2022), 2021.
- Carlotto, T., Chaffe, P. L. B., dos Santos, C. I., and Lee, S.: SW2D-GPU: A two-dimensional shallow water model accelerated by GPGPU, *Environ. Modell. Softw.*, 145, 105205, <https://doi.org/10.1016/j.envsoft.2021.105205>, 2021.
- Carroll, R. W. H., Bearup, L. A., Brown, W., Dong, W., Bill, M., and Williams, K. H.: Factors controlling seasonal groundwater and solute flux from snow-dominated basins, *Hydrol. Process.*, 32, 2187–2202, <https://doi.org/10.1002/hyp.13151>, 2018.
- Caviedes-Voullième, D. and Kesserwani, G.: Benchmarking a multi-resolution discontinuous Galerkin shallow water model: Implications for computational hydraulics, *Adv. Water Resour.*, 86, 14–31, <https://doi.org/10.1016/j.advwatres.2015.09.016>, 2015.
- Caviedes-Voullième, D., García-Navarro, P., and Murillo, J.: Influence of mesh structure on 2D full shallow water equations and SCS Curve Number simulation of rainfall/runoff events, *J. Hydrol.*, 448–449, 39–59, <https://doi.org/10.1016/j.jhydrol.2012.04.006>, 2012.
- Caviedes-Voullième, D., Fernández-Pato, J., and Hinz, C.: Cellular Automata and Finite Volume solvers converge for 2D shallow flow modelling for hydrological modelling, *J. Hydrol.*, 563, 411–417, <https://doi.org/10.1016/j.jhydrol.2018.06.021>, 2018.
- Caviedes-Voullième, D., Fernández-Pato, J., and Hinz, C.: Performance assessment of 2D Zero-Inertia and Shallow Water models for simulating rainfall-runoff processes, *J. Hydrol.*, 584, 124663, <https://doi.org/10.1016/j.jhydrol.2020.124663>, 2020a.
- Caviedes-Voullième, D., Gerhard, N., Sikstel, A., and Müller, S.: Multiwavelet-based mesh adaptivity with Discontinuous Galerkin schemes: Exploring 2D shallow water problems, *Adv. Water Resour.*, 138, 103559, <https://doi.org/10.1016/j.advwatres.2020.103559>, 2020b.
- Caviedes Voullième, D., Morales-Hernández, M., and Özgen-Xian, I.: SERGHEI (1.0), Zenodo [code], <https://doi.org/10.5281/zenodo.7041423>, 2022.
- Caviedes Voullième, D., Morales-Hernández, M., and Özgen-Xian, I.: Test cases for SERGHEI v1.0, Zenodo [data set], <https://doi.org/10.5281/zenodo.7041392>, 2022b.
- Cea, L. and Bladé, E.: A simple and efficient unstructured finite volume scheme for solving the shallow water equations in overland flow applications, *Water Resour. Res.*, 51, 5464–5486, <https://doi.org/10.1002/2014WR016547>, 2015.

- Cea, L., Garrido, M., and Puertas, J.: Experimental validation of two-dimensional depth-averaged models for forecasting rainfall-runoff from precipitation data in urban areas, *J. Hydrol.*, 382, 88–102, <https://doi.org/10.1016/j.jhydrol.2009.12.020>, 2010a.
- Cea, L., Garrido, M., Puertas, J., Jácome, A., Río, H. D., and Suárez, J.: Overland flow computations in urban and industrial catchments from direct precipitation data using a two-dimensional shallow water model, *Water Sci. Technol.*, 62, 1998–2008, <https://doi.org/10.2166/wst.2010.746>, 2010b.
- Chang, T.-J., Chang, Y.-S., and Chang, K.-H.: Modeling rainfall-runoff processes using smoothed particle hydrodynamics with mass-varied particles, *J. Hydrol.*, 543, 749–758, <https://doi.org/10.1016/j.jhydrol.2016.10.045>, 2016.
- Choi, B. H., Kim, D. C., Pelinovsky, E., and Woo, S. B.: Three-dimensional simulation of tsunami run-up around conical island, *Coast. Eng.*, 54, 618–629, <https://doi.org/10.1016/j.coastaleng.2007.02.001>, 2007.
- Clark, M. P., Bierkens, M. F. P., Samaniego, L., Woods, R. A., Uijlenhoet, R., Bennett, K. E., Pauwels, V. R. N., Cai, X., Wood, A. W., and Peters-Lidard, C. D.: The evolution of process-based hydrologic models: historical challenges and the collective quest for physical realism, *Hydrol. Earth Syst. Sci.*, 21, 3427–3440, <https://doi.org/10.5194/hess-21-3427-2017>, 2017.
- Coon, E., Svyatsky, D., Jan, A., Kikinzon, E., Berndt, M., Atchley, A., Harp, D., Manzini, G., Shelef, E., Lipnikov, K., Garimella, R., Xu, C., Moulton, D., Karra, S., Painter, S., Jafarov, E., and Molins, S.: Advanced Terrestrial Simulator, Computer Software, USDOE Office of Science (SC), Biological and Environmental Research (BER) (SC-23), <https://doi.org/10.11578/DC.20190911.1>, 2019.
- Costabile, P. and Costanzo, C.: A 2D SWEs framework for efficient catchment-scale simulations: hydrodynamic scaling properties of river networks and implications for non-uniform grids generation, *J. Hydrol.*, 599, 126306, <https://doi.org/10.1016/j.jhydrol.2021.126306>, 2021.
- Costabile, P., Costanzo, C., Ferraro, D., and Barca, P.: Is HEC-RAS 2D accurate enough for storm-event hazard assessment? Lessons learnt from a benchmarking study based on rain-on-grid modelling, *J. Hydrol.*, 603, 126962, <https://doi.org/10.1016/j.jhydrol.2021.126962>, 2021.
- Crompton, O., Katul, G. G., and Thompson, S.: Resistance formulations in shallow overland flow along a hillslope covered with patchy vegetation, *Water Resour. Res.*, 56, e2020WR027194, <https://doi.org/10.1029/2020wr027194>, 2020.
- David, A. and Schmalz, B.: A Systematic Analysis of the Interaction between Rain-on-Grid-Simulations and Spatial Resolution in 2D Hydrodynamic Modeling, *Water*, 13, 2346, <https://doi.org/10.3390/w13172346>, 2021.
- Dazzi, S., Vacondio, R., Palù, A. D., and Mignosa, P.: A local time stepping algorithm for GPU-accelerated 2D shallow water models, *Adv. Water Resour.*, 111, 274–288, <https://doi.org/10.1016/j.advwatres.2017.11.023>, 2018.
- Delestre, O., Lucas, C., Ksinant, P., Darboux, F., Laguerre, C., Vo, T., James, F., and Cordier, S.: SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies, *Int. J. Numer. Meth. Fl.*, 72, 269–300, <https://doi.org/10.1002/fld.3741>, 2013.
- Delestre, O., Darboux, F., James, F., Lucas, C., Laguerre, C., and Cordier, S.: FullSWOF: Full Shallow-Water equations for Overland Flow, *Journal of Open Source Software*, 2, 448, <https://doi.org/10.21105/joss.00448>, 2017.
- Demeshko, I., Watkins, J., Tezaur, I. K., Guba, O., Spatz, W. F., Salinger, A. G., Pawlowski, R. P., and Heroux, M. A.: Toward performance portability of the Albany finite element analysis code using the Kokkos library, *Int. J. High Perform. C.*, 33, 332–352, <https://doi.org/10.1177/1094342017749957>, 2018.
- Djemame, K. and Carr, H.: Exascale Computing Deployment Challenges, in: *Economics of Grids, Clouds, Systems, and Services*, Springer International Publishing, https://doi.org/10.1007/978-3-030-63058-4_19, pp. 211–216, 2020.
- Dullo, T. T., Darkwah, G. K., Gangrade, S., Morales-Hernández, M., Sharif, M. B., Kalyanapu, A. J., Kao, S.-C., Ghafoor, S., and Ashfaq, M.: Assessing climate-change-induced flood risk in the Conasauga River watershed: an application of ensemble hydrodynamic inundation modeling, *Nat. Hazards Earth Syst. Sci.*, 21, 1739–1757, <https://doi.org/10.5194/nhess-21-1739-2021>, 2021a.
- Dullo, T. T., Gangrade, S., Morales-Hernández, M., Sharif, M. B., Kao, S.-C., Kalyanapu, A. J., Ghafoor, S., and Evans, K. J.: Simulation of Hurricane Harvey flood event through coupled hydrologic-hydraulic models: Challenges and next steps, *J. Flood Risk Manag.*, 14, <https://doi.org/10.1111/jfr3.12716>, 2021b.
- Duran, A., Liang, Q., and Marche, F.: On the well-balanced numerical discretization of shallow water equations on unstructured meshes, *J. Comput. Phys.*, 235, 565–586, <https://doi.org/10.1016/j.jcp.2012.10.033>, 2013.
- Echeverribar, I., Morales-Hernández, M., Brufau, P., and García-Navarro, P.: 2D numerical simulation of unsteady flows for large scale floods prediction in real time, *Adv. Water Resour.*, 134, 103444, <https://doi.org/10.1016/j.advwatres.2019.103444>, 2019.
- Echeverribar, I., Morales-Hernández, M., Brufau, P., and García-Navarro, P.: Analysis of the performance of a hybrid CPU/GPU 1D2D coupled model for real flood cases, *J. Hydroinform.*, 22, 1198–1216, <https://doi.org/10.2166/hydro.2020.032>, 2020.
- Edwards, H. C., Trott, C. R., and Sunderland, D.: Kokkos: Enabling manycore performance portability through polymorphic memory access patterns, *J. Parallel Distr. Com.*, 74, 3202–3216, <https://doi.org/10.1016/j.jpdc.2014.07.003>, Domain-Specific Languages and High-Level Frameworks for High-Performance Computing, 2014.
- Fan, Y., Clark, M., Lawrence, D. M., Swenson, S., Band, L. E., Brantley, S. L., Brooks, P. D., Dietrich, W. E., Flores, A., Grant, G., Kirchner, J. W., Mackay, D. S., McDonnell, J. J., Milly, P. C. D., Sullivan, P. L., Tague, C., Ajami, H., Chaney, N., Hartmann, A., Hazenberg, P., McNamara, J., Pelletier, J., Perket, J., Rouholahnejad-Freund, E., Wagener, T., Zeng, X., Beighley, E., Buzan, J., Huang, M., Livneh, B., Mohanty, B. P., Nijssen, B., Safeeq, M., Shen, C., van Verseveld, W., Volk, J., and Yamazaki, D.: Hillslope Hydrology in Global Change Research and Earth System Modeling, *Water Resour. Res.*, 55, 1737–1772, <https://doi.org/10.1029/2018wr023903>, 2019.
- Fatichi, S., Vivoni, E. R., Ogden, F. L., Ivanov, V. Y., Mirus, B., Gochis, D., Downer, C. W., Camporese, M., Davison, J. H., Ebel, B., Jones, N., Kim, J., Mascaro, G., Niswonger, R., Restrepo, P., Rigon, R., Shen, C., Sulis, M., and Tarboton, D.: An overview of current applications, challenges, and future trends in distributed process-based models in hydrology, *J. Hydrol.*, 537, 45–60, <https://doi.org/10.1016/j.jhydrol.2016.03.026>, 2016.

- Fernández-Pato, J. and García-Navarro, P.: A 2D zero-inertia model for the solution of overland flow problems in flexible meshes, *J. Hydrol. Eng.*, 21, [https://doi.org/10.1061/\(asce\)he.1943-5584.0001428](https://doi.org/10.1061/(asce)he.1943-5584.0001428), 2016.
- Fernández-Pato, J., Caviedes-Voullième, D., and García-Navarro, P.: Rainfall/runoff simulation with 2D full shallow water equations: sensitivity analysis and calibration of infiltration parameters, *J. Hydrol.*, 536, 496–513, <https://doi.org/10.1016/j.jhydrol.2016.03.021>, 2016.
- Fernández-Pato, J., Martínez-Aranda, S., and García-Navarro, P.: A 2D finite volume simulation tool to enable the assessment of combined hydrological and morphodynamical processes in mountain catchments, *Adv. Water Resour.*, 141, 103617, <https://doi.org/10.1016/j.advwatres.2020.103617>, 2020.
- Gan, L., Fu, H., and Yang, G.: Translating novel HPC techniques into efficient geoscience solutions, *J. Comput. Sci.-Neth.*, 52, 101212, <https://doi.org/10.1016/j.jocs.2020.101212>, 2020.
- García-Alén, G., González-Cao, J., Fernández-Nóvoa, D., Gómez-Gesteira, M., Cea, L., and Puertas, J.: Analysis of two sources of variability of basin outflow hydrographs computed with the 2D shallow water model Iber: Digital Terrain Model and unstructured mesh size, *J. Hydrol.*, 612, 128182, <https://doi.org/10.1016/j.jhydrol.2022.128182>, 2022.
- García-Feal, O., González-Cao, J., Gómez-Gesteira, M., Cea, L., Domínguez, J., and Formella, A.: An Accelerated Tool for Flood Modelling Based on Iber, *Water*, 10, 1459, <https://doi.org/10.3390/w10101459>, 2018.
- García-Navarro, P., Murillo, J., Fernández-Pato, J., Echeverribar, I., and Morales-Hernández, M.: The shallow water equations and their application to realistic cases, *Environ. Fluid Mech.*, 19, 1235–1252, <https://doi.org/10.1007/s10652-018-09657-7>, 2019.
- George, D. L.: 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. Meth. Fl.*, 66, 1000–1018, <https://doi.org/10.1002/fld.2298>, 2010.
- Giardino, J. R. and Houser, C.: Introduction to the critical zone, in: *Developments in Earth Surface Processes*, vol. 19, chap. 1, edited by: J. R. Giardino, C. H., Elsevier B. V., Amsterdam, the Netherlands, <https://doi.org/10.1016/b978-0-444-63369-9.00001-x>, 2015.
- Ginting, B. M.: Central-upwind scheme for 2D turbulent shallow flows using high-resolution meshes with scalable wall functions, *Comput. Fluids*, 179, 394–421, <https://doi.org/10.1016/j.compfluid.2018.11.014>, 2019.
- Gottardi, G. and Venutelli, M.: An accurate time integration method for simplified overland flow models, *Adv. Water Resour.*, 31, 173–180, <https://doi.org/10.1016/j.advwatres.2007.08.004>, 2008.
- Govindaraju, R. S., Kavvas, M. L., and Jones, S. E.: Approximate Analytical Solutions for Overland Flows, *Water Resour. Res.*, 26, 2903–2912, <https://doi.org/10.1029/WR026i012p02903>, 1990.
- Grant, R. and the Ecosys development team: The Ecosys Modelling Project, <https://ecosys.ualberta.ca/>, last access: 22 August 2022.
- Grant, R. F., Barr, A. G., Black, T. A., Gaumont-Guay, D., Iwashita, H., Kidson, J., McCaughey, H., Morgenstern, K., Murayama, S., Nesic, Z., Saigusa, N., Shashkov, A., and Zha, T.: Net ecosystem productivity of boreal jack pine stands regenerating from clearcutting under current and future climates, *Glob. Change Biol.*, 13, 1423–1440, <https://doi.org/10.1111/j.1365-2486.2007.01363.x>, 2007.
- Grete, P., Glines, F. W., and O’Shea, B. W.: K-Athena: A Performance Portable Structured Grid Finite Volume Magnetohydrodynamics Code, *IEEE T. Parall. Distr.*, 32, 85–97, <https://doi.org/10.1109/tpds.2020.3010016>, 2021.
- Halver, R., Meinke, J. H., and Sutmann, G.: Kokkos implementation of an Ewald Coulomb solver and analysis of performance portability, *J. Parallel Distr. Com.*, 138, 48–54, <https://doi.org/10.1016/j.jpdc.2019.12.003>, 2020.
- Hartanto, I., Beevers, L., Popescu, I., and Wright, N.: Application of a coastal modelling code in fluvial environments, *Environ. Modell. Softw.*, 26, 1685–1695, <https://doi.org/10.1016/j.envsoft.2011.05.014>, 2011.
- Hervouet, J.-M. and Petitjean, A.: Malpasset dam-break revisited with two-dimensional computations, *J. Hydraul. Res.*, 37, 777–788, <https://doi.org/10.1080/00221689909498511>, 1999.
- Hiver, J.: Adverse-Slope and Slope (bump), in: *Concerted Action on Dam Break Modelling: Objectives, Project Report, Test Cases, Meeting Proceedings*, edited by: Soares-Frazão, S., Morris, M., and Zech, Y., vol. CD-ROM, Université Catholique de Louvain, Civil Engineering Department, Hydraulics Division, Louvain-la-Neuve, Belgium, 2000.
- Hou, J., Liang, Q., Simons, F., and Hinkelmann, R.: A stable 2D unstructured shallow flow model for simulations of wetting and drying over rough terrains, *Comput. Fluids*, 82, 132–147, <https://doi.org/10.1016/j.compfluid.2013.04.015>, 2013a.
- Hou, J., Simons, F., Mahgoub, M., and Hinkelmann, R.: A robust well-balanced model on unstructured grids for shallow water flows with wetting and drying over complex topography, *Comput. Method. Appl. M.*, 257, 126–149, <https://doi.org/10.1016/j.cma.2013.01.015>, 2013b.
- Hou, J., Liang, Q., Zhang, H., and Hinkelmann, R.: An efficient unstructured MUSCL scheme for solving the 2D shallow water equations, *Environ. Modell. Softw.*, 66, 131–152, <https://doi.org/10.1016/j.envsoft.2014.12.007>, 2015.
- Hou, J., Wang, R., Liang, Q., Li, Z., Huang, M. S., and Hinkelmann, R.: Efficient surface water flow simulation on static Cartesian grid with local refinement according to key topographic features, *Comput. Fluids*, 176, 117–134, <https://doi.org/10.1016/j.compfluid.2018.03.024>, 2018.
- Hou, J., Kang, Y., Hu, C., Tong, Y., Pan, B., and Xia, J.: A GPU-based numerical model coupling hydrodynamical and morphological processes, *Int. J. Sediment Res.*, 35, 386–394, <https://doi.org/10.1016/j.ijsrc.2020.02.005>, 2020.
- Hubbard, S. S., Williams, K. H., Agarwal, D., Banfield, J., Beller, H., Bouskill, N., Brodie, E., Carroll, R., Dafflon, B., Dwivedi, D., Falco, N., Faybishenko, B., Maxwell, R., Nico, P., Steefel, C., Steltzer, H., Tokunaga, T., Tran, P. A., Wainwright, H., and Varadharajan, C.: The East River, Colorado, Watershed: A Mountainous Community Testbed for Improving Predictive Understanding of Multiscale Hydrological-Biogeochemical Dynamics, *Vadose Zone J.*, 17, 180061, <https://doi.org/10.2136/vzj2018.03.0061>, 2018.
- Jain, M. K. and Kothiyari, U. C.: A GIS based distributed rainfall-runoff model, *J. Hydrol.*, 299, 107–135, 2004.
- Jeong, W., Yoon, J.-S., and Cho, Y.-S.: Numerical study on effects of building groups on dam-break flow

- in urban areas, *J. Hydro-Environ. Res.*, 6, 91–99, <https://doi.org/10.1016/j.jher.2012.01.001>, 2012.
- Jodhani, K. H., Patel, D., and Madhavan, N.: A review on analysis of flood modelling using different numerical models, *Mater. Today-Proc.*, <https://doi.org/10.1016/j.matpr.2021.07.405>, 2021.
- Kesserwani, G. and Liang, Q.: Well-balanced RKDG2 solutions to the shallow water equations over irregular domains with wetting and drying, *Comput. Fluids*, 39, 2040–2050, <https://doi.org/10.1016/j.compfluid.2010.07.008>, 2010.
- Kesserwani, G. and Liang, Q.: Dynamically adaptive grid based discontinuous Galerkin shallow water model, *Adv. Water Resour.*, 37, 23–39, <https://doi.org/10.1016/j.advwatres.2011.11.006>, 2012.
- Kesserwani, G. and Sharifian, M. K.: (Multi)wavelets increase both accuracy and efficiency of standard Godunov-type hydrodynamic models: Robust 2D approaches, *Adv. Water Resour.*, 144, 103693, <https://doi.org/10.1016/j.advwatres.2020.103693>, 2020.
- Kesserwani, G. and Sharifian, M. K.: (Multi)wavelet-based Godunov-type simulators of flood inundation: static versus dynamic adaptivity, *Adv. Water Resour.*, 171, 104357, <https://doi.org/10.1016/j.advwatres.2022.104357>, 2022.
- Kesserwani, G., Shaw, J., Sharifian, M. K., Bau, D., Keylock, C. J., Bates, P. D., and Ryan, J. K.: (Multi)wavelets increase both accuracy and efficiency of standard Godunov-type hydrodynamic models, *Adv. Water Resour.*, 129, 31–55, <https://doi.org/10.1016/j.advwatres.2019.04.019>, 2019.
- Kim, B., Sanders, B. F., Schubert, J. E., and Famiglietti, J. S.: Mesh type tradeoffs in 2D hydrodynamic modeling of flooding with a Godunov-based flow solver, *Adv. Water Resour.*, 68, 42–61, <https://doi.org/10.1016/j.advwatres.2014.02.013>, 2014.
- Kirstetter, G., Delestre, O., Lagrée, P.-Y., Popinet, S., and Josserand, C.: B-flood 1.0: an open-source Saint-Venant model for flash-flood simulation using adaptive refinement, *Geosci. Model Dev.*, 14, 7117–7132, <https://doi.org/10.5194/gmd-14-7117-2021>, 2021.
- Kobayashi, K., Kitamura, D., Ando, K., and Ohi, N.: Parallel computing for high-resolution/large-scale flood simulation using the K supercomputer, *Hydrological Research Letters*, 9, 61–68, <https://doi.org/10.3178/hrl.9.61>, 2015.
- Kollet, S., Sulis, M., Maxwell, R. M., Paniconi, C., Putti, M., Bertoldi, G., Coon, E. T., Cordano, E., Endrizzi, S., Kikinzon, E., Mouche, E., Mügler, C., Park, Y.-J., Refsgaard, J. C., Stisen, S., and Sudicky, E.: The integrated hydrologic model intercomparison project, IH-MIP2: A second set of benchmark results to diagnose integrated hydrology and feedbacks, *Water Resour. Res.*, 53, 867–890, <https://doi.org/10.1002/2016wr019191>, 2017.
- Kuffour, B. N. O., Engdahl, N. B., Woodward, C. S., Condon, L. E., Kollet, S., and Maxwell, R. M.: Simulating coupled surface–subsurface flows with ParFlow v3.5.0: capabilities, applications, and ongoing development of an open-source, massively parallel, integrated hydrologic model, *Geosci. Model Dev.*, 13, 1373–1397, <https://doi.org/10.5194/gmd-13-1373-2020>, 2020.
- Lacasta, A., Morales-Hernández, M., Murillo, J., and García-Navarro, P.: An optimized GPU implementation of a 2D free surface simulation model on unstructured meshes, *Adv. Eng. Softw.*, 78, 1–15, <https://doi.org/10.1016/j.advengsoft.2014.08.007>, 2014.
- Lacasta, A., Morales-Hernández, M., Murillo, J., and García-Navarro, P.: GPU implementation of the 2D shallow water equations for the simulation of rainfall/runoff events, *Environ. Earth Sci.*, 74, 7295–7305, <https://doi.org/10.1007/s12665-015-4215-z>, 2015.
- Lawrence, B. N., Rezný, M., Budich, R., Bauer, P., Behrens, J., Carter, M., Deconinck, W., Ford, R., Maynard, C., Mullerworth, S., Osuna, C., Porter, A., Serradell, K., Valcke, S., Wedi, N., and Wilson, S.: Crossing the chasm: how to develop weather and climate models for next generation computers?, *Geosci. Model Dev.*, 11, 1799–1821, <https://doi.org/10.5194/gmd-11-1799-2018>, 2018.
- Leiserson, C. E., Thompson, N. C., Emer, J. S., Kuszmaul, B. C., Lamson, B. W., Sanchez, D., and Schardl, T. B.: There’s plenty of room at the Top: What will drive computer performance after Moore’s law?, *Science*, 368, 6495, <https://doi.org/10.1126/science.aam9744>, 2020.
- Li, Z., Özgen-Xian, I., and Maina, F. Z.: A mass-conservative predictor-corrector solution to the 1D Richards equation with adaptive time control, *J. Hydrol.*, 592, 125809, <https://doi.org/10.1016/j.jhydrol.2020.125809>, 2021.
- Liang, D., Lin, B., and Falconer, R. A.: A boundary-fitted numerical model for flood routing with shock-capturing capability, *J. Hydrol.*, 332, 477–486, <https://doi.org/10.1016/j.jhydrol.2006.08.002>, 2007.
- Liang, Q., Hou, J., and Xia, X.: Contradiction between the C-property and mass conservation in adaptive grid based shallow flow models: cause and solution, *Int. J. Numer. Meth. Fl.*, 78, 17–36, <https://doi.org/10.1002/ffd.4005>, 2015.
- Liang, Q., Smith, L., and Xia, X.: New prospects for computational hydraulics by leveraging high-performance heterogeneous computing techniques, *J. Hydrodyn Ser. B*, 28, 977–985, [https://doi.org/10.1016/S1001-6058\(16\)60699-6](https://doi.org/10.1016/S1001-6058(16)60699-6), 2016.
- Lichtner, P. C., Hammond, G. E., Lu, C., Karra, S., Bisht, G., Andre, B., Mills, R., and Kumar, J.: PFLORAN user manual: A massively parallel reactive flow and transport model for describing surface and subsurface processes, Tech. rep., Los Alamos National Laboratory, New Mexico, USA, 2015.
- Liu, P. L. F., Cho, Y.-S., Briggs, M. J., Kanoglu, U., and Synolakis, C. E.: Runup of solitary waves on a circular island, *J. Fluid Mech.*, 302, 259–285, <https://doi.org/10.1017/s0022112095004095>, 1995.
- Loukili, Y. and Soulaïmani, A.: Numerical Tracking of Shallow Water Waves by the Unstructured Finite Volume WAF Approximation, *International Journal for Computational Methods in Engineering Science and Mechanics*, 8, 75–88, <https://doi.org/10.1080/15502280601149577>, 2007.
- Lynett, P. J., Wu, T.-R., and Liu, P. L.-F.: Modeling wave runup with depth-integrated equations, *Coast. Eng.*, 46, 89–107, [https://doi.org/10.1016/s0378-3839\(02\)00043-1](https://doi.org/10.1016/s0378-3839(02)00043-1), 2002.
- MacDonald, I., Baines, M., Nichols, N., and Samuels, P. G.: Comparison of some Steady State Saint-Venant Solvers for some Test Problems with Analytic Solutions, Tech. rep., University of Reading, 1995.
- Maneta, M. P. and Silverman, N. L.: A spatially distributed model to simulate water, energy, and vegetation dynamics using information from regional climate models, *Earth Interact.*, 17, 11.1–11.44, 2013.
- Mann, A.: Core Concept: Nascent exascale supercomputers offer promise, present challenges, *P. Natl. Acad. Sci. USA*, 117, 22623–22625, <https://doi.org/10.1073/pnas.2015968117>, 2020.

- Martínez-Aranda, S., Fernández-Pato, J., Caviedes-Voullième, D., García-Palacín, I., and García-Navarro, P.: Towards transient experimental water surfaces: A new benchmark dataset for 2D shallow water solvers, *Adv. Water Resour.*, 121, 130–149, <https://doi.org/10.1016/j.advwatres.2018.08.013>, 2018.
- Matsuyama, M. and Tanaka, H.: An experimental study of the highest run-up height in the 1993 Hokkaido Nansei-oki earthquake tsunami, *ITS Proceedings*, 879–889, 2001.
- Morales-Hernández, M., García-Navarro, P., and Murillo, J.: A large time step 1D upwind explicit scheme ($CFL > 1$): Application to shallow water equations, *J. Comput. Phys.*, 231, 6532–6557, <https://doi.org/10.1016/j.jcp.2012.06.017>, 2012.
- Morales-Hernández, M., Hubbard, M., and García-Navarro, P.: A 2D extension of a Large Time Step explicit scheme ($CFL > 1$) for unsteady problems with wet/dry boundaries, *J. Comput. Phys.*, 263, 303–327, <https://doi.org/10.1016/j.jcp.2014.01.019>, 2014.
- Morales-Hernández, M., Sharif, M. B., Gangrade, S., Dullo, T. T., Kao, S.-C., Kalyanapu, A., Ghafoor, S. K., Evans, K. J., Madadi-Kandjani, E., and Hodges, B. R.: High-performance computing in water resources hydrodynamics, *J. Hydroinform.*, <https://doi.org/10.2166/hydro.2020.163>, 2020.
- Morales-Hernández, M., Sharif, M. B., Kalyanapu, A., Ghafoor, S., Dullo, T., Gangrade, S., Kao, S.-C., Norman, M., and Evans, K.: TRITON: A Multi-GPU open source 2D hydrodynamic flood model, *Environ. Modell. Softw.*, 141, 105034, <https://doi.org/10.1016/j.envsoft.2021.105034>, 2021.
- Moulinec, C., Denis, C., Pham, C.-T., Rougé, D., Hervouet, J.-M., Razafindrakoto, E., Barber, R., Emerson, D., and Gu, X.-J.: TELEMAC: An efficient hydrodynamics suite for massively parallel architectures, *Comput. Fluids*, 51, 30–34, <https://doi.org/10.1016/j.compfluid.2011.07.003>, 2011.
- Mügler, C., Planchon, O., Patin, J., Weill, S., Silvera, N., Richard, P., and Mouche, E.: Comparison of roughness models to simulate overland flow and tracer transport experiments under simulated rainfall at plot scale, *J. Hydrol.*, 402, 25–40, <https://doi.org/10.1016/j.jhydrol.2011.02.032>, 2011.
- Murillo, J. and García-Navarro, P.: Weak solutions for partial differential equations with source terms: Application to the shallow water equations, *J. Comput. Phys.*, 229, 4327–4368, <https://doi.org/10.1016/j.jcp.2010.02.016>, 2010.
- Murillo, J. and García-Navarro, P.: Augmented versions of the HLL and HLLC Riemann solvers including source terms in one and two dimensions for shallow flow applications, *J. Comput. Phys.*, 231, 6861–6906, <https://doi.org/10.1016/j.jcp.2012.06.031>, 2012.
- Murillo, J., García-Navarro, P., and Burguete, J.: Time step restrictions for well-balanced shallow water solutions in non-zero velocity steady states, *Int. J. Numer. Meth. Fl.*, 60, 1351–1377, <https://doi.org/10.1002/ffd.1939>, 2009.
- Navas-Montilla, A. and Murillo, J.: 2D well-balanced augmented ADER schemes for the Shallow Water Equations with bed elevation and extension to the rotating frame, *J. Comput. Phys.*, 372, 316–348, <https://doi.org/10.1016/j.jcp.2018.06.039>, 2018.
- Nikolos, I. and Delis, A.: An unstructured node-centered finite volume scheme for shallow water flows with wet/dry fronts over complex topography, *Comput. Method. Appl. M.*, 198, 3723–3750, <https://doi.org/10.1016/j.cma.2009.08.006>, 2009.
- Özgen, I., Liang, D., and Hinkelmann, R.: Shallow water equations with depth-dependent anisotropic porosity for subgrid-scale topography, *Appl. Math. Model.*, 40, 7447–7473, <https://doi.org/10.1016/j.apm.2015.12.012>, 2015a.
- Özgen, I., Teuber, K., Simons, F., Liang, D., and Hinkelmann, R.: Upscaling the shallow water model with a novel roughness formulation, *Environ. Earth. Sci.*, 74, 7371–7386, <https://doi.org/10.1007/s12665-015-4726-7>, 2015b.
- Özgen-Xian, I., Kesserwani, G., Caviedes-Voullième, D., Molins, S., Xu, Z., Dwivedi, D., Moulton, J. D., and Steefel, C. I.: Wavelet-based local mesh refinement for rainfall–runoff simulations, *J. Hydroinform.*, 22, 1059–1077, <https://doi.org/10.2166/hydro.2020.198>, 2020.
- Özgen-Xian, I., Xia, X., Liang, Q., Hinkelmann, R., Liang, D., and Hou, J.: Innovations Towards the Next Generation of Shallow Flow Models, *Adv. Water Resour.*, 149, 103867, <https://doi.org/10.1016/j.advwatres.2021.103867>, 2021.
- Paniconi, C. and Putti, M.: Physically based modeling in catchment hydrology at 50: Survey and outlook, *Water Resour. Res.*, 51, 7090–7129, <https://doi.org/10.1002/2015WR017780>, 2015.
- Park, S., Kim, B., and Kim, D. H.: 2D GPU-Accelerated High Resolution Numerical Scheme for Solving Diffusive Wave Equations, *Water*, 11, 1447, <https://doi.org/10.3390/w11071447>, 2019.
- Petaccia, G., Soares-Fraza, S., Savi, F., Natale, L., and Zech, Y.: Simplified versus Detailed Two-Dimensional Approaches to Transient Flow Modeling in Urban Areas, *J. Hydraul. Eng.*, 136, 262–266, [https://doi.org/10.1061/\(asce\)hy.1943-7900.0000154](https://doi.org/10.1061/(asce)hy.1943-7900.0000154), 2010.
- Roe, P.: Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.*, 43, 357–372, [https://doi.org/10.1016/0021-9991\(81\)90128-5](https://doi.org/10.1016/0021-9991(81)90128-5), 1981.
- Schulthess, T. C.: Programming revisited, *Nat. Phys.*, 11, 369–373, <https://doi.org/10.1038/nphys3294>, 2015.
- Schwanenberg, D. and Harms, M.: Discontinuous Galerkin Finite-Element Method for Transcritical Two-Dimensional Shallow Water Flows, *J. Hydraul. Eng.*, 130, 412–421, [https://doi.org/10.1061/\(ASCE\)0733-9429\(2004\)130:5\(412\)](https://doi.org/10.1061/(ASCE)0733-9429(2004)130:5(412)), 2004.
- Serrano-Pacheco, A., Murillo, J., and Garcia-Navarro, P.: A finite volume method for the simulation of the waves generated by landslides, *J. Hydrol.*, 373, 273–289, <https://doi.org/10.1016/j.jhydrol.2009.05.003>, 2009.
- Sharif, M. B., Ghafoor, S. K., Hines, T. M., Morales-Hernández, M., Evans, K. J., Kao, S.-C., Kalyanapu, A. J., Dullo, T. T., and Gangrade, S.: Performance Evaluation of a Two-Dimensional Flood Model on Heterogeneous High-Performance Computing Architectures, in: *Proceedings of the Platform for Advanced Scientific Computing Conference, ACM*, <https://doi.org/10.1145/3394277.3401852>, 2020.
- Shaw, J., Kesserwani, G., Neal, J., Bates, P., and Sharifian, M. K.: LISFLOOD-FP 8.0: the new discontinuous Galerkin shallow-water solver for multi-core CPUs and GPUs, *Geosci. Model Dev.*, 14, 3577–3602, <https://doi.org/10.5194/gmd-14-3577-2021>, 2021.
- Simons, F., Busse, T., Hou, J., Özgen, I., and Hinkelmann, R.: A model for overland flow and associated processes within the Hydroinformatics Modelling System, *J. Hydroinform.*, 16, 375–391, <https://doi.org/10.2166/hydro.2013.173>, 2014.
- Singh, J., Altinakar, M. S., and Ding, Y.: Numerical Modeling of Rainfall-Generated Overland Flow Using Nonlin-

- ear Shallow-Water Equations, *J. Hydrol. Eng.*, 20, 04014089, [https://doi.org/10.1061/\(ASCE\)HE.1943-5584.0001124](https://doi.org/10.1061/(ASCE)HE.1943-5584.0001124), 2015.
- Sivapalan, M.: From engineering hydrology to Earth system science: milestones in the transformation of hydrologic science, *Hydrol. Earth Syst. Sci.*, 22, 1665–1693, <https://doi.org/10.5194/hess-22-1665-2018>, 2018.
- Sætra, M. L., Brodtkorb, A. R., and Lie, K.-A.: Efficient GPU-Implementation of Adaptive Mesh Refinement for the Shallow-Water Equations, *J. Sci. Comput.*, 63, 23–48, <https://doi.org/10.1007/s10915-014-9883-4>, 2015.
- Smith, L. S. and Liang, Q.: Towards a generalised GPU/CPU shallow-flow modelling tool, *Comput. Fluids*, 88, 334–343, <https://doi.org/10.1016/j.compfluid.2013.09.018>, 2013.
- Soares-Frazão, S.: Experiments of dam-break wave over a triangular bottom sill, *J. Hydraul. Res.*, 45, 19–26, <https://doi.org/10.1080/00221686.2007.9521829>, 2007.
- Soares-Frazão, S. and Zech, Y.: Dam-break flow through an idealised city, *J. Hydraul. Res.*, 46, 648–658, <https://doi.org/10.3826/jhr.2008.3164>, 2008.
- Steeffel, C. I.: CrunchFlow: Software for modeling multicomponent reactive flow and transport. Tech. rep., Lawrence Berkeley National Laboratory, California, USA, 2009.
- Steffen, L., Amann, F., and Hinkelmann, R.: Concepts for performance improvements of shallow water flow simulations, in: Proceedings of the 1st IAHR Young Professionals Congress, online, ISBN 978-90-82484-6-63, 2020.
- Stoker, J.: *Water Waves: The Mathematical Theory with Applications*, New York Interscience Publishers, Wiley, ISBN 978-0-471-57034-9, 1957.
- Su, B., Huang, H., and Zhu, W.: An urban pluvial flood simulation model based on diffusive wave approximation of shallow water equations, *Hydrol. Res.*, 50, 138–154, <https://doi.org/10.2166/nh.2017.233>, 2017.
- Suarez, E., Eicker, N., and Lippert, T.: Modular Supercomputing Architecture: From Idea to Production, in: Contemporary High Performance Computing, CRC Press, blackboxPlease add the place of publication., <https://doi.org/10.1201/9781351036863-9>, pp. 223–255, 2019.
- Tatard, L., Planchon, O., Wainwright, J., Nord, G., Favis-Mortlock, D., Silvera, N., Ribolzi, O., Esteves, M., and Huang, C. H.: Measurement and modelling of high-resolution flow-velocity data under simulated rainfall on a low-slope sandy soil, *J. Hydrol.*, 348, 1–12, <https://doi.org/10.1016/j.jhydrol.2007.07.016>, 2008.
- Thacker, W.: Some exact solutions to the nonlinear shallow-water wave equations, *J. Fluid Mech.*, 107, 499–508, <https://doi.org/10.1017/S0022112081001882>, 1981.
- The third international workshop on long-wave runup models: http://isec.nacse.org/workshop/2004_cornell/bmark2.html (last access: 22 August 2022), 2004.
- Toro, E.: *Shock-Capturing Methods for Free-Surface Shallow Flows*, Wiley, ISBN 978-0-471-98766-6, 2001.
- Trott, C., Berger-Vergiat, L., Poliakoff, D., Rajamanickam, S., Lebrun-Grandie, D., Madsen, J., Awar, N. A., Gligoric, M., Shipman, G., and Womeldorff, G.: The Kokkos EcoSystem: Comprehensive Performance Portability for High Performance Computing, *Comput. Sci. Eng.*, 23, 10–18, <https://doi.org/10.1109/mcse.2021.3098509>, 2021.
- Turchetto, M., Palu, A. D., and Vacondio, R.: A general design for a scalable MPI-GPU multi-resolution 2D numerical solver, *IEEE T. Parall. Distr.*, 31, <https://doi.org/10.1109/tpds.2019.2961909>, 2019.
- Vacondio, R., Palù, A. D., and Mignosa, P.: GPU-enhanced Finite Volume Shallow Water solver for fast flood simulations, *Environ. Modell. Softw.*, 57, 60–75, <https://doi.org/10.1016/j.envsoft.2014.02.003>, 2014.
- Vacondio, R., Palù, A. D., Ferrari, A., Mignosa, P., Aureli, F., and Dazzi, S.: A non-uniform efficient grid type for GPU-parallel Shallow Water Equations models, *Environ. Modell. Softw.*, 88, 119–137, <https://doi.org/10.1016/j.envsoft.2016.11.012>, 2017.
- Valiani, A., Caleffi, V., and Zanni, A.: Case Study: Malpasset Dam-Break Simulation using a Two-Dimensional Finite Volume Method, *J. Hydraul. Eng.*, 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)), 2002.
- Vanderbauwhede, W.: Making legacy Fortran code type safe through automated program transformation, *J. Supercomput.*, 78, 2988–3028, 2021.
- Vanderbauwhede, W. and Davidson, G.: Domain-specific acceleration and auto-parallelization of legacy scientific code in FORTRAN 77 using source-to-source compilation, *Comput. Fluids*, 173, 1–5, 2018.
- Vanderbauwhede, W. and Takemi, T.: An investigation into the feasibility and benefits of GPU/multicore acceleration of the weather research and forecasting model, in: 2013 International Conference on High Performance Computing and Simulation (HPCS), Helsinki, Finland, IEEE, <https://doi.org/10.1109/hpcsim.2013.6641457>, 2013.
- Vater, S., Beisiegel, N., and Behrens, J.: A limiter-based well-balanced discontinuous Galerkin method for shallow-water flows with wetting and drying: Triangular grids, *Int. J. Numer. Meth. Fl.*, 91, 395–418, <https://doi.org/10.1002/fld.4762>, 2019.
- Wang, Y., Liang, Q., Kesserwani, G., and Hall, J. W.: A 2D shallow flow model for practical dam-break simulations, *J. Hydraul. Res.*, 49, 307–316, <https://doi.org/10.1080/00221686.2011.566248>, 2011.
- Wang, Z., Walsh, K., and Verma, B.: On-Tree Mango Fruit Size Estimation Using RGB-D Images, *Sensors*, 17, 2738, <https://doi.org/10.3390/s17122738>, 2017.
- Watkins, J., Tezaur, I., and Demeshko, I.: A Study on the Performance Portability of the Finite Element Assembly Process Within the Albany Land Ice Solver, Springer International Publishing, Cham, 177–188, https://doi.org/10.1007/978-3-030-30705-9_16, 2020.
- Weill, S.: *Modélisation des échanges surface/subsurface à l'échelle de la parcelle par une approche darcéenne multidomaine*, PhD thesis, Ecole des Mines de Paris, 2007.
- Wittmann, R., Bungartz, H.-J., and Neumann, P.: High performance shallow water kernels for parallel overland flow simulations based on FullSWOF2D, *Comput. Math. Appl.*, 74, 110–125, <https://doi.org/10.1016/j.camwa.2017.01.005>, 2017.
- Xia, J., Falconer, R. A., Lin, B., and Tan, G.: Numerical assessment of flood hazard risk to people and vehicles in flash floods, *Environ. Modell. Softw.*, 26, 987–998, <https://doi.org/10.1016/j.envsoft.2011.02.017>, 2011.
- Xia, X. and Liang, Q.: A new efficient implicit scheme for discretising the stiff friction terms in the shallow water equations, *Adv. Water Resour.*, 117, 87–97, <https://doi.org/10.1016/j.advwatres.2018.05.004>, 2018.

- Xia, X., Liang, Q., Ming, X., and Hou, J.: An efficient and stable hydrodynamic model with novel source term discretization schemes for overland flow and flood simulations, *Water Resour. Res.*, 53, 3730–3759, <https://doi.org/10.1002/2016WR020055>, 2017.
- Xia, X., Liang, Q., and Ming, X.: A full-scale fluvial flood modelling framework based on a High-Performance Integrated hydrodynamic Modelling System (HiPIMS), *Adv. Water Resour.*, 132, 103392, <https://doi.org/10.1016/j.advwatres.2019.103392>, 2019.
- Yu, C. and Duan, J.: Two-dimensional depth-averaged finite volume model for unsteady turbulent flow, *J. Hydraul. Res.*, 50, 599–611, <https://doi.org/10.1080/00221686.2012.730556>, 2012.
- Yu, C. and Duan, J.: Simulation of Surface Runoff Using Hydrodynamic Model, *J. Hydrol. Eng.*, 22, 04017006, [https://doi.org/10.1061/\(asce\)he.1943-5584.0001497](https://doi.org/10.1061/(asce)he.1943-5584.0001497), 2017.
- Zhao, J., Özgen Xian, I., Liang, D., Wang, T., and Hinkelmann, R.: An improved multislope MUSCL scheme for solving shallow water equations on unstructured grids, *Comput. Math. Appl.*, 77, 576–596, <https://doi.org/10.1016/j.camwa.2018.09.059>, 2019.
- Zhou, F., Chen, G., Huang, Y., Yang, J. Z., and Feng, H.: An adaptive moving finite volume scheme for modeling flood inundation over dry and complex topography, *Water Resour. Res.*, 49, 1914–1928, <https://doi.org/10.1002/wrcr.20179>, 2013.