

A breakthrough in fast flood simulation

B. van den Bout^{a,*}, V.G. Jetten^a, C.J. van Westen^a, L. Lombardo^a

^a University of Twente, Faculty of Geo-Information Science and Earth Observation, Twente University, Hengelosestraat 99, 7514AE, Enschede, the Netherlands



ARTICLE INFO

Handling Editor: Daniel P Ames

Keywords:

Floods
Flow networks
Algorithms
Forecasting

ABSTRACT

The current status of technological advancement does not allow to generate detailed spatial flood forecasts. This hinders warning-systems, interactive planning tools and detailed forecasts. Our novel method computes flood hazard maps over three orders of magnitude faster than current state-of-the-art methods. It applies physically-based principles of steady-state flow to evade dynamic aspects of flood simulations. It estimates the relevant information for flood hazard, such as peak flow height, velocity and flood arrival time. Performance indicators show similar or exceeding accuracy compared to traditional flow models depending on the type of and data. In our tests, computation is reduced 1500 times. The method provides new perspective for the field of flood hazards, flood risk reduction through new types of early-warning systems, and user-interactive hazard assessment systems. As climate change is expected to aggravate flood hazard, the presented method can bring efficiency to flood simulation. The method is freely available at www.fastflood.org.

1. Introduction

Floods are the most frequently occurring type of disaster, percent of all disasters occurring in the past decades consist alluvial, flash floods and pluvial floods (McGlade et al., 2014). The immense world-wide economic and social impacts of flooding require more efforts in flood risk reduction, where fast and accurate hazard and assessment tools play a crucial role (Stelling and Verma, 2011). Physically-based numerical simulation tools have been developed for Disaster Risk Reduction (DRR) and Early Warning Systems (EWS) in both national scale and local scale. Such models generally employ an adaptation of the Saint-Venant equations for shallow water flow, combined with force descriptions related to gravity, pressure and friction (Delestre et al., 2014). When parameterized with elevation, surface and sub-surface related data, a full dynamic simulation is carried out. These models provide relevant information such as maximum flow height, maximum flow velocity and flood arrival time (Sanders, 2017).

The applicability of physically-based flood modelling has suffered from its computational demands. Efficient, simplified methods remained popular in hazard assessment and engineering applications (Rahman et al., 2011). These methods approximate the full physical system of water flow, as described by the navier-stokes equations, by implementing assumptions on the spatial or physics component of the model. The most often used set of assumptions are those of shallow water flow, depth-averaging the vertical velocity and force components, applying

vcross_16pt_pres
s_logo

pressure and gravity as a body force, and linearly approximating terrain normal forces. The resulting Saint Venant equations are industry standard in flood hazard modelling, forecasting and loss the globe. Beyond this, numerous further approximations simplify the physics, going into empirical approximations, or the spatial representation of the processes (Table 2). In the spatio-temporal physically-based simulation methods, the kinematic wave reduce the number of considered forces in even further. The quality loss and efficiency improvement can result in a more fit-for-purpose modelling system (Bout and Jetten, 2018). Steady state models simplify by approximating a state of equal incoming and outgoing discharge (q), resulting in zero net height change over time. Because of this, the time-dimension can be ignored in numerical integration. Empirical methods such as the rational method, unit hydrograph method or curve number method are used commonly in engineering due to their rapid speed. These tools predict only peak discharge, and are much less dependent on data quality and spatial details. Heuristic approaches exist in the form of fast approximate cellular automata (Gibson et al., 2016; Jamali et al., 2019) and machine learning models (Mosavi et al., 2018). These computational systems lose their physical basis but can be designed to quickly replicate the flowing behavior of water. Full spatial dynamical simulations have also seen improvements to efficiency. Techniques such as quad-tree domain subdivision, supercell water redistribution and parallelization on CPU and GPU compute power have been adopted for a significant reduction in

* Corresponding author.

E-mail address: b.vandenbout@utwente.nl (B. van den Bout).

Table 1

A short overview of some of the types of approximations used in flood modelling to increase efficiency while maintaining purpose-appropriate accuracy. More extensive databases are available at, for example, CSDMS ([Tucker et al., 2022](#); https://csdms.colorado.edu/wiki/Model_download_portal).

Physically-based, spatio-dynamical models			
Model	Type	Primary Assumption/simplification	Example models
Navier Stokes	Dynamic	Continuity of the fluid	Flo2D (Flo2D Software, 2006), Delft3D (Deltires, 2015), salva, fluidsim, FluidX3D, openFOAM (Jasak et al., 2007), Mike flood (DHI, 2014), TUFlow 3D (Syme, 2001)
	Physically-based	Mass and momentum conservation	Celeris (Tavakkol & Lyne, 2017), Mike21 (DHI, 2014)
	Spatial		
Boussinesq	Dynamic	Continuity of the flow, conservation of mass and momentum	
	Physically-based	Forces act on a depth-averaged water body	
	Spatial	Vertical velocity profile assumed	
Saint Venant	Dynamic	Velocity is depth averaged, and conforms to terrain	
	Physically-based	Viscosity ignored	
	Spatial		
Diffusive Wave	Dynamic	Continuity of the flow, conservation of mass and momentum	Flo2D (Flo2D Software, 2006), Delft2D (Deltires, 2014b), Hec-RAS (Brunner, 2002), LISEM (Bout and Jetten, 2018), FullSwof2D (Delestre et al., 2014), Mike Flood (DHI, 2014), TUFlow 2D (Syme, 2001), SOBEK (Deltires, 2014a), HE 2D (Iwah, 2017)
	Physically-based	Depth-averaged, terrain conforming flow and forces	
	Spatial	Inertia and viscosity terms ignored	
Kinematic Wave	Dynamic	Continuity of the flow, conservation of mass and momentum	LISEM (Bout and Jetten, 2018), PCRaster, HECHMS (Brunner, 2002), Glofas (Alfieri et al., 2013), Mike Hydro (DHI, 2014), TUFlow 1D (Syme, 2001), KWAVE (Rengers et al., 2016), TopoFlow (Peckham et al., 2017), TopModel (Wolock, 1993)
	Physically-based	Depth-averaged, terrain conforming flow and forces	
	Spatial	Inertia and Pressure terms ignored	
Steady-State	Not dynamic	Flow path requires directional network	Floodos 2D (Croissant et al., 2014), FastFlood (Steer et al., 2022)
	Physically-based	Continuity of the flow, conservation of mass and momentum	
	Spatial	Depth-averaged, terrain conforming flow and forces	
Non-spatial discharge approximations			
Model	Type	Primary Assumption/simplification	Example models
Curve Number	Not dynamic	Runout can be estimated from approximating a 1-to-1 relationship between inflow (precipitation) and outflow, while withholding initial storage and retention.	SCS-CN (SCS, 1985)
	Empirical		
	Non Spatial		
Rational Method	Not dynamic	Approximates a non-spatial steady-state flow, and corrects for storage and infiltration through an empirical runoff coefficient	Schaake et al. (1967), HydroCAD (Augustin, 2021)
	Empirical		
	Non Spatial		
Bucket method	Dynamic	A simple linear infiltration process with a threshold after which input precipitation equals precipitation	PCR-GLOBWB (Satunadja et al., 2018)
	Empirical		
	Non Spatial		
Unit hydrograph method	Dynamic	Normalized hydrographs are used to estimate discharge for new events with altered volumes and durations.	SCS Unit hydrograph model (SCS, 1985)
	Empirical		
	Non Spatial		
Empirical spatial methods			
Model	Type	Primary Assumption/simplification	Example models
Cellular automata	Dynamic	A cellular automata, mimicking the processes of water flow (divergent flow, filling depressions, following slope directions). Coefficients in the automata must be trained.	Caffé (Jamali et al., 2019)
	Cellular		
	Automata		
Machine Learning	Spatial		
	Not dynamic	A neural network trained on provided real simulation data that reproduces flood estimates for variations of the seen area.	Mosavi et al. (2018); Berkahn et al. (2019); FloodGAN (Hofmann & Schuttrumpf); Ahmad et al., 2022
	Neural Network		
	Spatial		
Machine learning aided differential solver	Not dynamic	Approximate method for the physically-based methods described above. This method solves the physically-based differential equations using fast, small, trained neural networks.	SPNets (Schenck and Fox, 2018), Raissi et al. (2020)
Particle-based Stochastic flow fields	Neural Network		
	Spatial		
	Not dynamic	Stochastically located particles, representing the water volumes on the terrain, follow empirical rules to mimic water flow over a landscape.	r.simwater (Mitasova et al., 2004)
Flow network accumulation routing	Empirical		
	Spatial		
	Not dynamic	Drainage networks can be produced by a variety of algorithms, including D4, D8 and D-Infinity. Each of these can be used for flow accumulation and estimation of flow networks.	D4, D8, DInfinity, r.terrassim (Arge et al., 2001)(available in many GIS systems)
Spatial approximation types			
Model	Spatial type	Primary Assumption/simplification	Example models
Lumped hydrology/ flow	Dynamic	0D, single value output	SWAT (Williams et al., 1984), D-Hydrology (Deltires, 2014a), SWMM (Gironás et al., 2010)
	Physically-based		
	Not spatial		
1D Flow networks	Dynamic	1D, output along a segment (usually a channel or river)	HecRAS (Brunner, 2002), Mike (Syme, 2001)
	Physically-based		
	Spatial		
Coupled 1D-2D models	Dynamic	1D/2D, output as a spatial field	Mike (DHI, 2014), Hec (Brunner, 2002), Tuflow (Syme, 2001), Deflt2D/D-Hydrology (Deltires, 2014a), LISEM (Bout and Jetten, 2018)
	Physically-based		
	Spatial		
Coupled 2D-3D models	Dynamic	2D/3D, output as a volumetric field	Mike (DHI, 2014), Hec (Brunner, 2002), Tuflow (Syme, 2001), Deflt2D/3D (Deltires, 2014a)

(continued on next page)

Table 1 (continued)

Coupled 2D-1D models	Physically-based Spatial Dynamic Physically-based Spatial	Multiple similar 2D model setups are linked through boundary conditions. Often nested areas are chosen with smaller areas with higher resolution nested in larger model domains of lower resolution.	See 2D flood models
Quad Tree structure	Dynamic Physically-based Spatial	An adaptive quad-tree structure is used to utilize larger, more efficient, grid cells in areas that are not of priority and have homogeneous properties.	3DI water management (Liang et al., 2004)
Super-cell water redistribution	Dynamic Physically-based Spatial	The model is run at a lower spatial resolution, and later the volume of flood water is redistributed over the higher-resolution elevation data.	CaMa-Flood (Yamazaki, 2014)
GPU Flexible mesh compute	Dynamic Physically-based Spatial	A flexible triangular grid is constructed with variable element area. Areas of less importance get larger, more efficient, grid structure, and GPU compute power is used to increase efficiency further.	Triton (Morales-Hernández et al., 2021), Liu et al. (2018)
Developed method Model	Spatial type	Primary Assumption/simplification	
FastFlood method (Presented here)	Not dynamic Physically-Based Spatial	Flow direction networks approximate actual flow directions A steady-state can be compensated to approximate real peak flow An inverse relationship can approximate peak flow height from peak discharge	FastFlood.org

computation times (Kalyanapu et al., 2011; Liang et al., 2008; Yu and Lane, 2006). A wider overview of methods in the literature is provided in Table 1.

Despite the availability of the models described in Table 1, flood simulation remains a computationally intensive task. Within the existing methods, the fastest of these suffer from applicability to various flood types and still can not offer real-time forecasts or rapid iterative mitigation design in many contexts (Liu et al., 2018; Jamali et al., 2019; Berkahn et al., 2019; Ming et al., 2020). While machine learning approaches are promising, their application is not transferable between areas, and alterations in the landscape context often require retraining of the model (Hofmann and Schüttumpf, 2021). In addition, the flood problem in general has complex spatial dependencies that poses an issue for many types of convolutional neural networks (Karim et al., 2023).

In this article, we present a new method for computation of flood hazard maps that decreases simulation time by a factor 1514 on our five study sites. Our method (the FastFlood model) uses a compensated steady-state flow to estimate peak throughflow in the spatial domain. We are able to efficiently solve the steady-state flow field using the Fast Sweeping Method for terrain hydrological correction, and flow accumulation. Steady state conditions are not commonly met in nature, as rainfall and infiltration are dynamic processes. However, we apply an innovative compensation term based on catchment shape properties and flow velocities. In section two, we describe the method and related algorithms. In section three, we compare the method with full simulation, observations and other fast approximate methods for various type of flood events. In section 4 we discuss the potential usage, strengths and benefits of the new method.

The developed tool has been published as the fastflood model, available for free on www.fastflood.org (Fig. 1)

2. Materials and methods

The method uses a combination of four innovative concepts, which are further expanded in the methods section.

- Fast steady-state flow accumulation solver
- Inversion of flow accumulation field
- Compensation for partial steady-state (when applicable)
- Adaptive solver to refine pressure-driven inundation

The data requirements for the method are limited compared to more

advanced flood simulation models. Our models requires data on the event duration, and rainfall intensity (can be spatial), Mannings Surface roughness (can be spatial) and elevation of the terrain. These datasets can be obtained from global datasets of rainfall (e.g. GPM), elevation (e.g. SRTM), and land cover (e.g. Corine, GlobCover), which gives the opportunity to use the model in data scarce regions. More detailed datasets (e.g. LiDAR-based elevation data) and field measurements and verification, however, are critical in obtaining high quality data. Note that, because of the steady-state assumption, the model is not fully dynamic. The presented method attempts to circumvent the need for full dynamic simulation, instead directly estimating several variables that are of high importance. Outputs include peak flow height, peak flow velocity, peak discharge and flood arrival time.

2.1. Steady state flow accumulation

The steady-state solver is based on flow theory, which is closely linked to the Saint-Venant equations and considers flow states with time-constant input, and a resulting non-dynamic flow states (de Saint-Venant, 1871). In such cases, a constant velocity field is present, and mass conservation results in inflow equal to outflow ($Q_{in} = Q_{out}$, where Q is water discharge). A special property of these flow states is that accumulation of water sources through the network is equal to the steady-state discharge, as the accumulation of all upstream incoming flows must equal local outflow.

We developed an extremely efficient algorithm for steady-state flow using multi-directional sweeping, based on the fast sweeping method (Zhao, 2005). Traditional methods loop over elements of the domain and transfer fluxes to direct neighbors. Our implementation sweeps the data through the velocity-field in-place, allowing for material to travel large stretches of space in a single iteration. In particular, as long as the principle directions are the same, the material flux can be accumulated through the network within a single iteration. For more general application, material moves along similar directions in a single iteration. For more details on this technique, see appendix A.

As a result, typical flow accumulation over terrain is solved an order of magnitude faster. In the example of Fig. 2, only requiring 21 iterations as opposed to 820, while the terrain is far from a smooth ideal case.

We use this algorithm first to create a monotonically-increasing elevation model (hydrologically-corrected). Using the same multi-directional sweep algorithm, we can accumulate direction-specific slope to reconstruct the elevation model. By limiting the slope in x-

Table 2

Overview of the properties of the study sites used in this work.

Region	Area (km ²)	Pixel size (m)	Pixel count	Validation data type	Water source	Elevation Source
Italy (Fella)	180	20	450,000	Flood extent and Discharge	Precipitation	LIDAR
Dominica (Grande-Bay)	34.7	10	347,123	Flood extent	Precipitation	LIDAR
Tajikistan (Kafirnigan)	71.7	30	79,684	Flood extent	Discharge boundary condition	SRTM
Netherlands (Levee 41)	875	40	546,875	Flood extent reference simulation	Discharge boundary condition	LIDAR
St. Lucia	617	10	6,173,205	None	Precipitation	Contour Lines

and y direction to be strictly positive, we can efficiently reconstruct the elevation model without depressions.

Using the x- and y-direction gradient of the corrected elevation model as velocity field, we can now accumulate precipitation through the network to get flow accumulation (which equals steady-state flow). Our usage of terrain slope as a directional flow network means that, for now, we ignore pressure forces but only consider gravitational and frictional forces for the direction of movement. This will be compensated in a later stage.

2.2. Inversed flow accumulation

As a result of the steady-state simulation described above, we have obtained the steady-state discharge in space. Natural systems however, encounter dynamic precipitation, and are often too large to be covered by an approximate steady-state flow. Instead, a discharge wave will propagate through the flow network. We can define the flow accumulation as (Equation (1)).

$$AF(R) = Q_{af} \quad 1$$

where $AF(x)$ is the flow accumulation algorithm, which takes a source material as input, and returns the accumulated throughput of that source material when it follows the velocity-field, R is the precipitation rate and Q_{af} is the steady-state discharge.

To invert the flow accumulation values and obtain flow heights, we employ Manning's surface flow law, which is an inversion of the momentum balance of kinematic flow. By approximating the hydraulic radius as the flow height (shallow floodplain assumption), and solving for flow height, the expression becomes (Equation (2)).

$$h_{af} = \left(\frac{q_{af}}{\Delta x} \frac{n}{\sqrt{s}} \right)^{\frac{3}{5}} \quad 2$$

where h_{af} is the flow height estimated from the discharge value, n is Manning's surface roughness coefficient, Δx is the width of the gridcells and s is the slope gradient of the terrain in the down-slope direction. The methods described above are the principal components responsible for the speed of the presented method. The fast estimation of steady-state velocity-fields linked with inverted flow accumulation-fields provides an extremely fast method for estimation of steady-state flow heights. Spatial precipitation can be used as input, as the flow accumulation algorithm will then estimate the steady-state flow accordingly.

2.3. Compensation for partial steady state

In order to compensate for the partial steady-state, we carry out a compensation scheme based on spatial properties of flow networks. When observing the shapes of catchments, it has been often noted that there is a power-type relationship between the distance to an outlet and the contributing area at that distance (e.g. a circle with an outlet at its center has more area further away from the outlet, growing quadratically with distance). When taking s as the distance from the point under consideration, we can normalize an exponential distribution s^b (Equation (3)).

$$PDF(s) = (1 - b) \left(\frac{s}{s_{max}} \right)^b \quad 3$$

where PDF is the probability density (normalized), s is the normalized distance (originally in meters, but normalized from 0 to 1, where 1 represents s_{max}), s_{max} is the maximum distance (meters) and b is the catchment shape parameter (–).

From our flow accumulation algorithm, we can define two conditions which let us determine the values of the catchment shape parameter, and the catchment maximum distance. The first condition is the average distance from the gridcells in the catchment to the outlet, which can be obtained through integration of the PDF, or through the flow accumulation method (Equation (4)).

$$\langle s \rangle = \frac{\int_0^{s_{max}} s (1 - b) \left(\frac{s}{s_{max}} \right)^b ds}{\int_0^{s_{max}} (1 - b) \left(\frac{s}{s_{max}} \right)^b ds} = \frac{(1 + b) s_{max}}{2 + b} \quad 4$$

We can acquire this value from the flow accumulation step as well. As $AF(1)$ represents for each location the number of pixels that eventually flow to it, $AF(AF(1))$ is the total number of pixels traveled, and $\frac{AF(AF(1))}{AF(1)}$ is the average number of pixels traveled. Then, multiplying with the cellsize gives the average travel distance.

$$\langle s \rangle = \frac{\Delta x AF(AF(1))}{AF(1)} \quad 5$$

The second condition comes from the fact that our methods assume there is a single outlet gridcell, which has specified area. This is the only area that lies between a distance of 0 and Δx . To obtain gridcell density, we scale the PDF with the total number of gridcells ($AF(1)$ at the cell under consideration) and integrate from a distance of 0 to Δx . The integration limits are normalized in space to match the PDF, resulting in equation (6).

$$1 = \int_0^{\Delta x} AF(1) (1 - b) \left(\frac{s}{s_{max}} \right)^b ds = AF(1) \left(\frac{\Delta x}{s_{max}} \right)^{1+b} \quad 6$$

The system of equations (4) and (6) can be solved for b for each gridcell within the model domain, giving an approximate shape for each pixels upstream catchment.

The peak flow of each location depends on the partial steady state of all contributing locations. For each location, we need to know for all the contributing area if it had time to reach that location or not. In an ideal linear kinematic wave system, the peak discharge depends on maximum discharge build-up in relative catchment space (Fig. 3).

Consider now the discharge progressing through the catchment at distance s . Each infinitesimal interval of s represents a set of points within the catchment that are an equal distance removed from the point under consideration. We can express the total discharge currently passing through that set of points as equation (6). The total discharge currently propagating in an infinitesimal interval of s is equal to the average unit discharge at that interval ($Q_{unit,t}$) multiplied by the total flow width at that distance s ($\propto PDF(t)$).

$$Q_t(s) = PDF(s) Q_{unit,t}(s) \quad 7$$

where Q_t is the total discharge in an interval of relative space and $Q_{unit,t}$

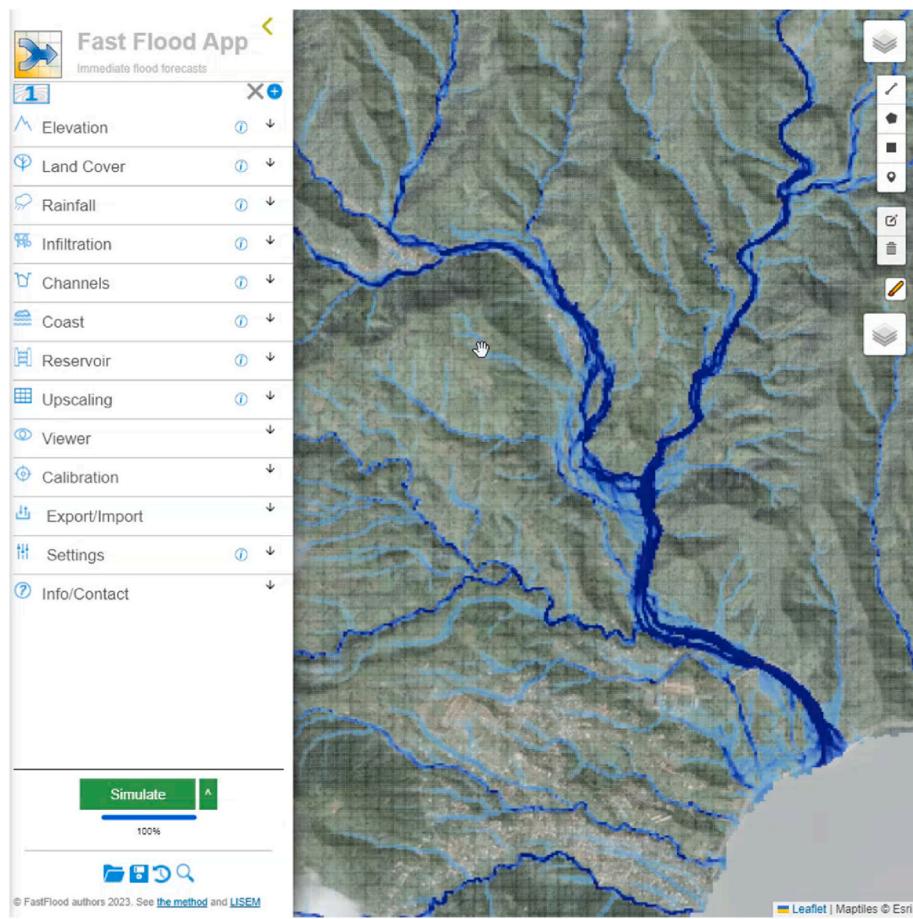


Fig. 1. The FastFlood.org website, featuring the methodology as described in this work.

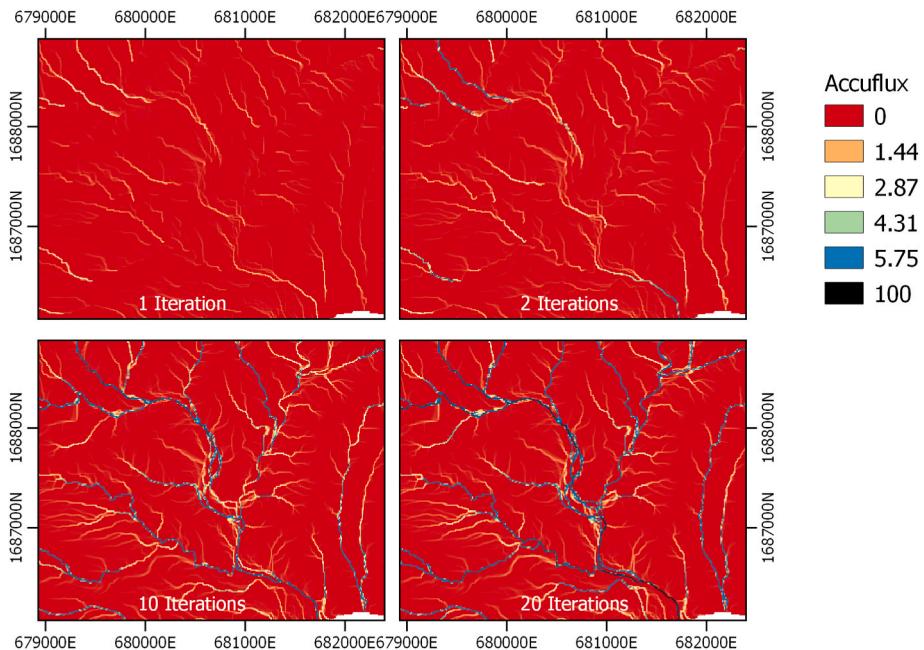


Fig. 2. The accumulation of flow after several iterations of our multi-directional sweeping algorithm. Area is the Grand-Bay catchment on Dominica.

is the discharge per unit flow width along a relative-space interval of the catchment.

For any location, a maximal discharge slice might be found on the

border between areas of full and partial steady state. In a linear kinematic wave system, this maximal discharge slice will propagate through the system and become the peak discharge at outlet.

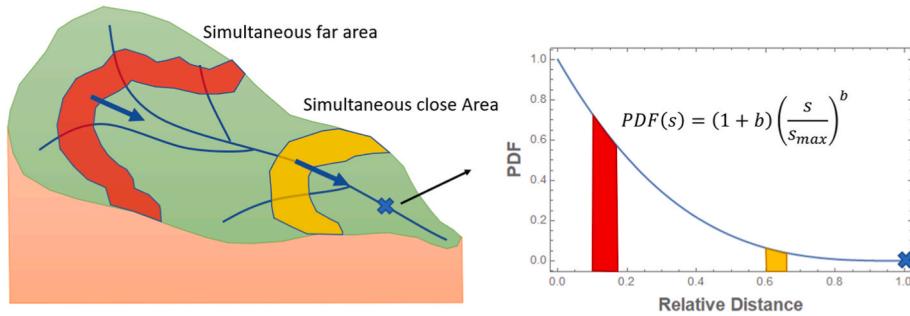


Fig. 3. There is a direct proportionality between the discharge progression in the flow accumulation scheme and the frequency of pixels with a relative time to arrival for the observation point. To match the visualization, we flip the relative distance for the PDF on the horizontal axes.

In a rainfall-fed context, areas upstream are fed by a smaller area, so their discharge should be lowered based on the relative size of the area attributing to flow at that point. We define the steady state distance (the approximate relative distance traveled by water) as s_{ss} . This distance is estimated using the duration of the event, and the average flow velocities.

When $s \leq s_{ss}$, all contributing area for those gridcells have reached a full steady-state flow. Discharge through infinitesimal interval at s can be estimated as the cumulative upstream area multiplied with the precipitation rate.

$$Q_t(s) = CDF(s) R \bullet s \leq s_{ss} \quad 8$$

where Q_t is the total discharge going through an interval of relative space and R is the precipitation rate in the catchment, and $CDF(s)$ is the cumulative distribution ($CDF(s) = s^{1+b}$).

When $s > s_{ss}$, only a part of the upstream area contributes to the discharge. We can estimate the discharge by taking the total contributing area and the contributing area within the steady-state distance.

$$Q_t(s) = CDF(s) - CDF(s - s_{ss}) R \bullet s > s_{ss} \quad 9$$

This assumes precipitation is distributed homogeneously through a catchment. For our scheme, we are interested in $\max(Q_t(s))$ within $[0, 1]$ in relative distance space. The ratio between this maximum value, the maximum discharge that will eventually propagate to the outlet, and the full steady state discharge becomes the compensation factor.

$$f_{ss} = \frac{\max(Q_t(s))}{CDF(1)R} \quad 10$$

The maximum of $Q_t(s)$ can typically be easily found, as the catchment shape PDF is limited to be monotonically increasing with distance. The largest possible contribution will always come from the furthest upstream section of the catchment, the interval $[0, s_{ss}]$. Under this condition, the maximum occurs where $s = s_{ss}$.

$$\max(Q_t(s)) = Q_t(s_{ss}) = CDF(s_{ss}) \quad b \leq 1 \quad 11$$

So, finally, our compensation factor becomes

$$f_{ss} = \frac{\max(Q_t(s))}{CDF(1)R} = CDF(s_{ss}) \quad s_{max} > s_{ss} \quad 12$$

$$f_{ss} = 1 \quad s_{max} \leq s_{ss} \quad 13$$

where f_{ss} is the correction factor that compensates for the partial steady-state flow. Finally, our corrected peak discharge becomes

$$q_c = f_{ss} AF((\Delta x)^2 R) \quad 14$$

where q_c is the compensated partial steady-state discharge, f_{ss} is the compensation factor derived from the catchment shape approximation, Δx is the gridcell size and R is the precipitation rate.

2.4. Adaptive solver to refine pressure-driven inundation

Our final step uses a diffusive wave solver that includes gravity, friction, pressure and advection, but ignored inertial terms. Utilizing a Darcy-Weisbach friction law and linearized gravity term, the following equations describe the momentum balance (Te Chow, Maidment and Mays, 1962) (Equations (15) and (16)).

$$\frac{\partial u_x}{\partial t} \Big|_{\text{steady state}} = 0 = g \frac{dz}{dx} + g \frac{d(\frac{1}{2}h^2)}{dx} - g n \frac{u_x | \vec{u}|}{h^3} \quad 15$$

$$\frac{\partial u_y}{\partial t} \Big|_{\text{steady state}} = 0 = g \frac{dz}{dx} + g \frac{d(\frac{1}{2}h^2)}{dx} - g n \frac{u_y | \vec{u}|}{h^3} \quad 16$$

where u is the velocity ($m s^{-1}$), z is the elevation of the terrain (m), h is the flow height (m), g is the gravitational acceleration ($m s^{-2}$), and n is Manning's Surface roughness coefficient ($s m^{-\frac{1}{2}}$). These equations lack the inertial part of the Saint-Venant equations. Such simplifications have been defended extensively in theoretical and modelling work using kinematic and diffusive wave equations in the literature (Miller, 1984). As a result of the missing terms, they can be solved analytically for the velocity u , when the elevation and water height fields are known (Equation (17)).

$$u_x = \sqrt{\frac{h^{\frac{3}{2}}}{n} \left(-\frac{dz}{dx} - \frac{d(\frac{1}{2}h^2)}{dx} \right)} \quad u_y = \sqrt{\frac{h^{\frac{3}{2}}}{n} \left(-\frac{dz}{dx} - \frac{d(\frac{1}{2}h^2)}{dx} \right)} \quad 17$$

Finally, a mass conservation equation can be applied to the flow heights. Iteratively updating flow height will finally reach a steady state, as increased water heights increase outflow and the flow heights converge to an equilibrium (Equation (18)).

$$h_{i+1} = h_i + dt \left(\frac{d(h_i u_x)}{dx} + \frac{d(h_i u_y)}{dy} + \max(0, R - I) \right) \quad 18$$

We apply these equations to the inverted flow height coming from the flow accumulation compensated for a partial steady-state. However, equation 15 shows that the assumed relationship between flow height and discharge is non-linear, and flow heights in this steady-state are not conserved, while discharges are. For this reason, we adapt equation (18) to flow according to diffusive wave principles, but conserve discharge instead. Finally, we apply this equation instead (Equation (19)).

$$(h_{i+1})^{\frac{5}{3}} = (h_i)^{\frac{5}{3}} + dt \left(\frac{d((h_i)^{\frac{5}{3}} u_x)}{dx} + \frac{d((h_i)^{\frac{5}{3}} u_y)}{dy} + \max(0, R - I) \right) \quad 19$$

In order to improve computational speed, we employ a numerical scheme with gradually decreasing artificial velocity. That is, in case of small flow velocities in some locations, and high velocities in others, the global timestep must be small, resulting in many required steps to advect

Table 3

Accuracy and computation time results for both the presented FastFlood method, and full dynamic simulation. Accuracy is shown as percentage of correctly labeled pixels: flood or no flood.

% Accuracy	UC Full (%)	CAL Full (%)	UC FastFlood (%)	CAL FastFlood (%)	Area (km ²)	Grid cell size (m)	CT - Full (seconds)*	CT -FastFlood (seconds)*	Sens – Full (%)**	Sens – FastFlood (%)**
Italy	92.10	92.26	93.8	94.95	180	20	2.2	2901	12.3	11.2
Dominica	95.34	95.69	94.52	95.54	34.7	10	1.3	1178	6.3	8.4
Tajikistan	86.78	87.08	96.87	98.23	71.7	30	0.42	195	–	–
Netherlands	100	100	99.13	99.91	875	40	2.1	880	–	–
St. Lucia	–	–	–	–	617	10	47	29,940	–	–

UC = Uncalibrated, CAL = Calibrated, Full = full dynamic simulation, FastFlood = Super-Fast-Flood-Simulation, CT = Compute Time *Average run-time per single simulation on AMD Threadripper 3970x, ** Sensitivity pertains to the average relative change in flood area compared to a relative change in input parameter.

the material through the model domain. In the case of steady-state modelling, we can adapt our model to instead initially have a fixed ratio between flux and the water volume within a gridcell. During the simulation, this constant fraction is linearly altered to become the actual fraction. This helps move large volumes of water initially while maintaining accuracy in the final iterations.

2.5. Method summary

Thus, our final approach contains the following steps: i) run a minimal steady-state solver to find a stable directional velocity field and carry out flow accumulation on the velocity field to efficiently solve for steady-state discharge, ii) invert discharge to estimate steady-state flow height, iii) run a compensation scheme for the partial steady-state flow of the event. iv) Compensate for pressure-based inundation through an adaptive diffusive-wave solver. Additional features such as reservoirs and channels with confined 1D flow can be implemented with relative ease by adapting the flow accumulation algorithm.

Within the space of full and approximate flood simulation tools, the FastFlood method does not directly equate with an existing model. Other a-temporal models exist that use some efficient method for flow routing, but these are not physically-based in the same manner. The concept of using compensated steady-state flow is also not used in any of the mentioned tools. As described in the discussion, this brings unique benefits and sensitivities to the model.

2.6. Rainfall selection

The steady-state algorithm requires, for each location, a single precipitation rate to function. Rainfall intensities might fluctuate, and lower precipitation might have a much longer duration compared to higher-intensities. In order to best represent an event, the duration t and intensity R might be chosen to maximize the product $R_{tot} = t R$ within the constraints of the temporal record.

2.7. Study sites and events

In this article, we verify the applicability and accuracy of the developed model on a set of three study-sites and events. These events have been chosen to represent some of the various types of context surrounding flood events.

The first study site is a watershed located on the South-East of the Caribbean island Dominica. Here, around the town of Grande-Bay, Category-5 hurricane Maria made landfall in autumn 2017. With over 500 mm of rainfall in 24 h, flash floods destroyed vast parts of the island. The selected catchment is described by Lidar-based elevation data (resampled to 10 m), Mannings surface roughness derived from field observations, the USGS field manual and a sentinel-2 based land use map. Rainfall data is available from Canefield airport. For a further description of the area and event, see also (Briones, 2019; Van Den Bout and Jetten, 2020).

The second event is a part of the Fella River basin, located in the Italian alps, with Ponteba chosen as the outlet. On august 29, 2003, an

intense cloud-burst resulted in major flashfloods throughout the Fella basin, destroying houses and infrastructure throughout the steep valleys. Here again, mapped flood extent is available, in combination with LIDAR elevation data, resampled to 20 m resolution. Additionally, land use and infiltration behavior was studied by. For a further description of the area and event, see also (Borga et al., 2007; Bout and Jetten, 2018)

The third site is a river-segment located in central Tajikistan, just South of the capital Dushanbe. In this area, a 2016 precipitation event resulted in flooding along the Kafimigan river, just south of the capital Dushanbe. Here, 30-m global SRTM elevation was filtered for high-frequency noise to improve quality of the terrain data. Mapped flood extents are available for a section of the river. Incoming discharge is provided as a boundary conditions on the upstream side of the river. We employ the discharge records for this area. The relevant return period discharge is provided as a boundary condition for both the presented FastFlood method and the full dynamic simulation. For more information on the event and its impact, see also (van Westen, 2019)

The fourth and final study site is a segment of the River Maas, located in the central parts of the Netherlands. In this particular area, a circular levee-system (dijkkring 41), protects a sub-sealevel region of land from water in the Maas and Waal river. High-resolution fully dynamic modelling has been carried out for various levee breaching scenario's. Here, we utilize one of these scenarios as a reference for our developed method. We employ a boundary condition on the eastern side of the Maas, based on the assumptions from the scenarios in the national flood defense program of the Netherlands (Alkema and Middelkoop, 2005).

2.8. Hydrology and calibration method

Hydrology was simulated identically for both the FastFlood method and the full simulations. In the case of Grande-Bay and Fella, we employ a fixed infiltration percentage based on literature sources on the event. By making this choice, we attempt to provide a fair comparison of the methods, as both can be linked to various more detailed hydrological systems that are beyond the scope of this work. For both the Dushanbe and Maas events, we assume infiltration and evapotranspiration during the event might be ignored. Calibration was carried out for both the presented method and the full dynamic flow simulations. A brute-force calibration approach was applied to the commonly chosen parameters. In this case, 6 parameter values were chosen on regular intervals between 50 and 150% of original values. The presented method was calibrated on 3 parameters: event duration, event net precipitation (infiltration is already subtracted), and Mannings Surface roughness coefficient. The full dynamical simulation was calibrated on 2 parameters: Mannings Surface roughness coefficient and infiltration percentage.

3. Results

3.1. Application, accuracy and speed

Below, we show a comparison with full dynamic simulations and mapped flood extent for various flood events in different environmental settings around the word (Italy, Dominica, Tajikistan and the

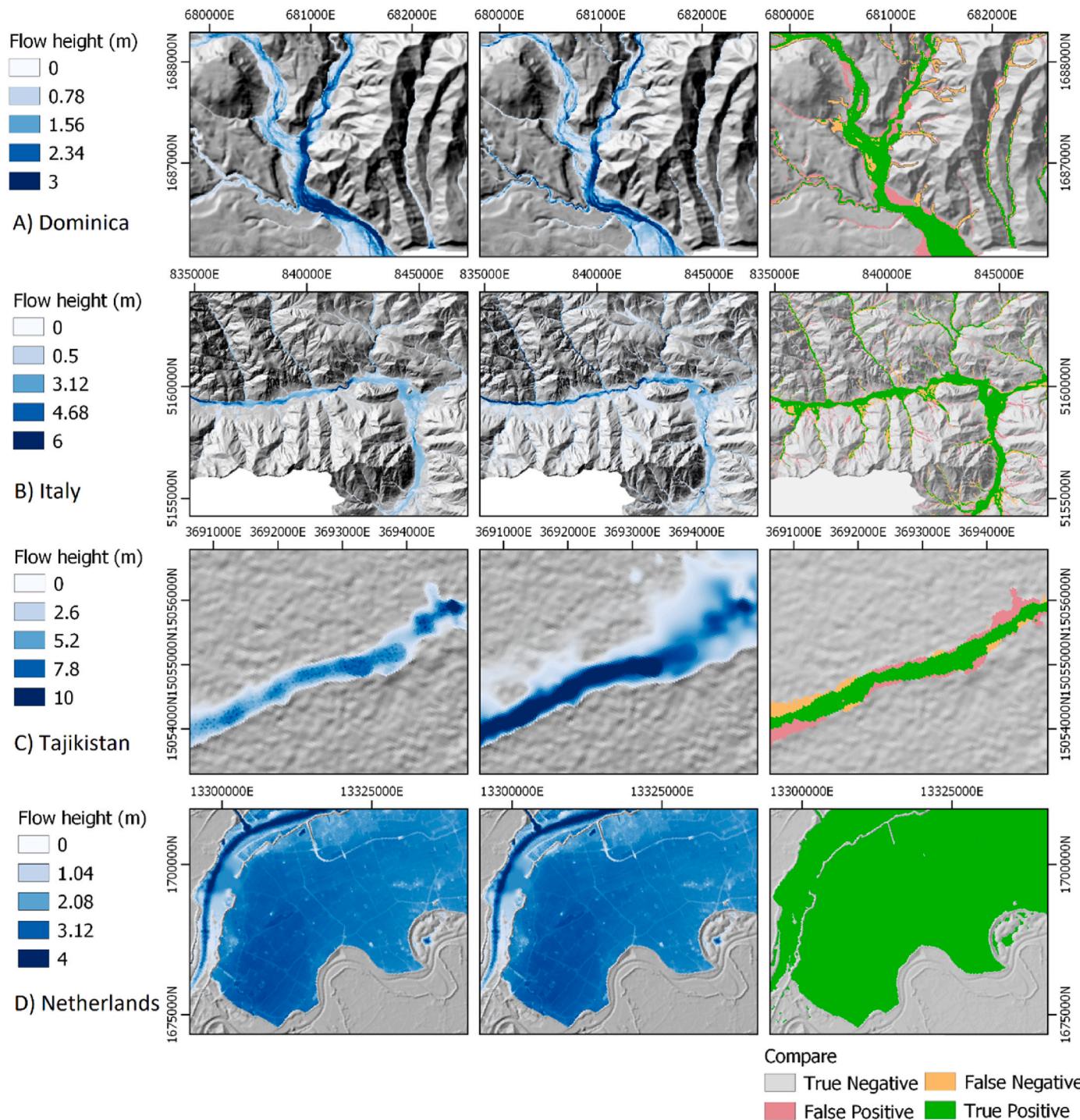


Fig. 4. Left) Maximum flow height for the presented FastFlood method. Middle) Maximum flow height for full dynamic simulation. Right) Comparison of FastFlood results and the observed flood extent. For the Maas flood (bottom), comparison is with reference high-resolution numerical simulation.

Netherlands). These events were chosen to cover a range of flood types (flash floods, riverine flooding, levee systems), data quality contexts (high-resolution lidar data at 10 m resolution or global SRTM elevation at 30 m resolution), and domain sizes (between 34 and 617 km²). See the methodology and data section for a description of these areas, and the studied events (Table 3).

Over-all, the presented method is able to perform with an accuracy of 75.1%, as compared to an average accuracy of 93.8% for the full simulations. The performance is roughly equal in those cases with good elevation, whereas the results are worse for the case in Tajikistan, where

a low quality SRTM DEM was used. Computation time was, on average, reduced by two orders of magnitude (avg. 1514.5x). Despite the similarities in accuracy for the super-fast flood simulation and full dynamic flow simulation, there are aspects where the methods differ considerably. In many cases, elevation data contains hydrologically inconsistencies known as local depressions, often due to elevation model errors. Flood simulations suffer from these, as they influence total outflow and peak flow rates (Bout and Jetten, 2018). The narrow streams feeding the Fella river (Fig. 4C, Italy study case) are poorly captured in the elevation data, and water is effectively captured. Our

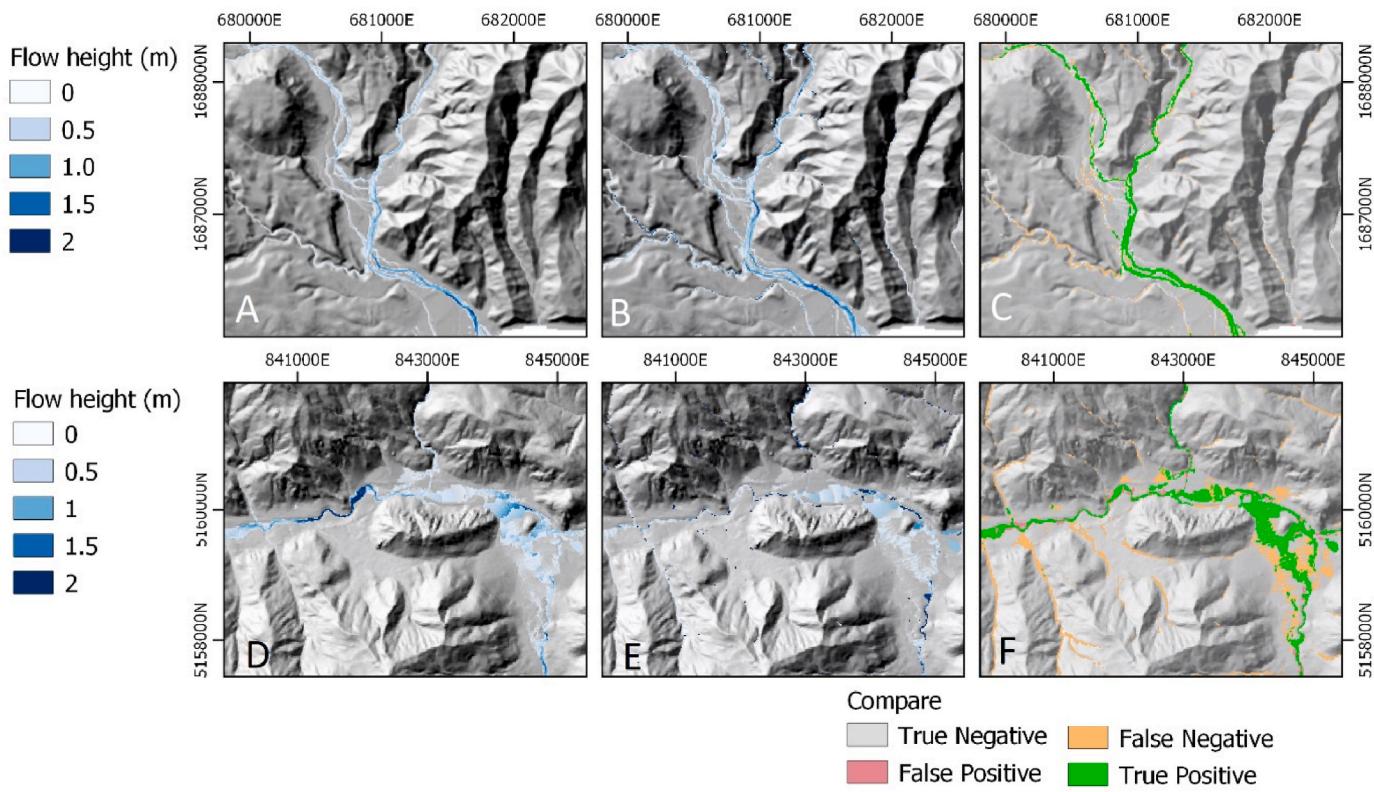


Fig. 5. A comparison of the presented FastFlood method and full dynamical simulation for a smaller event (1/10th the size of the event described earlier) for ABC (Dominica and DEF) Italy. A/D) FastFlood maximum flow height results. B/E) Full simulation maximum flow height results. C/F) Comparison between FastFlood and full dynamic simulation flood impact area.

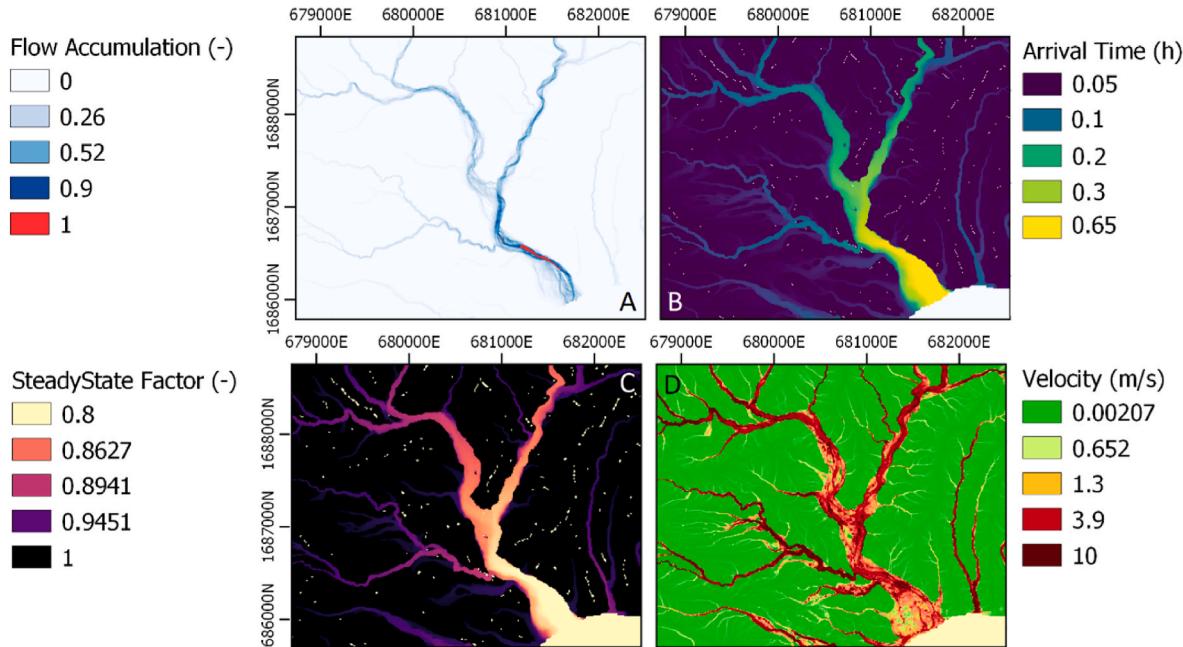


Fig. 6. Additional outcomes obtained from the presented FastFlood method for Grande-Bay on Dominica. A) Flow accumulation, which in the steady-state assumption equals discharge. B) Flood arrival time, C) Compensation factor for the partial steady-state of the event. See the methodology for further details on this. D) Velocity during peak flow.

method solves this problem by automatically considering a hydrologically corrected dem when specified by the user.

The simulated events all feature highly extreme occurrences of

precipitation or discharge. To verify behavior of the model in the context of small precipitation events, the calibrated models are validated on events with 10 times reduced precipitation input. Here, reference flow

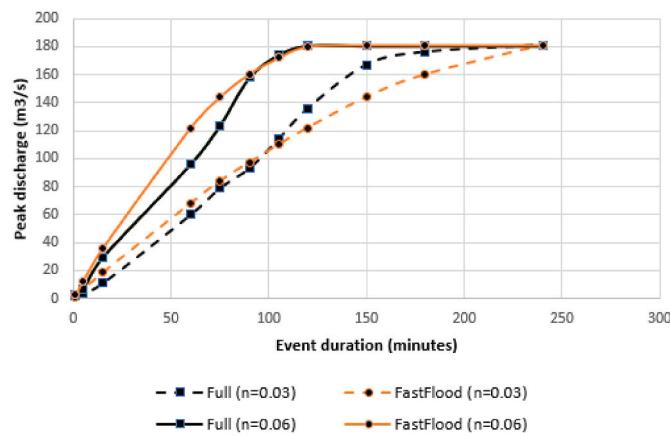


Fig. 7. A comparison of peak discharge predicted by the FastFlood model and full simulation for events of constant intensity but increasing intensity.

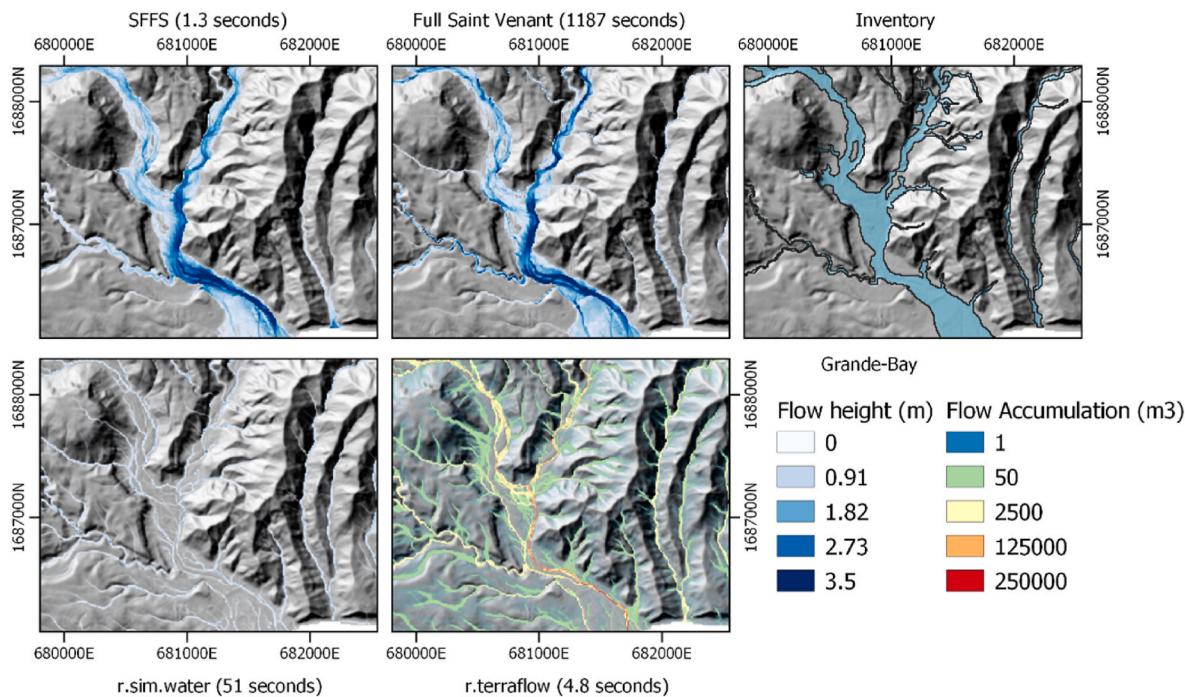


Fig. 8. Comparison of the presented method with full dynamics simulation, mapped flood extent (inventory), r.sim.water, and r.terraflow.

extents are not available, but reference simulations using full dynamic models are used instead (Fig. 5).

Besides the most commonly used hazard information (peak flow height and peak flow velocity), other information can be extracted from the model, such as flood arrival time, compensation factor for partial steady-state, b-coefficient, velocity field and the flow accumulation. For a more detailed explanation of these variables, please refer to the methods section.

The steady-state compensation is a critical feature in large-scale areas where a steady state is very far from being reached (see Fig. 6). In the case of the Grande-Bay area, a steady state is reached for most of the area besides the stream areas. The steady-state compensation remains close 1. In case of very large catchments, steady-state discharge will be a much more significant deviation (over-estimation) of the actual peak discharge and steady-state correction factors will be very small ($\ll 1$). In such cases, calibration of parameters such as Mannings N is of high importance. For the Grande-Bay area, the full simulation and fast-flood estimate of peak discharge for events of various durations is shown in Fig. 7. The discharge point observed here is the outlet of the main channel at the coastline.

3.2. Comparative analysis

We compare our method with full dynamic Saint-Venant simulation, as well as popular methods for rapid flood hazard mapping: r.sim.water and r.terraflow. R.sim.water uses a stochastic particle-grid hybrid approach to map flow heights for a specified precipitation input. R.terraflow uses flow accumulation in multiple directions to estimate steady-state discharge.

The results of the comparison (Fig. 8), show that our presented method correctly implements water dynamics, including pressure-driven flow paths. Both r.sim.water and r.terraflow are limited by their flow paths, and show significantly longer computation time.

Other methods in the literature that focus on fast flood simulation show distinct limitations compared to the FastFlood method. Cellular automata can be leveraged for fast approximate flood height maps, and show a reduction in simulation time between 250 and 1100 times (Jamali et al., 2019). However, these methods are only partially based on physical principles, and require additional empirical constants to be calibrated. Because if this, they cannot be used as easily in alternative

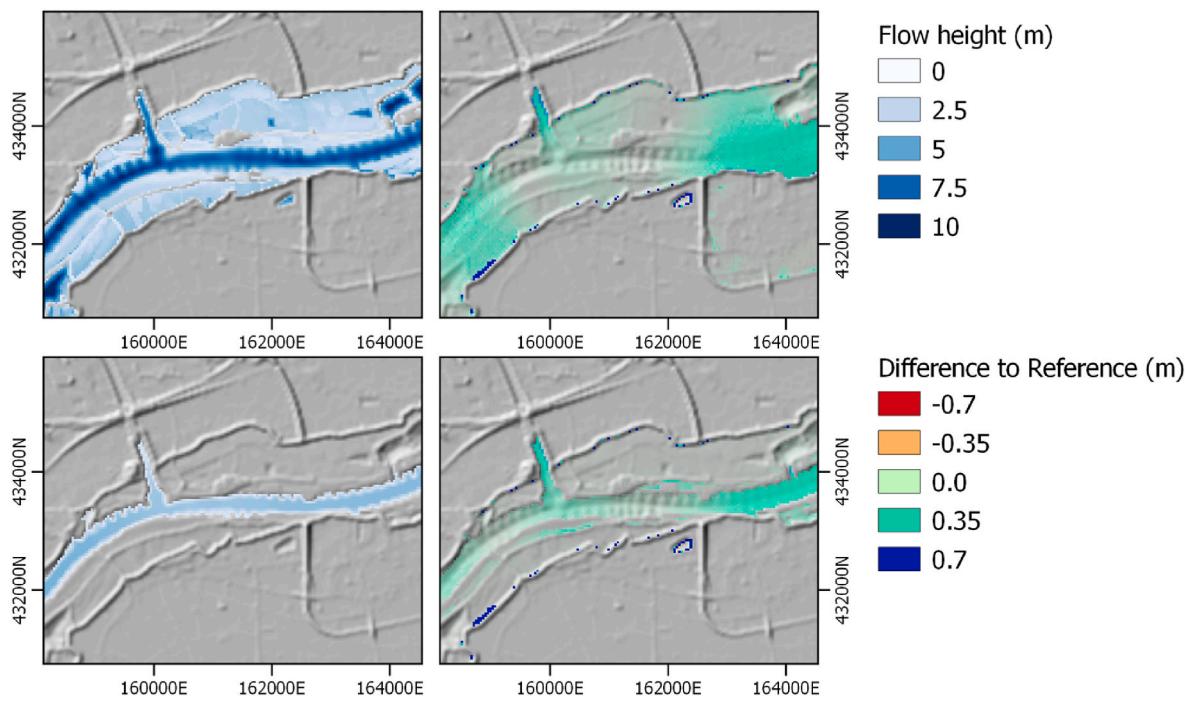


Fig. 9. Comparison of simulated flow heights in the Waal river in the Netherlands for high discharge (top, 1200 m³/s) and low discharge (bottom, 240 m³/s).

scenario's where physical parameters of the landscape change. Additionally, cellular automata have not been applied to flash floods and riverine flooding, and have not been shown to work well with precipitation as a water source. Machine-learning based methods have similarly shown potential for fast flood simulation by learning networks based on full simulation results (Kabir et al., 2020). Currently, extensive training on large datasets of fully-simulated scenarios is required. After training, these methods show a reduction of simulation time similar to our method, but do not show the general applicability (Chu et al., 2020). These methods can not automatically adapt to new scenarios (e.g. implementation of levee-system, altered land use or application in another river-system). Such changes require expensive retraining of the neural networks.

In application to river systems, the method shows high accuracy in predicting spatial water heights in channel topography. For both high and low discharge in the Waal river, the new method predicts flow heights with a total flow height deviation of 4.6% for the high flow scenario and 4.3 percent for the low flow scenario (Fig. 9).

3.3. Discussion

The OpenLISEM FastFlood model is based on various assumptions that require thorough consideration in case of application, such as the invertibility of flow accumulation to flow heights. Despite these assumptions, the over-all performance of the method is excellent compared to a state-of-the-art implementation of full dynamic flow simulation. In particular in consideration of the computation time, which was on average decreased by a factor 1514. The results of the simulation are limited as a FastFlood method does not produce full dynamic output. However, the balance between quality, flexibility and compute time can be expected to benefit several common use-cases. The provided output (e.g. peak flow height, peak flow velocity, flood arrival time), are sufficient for a large portion of flood hazard studies. In hazard and risk assessment, estimation of annual risk is often directly carried out using peak flow heights, peak flow velocities, and vulnerability functions. It does remain important to note that the aim of the FastFlood method is not to provide the most accurate flood hazard prediction. However, as long as the validity of the underlying assumptions are met,

such as for our case studies, the application to risk reduction planning or early warning can be considered.

The computational time of the methods on consumer-level pc-hardware is an indication of new potential for flood hazard decision making. For areas smaller than 50km² at 10 m resolution, computational times are smaller than 10 s even on common laptops. This is fast enough that a web-platform or application could provide some interactivity to users, with fast feedback on the effect of certain risk reduction alternatives in flood characteristics, allowing for fast iteration and efficient workflows. For many of the input parameters, there is no need to recalculate the entire simulation. Instead, near-instant results might be obtained in case of changes to surface roughness or event duration. Additionally, results from similarly-sized events can be used as preconditioning input for the model, and reduce compute time more. Further potential is visible in the field of early-warning systems. Running spatial rainfall input through flood simulations, taking into account ensembles of uncertain weather prediction, was a huge bottleneck but might now be possible (Bhola et al., 2018). Finally, probabilistic flood hazard and risk assessment, often requiring thousands of simulations, can be sped up to become practically applicable. Many alternative scenario's including altered land cover, reservoirs, flow barriers and climate change can be simulated in a fraction of the time.

4. Conclusions

Our FastFlood model has the potential to alter the field of flood modelling. It shows two orders of magnitude of increase in simulation speed, while obtaining highly similar accuracy as full dynamic models in our presented case studies. With the reported increase in simulation speed, many avenues of research and application might be unlocked, from ensemble modelling for uncertainty, user-interactivity in web interfaces, or real-time modelling of large areas in early warning systems. Beyond that, the presented method still shows several points of potential improvement that the wider flood community might use to yield further improvements, such as the influence of water arrival time on peak flows and a detailed multi-area study in the scheme for partial steady state compensation.

Software availability

Name of the software: FastFlood
 Developer: Bastian van den Bout.
 Contact Information: b.vandenbout@utwente.nl.
 Year first available: 2023.
 Programming language: c++/Javascript.
 Cost: free.
 Licence: GPL-3 licence (interface and LISEM algorithms only)
 Software availability: www.fastflood.org for the free, implemented simulation platform. Source code for the algorithms and the test datasets are part of the LISEM project, www.lisemmodel.com and <https://github.com/bastianvandenbout/LISEM>.

Appendix A

Fast Sweeping Method for geo-applications

The Fast Sweeping Method (FSM) is a numerical technique originally developed for usage with the Eikonal equation. The Eikonal equation described the mathematical condition that might be solved to obtain distance-fields.

$$|\nabla u(x)| = 1$$

20

where u is the distance field.

Solving such a relationship non-trivial due to the presence of the absolute sign. A naïve solution works as follows (F. 10)

LET S BE SOLUTION ARRAY (NxN), INITIALLY LARGE NUMBER WHILE(CHANGES) CHANGES = FALSE FOR EVERY CELL GET SOLUTIONS FOR ALL NEIGHBORS IF NEIGHBOR CELL WITH SMALLER DISTANCE SNEW = F(SMALLER NEIGHBOR DISTANCES) IF SNEW < OLD S[cell] = SNEW CHANGES = TRUE

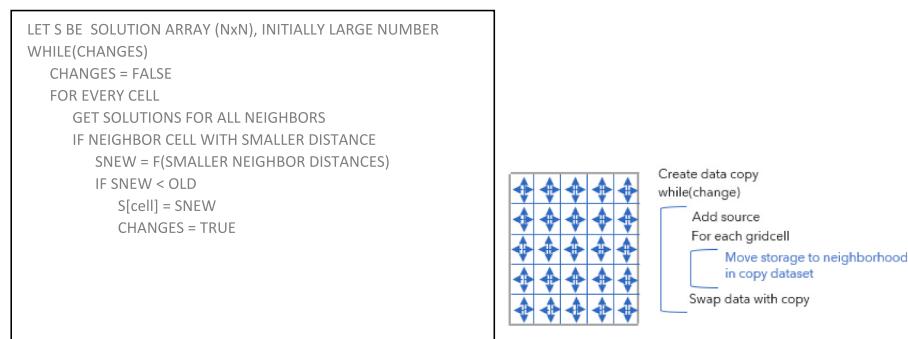


Fig. 10. Pseudo-code representation of a straight-forward solution to the Eikonal equation. F is a function that calculates the new distance solution from the values of all smaller neighbors.

This solution is conceptually simple, but computationally intensive. Each iteration of the ‘while’ loop, the solution only advances approximately one gridcell. This means that the algorithm becomes O (n^3) efficient, as the largest distance between any source point and a gridcell often depends on the length of the grid. Furthermore, each time it visits a gridcell, it must read all neighboring values.

Zhao (2005) realized a critical insight, the solution is spatially one-sided. The solution for a particular grid-cell depends only on those gridcells which have a final solution that is smaller (or closer to the source points). If we could visit the gridcells in order from small to greater distance, we could calculate the final solution in a single iteration. The fast sweeping method leverages this concept as follows (F. 11).

LET S BE SOLUTION ARRAY (NxN), INITIALLY LARGE NUMBER WHILE(CHANGES) CHANGES = FALSE FOR EACH DIRECTION (DOWN,LEFT, RIGHT,UP) FOR EVERY CELL, DIRECTIONALLY GET SOLUTIONS FOR ALL NEIGHBORS IF NEIGHBOR CELL WITH SMALLER DISTANCE SNEW = F(SMALLER NEIGHBOR DISTANCES) IF SNEW < OLD S[cell] = SNEW CHANGES = TRUE

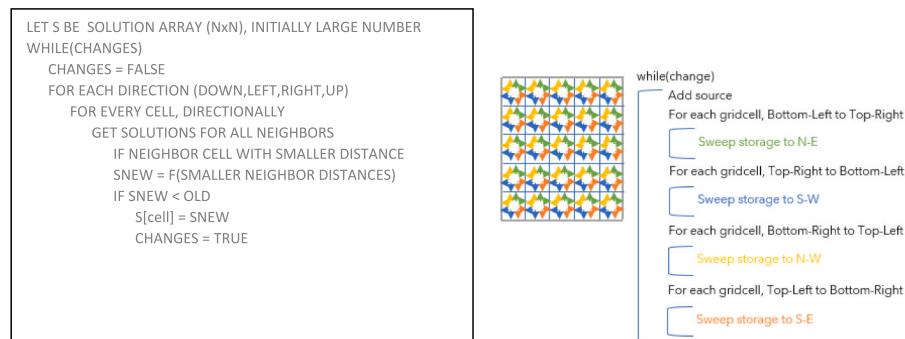


Fig. 11. Pseudo-code for a fast sweeping solution to the Eikonal equation.

By visiting the grid cells in each directionality of the raster (Down, Up, Left and Right), we use the natural order of distances to optimize the order in which the grid cells are visited.

Many others have discussed this optimization, and have both compared, improved, parallelized it for various use-cases (Zhao, 2007; Qian et al., 2007; Detrixhe et al., 2013; Bak et al., 2010).

DEM Correction

We employ the Fast Sweeping method for hydrological correction of the Digital Elevation Model (DEM). DEMs are two-dimensional rasters containing gridded surface elevation. Due to sensor noise or sampling issues, these rasterized datasets often exhibit undesirable features. One of these is local depressions. Localized minima in the surface represented by the DEM. In reality, localized minima prohibit water flow until they are filled, often forming lakes. In practice, such as the Grande-Bay catchment in Dominica, no real local depressions exist at a 10-m scale. All of the depressions (often a single pixel in size, and 112 in total) are because of small issues with the elevation data. To improve the quality of flow simulation, a correction is then applied to fill these depressions that will restore the DEM to be hydrologically correct.

The classical algorithm for hydrological correction constructs a drainage network, a connect graph of all the gridcells, where each gridcell connects to the lowest neighbor (Mark, 1984). Local depressions have at least a single pixel without a smaller neighbor, and therefore become disconnected from the larger graph. These are finally reconnected by routing the water to the lowest outflow point. This solution is robust, but due to the large sizes of the graphs, computationally demanding.

Our solution employs the same conceptual trick as the Fast Sweeping Method. Imagine a slice of terrain, which contains some local depression. Analogous to the Fast Sweeping method for the Eikonal equation, we would prefer to end with a solution that is monotonically increasing away from the source (our elevation at the boundary of the grid). Thus, we can apply a function that does not provide a distance increase, but elevation increase based on modified slope

$$S_{mod}^{i,j} = \max(\delta, S(x)) \quad 21$$

$$|\nabla u(x)| = S_{mod}(x) \quad 22$$

where $S_{mod}(x)$ is a modified slope function that replicates a hydro-correction elevation model when used in the eikonal equation, $S(x)$ is the local slope and δ is the minimum increase in elevation per gridcell that the final hydro-corrected elevation dataset will have (δ must be some non-negative constant).

We can set the gridcells at the boundary of our raster as source point values with known elevation (as done previously for source points with known zero distance). Solving this system gives us a corrected elevation model.

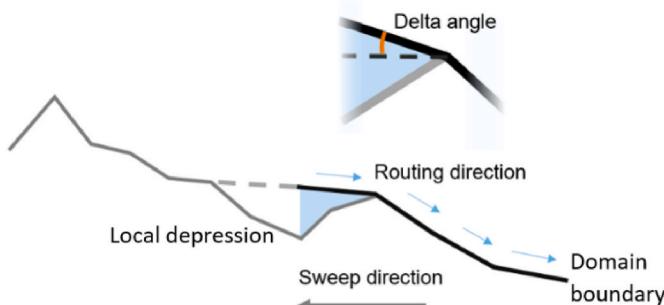


Fig. 12. A visual representation of how the fast sweeping method can be used to hydro-correct an elevation model.

Additionally, with the constant δ that was introduced, the minimum increase in elevation per grid-cell can be set, which will allow for automatic determination of flow direction to the nearest outflow point.

Multi-directional Drainage Network

A multi-directional flow network is a field of directional values, whereby each location in the dataset contains a flow direction. Each of these flow directions link the entire space into a single linked network.

Using standard methods for hydro-correction, the local depressions are filled with a single value, resulting in zero gradient. With our developed method, the new elevation raster is monotonically increased when starting at the edge of the model domain. As a result, the local gradient of the elevation field always points in the direction of some neighboring grid cell, which necessarily has a lower elevation. As this is true for all gridcells, except those at the boundary, the gradients of the hydro-corrected elevation model must form a valid multi-directional flow network.

$$f_x = \frac{\frac{dz}{dx}}{\sqrt{\frac{dz^2}{dx^2} + \frac{dz^2}{dy^2}}}, f_y = \frac{\frac{dz}{dy}}{\sqrt{\frac{dz^2}{dx^2} + \frac{dz^2}{dy^2}}} \quad 23$$

where f_x and f_y are

The resulting flow network can then be used to accumulate material through 2D space. Again, here we use the sweeping method to propagate the material, as shown in Figure 13

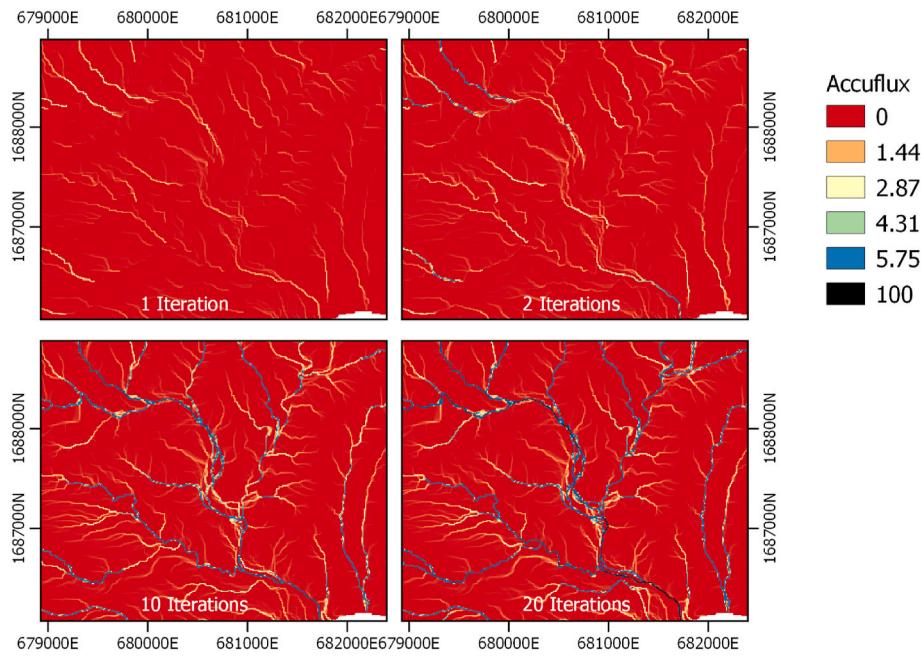


Fig. 13. The results for the Grande-Bay study case on Dominica. Instead of the usual 800 iterations to complete a flow accumulation, only 25 are required to obtain the final results.

D4 Drainage Network

Multi-directional flow networks can be useful in a variety of applications, but in some cases a D4 network can be beneficial. The multidirectional network can be partially or completely converted by selecting the neighboring grid cell in the direction of steepest descent.

References

- Ahmad, M., Al Mehedi, M.A., Yazdan, M.M.S., Kumar, R., 2022. Development of machine learning flood model using artificial neural network (ANN) at var river. *Liquids* 2 (3), 147–160.
- Alfieri, L., Burek, P., Dutra, E., Krzeminski, B., Muraro, D., Thielen, J., Pappenberger, F., 2013. GloFAS-global ensemble streamflow forecasting and flood early warning. *Hydrol. Earth Syst. Sci.* 17 (3), 1161–1175.
- Alkema, D., Middelkoop, H., 2005. The influence of floodplain compartmentalization on flood risk within the Rhine–Meuse Delta. *Nat. Hazards* 36 (1–2), 125–145.
- Arge, L., Toma, L., Vitter, J.S., 2001. I/O-efficient algorithms for problems on grid-based terrains. *J. Exp. Algorithmics* 6, 1–es.
- Augustin, I.C., 2021. Hydrologic Modeling of a Small Wetland Complex to Inform Estimates of Phosphorus Retention.
- Bak, S., McLaughlin, J., Renzi, D., 2010. Some improvements for the fast sweeping method. *SIAM J. Sci. Comput.* 32 (5), 2853–2874.
- Berkhahn, S., Fuchs, L., Neuweiler, I., 2019. An ensemble neural network model for real-time prediction of urban floods. *J. Hydrol.* 575, 743–754.
- Bhola, P.K., Leandro, J., Disse, M., 2018. Framework for offline flood inundation forecasts for two-dimensional hydrodynamic models. *Geosciences* 8 (9), 346.
- Borga, M., Boscolo, P., Zanon, F., Sangati, M., 2007. Hydrometeorological analysis of the 29 august 2003 flash flood in the eastern Italian alps. *J. Hydrometeorol.* 8 (5), 1049–1067.
- Bout, B., Jetten, V., 2018. The validity of flow approximations when simulating catchment-integrated flash floods. *J. Hydrol.* 556, 674–688.
- Briones, F., 2019. The endless hurricane: documenting life in the shelters, after Maria hit Dominica. *Disaster Prev. Manag.* Int. J. 28 (5), 616–622.
- Brunner, G.W., 2002. Hec-ras (river analysis system). In: North American Water and Environment Congress & Destructive Water. ASCE, pp. 3782–3787.
- Chu, H., Wu, W., Wang, Q., Nathan, R., Wei, J., 2020. An ANN-based emulation modelling framework for flood inundation modelling: application, challenges and future directions. *Environ. Model. Software* 124, 104587.
- Croissant, T., Lague, D., Davy, P., 2014. Calibration of the 2D hydrodynamic model floods and implications of distributed friction on sediment transport capacity. *AGU Fall Meeting Abstracts 2014*, EP51E–3565.
- de Saint-Venant, B., 1871. Theory of unsteady water flow, with application to river floods and to propagation of tides in river channels. *French Academy of Science* 73, 148–154.
- Delestre, O., Cordier, S., Darboux, F., Du, M., James, F., Laguerre, C., Planchon, O., 2014. FullSWOF: a software for overland flow simulation. In: *Advances in Hydroinformatics*. Springer, pp. 221–231.
- Deltares, 2014a. SOBEK User Manual. Delft, the Netherlands. http://content.oss.deltares.nl/delft3d/manuals/SOBEK_User_Manual.pdf.
- Deltares, 2014b. Delft3D-Flow, Simulation of Multi-Dimensional Hydrodynamic Flows and Transport Phenomena, Including Sediments, User Manual, p. 684. Version 3.15.34158, May 2014.
- Deltares, 2015. Delft3D Flexible Mesh Suite, D-Flow FM in Delta Shell, User Manual, Version 1.1.148, p. 376. November 2015.
- Detrixhe, M., Gibou, F., Min, C., 2013. A parallel fast sweeping method for the Eikonal equation. *J. Comput. Phys.* 237, 46–55.
- De Roo, A.P.J., Wesseling, C.G., Van Deursen, W.P.A., 2000. Physically based river basin modelling within a GIS: the LISFLOOD model. *Hydrol. Process.* 14 (11–12), 1981–1992.
- DHI, 2014. MIKE 21 Flow Model FM Hydrodynamic Module, p. 134. User Manual. August 2014.
- Gibson, M.J., Savic, D.A., Djordjevic, S., Chen, A.S., Fraser, S., Watson, T., 2016. Accuracy and computational efficiency of 2D urban surface flood modelling based on cellular automata. *Procedia Eng.* 154, 801–810.
- Gironas, J., Roesner, L.A., Rossman, L.A., Davis, J., 2010. A new applications manual for the storm water management model(SWMM). *Environ. Model. Software* 25 (6), 813–814.
- Hofmann, J., Schütttrumpf, H., 2021. floodGAN: using deep adversarial learning to predict pluvial flooding in real time. *Water* 13 (16), 2255.
- itwh, 2017. HYSTEM-EXTRAN 2D Modellbeschreibung (HYSTEM-EXTRAN 2D model description). In: Institut für technisch-wissenschaftliche Hydrologie GmbH. Hannover 2017.
- Jamali, B., Bach, P.M., Cunningham, L., Deletic, A., 2019. A Cellular Automata fast flood evaluation (CA-ffé) model. *Water Resour. Res.* 55 (6), 4936–4953.
- Jasak, H., Jemcov, A., Tukovic, Z., 2007. OpenFOAM: a C++ library for complex physics simulations. In: International Workshop on Coupled Methods in Numerical Dynamics. IUC Dubrovnik Croatia, pp. 1–20, 1000.
- Kabir, S., Patidar, S., Xia, X., Liang, Q., Neal, J., Pender, G., 2020. A deep convolutional neural network model for rapid prediction of fluvial flood inundation. *J. Hydrol.* 590, 125481.
- Kalyanapu, A.J., Shankar, S., Pardyjak, E.R., Judi, D.R., Burian, S.J., 2011. Assessment of GPU computational enhancement to a 2D flood model. *Environ. Model. Software* 26 (8), 1009–1016.
- Karim, F., Armin, M.A., Ahmed-Aristizabal, D., Tychsen-Smith, L., Petersson, L., 2023. A review of hydrodynamic and machine learning approaches for flood inundation modeling. *Water* 15 (3), 566.
- Leijnse, T., Nederhoff, K., Van Dongeren, A., McCall, R.T., Van Ormondt, M., 2020. Improving computational efficiency of compound flooding simulations: the SFINCS model with subgrid features. *AGU Fall Meeting Abstracts 2020*, NH022–N02006.

- Liang, Q., Borthwick, A.G.L., Stelling, G., 2004. Simulation of dam-and dyke-break hydrodynamics on dynamically adaptive quadtree grids. *Int. J. Numer. Methods Fluid.* 46 (2), 127–162.
- Liang, Q., Du, G., Hall, J.W., Borthwick, A.G., 2008. Flood inundation modeling with an adaptive quadtree grid shallow water equation solver. *J. Hydraul. Eng.* 134 (11), 1603–1610.
- Liu, Q., Qin, Y., Li, G., 2018. Fast simulation of large-scale floods based on GPU parallel computing. *Water* 10 (5), 589.
- Mark, D.M., 1984. Part 4: mathematical, algorithmic and data structure issues: automated detection of drainage networks from digital elevation models. *Cartographica: Int. J. Geogr. Inf. Geovisualization* 21 (2–3), 168–178.
- McGlade, J., Bankoff, G., Abrahams, J., Cooper-Knock, S.J., Cotecchia, F., Desanker, P., ... & Wood, M. (2019). Global assessment report on disaster risk reduction 2019.
- Miller, J.E., 1984. *Basic Concepts of Kinematic-Wave Models* (2330-7102) (Retrieved from).
- Mitasova, H., Thaxton, C., Hofierka, J., McLaughlin, R., Moore, A., Mitas, L., 2004. Path sampling method for modeling overland water flow, sediment transport, and short term terrain evolution in Open Source GIS. *Dev. Water Sci.* 55, 1479–1490 (Elsevier).
- Ming, X., Liang, Q., Xia, X., Li, D., Fowler, H.J., 2020. Real-time flood forecasting based on a high-performance 2-D hydrodynamic model and numerical weather predictions. *Water Resour. Res.* 56 (7), e2019WR025583.
- Morales-Hernández, M., Sharif, M.B., Kalyanapu, A., Ghafoor, S.K., Dullo, T.T., Gangrade, S., et al., 2021. TRITON: a Multi-GPU open source 2D hydrodynamic flood model. *Environ. Model. Software* 141, 105034.
- Mosavi, A., Ozturk, P., Chau, K.-w., 2018. Flood prediction using machine learning models: literature review. *Water* 10 (11), 1536.
- Peckham, S.D., Stoica, M., Jafarov, E., Endalamaw, A., Bolton, W.R., 2017. Reproducible, component-based modeling with TopoFlow, a spatial hydrologic modeling toolkit. *Earth Space Sci.* 4 (6), 377–394.
- Qian, J., Zhang, Y.T., Zhao, H.K., 2007. A fast sweeping method for static convex Hamilton-Jacobi equations. *J. Sci. Comput.* 31, 237–271.
- Rahman, A., Haddad, K., Zaman, M., Kuczera, G., Weinmann, P., 2011. Design flood estimation in ungauged catchments: a comparison between the probabilistic rational method and quantile regression technique for NSW. *Aust. J. Water Resour.* 14 (2), 127–139.
- Raissi, M., Yazdani, A., Karniadakis, G.E., 2020. Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* 367 (6481), 1026–1030.
- Rengers, F.K., McGuire, L.A., Kean, J.W., Staley, D.M., Hobley, D.E.J., 2016. Model simulations of flood and debris flow timing in steep catchments after wildfire. *Water Resour. Res.* 52 (8), 6041–6061.
- Sanders, B.F., 2017. Hydrodynamic modeling of urban flood flows and disaster risk reduction. In: Oxford Research Encyclopedia of Natural Hazard Science.
- Schaake Jr., J.C., Geyer, J.C., Knapp, J.W., 1967. Experimental examination of the rational method. *J. Hydraul. Div.* 93 (6), 353–370.
- Steer, P., Davy, P., Lague, D., Bernard, T., Feliciano, H., 2022. May). FastFlood: a fast and simple 2D hydrodynamic or hydrostatic numerical solution to river flow in landscape evolution models. In: EGU General Assembly Conference Abstracts, pp. EGU22-3142.
- Stelling, G.S., Verwey, A., 2006. Numerical Flood Simulation. *Encyclopedia of hydrological sciences*.
- SCS, U., 1985. National Engineering Handbook, Section 4: Hydrology. US Soil Conservation Service, USDA, Washington, DC.
- Syme, W.J., 2001. February). TUFLOW-Two & Onedimensional unsteady flow Software for rivers, estuaries and coastal waters. In: IEAust Water Panel Seminar and Workshop on 2d Flood Modelling (Sydney).
- Tavakkol, S., Lynett, P., 2017. Celeris: a GPU-accelerated open source software with a Boussinesq-type wave solver for real-time interactive simulation and visualization. *Comput. Phys. Commun.* 217, 117–127.
- Te Chow, V., Maidment, D., Mays, L., 1962. Applied hydrology. *J. Eng. Educ.* 308, 1959.
- Tucker, G.E., Hutton, E.W., Piper, M.D., Campforts, B., Gan, T., Barnhart, K.R., et al., 2022. CSDMS: a community platform for numerical modeling of Earth surface processes. *Geosci. Model Dev. (GMD)* 15 (4), 1413–1439.
- Van Den Bout, B., Jetten, V., 2020. Catchment-scale multi-process modeling with local time stepping. *Environ. Earth Sci.* 79 (8), 1–15.
- van Westen, C., 2019. Procedure for the Development of Multi-hazard Risk Profiles at the District Level in Tajikistan.
- Williams, J.R., Jones, C.A., Dyke, P.T., 1984. A modeling approach to determining the relationship between erosion and soil productivity. *Transactions of the ASAE* 27 (1), 129–144.
- Wolock, D.M., 1993. Simulating the variable-source-area concept of streamflow generation with the watershed model TOPMODEL. *Water-Resources Investigations Report* 93, 4124.
- Yamazaki, D., 2014. The Global Hydrodynamic Model CaMa-Flood. JAMSTEC–Japan Agency for Marine-Earth Science and Technology, version 3.6. 2. http://hydro.iis.u-tokyo.ac.jp/~yamadai/CaMa-Flood_v3.6.
- Yu, D., Lane, S.N., 2006. Urban fluvial flood modelling using a two-dimensional diffusion-wave treatment, part 2: development of a sub-grid-scale treatment. *Hydrol. Process.: Int. J.* 20 (7), 1567–1583.
- Zhao, H., 2005. A fast sweeping method for eikonal equations. *Math. Comput.* 74 (250), 603–627.
- Zhao, H., 2007. Parallel implementations of the fast sweeping method. *J. Comput. Math.* 421–429.