GraphFlood 1.0: an efficient algorithm to approximate 2D hydrodynamics for Landscape Evolution Models

Boris Gailleton[1], Philippe Steer[1], Philippe Davy[1], Wolfgang Schwanghart[2], and Thomas Bernard[1]

[1]Université de Rennes, CNRS, Géosciences Rennes - UMR 6118, F-35000 Rennes, France
[2]University of Potsdam, Potsdam, Germany

Correspondence: Boris Gailleton (boris.gailleton@univ-rennes.fr)

Abstract. Computing hydrological fluxes at the Earth's surface is crucial for landscape evolution models, topographic analysis, and geographic information systems. However, existing formalisms, like single or multiple flow algorithms, often rely on ad-hoc rules based on local topographic slope and drainage area, neglecting the physics of water flow. While more physics-oriented solutions offer accuracy (e.g. shallow water equations), their computational costs limit their use in term of spatial and temporal scales. In this conrtibution, we introduce GraphFlood, a novel and efficient iterative method for computing river depth and water discharge in 2D on a digital elevation model (DEM). Leveraging the Directed Acyclic Graph ~~(DAG)~~ structure of surface water flow, GraphFlood iteratively solves the 2D shallow water equations. This algorithm aims to find the correct hydraulic surface by balancing discharge input and output over the topography. At each iteration, we employ fast ~~DAG~~ graph theory algorithms to calculate flow accumulation on the hydraulic surface, approximating discharge input. Discharge output is then computed using the Manning flow resistance equation, similar to the River.lab model (Davy and Lague, 2009). Iteratively, the divergence of discharges increments flow depth until reaching a stationary state. This algorithm can also solve for flood wave propagation by approximating the input discharge function of the immediate upstream neighbours. We validate water depths obtained with the stationary solution against analytical solutions for rectangular channels and the River.lab and Caesar Lisflood models for natural DEMs. GraphFlood demonstrates significant computational advantages over previous hydrodynamic models, with approximately a 10-fold speed-up compared to the River.lab model (Davy and Lague, 2009). Additionally, its computational time scales slightly more than linearly with the number of cells, making it suitable for large DEMs exceeding 10^6 - 10^8 cells. We demonstrate the versatility of GraphFlood in integrating realistic hydrology into various topographic and morphometric analyses, including channel width measurement, inundation pattern delineation, floodplain delineation, and the classification of hillslope, colluvial, and fluvial domains. Furthermore, we discuss its integration potential in landscape evolution models, highlighting its simplicity of implementation and computational efficiency.

1 Introduction

River dynamics encompass key processes of landscape evolution at different temporal and spatial scales. Rivers transfer sediments downstream, they control the baselevel of hillslopes, and set the pace of denudation rates (e.g. Clubb et al., 2019). Modelling landscape evolution and the development of fluvial landforms, in particular, thus requires a sound representation of how rivers erode, transport and deposit material. As landscape evolution models are used to simulate the dynamics of topography over 10^5-10^7 years and at continental scales (Salles et al., 2023), accounting for short-term processes (e.g. daily variations of discharge, flood) at local scales remains a methodological and numerical challenge. Simulating flow in open environments in two or three dimensions requires sophisticated numerical methods which are computationally demanding and which are thus mostly inapt for the challenge of simulating landscape evolution over geological time scales (Davy et al., 2017). Instead, a common approach to model water flow across landscapes consists in applying the single or multiple flow algorithms (e.g. Tarboton, 1997; O'Callaghan and Mark, 1984). These techniques route water along topographic gradients towards one or multiple neighboring pixels in a DEM and approximate discharge by drainage area weighted by precipitation rates (Adams et al., 2020). The approximation of steady flow using drainage-area based discharge has been the cornerstone of integrating hydrodynamics in long-term erosion laws (e.g. Whipple and Tucker, 1999). This approach has the compelling advantage that it reduces flow patterns to a network of flow lines, and has been widely used to establish empirical scaling laws relating drainage area to channel steepness and uplift (Wobus et al., 2006), or to unravel landscape evolution from the planform shape of the river networks (Schumm et al., 2000; Willett et al., 2014). Moreover, these methods rely on efficient algorithms, which leverage graph theory to compute drainage area (e.g. Braun and Willett, 2013; Anand et al., 2020), flow across complex terrain (e.g. Barnes et al., 2014; Cordonnier et al., 2018; Barnes et al., 2021; Schwanghart and Scherler, 2017) or geomorphological metrics (e.g. Gailleton et al., 2019; Mudd et al., 2018; Grieve et al., 2018; Schwanghart et al., 2021). In particular the ~~Single Flow Direction (SFD)~~ single flow direction algorithm is thus the numerical workhorse for simulation software for landscape evolution (Hergarten, 2020; Braun and Sambridge, 1997; Willgoose et al., 1994; Campforts et al., 2017; Braun and Willett, 2013, e.g.) and numerical frameworks for quantitative geomorphology (e.g. Barnhart et al., 2020; Gailleton et al., 2023; Schwanghart and Scherler, 2014; Mudd et al., 2019).

However, reducing rivers to lines in landscape evolution models may overtly simplify the dynamics and feedbacks of fluvial processes (Armitage, 2019). In fact, the response of rivers to climate variability, tectonic movements or baselevel changes is more varied than the simple propagation of a wave of vertical changes through 1D network of lines. For example, changes in boundary conditions cause rivers to adjust their width (e.g. Dunne and Jerolmack, 2020; Baynes et al., 2022) and their planform flow pattern (e.g. Schuurman et al., 2013), both of which feedback on sediment fluxes (e.g. Davy and Lague, 2009). In addition, the past decade has seen the rising availability of high resolution lidar-derived DEMs (<1 m resolution). This means, however, that for a variety of geomorphological applications (e.g. Steer et al., 2022; Stammberger et al., 2024) rivers cannot be realistically represented by one pixel-wide paths (Figure 1 d). Several recent studies demonstrate the advantages of integrating 2D hydrodynamics to inform the study of landforms~~(Costabile et al., 2019; Costabile and Costanzo, 2021; Bernard et al., 2022)~~, even on long timescales ~~. Here, we present~~ (Costabile et al., 2019; Costabile and Costanzo, 2021; Bernard et al., 2022). However, these methods are difficult to upscale for more generalized analysis due to their reliance on closed-source software and are not straightforward for non-specialists to adapt and reuse.

Here we present GraphFlood, a new and efficient method, based on graph theory and finite differences, to fill this methodological gap and allow the efficient approximation of 2D hydrodynamics on high resolution topography and/or longer term landscape evolution model.

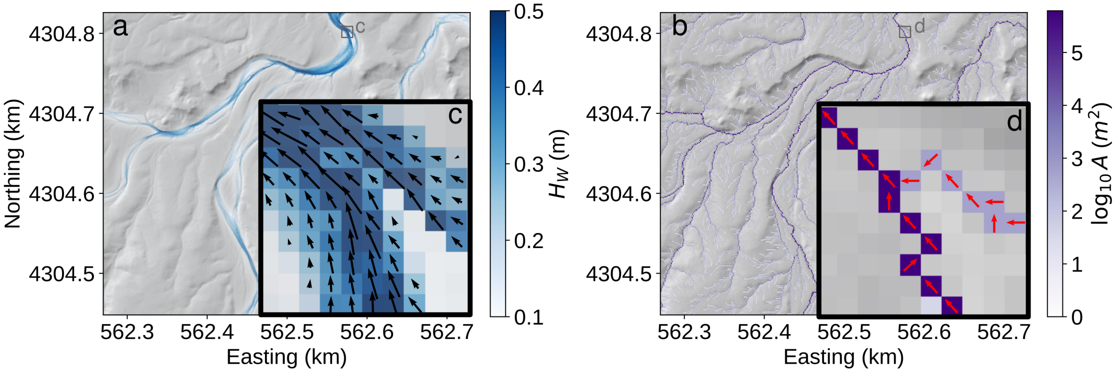


Figure 1. Comparison between water flows approximated with GraphFlood (a and c), calculating flow depth and discharge vectors, and with a classic drainage area based method (D8 Steepest descent route) (b and d). The panels detail a channel junction and highlight how GraphFlood models flow patterns and how these differ from one-pixel wide flows derived from the D8 algorithm. The black arrows on panel c represents the flow velocity vectors and are scaled to their magnitude. For panel d, the red arrows represent the D8 flow direction. h is the flow depth and A drainage area.

1.1 Existing solutions

A range of numerical models incorporating 2D to 3D hydrodynamics to study river systems and their morphological evolution exists, with widely different methods and levels of complexity, depending on the temporal and spatial scales of interest. Finite-element models are commonly used for reach-scale models, such as DELFT3D (Roelvink and Banning, 1995), HEC RAS (Brunner, 2002), BASEMENT (Vanzo et al., 2021) or TELEMAC (Villaret et al., 2013). These models are designed for simulating the evolution of fluvial landforms over scales of 1-100 km and over 1-100 years, and therefore fall outside the scope of this study.

Bates et al. (2010) developed a two-dimensional hydrodynamic model Lisflood-FP, solving for the 2D ~~Shallow Water Equations (SWE)~~shallow water equations. Their cellular-automata approach has been successfully incorporated in the landscape evolution model CAESAR Coulthard et al. (2013) to simulate reach-to-catchment scale fluvial hydro-morphodynamics (e.g. Yu and Coulthard, 2015; Liu and Coulthard, 2015; Coulthard and Van De Wiel, 2017). Lisflood-FP adopts a finite difference scheme on the bidirectional water fluxes between pixel. While it has been applied to catchment scales over potentially thousands of years (Liu and Coulthard, 2017, e.g.), its potential for longer-term and larger-scale studies remains hampered by the physics behind which explicitly ~~simulates wave propagation. Indeed~~and gradually transfer water from a cell to another. Specifically, any upstream change ~~of~~ in runoff input (e.g., precipitation) ~~needs to~~ must be gradually propagated downstream one pixel per computational time step. ~~While modelling~~ Although modeling non-steady flows is ~~important~~ crucial for simulating transient responses to individual storm events ~~(e.g. Van De Wiel and Coulthard, 2010), it represents~~ (e.g., Van De Wiel and Coulthard, 2010), it becomes a limiting factor ~~aiming for simulating longer time~~ when simulating over longer timescales or larger scales. Bates et al. (2010) and subsequent improvements by de Almeida et al. (2012) have been utilized in other landscape evolution framework (e.g. Barnhart et al., 2020) following the same principle.

An alternative to propagating wave is to focus on the stationary state of the river network (i.e., in equilibrium with the input field of runoff). The main challenge in estimating efficiently the stationary solution lies in spreading the flow to its equilibrium field. The latter depends on the final geometry of the hydraulic surface, which cannot be deduced from the geometry of the terrain alone. To address this point, Davy et al. (2017) developed an efficient particle-based solution to solve the ~~SWE~~shallow water equations. In this approach, precipitons (i.e. ~~,~~ elementary volumes of water) are ~~dropped on the landscapes and propagate following~~ released onto the landscape. Each precipiton follows a stochastic path down the hydraulic surface~~. Precipitons increase the water height along their path, bypassing the need to to propagate flood waves gradually~~, increasing the flow depth by a constant value, representing the total water influx. This increase is balanced by a decrease in flow depth, calculated using Manning's equations and where each precipiton has its own timestamp. The frequency at which precipitons pass through a cell determines the ~~amount of water received by this cell, balanced by a decrease of flow depth based on discharge calculated with Manning's equations~~final, stationary flow depth field. This method is ~~efficient in terms of computation time (Davy et al., 2017), and in particular in the fluvial domain having~~ computationally efficient in areas where flow converges and a high frequency of ~~precipiton passage~~precipitons passes through (e.g., fluvial valleys). However, ~~it has some physical and numerical drawbacks: i) each precipiton is on a different timeline making the isolation of snapshots through time challenging; ii) the fluvial domain receives many more precipitons than the hillslope domain, making their repeated passage numerically redundant while displaying slower convergence time on hillslopes; and iii)precipitons are independent one from another and only integrate information down their 1D flow path. A similar approach has been developed by Pelletier (2008), who~~ the efficiency and accuracy of Floodos depend on the frequency of precipiton passage. Accurately approximating the shallow water equation becomes challenging in regions with lower drainage areas or any domains where the frequency of precipiton passage is very low (e.g., flat areas, hilltops, hillslopes, smaller tributaries). This behaviour is accentuated in large DEMs, where the probability of precipiton passage is even lower.

Finally, Pelletier (2008) outlined the prototype of a highly-iterative solution that repeatedly runs ~~the MFD model on the terrain and the water surface~~a multiple flow direction flow routing a DEM (e.g. Tarboton, 1997). This process incrementally and arbitrarily increases the flow height until satisfying an equilibrium between flow depth and input discharge. ~~This approach is the starting point for our new algorithm~~However, it requires a significant number of iteration to ensure all the cells have converged to the final result.

1.2 A new solution based on graph theory

~~GraphFlood uses a novel approach to efficiently calculate the stationary solution for the whole landscape. Topography can numerically be described as a data structure where each location of a DEM~~ In this contribution, we introduce GraphFlood, a novel iterative approach that is both efficient and adaptable for solving the shallow water equations across entire landscapes. Numerically, each DEM location is linked to its ~~neighbours *via*~~ neighbors through unique directional connections~~*upstream* or *downstream*~~, either upstream or downstream. In graph theory, this ~~data structure is called a Directed Acyclic Graph (DAG)~~

~~and opens a range~~ structure is known as a directed acyclic graph, which allows for the application of efficient algorithms ~~applied to the propagation of information through a landscapes (see the review work of Heckmann et al., 2015). We leverage the DAG nature of the topography to propagate runoff through the whole landscape at every single time step using drainage area calculated on the hydraulic surface. Using the DAG structure, calculating drainage-area is very efficient and can be done in a single graph traversal following the downstream topological order (e.g. Anand et al., 2020; Braun and Willett, 2013; Gailleton et al., 2023;~~ ~~. Weighted~~ for information propagation through landscapes (e.g., Anand et al., 2020; Braun and Willett, 2013; Gailleton et al., 2023; Herga . Similar to Davy et al. (2017), GraphFlood assumes steady flow to focus on the stationary solution, meaning that flow is propagated across the landscape instantaneously. However, unlike Davy et al. (2017), whose accuracy and efficiency vary depending on the frequency of passage of discrete particles, GraphFlood leverages the graph structure to process the entire landscape in each iteration, including domains with low drainage areas. Runoff is propagated using drainage area weighted by precipitation rates, ~~drainage area determines the amount of water entering every cell of the system~~and local discharge is calculated using Manning's friction equations. At each iteration, ~~we calculate the discharge leaving the cells following a SWE, neglecting inertia (Davy et al., 2017). The balance of~~ the balance of input and output discharges ~~iteratively increments~~ is incrementally adjusted for every cell in the landscape, refining the flow depth until ~~reaching an equilibrium of the water surface~~hydraulic equilibrium is achieved. This global approach is scalable and allows for targeting larger DEMs without compromising the efficiency or accuracy of the algorithm.

In the following sections, we first describe the theory behind our method, ~~before explaining~~ then explain the algorithm and the associated finite difference scheme. ~~Different case studies are then tested~~ We then test different case studies to demonstrate the ~~potential of the methodfor flood modelling~~method's potential for flood modeling, morphometric analysis, and landscape evolution ~~modelling. Last~~modeling. Finally, we discuss the ~~limitation and next developments for~~ limitations and potential future developments of the model.

2 Theoretical background

2.1 ~~Shallow Water Equations~~

First, we outline the governing equations behind GraphFlood. We use the 2D ~~SWE~~ shallow water equations to approximate the physics of water flow in open-environment. ~~The equations are derived by integrating~~ They integrate the three-dimensional Navier-Stokes equations over the vertical dimension, assuming that the velocity field varies primarily in the horizontal direction~~, and~~ . Different variants of the shallow water equations are commonly used to model flooding beyond reach scale ~~(Bates, 2022)~~ (e.g. Bates et al., 2010; de Almeida et al., 2012; Davy et al., 2017; Bates, 2022). The 2D ~~SWEs~~ shallow water equations consist in a mass conservation equation and a momentum conservation equation. Using the notations of Davy et al. (2017), the mass conservation equation can be written:

$$\frac{\partial h}{\partial t} - \nabla \cdot (\boldsymbol{q}) = 0 \qquad (1)$$

h is the water depth in [L], t the time in [T] and q the discharge per unit width in $[\frac{L^2}{T}]$.

Neglecting inertia, (Manning et al., 1890) demonstrated that the momentum equation can be simplified into Manning's equations where flow velocity u (in $\frac{L}{T}$) is expressed as:

$$\boldsymbol{u} = \frac{h^{\alpha}}{n}\frac{\boldsymbol{s}}{\|\boldsymbol{s}\|^{\frac{1}{2}}} \tag{2}$$

where α is Manning's exponent, usually assumed equal to $\frac{2}{3}$, n is Manning's friction coefficient ~~, $xmax$ being the direction of the steepest hydraulic gradient~~and s the steepest gradient of the hydraulic surface. The hydraulic surface is $Z_h = Z + h$, the elevation of the flow depth h on the top of the topographic surface Z.

In order to insert equation 2 into equation 1, discharge per unit width and velocity are related *via* flow depth:

$$\boldsymbol{q} = \boldsymbol{u} \cdot h \tag{3}$$

Unlike similar methods (Bates et al., 2010, e.g.) or more sophisticated formulations (e.g. Brunner, 2002) incorporating additional physical elements (e.g. inertia, turbulence), our method is designed to be optimized when these components can be neglected (Davy et al., 2017). We use $\boldsymbol{Q}$ to refer to the volumetric flux in $[\frac{L^3}{T}]$ and the ~~indices $\boldsymbol{X_{in}}$ and $\boldsymbol{X_{out}}$~~ subscript $\boldsymbol{Q_{in}}$ and $\boldsymbol{Q_{out}}$ to refer respectively to quantities *entering* or *leaving* a given cell.

These equations can simulate the propagation of water through space and time dynamically, solving a transient flood wave. $\nabla \cdot \boldsymbol{q}$ is the difference between q_{in} made of q_{out} from upstream neighbours and q_{out} from the current cell to its downstream neighbours. For a constant input of q_{in} on a landscape (e.g. constant precipitation rates, fixed input discharge), the system has an equilibrium state - or stationary solution - where the water depth and hydraulic slope lead to a q_{out} balancing q_{in}. The total Q_{in} for the stationary state for a given location becomes the integration of all the source terms (e.g. precipitations, resurgence) over the drainage area upstream of a given location.

In this contribution, we refer to the *transient solution* when we seek to solve the transient propagation of Q through space and time and to the *stationary solution* when we are only interested in the equilibrated fields.

3 A graph-based iterative method

~~As stated in section 1.1, there are multiple ways to numerically solve for the SWE~~We present a numerical framework to solve the governing equations outlined in section 2. Our developed scheme applies different variants of an explicit finite difference scheme on a ~~graph (Braun and Willett, 2013; Barnhart et al., 2020; Gailleton et al., 2023)~~directed acyclic graph (see Braun and Willett, 201 . It aims to provide ~~a reasonably efficient and scalable~~ efficient solution suitable for large-scale DEMs and ~~LEMs~~landscape evolution models. Our iterative scheme is optimised for the stationary solution, but can be used for transient simulation. In the following, we detail the numerical graph structure ~~(DAG)~~ required by our method, we describe the finite difference scheme, explain the transient and stationary solutions and validate them against analytical solutions.

3.1 Numerical structure

We use the following terms adopted from graph theory (see Heckmann et al., 2015, for a comprehensive review about the use of graph theory applied to geomorphological applications): a discrete location is represented by a *node*, linked to its ~~*neighbor*~~*neighbour* nodes via *links*. The links are directed edges linking *donors* to their downstream *receivers*. In our referential donors have higher hydraulic surface ($Z + h$) than their receivers. The algorithm is compatible with any type of grid (e.g. hexagonal grid or triangular network), as long as the ~~DAG~~ directed acyclic graph structure defines the topology between the pixels or facets. Each link is characterized by a specific length ∂l representing the distance between the two neighbour nodes and a link width ∂w representing the local width. Each node represents a cell area A_c. The scheme also requires common ~~DAG algorithms: the~~ directed acyclic graph algorithms: *topological ordering*~~- an operation providing~~ , which provides a list of nodes sorted from upstream to downstream, and *sink filling*~~a method filling local minimas~~ , a method for filling local minima disconnected from the rest of the graph (e.g.~~lake~~, lakes, local noise). The ~~DAG can use both Single Flow Direction (SFD)~~ directed acyclic graph can utilize either a single flow direction topology (Braun and Willett, 2013), where each node has a single receiver (e.g., steepest descent or D8), or ~~Multiple Flow Direction (MFD) DAGs (e.g. Tarboton, 1997; Anand et al., 2020)~~a multiple flow direction directed acyclic graph where each nodes is linked to multiple receivers (e.g. Tarboton, 1997; Anand et al., 2020). This distinction is important ~~as most common operations on SFD DAGs are~~ because operations on single flow direction directed acyclic graphs are generally simpler and more efficient than ~~the MFD DAGs (e.g. Braun and Willett, 2013; Anand et al., 2020) . It~~ those on multiple flow direction directed acyclic graphs (e.g. Braun and Willett, 2013; Anand et al., 2020). However, it is worth noting ~~the latter catches~~ that the latter captures more details about flow topology and ~~tend~~ tends to increase the accuracy of the represented processes ~~(Armitage, 2019, e.g.)~~(e.g., Armitage, 2019).

In this contribution, we developed the method for regular grids. In the stationary case, we use the algorithms of Barnes et al. (2014) and Cordonnier et al. (2018) to ensure flow continuity and proceed to an initial filling of the local minimas (e.g. noise, lake). Topological sorting operations use a modified version of Braun and Willett (2013) for ~~SFD~~ single flow direction and a variant of Anand et al. (2020) for ~~MFD~~multiple flow direction. The modifications are minor changes of data structure that do not change the overall functioning while improving performance and readability (see Gailleton et al. (2023) for detailed implementations). One advantage of GraphFlood is that it can be implemented using existing computational frameworks for DEM analysis and LEM simulation (e.g. Schwanghart and Scherler, 2014; Gailleton and Mudd, 2021; Barnhart et al., 2020): the base of the algorithm only needs to calculate flow direction and topological order. A notable difference compared to existing framework is that we calculate the ~~DAG~~ directed acyclic graph using the hydraulic surface rather than the topography.

3.2 Iterative explicit finite difference scheme

We use an explicit finite difference scheme to solve equation 1. In the transient case, the numerical solution predicts flow depth change for every node i:

$$\frac{h_i^{t+1} - h_i^t}{\Delta t} = \frac{\sum\limits_{d=donors(i)} Q_{in_d} - \sum\limits_{r=receivers(i)} Q_{out_r}}{A_c} \tag{4}$$

where Q_{in_d} represent the discharge from a donor d to the node i and Q_{out_r} the discharge from the node i and a receiver r. For the latter, in the case of ~~SFD~~ single flow direction (i.e. single receiver), equation 3 becomes:

$$Q_{out_i} = \frac{\Delta W}{n} h_i^\alpha \sqrt{s_{ir}} \tag{5}$$

where i and r are respectively a given node and its single receiver and ΔW the flow width in the given direction. Because flow can only go through one link, ΔW is easy to determine. For example for our case of a regular grid, it is Δx in the y direction, Δy in the x direction and the diagonal length for the other cases. As noted by Coulthard et al. (2013), ~~MFD~~ multiple flow direction can become increasingly more complicated: multiple receivers mean ΔW "overlaps" and using the direct width of flow for each links can break the conservation of mass. Let's imagine a regular grid considering D8 neighbouring (cardinal and diagonal directions), a node that would discharge to all these directions would integrate twice the total flow width. Porting this formulation to ~~MFD~~ multiple flow direction requires then a correction factor. Equation 3 in ~~MFD DAG~~ multiple flow direction directed acyclic graph therefore becomes:

$$Q_{out_i} = \frac{C}{n} h_i^\alpha \cancel{\frac{\sum\limits_{j\,in\,receivers} s_{ij} \Delta W_{ij}}{\sqrt{S_{ijmax}}}} \frac{\sum\limits_{j\,in\,receivers} s_{ij} \Delta W_{ij}}{\sqrt{s_{ijmax}}} \tag{6}$$

where s_{ijmax} is the hydraulic slope in the direction of maximum descent.

By definition, for a given flow depth, both ~~SFD and MFD~~ single flow direction and multiple flow direction discharge should be equal. Therefore, the correction factor is:

$$C = \frac{s_{ijmax} \Delta W_{ijmax}}{\sum\limits_{j\,in\,receivers} S_{ij} \Delta W_{ij}} \tag{7}$$

~~The magnitude of Q_{out} flux is the same for MFD and SFD schemes, but the correction factor states the discharge need to be parted to multiple receivers proportional to $S_{ij} \Delta W_{ij}$~~ and equation 6 is then equivalent to equation 5, and the difference between the two solvers only remains in the partitioning of Q_{in} which becomes proportional to $s_{ij} \Delta W_{ij}$.

Both transient and stationary solutions follow that scheme to calculate the output discharge, the difference is the calculation Q_{in} for all nodes. The overall process is outlined on algorithm 1.

3.3 Transient solution

For the transient solution, $Q_{in_{di}}$ is $Q_{out_{di}}$ calculated between the donor and this node plus an eventual local external Q_{in} source term (*e.g.,* resurgence, precipitation, grid edge input). The method becomes similar to (Bates et al., 2010) - only that

Algorithm 1 Iterative stationary solver

```
Initialise ~~DAG~~ directed acyclic graph structure on hydraulic surface
while Convergence criterion[1] not met do
    Update ~~DAG~~ directed acyclic graph with hydraulic surface
    for each node n in downstream topological order do
        Calculate $s(n)$ and weight partitioning
        Determine $Q_{in}(n)$ from upstream nodes
        Calculate $Q_{out}(n)$
        Transfer $Q$ to receivers of $n$
    end for
    Increment $h_w$ for all nodes
end while
```

their formulation includes an approximation of inertia and have a D4 flow topology. Although straightforward and massively parallelisable (e.g. Apel et al., 2022), this method does not benefit from the ~~DAG~~ directed acyclic graph data structure as signals are propagated from one node to their immediate neighbours. If external Q_{in} is kept constant long enough, this solution converges toward a unique equilibrium stationary state and is not efficient if the intermediate transient steps are not important.

Like any explicit finite difference methods, higher time steps leads to less iterations and more efficient spread, but also more instability. Equation ~~6~~ 2 expresses the velocity of a flood wave and therefore its stability can be approximated using the Courant Friedrich Levy conditions (CFL):

$$C_r = \Delta t \frac{u_{max}}{\Delta x max} \tag{8}$$

where C_r is the Courant number.

The transient solution converges toward an equilibrium hydraulic surface and Q field. We estimate convergence based on both median h and $\frac{\Delta h}{\Delta t}$ for the whole landscape. We stopped the iterative process once the first plateaus and, when increment in flow depth becomes ~~lowerthan~~ lower than an acceptable *ad hoc* threshold (*e.g.* 10^{-9} m).

3.4 Stationary solution

The stationary solution optimises convergence towards the equilibrated solution - *i.e.* the steady state flow depth and discharge fields to an input runoff. Ultimately, the amount of water flowing through a landscape equates the runoff rate propagated into the drainage network. Numerically speaking, it falls down to calculating a weighted drainage area, a procedure already in use in GIS applications and LEMs when it comes to integrating the effect of spatial variations in precipitations (Leonard et al., 2023, e.g.). In the case of effective precipitations, each nodes receive a local $P(x,y)\Delta x\Delta y$~~, while in~~ . In reach mode, ~~given entry nodes receive an arbitrary~~ the model receives Q_{in} in the boundary cells corresponding to the upstream section of the river. In both cases, received water is then recursively transferred to all the downstream nodes following the topological order.

It effectively reduces the need to propagate a signal gradually from upstream to downstream one node at a time. However, the final hydraulic surface being different than the topographic surface, the algorithm needs to iterate to gradually build the hydraulic surface. From the first iteration, discharge is propagated through the full landscape and starts "piling up" h on the whole flow path. Every iteration recomputes the ~~DAGs~~ directed acyclic graphs from the updated hydraulic surface, effectively spreading Q_{in} towards its final geometry balanced by Q_{out}. Time step in the stationary mode is a numerical stability criterion modulating the magnitude of flow depth increment. ~~Because Q_{in} is independently determined from Q_{out}, the CFL stability criterion does not strictly apply and we test the model with a constant or a variable time step (then determined in respect to CFL conditions).~~ Similarly to the transient solution, we estimate convergence based on both median h and Δh between each iterations for the whole landscape and considered convergence reached once median $\Delta h < 1e-9$ m.

3.5 Validation

We validate the numerical scheme against an analytical solution (Figure 3) in the case of a rectangular channel (Bates et al., 2010; Davy et al., 2017). We combine equation 1 and equation 3 to obtain an analytical stationary flow depth noted h^*_W:

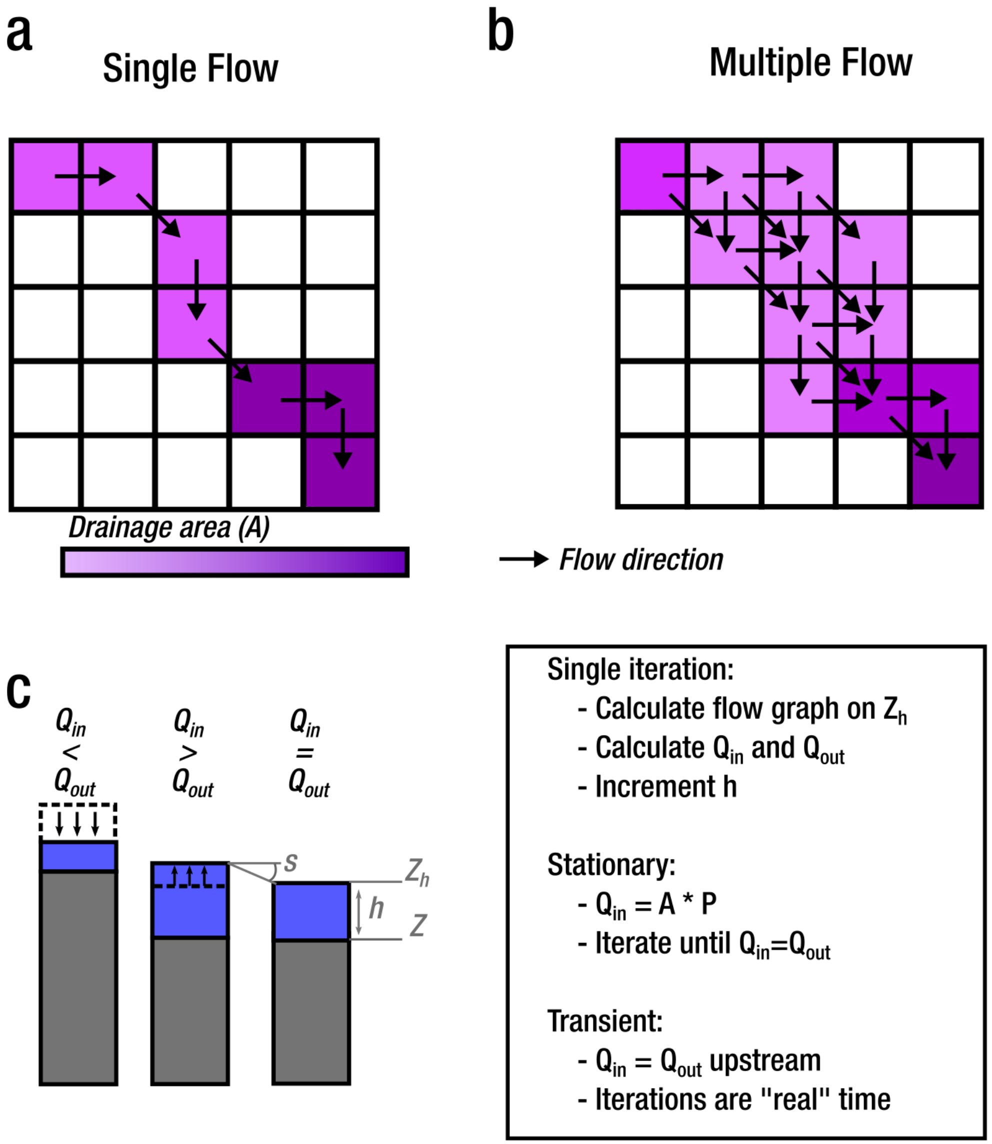
$$h^* = \frac{nQ_{in}}{dx\sqrt{s}}^{\frac{1}{\alpha}} \tag{9}$$

Equation 2 predicts that in the case of a rectangular channel with a constant slope S_0, the slope of the water surface s should be equal to S_0. Assuming a boundary condition of fixed hydraulic slope equals to S_0, we can determine h^* suitable for an analytical calibration.

We run GraphFlood with the transient and stationary solvers, and ~~MFD and SFD~~ multiple flow direction and single flow direction schemes on a $200\,m \times 40\,m$ rectangular channel with a regular $dx = 1\,m$ (more details in the figure caption). Figure 3a shows the results for all runs. Each simulation converges towards h^*, validating the numerical methods. The number of iterations to reach h^* - directly linked to the computational efficiency of the algorithms - is significantly higher for the transient model as it needs to propagate the flood wave through the whole channel one node per iteration. This behaviour is likely to worsen with the complexity of a natural river network where any junction would need catchment-wise upstream information before being equilibrated and being able to propagate signal downstream. Figure 3b zooms on the stationary models that reach stationary state in about 300-1000 iterations, roughly 400 times faster than the transient model. Adaptive time stepping based on the CFL condition slightly reduces the number of iterations required to reach the analytical solution and the ~~SFD~~ single flow direction model converges in less iterations than the ~~MFD~~ multiple flow direction model.

3.6 Test sites

We test GraphFlood on two lidar-derived DEMs and aim to explore the effect of different geographical contexts on the algorithm, both in term of relief and climate. Our first test site is located near Green River (Utah, USA), a low-relief area in an arid context with smooth hillslopes. The second test site is the Hanalei river catchment in Hawai (USA), with sharp relief made of volcanic rocks, steep hillslopes and entrenched valleys. The original spatial resolution of both DEMs is 1 m, provided pre-



Single iteration:
- Calculate flow graph on Z_h
- Calculate Q_{in} and Q_{out}
- Increment h

Stationary:
- $Q_{in} = A * P$
- Iterate until $Q_{in}=Q_{out}$

Transient:
- $Q_{in} = Q_{out}$ upstream
- Iterations are "real" time

Figure 2. ~~Comparison between hydrology approximated~~ Graphical representation of a single iteration with GraphFlood~~(~~. Panels a and ~~c), calculating~~ b show the flow ~~depth and discharge vectors~~routing structuring the graph, respectively for single flow and ~~with a classic drainage area based method (D8 Steepest descent route)~~multiple flow directions in map view. ~~The pannels zoomed on~~ Note that it only displays flow routing from a ~~channel junction highlight how GraphFlood allows~~ single source for clarity. Panel c illustrates in cross sectional view the ~~extraction~~ increment or decrement of ~~detailed~~ flow ~~pattern in all direction and magnitude compared to~~ depth at the ~~D8, linear networks~~ end of ~~drainage area~~the iteration depending on calculated/propagated discharges.

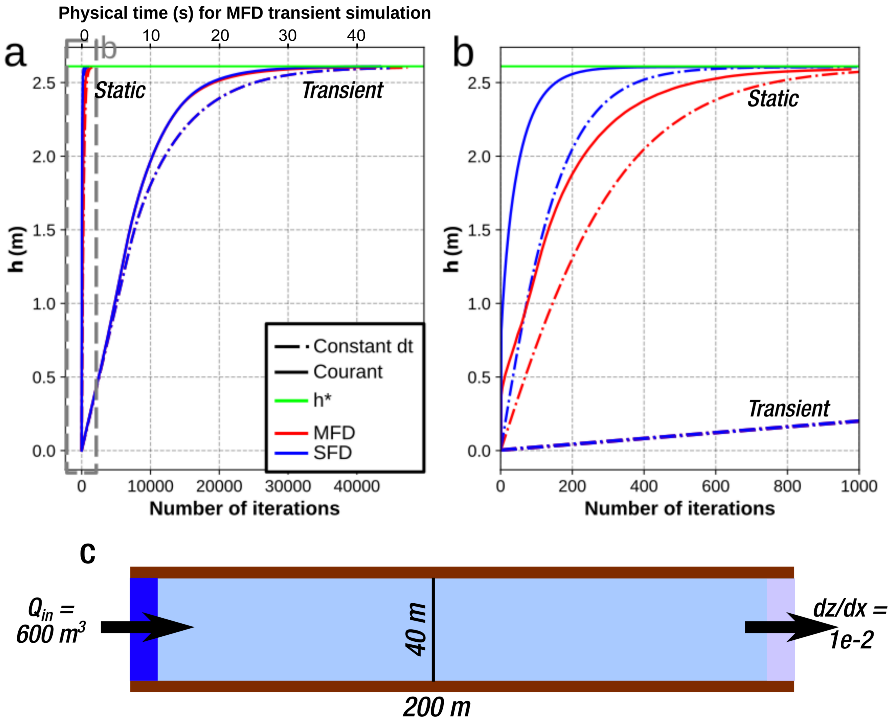


Figure 3. Validation tests for the ~~MFD~~ multiple flow direction and ~~SFD~~ single flow direction stationary and transient simulation for a given $Q_{in} = 15\, m^3.s^{-1}$. The scenarios with constant dt were set to $1e^{-3}$ seconds and the scenarios with CFL condition were calculated with $C_r = 3e^{-3}$. Both were chosen empirically as values balancing model performances, stability and cleanness of the final results. Panel a displays the full results for all the simulations while b zooms on the stationary model results. Panel c describes the model setting in map view. 1000 iterations of graphflood on this small rectangular channel take about 0.7 seconds for single flow direction and 1.2 second for multiple flow direction with a CPU intel i9-11950H.

processed from point clouds and provided by opentopography.org (OpenTopography, 2020, 2012). We also downsample the DEM of the Hanalei river catchment to a resolution of 5 m using a cubic resampling implemented by GDAL/OGR contributors (2023) to process a larger watershed and test GraphFlood on multiple resolutions.

4 Results

4.1 Numerical behavior for a single simulation

We first explore the behavior of the model during a single simulation, where we run the ~~MFD~~ multiple flow direction stationary algorithm on both test sites for a high-intensity rainfall rate of $100\; mm\; h^{-1}$. We deliberately chose an extreme rainfall rate to test the algorithm under high flow conditions during which multiple diverging river channels are activated.

We run the model to convergence (figure 4 - see caption for the full simulation parameters). In term of channel network topology, GraphFlood is able to reproduce diverging and converging flow patterns that follow converging and diverging channel networks. This behaviour is striking on Green River, where the broad valleys consist of an interwoven network of channels, but also well-captured on the clearer channel beds of Hanalei. GraphFlood in that way contrasts with drainage-area based flow patterns which by nature converge toward a single line of flow (e.g. fig. 1). In both cases the majority of the DEM pixels are displaying insignificant flow depth (<1 cm) as one should expecting from natural landscapes where rivers only represent small portions of the landscape.

GraphFlood reaches convergence in respectively 4000 and 3000 numerical iterations for Green river and Hanalei (fig. 5 a and b) based on the criterion outlined in sections 3.3 and 3.4. At first glance, this number is high, but we observe a huge discrepancy in the spatial and temporal patterns of convergence. The model converges asymptotically in the rivers where less than 200 iteration for Green River and less than 60 for Hanalei are enough, as illustrated by the striking spatial variations on figure 5 c and d. Low drainage area on the hillslopes induces lower increments of flow depth, which combined with high slopes explain the slower convergence on the hillslopes.

We test the sensitivity of the model to its numerical parameter Δt and its discretisation Δx. Δt controls the magnitude of h increment. Maximising it optimises the spreading of Q to its equilibrium field. However, our tests also highlight that while significant over-estimation provokes numerical divergence, slight overestimation converges to an underestimated final h. Spatial resolution of DEM, Δx, can be dictated by the availability of source data, but it can be interesting to reduce the resolution of a DEM in order to process larger area (if computing speed or memory are limiting factors). For this test, we use the Green River DEM resampled from $dx = 1\,m$ to $dx = 10\,m$. Flow patterns remain relatively similar from a resolution to another. However loss of details are observed at lower resolution as expected. Lowering resolution leads to lower hydraulic slopes on averaged and subsequently a decrease of Q_{out} and an increase of total volume of water stored on the DEM.

We also test the sensitivity to the physical parameters. Manning's coefficient is an empirical friction parameter reflecting the local surface condition (e.g. vegetation, bed roughness, see Arcement and Schneider (1989) for different measurements). Higher friction values predicts a higher and more distributed water surface required to reach the same Q_{out}. Higher input discharge or precipitation rates lead to higher flow velocity and therefore lower the stability condition, thus impacting speed of convergence.

4.2 Comparison with existing models

We compared GraphFlood with ~~previoous~~ previous models sharing similar applications (relatively large-scale and medium term hydrology): Caesar Lisflood (Coulthard et al., 2013) and River.Lab (formerly Eros/Floodos - Davy et al. (2017)). We ran the three models on Green River with a constant rainfall rate of 30 $mm\,h^{-1}$ and a classical friction coefficient of 0.033. We ran the three stationary simulations, as detailed in section 3.4. We compared the fields of flow depth by pairs of models (figure 6). Overall, the differences between the models are minimal, centered between 3 10^{-4} and 5 10^{-4} m. The differences can be linked to the differences in flow routing. Caesar Lisflood can only route flow to cardinal directions therefore the distribution of slopes is not exactly the same than GraphFlood and River.lab which include diagonals. River.Lab relies on a stack of consecutive 1D

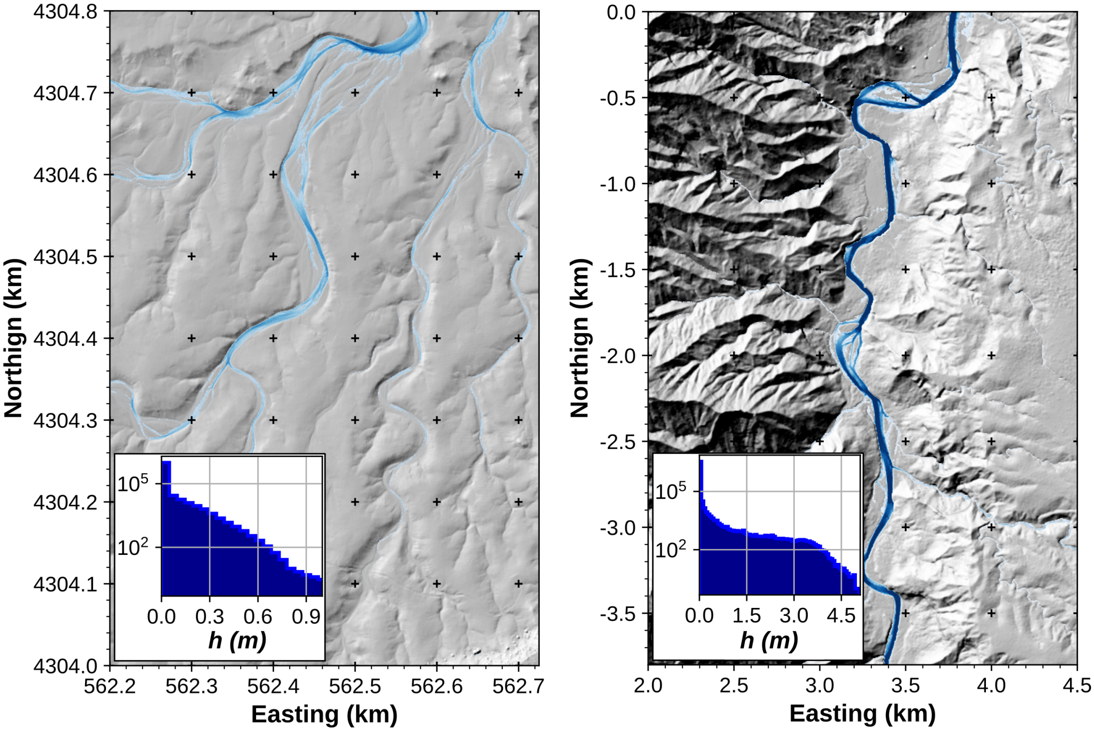


Figure 4. Flow depth field calculated with GraphFlood for fluvial valleys in Green River, Wyoming, USA (a) and Hanalei, Hawaii, USA (b). The maps are zoomed on major fluvial valleys for clarity. Both histograms show the distribution of water height for the ~~MFD~~ multiple flow direction stationary solutions calculated during a high storm event (precipitation rate = 100 mm/h). Note the logarithmic y scale on the histogram demonstrating the huge majority of points have low flow depth (< 1cm).

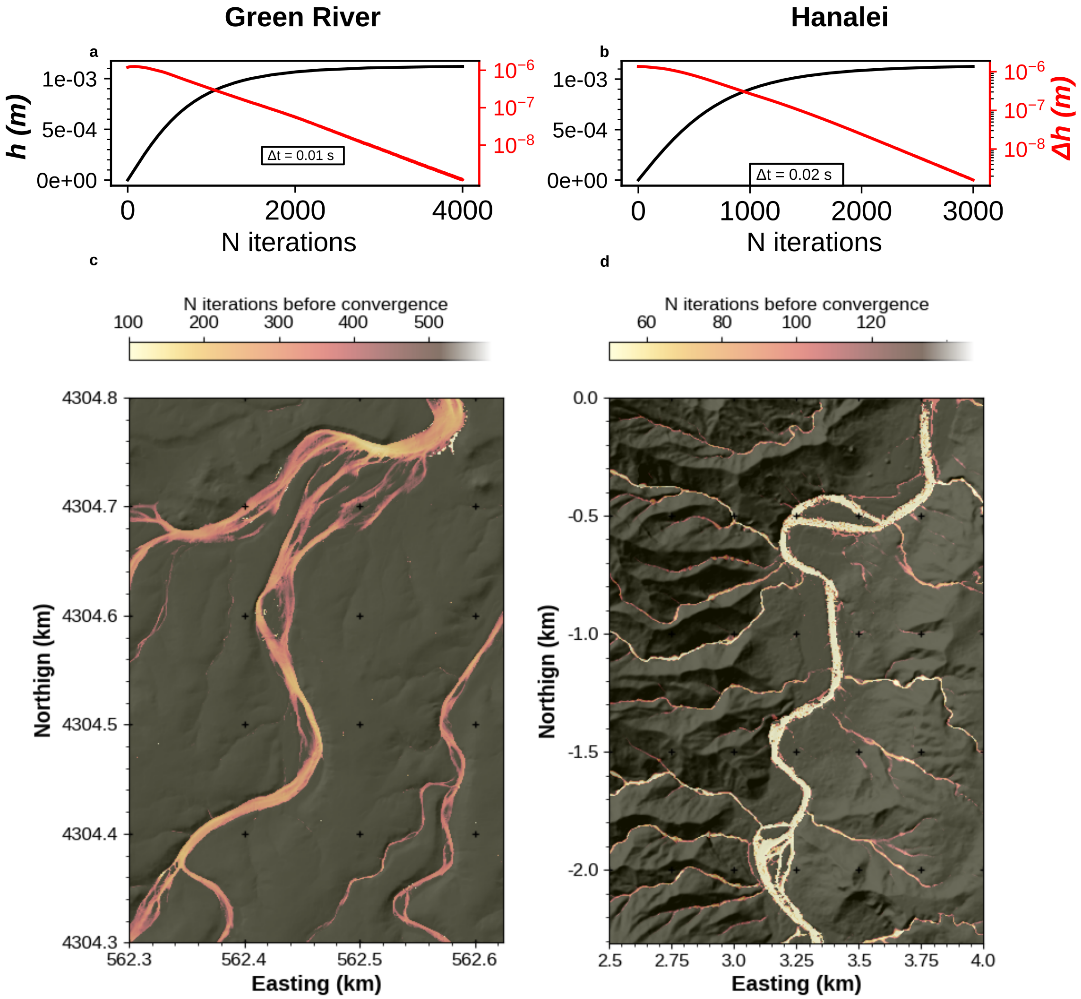


Figure 5. Rate of convergence for the simulation of figure 4 with respectively $\Delta t = 1 \times 10^{-2}\ s$ and $\Delta t = 2 \times 10^{-2}\ s$. On panels a and b, we show in black the median flow depth function of the number of numerical iterations and in red the changes in flow depth between each iterations. Panels c and d demonstrate the spatial variability in the rate of convergence. Note that GraphFlood converges significantly faster in fluvial domain. The number of iterations before convergence is defined as the first iteration reaching 95% of its equillibrium value .

stochastic paths on a 2D grid while GraphFlood offers a continuous solution in space and time, explaining the small differences in the final solutions.

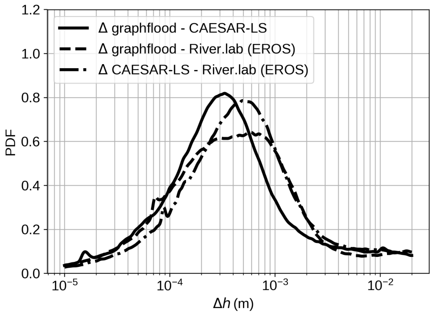


Figure 6. Benchmark comparing the difference in stationary field of flow depth between CAESAR-LISFLOOD, River.Lab (formerly EROS/FLOODOS) and GraphFlood. The data expresses the distribution of flow depth differences for each pairs of the models. The distributions are estimated using a Kernel Density Estimation.

5 Applications and potential

5.1 Flood extent

The computational efficiency of GraphFloods enables the rapid simulation of stationary flow depth and extents under different runoff intensities. We ran the model for effective precipitation rates ranging from $5\,mm\,h^{-1}$ - approximating low-flow conditions - to $300\,mm\,h^{-1}$ - extreme storm conditions. Figure 7 shows the flood extent for each different scenario on a per node basis. In addition to fast engineering application or flood risk assessment, ~~(Bates, 2022)~~(e.g., Bates, 2022), Bernard et al. (2022) noted that using flow metrics calculated from different precipitation rates could be used to determine the extent of flood plains and of the different channels of a river system. While more computationally demanding than geometrical method (e.g. Clubb et al., 2022), GraphFlood offers a physics-based method self-emerging the floodplain geometry. Low flow conditions in purple in Figure 7 emphasise the geometry of channel beds while higher, storm-related flow conditions in blue indicate the maximum extent of the floodplain. We only computed uniform precipitation rate scenarios, but GraphFlood can ingest spatially variable matrices of effective precipitations if coupled with more sophisticated precipitation/infiltration data or model.

5.2 Flood wave

While the model is primarily designed and optimised for the stationary state, we illustrate its capabilities to model the transient propagation of a flood wave (e.g. sudden increase of input discharge in reach mode) in Figure 8. We isolated a small section of a river from the Green River site and started from equilibrated low flow conditions (time=0s). We instantly increase the input discharge by a factor 3 and the different panels display the spatial propagation of the resulting flood wave through time.

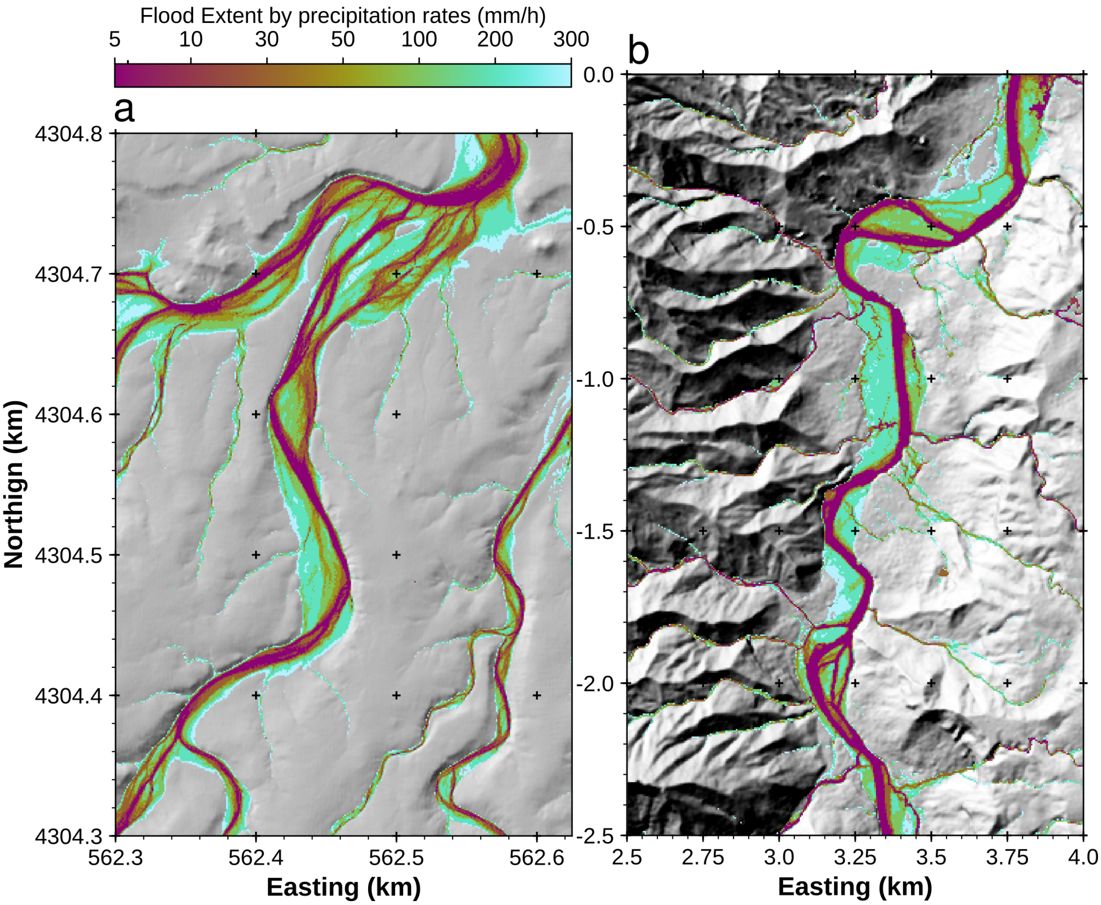


Figure 7. Flood extent at stationary solutions for different precipitation rates. The color represent the minimum precipitation rate at which the area is flooded by at least 10 cm of water. Note the self-emergence of bedforms and floodplains.

5.3 Hydromorphometry

One of the main technical challenge in topographic analysis studies is to determine from topographic data the transitions between the fluvial network, the colluvial channels, and the hillslopes. Such classification is useful for understanding landscape dynamics (e.g. Grieve et al., 2016; Hurst et al., 2019), to constrain geomorphological laws (Perron, 2011, e.g.). Landscape Evolution Models also routinely apply different process laws based on that transition (e.g. Perron, 2011), or to assess the response of landscape to tectonics or climate changes (e.g. Willett, 1999). A common approach consists in isolating breaks in the Slope-Area distributions to determine a critical drainage area value (DiBiase et al., 2010; Whipple et al., 2013, e.g.). A number of geometrical/empirical method have also been developed to isolate individual channel heads in higher resolution

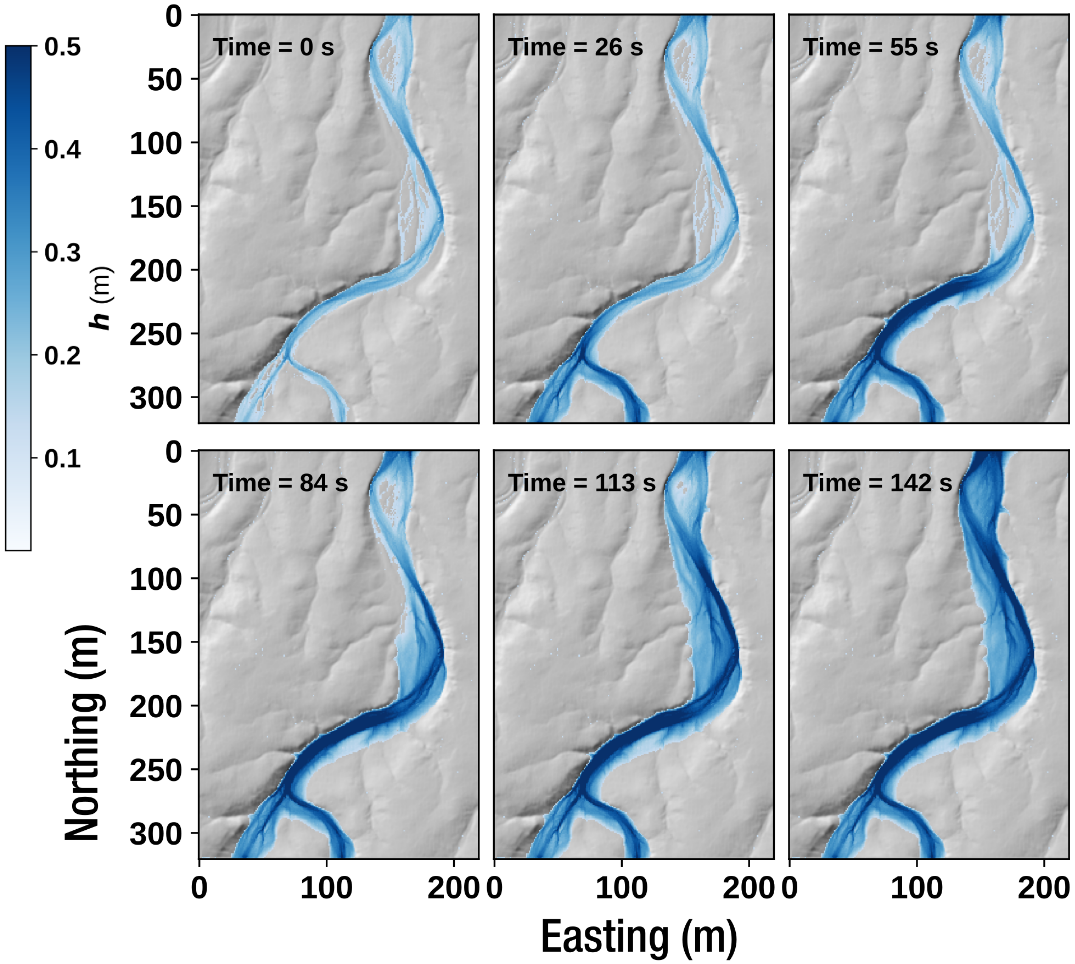


Figure 8. ~~Flood extent at stationary solutions for different precipitation rates~~Propagation of flood wave thorugh time using GraphFlood in transient mode. The ~~color represent the minimum precipitation rate at~~ initial conditions correspond to a steady flow for a total input of $3m^3/s$, which ~~the area~~ is ~~flooded by~~ triple at ~~least 10 cm~~ the start of ~~water~~the simulation. ~~Note~~ The times indicated on the ~~self-emergence of bedforms and floodplains~~different panels are the simulation times.

DEMs (Pelletier, 2013; Clubb et al., 2014; Lurin et al., 2023, e.g.). These methods intrinsic limitation is the use of surface topography: the latter by nature cannot express the actual geometry of water bodies there making them harder to detect.

Recent studies (Costabile et al., 2019; Costabile and Costanzo, 2021; Bernard et al., 2022) demonstrated that approaches explicitly approximating hydrodynamics effectively overcome that limitation by computing hydrology-derived geomorphological metrics from hydraulic surface and discharge. ~~They~~ In particuler, Bernard (2022) show that the slope-area relationship can incorporate hydrological information by replacing topographic slope by the hydraulic slope at equilibrium and D8 drainage area by a specific drainage area $a_s(r) = \frac{q}{r}$, where r is the runoff precipitation rate and q the discharge per unit width. ~~These methods show that $a_s(r)$ is very efficient to naturally separate river channels from colluvial channels and hillslopes. These metrics~~ s and $a_s(r)$ are naturally embedded within the ~~DAG~~ directed acyclic graph structure of GraphFlood allowing a more systematic and straightforward bulk computation. ~~We~~

First, as illustrated in Figure 7, applying GraphFlood with high precipitation rates proves to be an efficient method for determining the extent of the floodplain. Next, we extracted s ~~- $a_s(r)$~~ and $a_s(r)$ for both test sites and ~~separated hillslopes, colluvial and fluvial domains (see~~ applied thresholds based on the breaks in slope of the $\log a_s(r) - \log s$ plots to delineate different domains (Figure 9 ~~). For clarity, we use arbitrary thresholds from the s - $a_s(r)$ plots to determine the transitions. We also define the floodplains using the maximum extent of fluvial channels for high precipitation rates from figure 7~~a and b). Following the approach of Bernard (2022), we isolated domains I, II, and III, which correspond to the classic geomorphological features of convex hillslopes, concave valleys, and fluvial regions, respectively.

The s - $a_s(r)$ relationships for both catchments ~~globally show~~ (fig.9 c and d) generally exhibit patterns similar to those observed in classic Slope-Area techniques. In domain I, s increases and ~~plateaus in the hillslopes domain to then decrease with break in slopes in log space corresponding to colluvial and fluvial channels (e.g. Montgomery, 2001). However, we also observe low s~~ then plateaus before decreasing, with breaks in slope in the log-log space defining the transitions to domains II and III (e.g., Montgomery, 2001). Notably, domains I and III define hillslopes and fluvial areas, as discussed in Bernard (2022) . Domain II reveals a variety of patterns: (i) including convergent hillslopes that gradually concentrate flow into small channels; and (ii) and divergent branches of fluvial channels in partially flooded regions. For each domain, we calculated θ and k_w, which are equivalent to the concavity and steepness indices in Flint's law (Flint, 1974). The observed variation in θ values is greater than that typically seen in Flint's law (Gailleton et al., 2021). The significant scatter in the $\log s$ - ~~$a_s(r)$ areas, corresponding to flat surfaces isolated from the channel (e.g. elevated terraces). Both sites then show a noticeable break in slope corresponding to the colluvial domain where flow starts to converge towards proto-channels, followed by another less-pronounced break in slope expressing the switch to well define rivers domain. The addition~~ $\log a_s(r)$ plots is consistent with common Slope-Area plots. However, the incorporation of hydraulic information ~~to slope and area makes the distinction less sensitive to the threshold and direct visualisation of a_s give an already clear and physics-based separation of the different domains~~and the reduction of topographic noise by using the hydraulic surface allow us to link local outliers to specific morphological features. For example, low s and low $a_s(r)$ values reflect a flat surface disconnected from the active channel (e.g., a fluvial terrace), a feature that traditional methods might struggle to detect. The fluvial domains also ~~terminate with an interesting high surge of~~ exhibit an interesting surge in s for high $a_s(r)$~~corresponding to local accelerated flow that would not be caught by common S-A~~

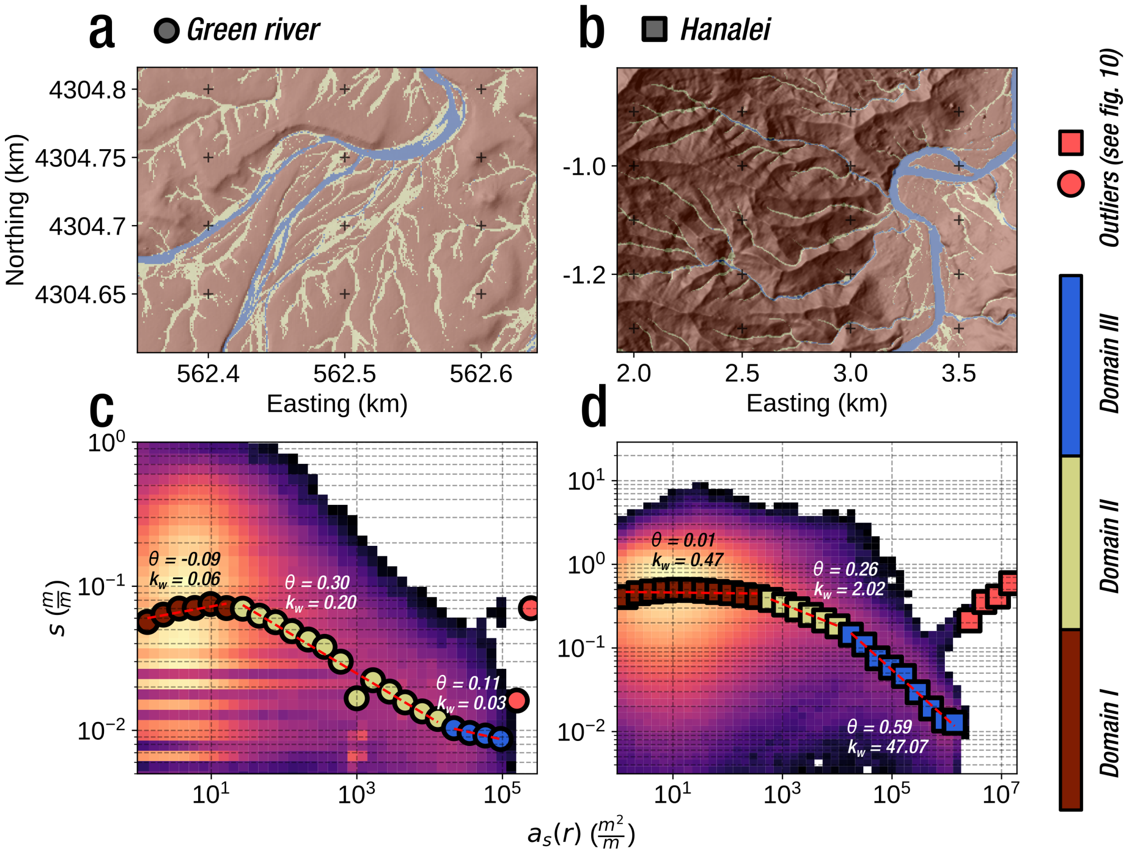


Figure 9. Classification of different domains on the test sites based on s - $a_s(r)$ relationship. a and b show the classification on area of interest for respectively Green River and Hanalei test sites. c and d show the corresponding $\log s$ - $\log a_s(r)$ plots. The colour of the points correspond to their domains. As described in the main text, we determined the separations of the domains using the breaks in slope in the binned data. Outliers, belonging to nodes showing high $a_s(r)$ and potentially high s are displayed on appendix figure A1.

~~plots~~, a novel feature compared to traditional Slope-Area plots, decoupled from the typical monotonic downstream increase in drainage area. We isolated these outliers using the last break in slope in figure9 c and d and visualized some of them in appendix figure A1. Few of these points represent numerical artifacts linked to local minima that artificially increase h and, consequently, the discharge and $a_s(r)$. Most of them correspond to areas of accelerated flow and concentrated discharge where channels narrow, branches converge, and potentially where hydraulic slopes increase due to topographic knickpoints.

~~Domainification of the landscape based on hydromorphometry. Using an approch based on Bernard (2022) as well as data in figure 7, we separate the domains into area affected by hillslopes, colluvial and fluvial domains. The domains are selected by applying cutoff values on the $s - a_s(r)$ plots - see main text for details about these values. Areas that are not fluvial but flooded at high flow are considered floodplains.~~

This last observation highlights the kind of additional information the hydrology-aware approach unravels. Bernard et al. (2022) built on earlier work restricted on hillslope (Gallant and Hutchinson, 2011) where $s \equiv \frac{dz}{dx}$ to develop this principle further and express a proxy for channel width, called specific width $w_s(r)$. The specific width is calculated from the ratio between ~~SFD~~ single flow direction drainage area (i.e. most convergent flow lines) and the specific drainage area (i.e. representing the flow field spread to its natural extent). As acknowledge by the authors, the challenge lies in the choice of the single flow path which will determine A: if the latter does not coincide with that main discharge field, the results are highly noisy and difficult to interpret. With the precipiton method, Bernard et al. (2022) suggest the calculation should be post-processed on the discharge field calculated at low-flow condition and following its maximum values.

We leverage GraphFlood integrated ~~DAG~~ directed acyclic graph data structure to optimise this process and generalise it to the 2D channel network. Indeed, using the ~~DAG~~ directed acyclic graph calculated from the equillibrated hydraulic surface, we repeat a stochastic walk to calculate A where the steepest receivers of each nodes is determined from the multiple flow receivers using the hydraulic surface and a probability function of these receivers' Q_{out}. Repeating this walk about 50 times and keeping track of node-wise $max(w_s)$ ensures all the channel pixels are visited. Figure 10 displays the resulting field of flow width where we simply apply a threshold to filter out unreasonable values happening when A gets out of the main channel for few nodes. This method effectively highlights fine-grained variations in flow width and allows its systematic, efficient extraction unravelling patterns of "width" knickpoints.

6 Discussion

6.1 Controls on numerical efficiency and accuracy

Computational efficiency to reach the stationary solution is one of the main advantage of GraphFlood and figure 12 provides a number or benchmarks function of the number of nodes of the DEM. However, computational efficiency depends on multiple factors making the efficiency partly case dependent.

First, part of the method relies on subjective choices. As demonstrated on figure 5, there are spatial discrepancies in GraphFlood convergence speed. A study focusing on fluvial domains (e.g. flood extent) often only require <100 iterations, while obtaining convergence for the entire landscape (e.g. separate the different process-based domains) can take up to few thou-

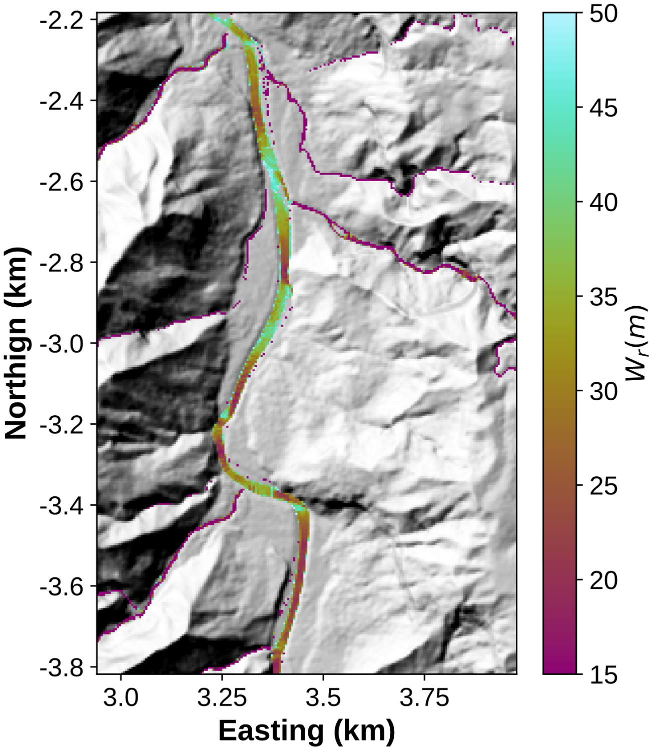


Figure 10. Effective width for a section of Hanalei river, reflecting channel widening and narrowing.

sands iterations. The time step also dictates the speed and accuracy of the algorithm. Maximising the time step reduces the number of iterations to reach convergence. Yet, it also impacts the accuracy, consistency, computational time and stability of the solution (i.e., a higher time step plateau to a fluctuating hydraulic surface).

Secondly, switching the model from ~~MFD to SFD~~ multiple flow direction to single flow direction mode reduces the number of operations to compute and therefore the computational time. However the resulting water surface is impacted by this choice due to the over-focusing of flow in the single flow routing (figure 11). The line concentrating all the flow overestimates Q_{in} while all the other channel nodes overestimate Q_{out} resulting in a global underestimation of h. The error on Green River is concentrated around 10%.

Finally the performances of GraphFlood are tightly linked to the numerical framework used for its implementation. The simplicity and versatility of GraphFlood make it straightforward to re-implement in different frameworks as long as they offer basic graph data structure and local minima handling. Computing the ~~DAG~~ directed acyclic graph and the related algorithms for each iteration accounts for a big part of of the computational time. Therefore, the implementations of these algorithms strongly impact the overall performances. For example, the exact same simulation takes approximately 250 ms or 800 ms in the python/c++ implementation or using MATLAB©/ TopoToolBox (Schwanghart and Scherler, 2014) respectively. The time consuming algorithms are the topological ordering (e.g. Anand et al., 2020; Braun and Willett, 2013; Carretier et al., 2016), the local minima resolver (e.g. Cordonnier et al., 2018; Barnes et al., 2014; Gailleton et al., 2023) and the receivers and donors computations as they need updtates at each iterations.

Detailed time-benchmark comparison with other methods can also quickly be misleading because of the divergence of scopes: GraphFlood focuses on steady flow which is conceptually too different to compare to transient solvers (e.g. Bates et al., 2010; Brunner, 2002). River.Lab (Davy et al., 2017, formerly Floodos,) also targets stationary solution. Bernard (2022) demonstrated that the method could reach the same orders of magnitude for the time required to get a convergent solution in the main rivers in specific cases where the influx of precipitons is optimised to enter only the main channel via discrete inlets from tributary junctions. However, the efficiency of this method decreases when simulating other parts of the landscape, such as hillslopes, due to the low frequency of precipitation passage on non-convergent areas.

Nevertheless it is worth noting the algorithm is scalable: Green River site converges in about 20 seconds for the main rivers, with less than 200 ms per iterations. We also tested GraphFlood on an 83 Million pixels DEM on a laptop with 32 Gb of memory and the model converged for the main rivers in about 20 hours with 100 seconds per iterations.

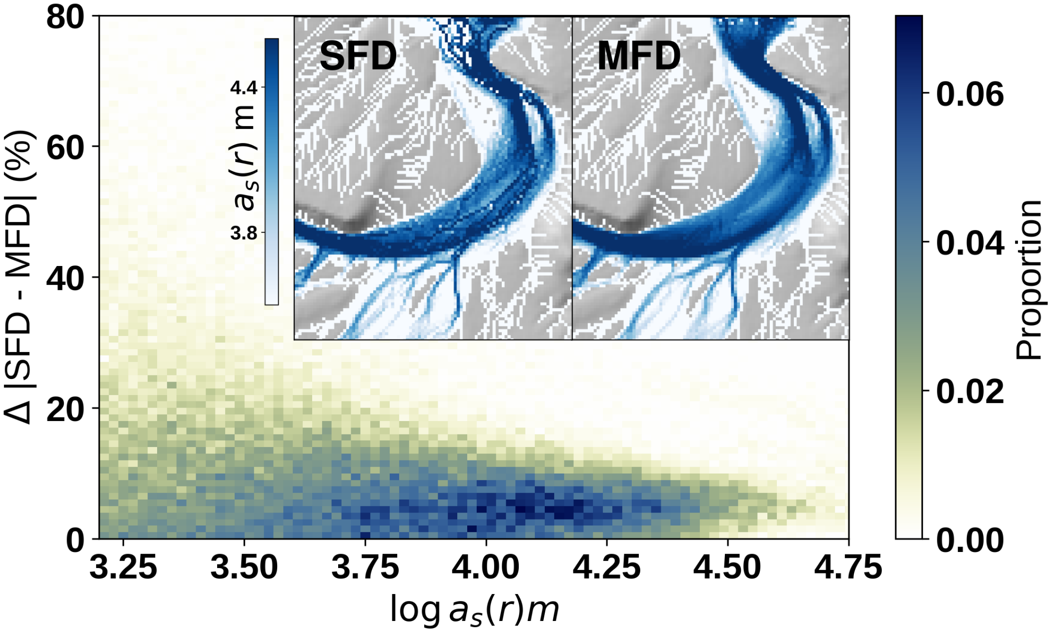


Figure 11. Differences in final results for Single flow solver and Multiple flow solvers. The ~~MFD~~ multiple flow direction solution is cleaner and has less artifacts. The magnitude of the differences is function of the frequency at which the D8 ~~SFD~~ single flow direction flow passes through a cell (proxied here by ~~MFD~~ multiple flow direction $a_s(r)$). While ~~SFD~~ single flow direction solvers are faster and simpler, their accuracy will be function of diverging flow patterns. Smaller Δt can reduce the differences.

6.2 Potential optimisations

An obvious optimization consists in developing a parallel version of GraphFlood. In this paper, we made the choice to remain on single threaded CPU for (i) simplicity, (ii) flexibility and (iii) favouring the possibility to run concurrent models to explore parameter space. Transient mode can be parallelised, even on GPU, as each node is independent from one another at a time t similar to Apel et al. (2022). Stationary GraphFlood, on the other hand, has a strong non-local component in the calculation of Q_{in} and would require significant modification to be partially parallelised, using for example Barnes et al. (2021) .

Another optimisation consists in improving our management of time stepping. CFL conditions only theoretically apply to our calculation of Q_{out}, but not on the propagation of Q_{in} in stationary mode. Alternative finite difference formulation like Runge-Kutta or an implicit formulation could allow larger time steps. However these methods would only increase the efficiency of a single iteration but would still suffer from the highly-iterative nature of the algorithm to reach an equilibrated hydraulic surface.

Finally, we can significantly reduce the computation time of studies interested in the fluvial domain only. As suggested in Bernard (2022) and illustrated in figure 4, GraphFlood converges significantly faster in areas with higher Q. The fluvial domain only represents a minor subset of the total number of nodes in a landscape and theoretically, focusing only on these nodes could significantly speed up the process. Induced sub-graph methods offer solutions to apply algorithms in a subset of a ~~DAG~~ directed acyclic graph without the need to process its entirety. In the case of rivers, it requires the identification of all the nodes of interest, *i.e.* downstream of a given discharge or drainage area threshold. Taking full advantage of this optimisation is challenging as it requires the dynamic identifications of the nodes of interest without processing the whole graph.

We developed an induced sub-graph method to take advantage of that optimisation. The principle remains the same than section 3.1, except that graph-realted operations are computed in a node-to-node basis (e.g. computing the ~~DAG~~ directed acyclic graph donors and receivers, handling of local minimas, topological ordering). A pre-computing step determines input points based on drainage area thresholds or arbitrary input points (Tarboton, 1997). These points are pushed in a priority queue sorting active nodes per decreasing elevation (opposite to Barnes et al. (2014)), ensuring that the most upstream node of interest that has not been processed yet is always the next in queue. The nodes are popped and processed from the priority queue sequentially. Once ~~Q_in and Q_out~~ Q_{in} and Q_{out} computed according to section 3.1, we push in the priority queue the receivers of the active node. The process is repeate until emptying the queue. Note that if a node has no receiver and is not a model edge, we gradually fill the local depression until finding an outlet, in a similar way to Davy et al. (2017) or Gailleton et al. (2023).

This version of the algorithm reproduces the results from the original one, except minor artifacts near the input points. One iteration takes 250 ms with GraphFlood and 15 ms with the induced graph method. For a discharge threshold of 36000 m^2 and a precipitation rate of 50 $mm\ yrs^{-1}$, the models converge for the main rivers in about 50 s for GraphFlood *vs* 3 s for the induced graph method demonstrating strong potential for studies focusing on the fluvial domain. The complexity of the algorithm is tied to the priority queue and is therefore $\mathcal{O}(n \log n)$ with n being the number of nodes in each traversal, meaning computation time increases non linearly as the drainage area threshold decreases. Figure 12 provides an extensive time benchmark comparing the efficiency of both methods in a global and per-iteration perspective.

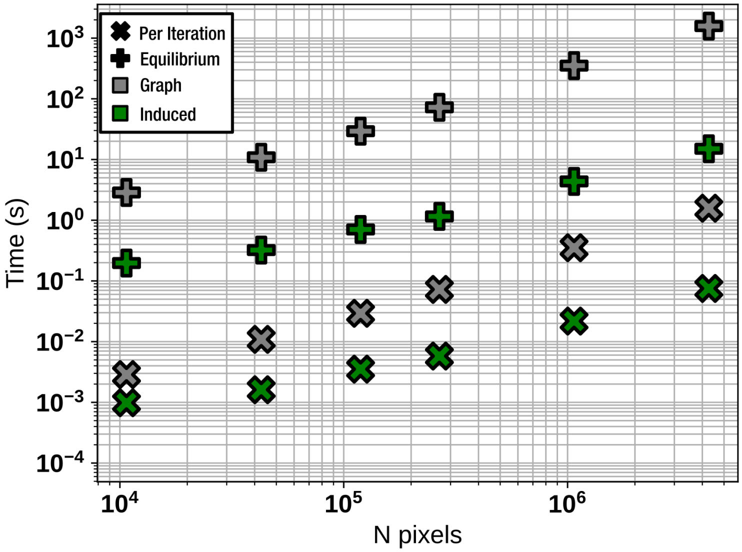


Figure 12. Time benchmark comparing the computational efficiency of GraphFlood and its induced graph variant for the Green River DEM resampled at various resolutions. The global convergence time represents the timing for converging the model for the fluvial and colluvial domains while the time per iterations is an important metric when considering GraphFlood for LEMs.

6.3 Potential for hydromorphometry and Landscape Evolution Models

Bernard et al. (2022) demonstrated the potential of informing common scaling laws used in tectonic geomorphology (e.g. Kirby and Whipple, 2012) with hydrodynamics. GraphFlood represents a step toward making the inclusion of hydrology more systematic in geomorphological analysis. For example $s - a_s(r)$ plots, as illustrated by both Bernard et al. (2022) and figure 9, isolate more signals than classic $S - A$ as per originally designed by Morisawa (1962) and Flint (1974). $a_s(r)$ is not strictly function of the downstream distance like A and has the potential to express a wider range of landform. Data points with high $a_s(r)$ and/or high s are likely to represent areas of increased stream power beyond the common geometrical knickpoint (e.g., increased discharge due to local channel narrowing) as demonstrated in appendix figure A1. Alternatively, low s and a_s testify of abnormally flat areas (i.e., flat areas not visited by rivers), which if calculated from multiple runoff rates could unravel families of terraces. Commonly used metrics linked to $S - A$ (e.g., concavity index, steepness index) are likely to express a wider range of signals when extracted from $s - a_s(r)$. Both our test sites and the study of Bernard (2022) show similar global patterns in $s - a_s(r)$ plots while displaying notably different values, regression coefficient and intercept, and absolute values.

Combined with effective width or the direct calculation of shear stress from h, hydromorphometrics can help identify and quantify new family of responses to perturbations. Alongside with geometrical knickpoints (e.g. Gailleton et al., 2019), area of channel narrowing or widening or accelerated flow can be caught unravelling wider ranges of landscapes responses to perturbations. Systematic calculations of all these metrics for multiple ranges of runoff rates could help redefining and completing global scaling laws comparing discharge, drainage area, channel width and hydraulic slopes.

GraphFlood's ability to extract metrics for various precipitation rates also opens possibilities for indirect metrics. For instance, Clubb et al. (2022) and Clubb et al. (2023) highlighted the importance of valley width in understanding landscape evolution. By using extremely high precipitation rates with GraphFlood, it becomes possible to flood the valley and systematically determine its width. Another potential application could involve gradually increasing precipitation rates to progressively flood a fluvial system from its bed to its floodplain, revealing multiple families of terraces.

GraphFlood allows the fast approximation of hydrodynamics, and therefore shear stress. Coupling GraphFlood with physics based morphodynamics (e.g. Davy and Lague, 2009; Minor et al., 2022) would allow the upscaling of short term fluvial dynamics to longer time scale and larger spatial scales.

7 Conclusion

This study introduces GraphFlood, an efficient algorithm for solving 2D hydrodynamics based on 2D shallow water equations and specifically tailored for large DEMs. By employing Manning's equation within a graph theory framework, GraphFlood iteratively computes a stationary flow depth and discharge equilibrated to prescribed runoff rates. Leveraging graph theory algorithms ensures numerical efficiency, enabling GraphFlood to compute solutions for rivers in just seconds for a million-pixel DEM. Validation against analytical solutions and established models demonstrates the accuracy of GraphFlood. The simplicity, efficiency, and versatility of GraphFlood position it as a promising engine for incorporating 2D hydrodynamics into large-scale topographic analysis and landscape evolution models. Future work could utilize GraphFlood to investigate river inundation patterns, systematically extract river width as a function of water discharge, or focus on classifying landscapes to better relate landscape shape to geomorphological processes.

Code availability. The static version of the code used in this contribution can be found in Gailleton (2024). Updates on newer versions and more material will be posted on https://github.com/bgailleton/Gailleton_et_al_2024_GraphFlood_esurf .

Data availability. The DEM utilised in this study are openly available from opentopography.org under the datasets OpenTopography (2012) and OpenTopography (2020).

Appendix A: Notations

Table A1. Nomenclature for Scientific Notations

Notation	Meaning	Dimension	Unit
Physical quantities			
Z	Topogrpahic surface	$[L]$	m
Z_h	Hydraulic surface	$[L]$	m
s	Hydraulic slope	$[L/L]$	m/m
h	Flow Depth	$[L]$	m
u	Flow velocity	$[L/T]$	m/s
q	Water discharge per unit width	$[L^2/T]$	m^2/s
Q	Volumetric water discharge	$[L^3/T]$	m^3/s
P	Precipitation rate	$[L/T]$	m/s
α	Manning's exponent	-	-
n	Manning's friction coefficient	$[TL^{\alpha-1}]$	$sm^{\alpha-1}$
t	Time	$[T]$	s
$a_s(r)$	Effective drainage area for a runoff rate	$[L]$	m
W_r	Effective width	$[L]$	m
Discrete Quantities			
i	Generic index of cell	-	-
$donors(i)$	list of cell directly upstream of i	-	-
$receivers(i)$	list of cell directly downstream of i	-	-
h^*	Analytically determined flow Depth	$[L]$	m
q_{in}	Water discharge per unit width entering a cell	$[L^3/T]$	m^3/s
q_{out}	Water discharge per unit width leaving a cell	$[L^3/T]$	m^3/s
Q_{in}	Volumetric water discharge entering a cell	$[L^3/T]$	m^3/s
Q_{out}	Volumetric water discharge leaving a cell	$[L^3/T]$	m^3/s
A_c	Surface area of a single cell	$[L^2]$	m^2
A	D8 Drainage Area	$[L^2]$	m^2
ΔW	Flow width for a given cell	$[L^2]$	m^2
C	Correction factor for Multiple flow partitionning	-	-
C_r	Courant Number	-	-

Appendix B: Outliers in s - $a_s(r)$

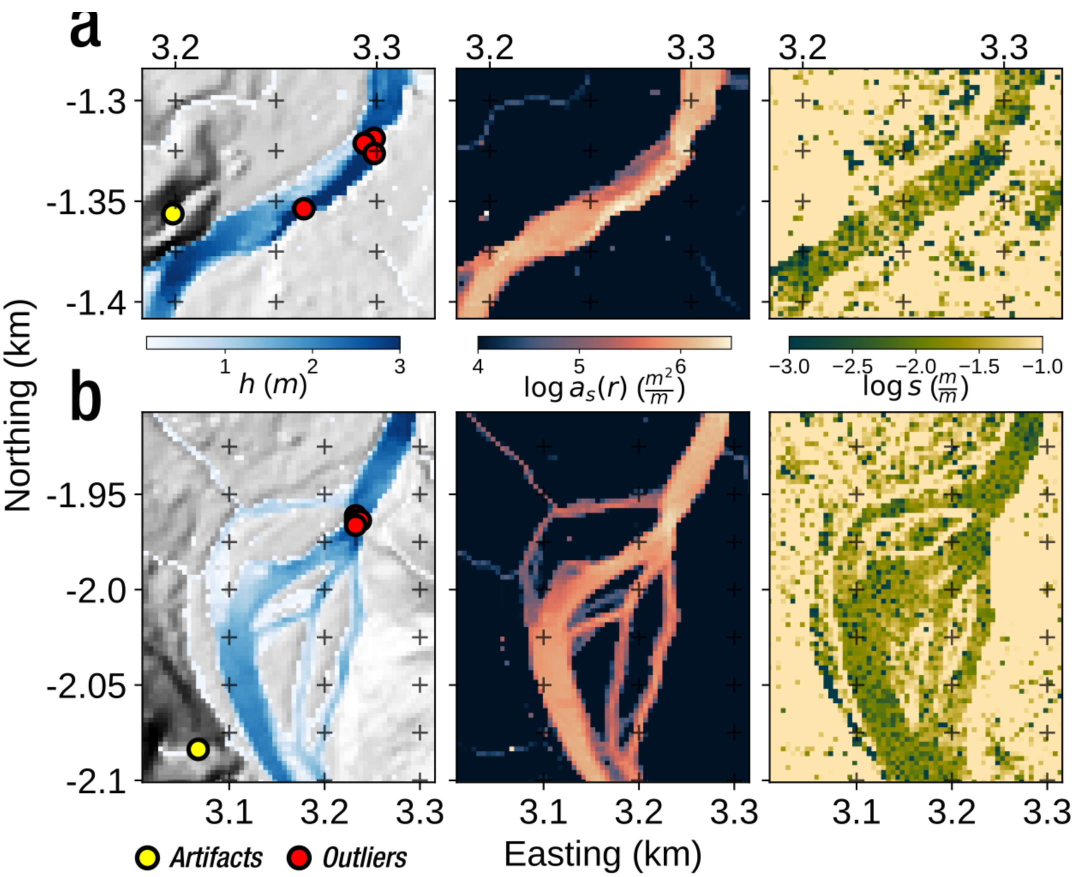


Figure A1. Zoom on some outliers for Hanalei test site, isolated using the data on figure 9. From the left to the right, the maps show the flow depth with the localisation of the outliers, the specific area and the hydraulic slope. Panel a displays outliers concentrating flow on narrowing section of the river or on its bends. Panel b shows the case of a converging branches. In both cases, outliers are accompanied with a slight increase in s.

Author contributions. PS and BG designed the concept of the GraphFlood algorithms. BG, PS, PD, TB and WS designed the study. BG wrote the code and ran the analysis. BG wrote the manuscript with the inputs of PS, WS, PD and TB.

Competing interests. Wolfgang Schwanghart is a member of the editorial board of Earth Surface Dynamics.

Acknowledgements. This research has been supported by the H2020 European Research Council (grant no. 803721). We thank Dimitri Lague, Guillaume Cordonnier, Ron Nativ, Fiona Clubb and Laure Guerit for constructive discussions, feedbacks and testing on GraphFlood. We thank two anonymous referees and handling editor Greg Hancock for constructive reviews that greatly improved the manuscript.

References

Adams, B. A., Whipple, K. X., Forte, A. M., Heimsath, A. M., and Hodges, K. V.: Climate controls on erosion in tectonically active landscapes, Science Advances, 6, https://doi.org/10.1126/sciadv.aaz3166, publisher: American Association for the Advancement of Science _eprint: https://advances.sciencemag.org/content/6/42/eaaz3166.full.pdf, 2020.

Anand, S. K., Hooshyar, M., and Porporato, A.: Linear layout of multiple flow-direction networks for landscape-evolution simulations, Environmental Modelling & Software, 133, 104 804, https://doi.org/10.1016/j.envsoft.2020.104804, 2020.

Apel, H., Vorogushyn, S., and Merz, B.: Brief communication: Impact forecasting could substantially improve the emergency management of deadly floods: case study July 2021 floods in Germany, Natural Hazards and Earth System Sciences, 22, 3005–3014, https://doi.org/10.5194/nhess-22-3005-2022, 2022.

Arcement, G. J. and Schneider, V. R.: Guide for selecting Manning's roughness coefficients for natural channels and flood plains, USGS Numbered Series 2339, U.S. G.P.O. ; For sale by the Books and Open-File Reports Section, U.S. Geological Survey,, https://doi.org/10.3133/wsp2339, code Number: 2339 Code: Guide for selecting Manning's roughness coefficients for natural channels and flood plains Publication Title: Guide for selecting Manning's roughness coefficients for natural channels and flood plains Reporter: Guide for selecting Manning's roughness coefficients for natural channels and flood plains Series: Water Supply Paper, 1989.

Armitage, J. J.: Short communication: Flow as distributed lines within the landscape, Earth Surface Dynamics, 7, 67–75, https://doi.org/10.5194/esurf-7-67-2019, publisher: Copernicus GmbH, 2019.

Barnes, R., Lehman, C., and Mulla, D.: Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models, Computers and Geosciences, 62, 117–127, https://doi.org/10.1016/j.cageo.2013.04.024, publisher: Pergamon _eprint: 1511.04463, 2014.

Barnes, R., Callaghan, K. L., and Wickert, A. D.: Computing water flow through complex landscapes – Part 3: Fill–Spill–Merge: flow routing in depression hierarchies, Earth Surface Dynamics, 9, 105–121, https://doi.org/10.5194/esurf-9-105-2021, publisher: Copernicus GmbH, 2021.

Barnhart, K. R., Hutton, E. W. H., Tucker, G. E., Gasparini, N. M., Istanbulluoglu, E., Hobley, D. E. J., Lyons, N. J., Mouchene, M., Nudurupati, S. S., Adams, J. M., and Bandaragoda, C.: Short communication: Landlab v2.0: a software package for Earth surface dynamics, Earth Surface Dynamics, 8, 379–397, https://doi.org/10.5194/esurf-8-379-2020, publisher: Copernicus GmbH, 2020.

Bates, P. D.: Flood Inundation Prediction, Annual Review of Fluid Mechanics, 54, 287–315, https://doi.org/10.1146/annurev-fluid-030121-113138, _eprint: https://doi.org/10.1146/annurev-fluid-030121-113138, 2022.

Bates, P. D., Horritt, M. S., and Fewtrell, T. J.: A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling, Journal of Hydrology, 387, 33–45, https://doi.org/10.1016/j.jhydrol.2010.03.027, 2010.

Baynes, E. R., Lague, D., Steer, P., and Davy, P.: Dynamic bedrock channel width during knickpoint retreat enhances undercutting of coupled hillslopes, Earth Surface Processes and Landforms, 47, 3629–3640, https://doi.org/10.1002/esp.5477, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/esp.5477, 2022.

Bernard, T.: Analyse haute résolution de la morphologie des paysages et des processus à partir de LiDAR aéroporté répété et simulation hydraulique, These de doctorat, Rennes 1, https://www.theses.fr/2022REN1B011, 2022.

Bernard, T. G., Davy, P., and Lague, D.: Hydro-Geomorphic Metrics for High Resolution Fluvial Landscape Analysis, Journal of Geophysical Research: Earth Surface, 127, e2021JF006 535, https://doi.org/10.1029/2021JF006535, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2021JF006535, 2022.

Braun, J. and Sambridge, M.: Modelling landscape evolution on geological time scales: a new method based on irregular spatial discretization, Basin Research, 9, 27–52, https://doi.org/https://doi.org/10.1046/j.1365-2117.1997.00030.x, 1997.

Braun, J. and Willett, S. D.: A very efficient O(n), implicit and parallel method to solve the stream power equation governing fluvial incision and landscape evolution, Geomorphology, 180-181, 170–179, https://doi.org/10.1016/j.geomorph.2012.10.008, 2013.

Brunner, G. W.: Hec-ras (river analysis system), in: North American water and environment congress & destructive water, pp. 3782–3787, ASCE, 2002.

Campforts, B., Schwanghart, W., and Govers, G.: Accurate simulation of transient landscape evolution by eliminating numerical diffusion: The TTLEM 1.0 model, Earth Surface Dynamics, 5, 47–66, https://doi.org/10.5194/esurf-5-47-2017, 2017.

Carretier, S., Martinod, P., Reich, M., and Godderis, Y.: Modelling sediment clasts transport during landscape evolution, Earth Surface Dynamics, 4, 237–251, https://doi.org/10.5194/esurf-4-237-2016, publisher: Copernicus GmbH, 2016.

Clubb, F. J., Mudd, S. M., Milodowski, D. T., Hurst, M. D., and Slater, L. J.: Objective extraction of channel heads from high-resolution topographic data, Water Resources Research, 50, 4283–4304, https://doi.org/10.1002/2013WR015167, iSBN: 9780771428975 _eprint: 2014WR016527, 2014.

Clubb, F. J., Mudd, S. M., Hurst, M. D., and Grieve, S. W.: Differences in channel and hillslope geometry record a migrating uplift wave at the Mendocino triple junction, California, USA, Geology, 48, 184–188, https://doi.org/10.1130/G46939.1, 2019.

Clubb, F. J., Weir, E. F., and Mudd, S. M.: Continuous measurements of valley floor width in mountainous landscapes, Earth Surface Dynamics, 10, 437–456, https://doi.org/10.5194/esurf-10-437-2022, publisher: Copernicus GmbH, 2022.

Clubb, F. J., Mudd, S. M., Schildgen, T. F., van der Beek, P. A., Devrani, R., and Sinclair, H. D.: Himalayan valley-floor widths controlled by tectonically driven exhumation, Nature Geoscience, 16, 739–746, https://doi.org/10.1038/s41561-023-01238-8, number: 8 Publisher: Nature Publishing Group, 2023.

Cordonnier, G., Bovy, B., and Braun, J.: A Versatile, Linear Complexity Algorithm for Flow Routing in Topographies with Depressions, Earth Surface Dynamics Discussions, 7, 1–18, https://doi.org/10.5194/esurf-2018-81, publisher: Copernicus GmbH, 2018.

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, Journal of Hydrology, 599, 126 306, https://doi.org/10.1016/j.jhydrol.2021.126306, 2021.

Costabile, P., Costanzo, C., De Bartolo, S., Gangi, F., Macchione, F., and Tomasicchio, G. R.: Hydraulic Characterization of River Networks Based on Flow Patterns Simulated by 2-D Shallow Water Modeling: Scaling Properties, Multifractal Interpretation, and Perspectives for Channel Heads Detection, Water Resources Research, 55, 7717–7752, https://doi.org/10.1029/2018WR024083, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2018WR024083, 2019.

Coulthard, T. J. and Van De Wiel, M. J.: Modelling long term basin scale sediment connectivity, driven by spatial land use changes, Geomorphology, 277, 265–281, https://doi.org/10.1016/j.geomorph.2016.05.027, 2017.

Coulthard, T. J., Neal, J. C., Bates, P. D., Ramirez, J., de Almeida, G. A. M., and Hancock, G. R.: Integrating the LISFLOOD-FP 2D hydrodynamic model with the CAESAR model: implications for modelling landscape evolution, Earth Surface Processes and Landforms, 38, 1897–1906, https://doi.org/10.1002/esp.3478, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/esp.3478, 2013.

Davy, P. and Lague, D.: Fluvial erosion/transport equation of landscape evolution models revisited, Journal of Geophysical Research: Earth Surface, 114, https://doi.org/10.1029/2008JF001146, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2008JF001146, 2009.

Davy, P., Croissant, T., and Lague, D.: A precipiton method to calculate river hydrodynamics, with applications to flood prediction, landscape evolution models, and braiding instabilities, Journal of Geophysical Research: Earth Surface, 122, 1491–1512, https://doi.org/10.1002/2016JF004156, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/2016JF004156, 2017.

de Almeida, G. A. M., Bates, P., Freer, J. E., and Souvignet, M.: Improving the stability of a simple formulation of the shallow water equations for 2-D flood modeling, Water Resources Research, 48, https://doi.org/10.1029/2011WR011570, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2011WR011570, 2012.

DiBiase, R. A., Whipple, K. X., Heimsath, A. M., and Ouimet, W. B.: Landscape form and millennial erosion rates in the San Gabriel Mountains, CA, Earth and Planetary Science Letters, 289, 134–144, https://doi.org/10.1016/j.epsl.2009.10.036, publisher: Elsevier, 2010.

Dunne, K. B. J. and Jerolmack, D. J.: What sets river width?, Science Advances, 6, eabc1505, https://doi.org/10.1126/sciadv.abc1505, publisher: American Association for the Advancement of Science, 2020.

Flint, J. J.: Stream gradient as a function of order, magnitude, and discharge, Water Resources Research, 10, 969–973, https://doi.org/10.1029/WR010i005p00969, iSBN: 0043-1397, 1974.

Gailleton, B.: Supporting code for GraphFlood 1.0: an efficient algorithm to approximate 2D hydrodynamics for Landscape Evolution Models, https://doi.org/10.5281/zenodo.11065794, 2024.

Gailleton, B. and Mudd, S.: Lsdtopotools/lsdtopytools: lsdtopytools, 2021.

Gailleton, B., Mudd, S. M., Clubb, F. J., Peifer, D., and Hurst, M. D.: A segmentation approach for the reproducible extraction and quantification of knickpoints from river long profiles, Earth Surface Dynamics, 7, 211–230, https://doi.org/10.5194/esurf-7-211-2019, publisher: Copernicus GmbH, 2019.

Gailleton, B., Mudd, S. M., Clubb, F. J., Grieve, S. W. D., and Hurst, M. D.: Impact of Changing Concavity Indices on Channel Steepness and Divide Migration Metrics, Journal of Geophysical Research: Earth Surface, 126, e2020JF006 060, https://doi.org/10.1029/2020JF006060, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2020JF006060, 2021.

Gailleton, B., Malatesta, L., Cordonnier, G., and Braun, J.: CHONK 1.0: landscape evolution framework: cellular automata meets graph theory, EGUsphere, 2023, 1–31, https://doi.org/10.5194/egusphere-2022-1394, 2023.

Gallant, J. C. and Hutchinson, M. F.: A differential equation for specific catchment area, Water Resources Research, 47, https://doi.org/10.1029/2009WR008540, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2009WR008540, 2011.

GDAL/OGR contributors: GDAL/OGR Geospatial Data Abstraction software Library, Open Source Geospatial Foundation, https://doi.org/10.5281/zenodo.5884351, 2023.

Grieve, S. W., Mudd, S. M., and Hurst, M. D.: How long is a hillslope?, Earth Surface Processes and Landforms, 41, 1039–1054, https://doi.org/10.1002/esp.3884, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/esp.3884, 2016.

Grieve, S. W. D., Hales, T. C., Parker, R. N., Mudd, S. M., and Clubb, F. J.: Controls on Zero-Order Basin Morphology, Journal of Geophysical Research: Earth Surface, 123, 3269–3291, https://doi.org/10.1029/2017JF004453, publisher: Blackwell Publishing Ltd, 2018.

Heckmann, T., Schwanghart, W., and Phillips, J. D.: Graph theory—Recent developments of its application in geomorphology, Geomorphology, 243, 130–146, https://doi.org/10.1016/j.geomorph.2014.12.024, 2015.

Hergarten, S.: Transport-limited fluvial erosion – simple formulation and efficient numerical treatment, Earth Surface Dynamics, 8, 841–854, https://doi.org/10.5194/esurf-8-841-2020, publisher: Copernicus GmbH, 2020.

Hergarten, S. and Neugebauer, H. J.: Self-Organized Critical Drainage Networks, Physical Review Letters, 86, 2689–2692, https://doi.org/10.1103/PhysRevLett.86.2689, publisher: American Physical Society, 2001.

Hurst, M. D., Grieve, S. W., Clubb, F. J., and Mudd, S. M.: Detection of channel-hillslope coupling along a tectonic gradient, Earth and Planetary Science Letters, 522, 30–39, https://doi.org/10.1016/j.epsl.2019.06.018, 2019.

Kirby, E. and Whipple, K. X.: Expression of active tectonics in erosional landscapes, vol. 44, https://doi.org/10.1016/j.jsg.2012.07.009, iSSN: 01918141 Publication Title: Journal of Structural Geology, 2012.

Leonard, J. S., Whipple, K. X., and Heimsath, A. M.: Isolating climatic, tectonic, and lithologic controls on mountain landscape evolution, Science Advances, 9, eadd8915, https://doi.org/10.1126/sciadv.add8915, publisher: American Association for the Advancement of Science, 2023.

Liu, B. and Coulthard, T. J.: Mapping the interactions between rivers and sand dunes: Implications for fluvial and aeolian geomorphology, Geomorphology, 231, 246–257, https://doi.org/10.1016/j.geomorph.2014.12.011, 2015.

Liu, B. and Coulthard, T. J.: Modelling the interaction of aeolian and fluvial processes with a combined cellular model of sand dunes and river systems, Computers & Geosciences, 106, 1–9, https://doi.org/10.1016/j.cageo.2017.05.003, 2017.

Lurin, A., Marc, O., Meunier, P., and Carretier, S.: A Robust Channel Head Extraction Method Based on High-Resolution Topographic Convergence, Suitable for Both Slowly and Fastly Eroding Landscapes, Journal of Geophysical Research: Earth Surface, 128, https://doi.org/10.1029/2022JF006999, 2023.

Manning, R., Griffith, J. P., Pigot, T., and Vernon-Harcourt, L. F.: On the flow of water in open channels and pipes, 1890.

Minor, M., Davy, P., Howarth, J., and Lague, D.: Multi Grain-Size Total Sediment Load Model Based on the Disequilibrium Length, Journal of Geophysical Research: Earth Surface, 127, https://doi.org/10.1029/2021JF006546, 2022.

Montgomery, D. R.: Slope distributions, threshold hillslopes, and steady-state topography, American Journal of science, 301, 432–454, 2001.

Morisawa, M. E.: Quantitative Geomorphology of Some Watersheds in the Appalachian Plateau, GSA Bulletin, 73, 1025–1046, https://doi.org/10.1130/0016-7606(1962)73[1025:QGOSWI]2.0.CO;2, 1962.

Mudd, S. M., Clubb, F. J., Gailleton, B., and Hurst, M. D.: How concave are river channels ?, Earth Surface Dynamics Discussions, 25, 1–34, https://doi.org/10.5194/esurf-2018-7, 2018.

Mudd, S. M., Clubb, F. J., Grieve, S. W. D., Milodowski, D. T., Hurst, M. D., Gailleton, B., and Valters, D. A.: LSDTopoTools2, https://doi.org/10.5281/ZENODO.3245041, 2019.

O'Callaghan, J. F. and Mark, D. M.: The extraction of drainage networks from digital elevation data., Computer Vision, Graphics, & Image Processing, 28, 323–344, https://doi.org/10.1016/S0734-189X(84)80011-0, publisher: Elsevier, 1984.

OpenTopography: Hawaii Kauai Survey, https://doi.org/10.5069/G91V5BWJ, 2012.

OpenTopography: Interpreting Fluvial Processes from Channel-Belt Deposits, Utah 2018, https://doi.org/10.5069/G9J964J3, 2020.

Pelletier, J. D.: Quantitative Modeling of Earth Surface Processes, Cambridge University Press, 2008.

Pelletier, J. D.: A robust, two-parameter method for the extraction of drainage networks from high-resolution digital elevation models (DEMs): Evaluation using synthetic and real-world DEMs, Water Resources Research, 49, 75–89, https://doi.org/10.1029/2012WR012452, 2013.

Perron, J. T.: Numerical methods for nonlinear hillslope transport laws, Journal of Geophysical Research: Earth Surface, 116, https://doi.org/10.1029/2010JF001801, _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2010JF001801, 2011.

Roelvink, J. A. and Banning, G. K. F. M. V.: Design and development of DELFT3D and application to coastal morphodynamics, Oceanographic Literature Review, 11, 925, https://www.infona.pl//resource/bwmeta1.element.elsevier-1ca19bb6-25b9-3bf5-bfe9-e96a7027c553, 1995.

Salles, T., Husson, L., Rey, P., Mallard, C., Zahirovic, S., Boggiani, B. H., Coltice, N., and Arnould, M.: Hundred million years of landscape dynamics from catchment to global scale, Science, 379, 918–923, 2023.

Schumm, S. A., Dumont, J. F., and Holbrook, J. M.: Active Tectonics and Alluvial Rivers, Cambridge University Press, Cambridge, UK ; New York, NY, 2000.

Schuurman, F., Marra, W. A., and Kleinhans, M. G.: Physics-based modeling of large braided sand-bed rivers: Bar pattern formation, dynamics, and sensitivity, Journal of Geophysical Research: Earth Surface, 118, 2509–2527, https://doi.org/10.1002/2013JF002896, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/2013JF002896, 2013.

Schwanghart, W. and Scherler, D.: Short Communication: TopoToolbox 2 - MATLAB-based software for topographic analysis and modeling in Earth surface sciences, Earth Surface Dynamics, 2, 1–7, https://doi.org/10.5194/esurf-2-1-2014, 2014.

Schwanghart, W. and Scherler, D.: Bumps in river profiles: Uncertainty assessment and smoothing using quantile regression techniques, Earth Surface Dynamics, 5, 821–839, https://doi.org/10.5194/esurf-5-821-2017, 2017.

Schwanghart, W., Molkenthin, C., and Scherler, D.: A systematic approach and software for the analysis of point patterns on river networks, Earth Surface Processes and Landforms, 46, 1847–1862, https://doi.org/10.1002/esp.5127, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/esp.5127, 2021.

Stammberger, V., Jacobs, B., and Krautblatter, M.: Hyperconcentrated flows shape bedrock channels, Communications Earth & Environment, 5, 1–15, https://doi.org/10.1038/s43247-024-01353-3, publisher: Nature Publishing Group, 2024.

Steer, P., Guerit, L., Lague, D., Crave, A., and Gourdon, A.: Size, shape and orientation matter: fast and semi-automatic measurement of grain geometries from 3D point clouds, Earth Surface Dynamics, 10, 1211–1232, https://doi.org/10.5194/esurf-10-1211-2022, publisher: Copernicus GmbH, 2022.

Tarboton, D. G.: A new method for the determination of flow directions and upslope areas in grid digital elevation models, Tech. Rep. 2, https://doi.org/10.1029/96WR03137, publication Title: Water Resources Research Volume: 33, 1997.

Van De Wiel, M. J. and Coulthard, T. J.: Self-organized criticality in river basins: Challenging sedimentary records of environmental change, Geology, 38, 87–90, https://doi.org/10.1130/G30490.1, 2010.

Vanzo, D., Peter, S., Vonwiller, L., Bürgler, M., Weberndorfer, M., Siviglia, A., Conde, D., and Vetsch, D. F.: basement v3: A modular freeware for river process modelling over multiple computational backends, Environmental Modelling & Software, 143, 105 102, https://doi.org/10.1016/j.envsoft.2021.105102, 2021.

Villaret, C., Hervouet, J.-M., Kopmann, R., Merkel, U., and Davies, A. G.: Morphodynamic modeling using the Telemac finite-element system, Computers & Geosciences, 53, 105–113, https://doi.org/10.1016/j.cageo.2011.10.004, 2013.

Whipple, K. X. and Tucker, G. E.: Dynamics of the stream-power river incision model: Implications for height limits of mountain ranges, landscape response timescales, and research needs, Journal of Geophysical Research: Solid Earth, 104, 17 661–17 674, https://doi.org/10.1029/1999JB900120, 1999.

Whipple, K. X., DiBiase, R. A., and Crosby, B. T.: Bedrock Rivers, in: Treatise on Geomorphology, vol. 9, pp. 550–573, Fluvial Geomorphology, https://doi.org/10.1016/B978-0-12-374739-6.00254-2, 2013.

Willett, S. D.: Orogeny and orography: The effects of erosion on the structure of mountain belts, Journal of Geophysical Research: Solid Earth, 104, 28 957–28 981, 1999.

Willett, S. D., McCoy, S. W., Taylor Perron, J., Goren, L., and Chen, C. Y.: Dynamic reorganization of River Basins, Science, 343, 1248 765, https://doi.org/10.1126/science.1248765, iSBN: 1095-9203 (Electronic)$\backslash$r0036-8075 (Linking), 2014.

Willgoose, G., Bras, R., and Rodriguez-Iturbe, I.: Hydrogeomorphology modelling with a physically based river basin evolution model, 1994.

Wobus, C., Whipple, K. X., Kirby, E., Snyder, N., Johnson, J., Spyropolou, K., Crosby, B., and Sheehan, D.: Tectonics from topography: procedurses, promise, and pitfalls, Geological Society of America Special Paper, 398, 55–74, https://doi.org/10.1130/2006.2398(04)., 2006.

Yu, D. and Coulthard, T. J.: Evaluating the importance of catchment hydrological parameters for urban surface water flood modelling using a simple hydro-inundation model, Journal of Hydrology, 524, 385–400, https://doi.org/10.1016/j.jhydrol.2015.02.040, 2015.