Efficient Debris-flow Simulation for Steep Terrain Erosion

ARYAMAAN JAIN, Inria, Université Côte d'Azur, France
BEDRICH BENES, Computer Science, Purdue University, USA
GUILLAUME CORDONNIER, Inria, Université Côte d'Azur, France

Erosion simulation is a common approach used for generating and authoring mountainous terrains. While water is considered the primary erosion factor, its simulation fails to capture steep slopes near the ridges. In these low-drainage areas, erosion is often approximated with slope-reducing erosion, which yields unrealistically uniform slopes. However, geomorphology observed that another process dominates the low-drainage areas: erosion by debris flow, which is a mixture of mud and rocks triggered by strong climatic events. We propose a new method to capture the interactions between debris flow and fluvial erosion thanks to a new mathematical formulation for debris flow erosion derived from geomorphology and a unified GPU algorithm for erosion and deposition. In particular, we observe that sediment and debris deposition tend to intersect river paths, which motivates the design of a new, approximate flow routing algorithm on the GPU to estimate the water path out of these newly formed depressions. We demonstrate that debris flow carves distinct patterns in the form of erosive scars on steep slopes and cones of deposited debris competing with fluvial erosion downstream.

CCS Concepts: • **Computing methodologies** → **Shape modeling**.

Additional Key Words and Phrases: Terrain, Erosion, Deposition, Debris Flow, Simulation, Flow Routing

ACM Reference Format:
Aryamaan Jain, Bedrich Benes, and Guillaume Cordonnier. 2024. Efficient Debris-flow Simulation for Steep Terrain Erosion. *ACM Trans. Graph.* 43, 4, Article 58 (July 2024), 11 pages. https://doi.org/10.1145/3658213

1 INTRODUCTION

Natural landscapes are fascinating because of their monumental presence and striking diversity. Our deepening understanding of *orogeny* – the geological origins of mountainous landscapes – allows us now to simulate the emergence of consistent mountain ranges.

One of such examples is fluvial erosion, which is inspired by geomorphology and successfully applied to computer graphics to generate massive ranges at scales up to thousands of kilometers with simple and efficient simulation models [Cordonnier et al. 2016; Schott et al. 2023]. However, the drawback of using a simple model becomes apparent when fluvial erosion is used at intermediate scales (*e.g.,* a few valleys of up to 10 km), where the generated terrain is smooth and uniform [Perron et al. 2009], and does not exhibit high-frequency landforms.

We observe typical erosion patterns in high valleys and ridges that emerge from another erosion process. In these *low-drainage* regions where the action of water alone is not sufficient to carve significant landforms, a mixture of mud and rocks called *debris flow* [Iverson 1997] is occasionally triggered by exceptional climatic events such as downpours, and flows down with an extreme erosive capacity.

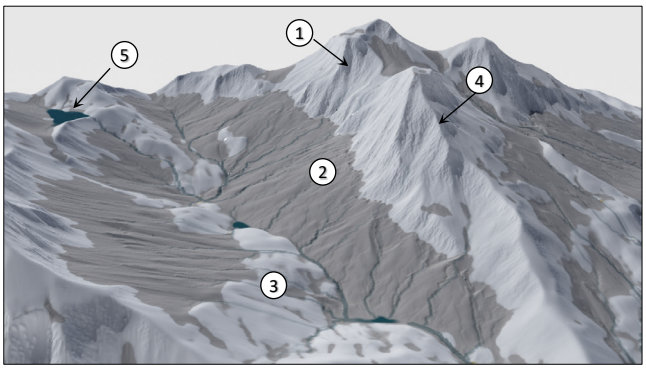


Fig. 1. A landscape carved by simulated erosion and deposition. (1) Deep scars of debris flow erosion, (2) deposit fan, (3) low-slope fluvially shaped hill, (4) sharp ridges carved by debris flow erosion, (5) lake found by our depression routing algorithm.

Improving upon the work on *fluvial erosion*, validated at what we call the *mountain range scale* (with a cell size Δx above $50 - 100\,m$), we now focus on a *valley scale* (Δx around $5 - 30\,m$), where we observe different erosion patterns depending on the distance to the ridges [Neely and DiBiase 2023]. Debris flow leaves sharp erosive scars on steep mountain slopes while, farther, fluvial erosion flattens the main valleys. Fluvial erosion competes with debris flow deposition at the bottom of the steepest slopes, yielding a smooth transition between the two erosion regimes. To capture these intricate details (Figure 1), we propose a new unified erosion and deposition simulation algorithm that integrates fluvial processes and debris flow. We explore the interaction between these two processes and provide additional control parameters such as vegetation cover. We observe that deposition leaves several *depressions* – or local minima in the topography – that severely reduce the ability of our algorithm to transport water and sediments. While several precise yet slow solutions to this problem exist in the CPU, we propose the first approximate algorithm on the GPU to compute a path out of the depressions. Eventually, we present how we derive our model for debris flow from different geological observations and show in particular that debris flow erosion generalizes the slope limiting effect of *thermal erosion* (a term mainly found in graphics, defined by Musgrave et al. [1989] as "a catch-all term for any process that knocks material loose, which material then falls down to pile up at the bottom of an incline").

Authors' addresses: Aryamaan Jain, aryamaan.jain@inria.fr, Inria, Université Côte d'Azur, Sophia-Antipolis, France; Bedrich Benes, bbenes@purdue.edu, Computer Science, Purdue University, West Lafayette, USA; Guillaume Cordonnier, guillaume.cordonnier@inria.fr, Inria, Université Côte d'Azur, Sophia-Antipolis, France.

In summary, our main contributions are 1) A novel model for debris flow erosion and deposition derived from geomorphological observations; 2) A unified GPU erosion and deposition simulation that encompasses both fluvial and debris flow processes; and 3) A GPU algorithm to approximate flow paths in topographies with depressions.

2 RELATED WORK

While there exists a wide variety of methods to model terrain in computer graphics, spanning from procedural to data-driven or simulations [Galin et al. 2019], here we will focus on simulations for their capacity to produce physically consistent landscapes from little data.

Hydraulic erosion. The first physically-based terrain generation methods observed the strong impact of water incision in landscape morphologies and proposed hydraulic erosion as a solution to improve the realism of procedural terrains [Musgrave et al. 1989]. These methods simulate a shallow layer of water running over the terrain with approximated flow dynamics [Benes 2007; Krištof et al. 2009] and model the erosive interactions between the water and the underlying rocks [Wojtan et al. 2007]. Hydraulic erosion is easily parallelizable on the GPU [Vanek et al. 2011] and was improved with layered-based terrain representations, yielding strata of various erosion resistance [Roudier et al. 1993] or enabling sediment deposition [Benes and Forsbach 2001]. While the interactions between water dynamics and erosion are key to the physical realism of hydraulic erosion – particularly striking when modeling natural scenes at the scale of the river bed –, they also limit the spatial and temporal scales to the ones of the water dynamics. This limitation prevents the use of hydraulic erosion for large-scale terrains or forces inconsistent up-scaling.

Fluvial erosion. In contrast, fluvial erosion borrowed from geomorphology [Braun and Willett 2013; Whipple and Tucker 1999] abstracts water dynamics into a simpler term, the *drainage area*, which acts as a proxy for the water discharge. With this simplification, the numerical constraints come from the erosion itself, which enabled the generation of large-scale mountain ranges [Cordonnier et al. 2016]. At this scale, it is impossible to neglect the effect of tectonic forces that combine with erosion to shape the landscapes. Tectonic uplift – or the growth rate of mountains – was therefore used as a control mechanism through a sculpting metaphor [Cordonnier et al. 2017] or a GPU implementation [Schott et al. 2023]. However, fluvial erosion is designed and validated for large-scale landscapes, and attempts at reducing the resolution do not bring additional details and fail to carve steep slopes convincingly. This was particularly visible in a recent work on glacial erosion [Cordonnier et al. 2023] where the authors observed that fluvial erosion could not realistically erode the steep cliffs left after the retreat of glaciers and instead proposed a first empirical approximation for debris flow. We build upon this idea, propose a new geologically-based debris flow model, and couple it with sediment deposition.

Thermal erosion. The need for an erosion term specific to the steep slopes left at the banks of eroding rivers was already observed from the original work of Musgrave et al. [1989], who introduced thermal erosion, named from the assumption that thermal shocks unseal rocks that falls when above a repose angle, and settle downslope. We will show that debris flow erosion is a generalization of thermal erosion and enhance the uniform slopes left by thermal erosion with intricate erosion scars.

Debris flow in geology. Debris flows are well known in geomorphology, but mostly for their immediate threat to human properties. Rocks and sediments, loosely attached to steep slopes, mix with water during strong storms and suddenly detach and flow as a destructive avalanche of mud and stones [Takahashi 1981]. Therefore, several studies emerged on the simulation of the triggering, avalanche, and deposition of individual debris flow events [Hutter et al. 1994; Takahashi 1978] with a strong focus on accurate and predictable physics [Iverson 1997; Xia et al. 2023]. However, these models are too complex to be integrated in long-term periods to account for the accumulated erosion of thousands of debris flow events. The observation that debris flow erosion is critical in steep low drainage area [Stock and Dietrich 2006] motivated the formulation of simpler models to simulate the evolution of steep landscapes [McGuire et al. 2022]. We extend these laws, originally adapted to geomorphological experiments in 1D convex valleys, and adapt them to the broader diversity of terrains encountered in computer graphics.

3 OVERVIEW

Our focus is on debris flow, which cannot be modeled independently from other processes responsible for mountain formation. Therefore, we briefly review the geological origins and equations of all processes we consider.

Mountain formation from geomorphology. Mountains emerge from the mutual influence of various geological processes (Figure 2). The collision of tectonic plates induces the compression of the Earth's crust accommodated by multi-scale folding and faulting [Willett et al. 1993]. The ensuing vertical growth rate is called the *uplift* $\mathcal{U}$ and has been used in computer graphics to control the overall shape of the mountain [Cordonnier et al. 2017; Schott et al. 2023].

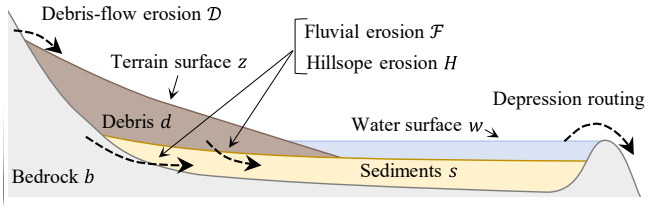


Fig. 2. We model interleaved erosion processes, which affect different terrain layers: debris flow erosion turns bedrock b into debris d, while hillslope and fluvial erosion turn bedrock and debris into sediments s. Depression routing allows us to compute the path of water and transported material out of local minima and deduce the water surface w.

Counteracting the effects of uplift, rainwater flows over the terrain, accumulates, erodes it, and forms valleys. This effect is modeled with *fluvial erosion* $\mathcal{F}$ that explains at large scale the distribution of mountains and valleys [Whipple and Tucker 1999]. We extend

the computer graphics implementations of the stream power fluvial erosion [Cordonnier et al. 2016; Schott et al. 2023] with sediment deposition, borrowing a law from geomorphology [Yuan et al. 2019]:

$$\mathcal{F} = -k_f\, Q_w^m \|\nabla z\|^n + k_d\, \frac{Q_s}{Q_w}, \tag{1}$$

where k_f and k_d are erosion and deposition constants, $m = 0.4$ and $n = 1$ are power law exponents, ∇z is the terrain surface gradient (with respect to horizontal coordinates, which means that $\|\nabla z\|$ is the terrain slope), Q_w (units $\mathrm{m^3 y^{-1}}$) is the water volumetric flow rate or discharge, and Q_s is the sediment volumetric flow rate. In particular, Q_w and Q_s are computed from the upstream domain A, which is defined, at a point $\mathbf{x}$, as the set of points $\mathbf{y}$ *upstream* from $\mathbf{x}$, *i.e.,* there exists a path of strictly increasing elevations between $\mathbf{x}$ and $\mathbf{y}$. Note that a rigorous definition of A should be relative to a channel width instead of a single point, but this single point approximation is valid under the assumption that the channel width is lower than then cell size [Yuan et al. 2019]. Then, the discharge is the upstream sum of precipitation p:

$$Q_w = \iint_A p\, \mathrm{d}x^2. \tag{2}$$

Assuming $p = 1$, Eq. 2 gives the drainage area, which, with the leftmost part of Eq. 1, yields the common expression for the stream power erosion. The sediment volumetric flow rate is given by the total upstream balance of fluvial erosion:

$$Q_s = -\iint_A \mathcal{F}\, \mathrm{d}x^2. \tag{3}$$

The upstream balance of fluvial erosion is always negative (the deposition amount is limited by upstream erosion). Therefore, Q_s is always positive.

Fluvial erosion cannot explain the erosion of low-drainage regions close to the ridges [Lague and Davy 2003]. In these regions, material creep flow tends to smoothen the ridges, a phenomenon modeled with *hillslope erosion* $\mathcal{H}$ [Braun and Sambridge 1997]:

$$\mathcal{H} = k_h\, \Delta z, \tag{4}$$

where k_h is a hillslope erosion coefficient, and Δz is the Laplacian of the terrain surface, which we discretize explicitly in time with a standard 4-neighbor Laplacian kernel.

In steeper regions, slopes are subject to debris flow, which yields a specific and intense erosion [Stock and Dietrich 2006]. We will propose a new model for debris flow erosion and deposition $\mathcal{D}$ in Section 4, and take advantage of the similarities between this model and fluvial erosion (Section 1) to model them through a unified algorithm in Section 5. In particular, our algorithm computes the propagation of water, sediment, and debris along flow paths, often interrupted by *depressions* or local minima in the topography [Scott and Dodgson 2021]. This interruption is not realistic, as, in nature, the water accumulates, forms a pond or a lake, and eventually outflows. Therefore, in Section 6, we propose an approximate GPU algorithm to compute the path of water and sediments out of depressions.

Multi-layered deposition. We handle sediment and debris deposition thanks to a layered terrain structure [Benes and Forsbach 2001]. Our heightmap embeds the bedrock altitude b. The sediment and debris thicknesses s and d, and their aggregation gives the surface elevation $z = b + s + d$. Erosion acts directly on the terrain surface (with the notation $\square^{-/+} = \min/\max(0, \square)$):

$$\frac{\partial z}{\partial t} = \mathcal{F}^- + \mathcal{D}^- + \mathcal{H}^-, \tag{5}$$

and then any decrease in elevation δz is propagated, first to d and s in proportion (*e.g.,* $\delta d = d/(d+s)\delta z$), then to the bedrock if both d and s vanish. In contrast, deposition and uplift directly act on the corresponding layers:

$$\frac{\partial b}{\partial t} = \mathcal{U}\,;\ \frac{\partial s}{\partial t} = \mathcal{F}^+ + \mathcal{H}^+ \text{ and } \frac{\partial d}{\partial t} = \mathcal{D}^+. \tag{6}$$

Depression routing also allows us to compute a water surface altitude w used to render and compute deposition inside lakes.

4 A NEW DEBRIS-FLOW EROSION MODEL

We propose a new expression for the erosion of steep slopes by debris flow. In nature, debris flow erosion results in constant slope profiles in low-drainage regions [Stock and Dietrich 2006], similar to thermal erosion [Musgrave et al. 1989], but it also leaves small erosion channels indicating a relationship between erosion and the local accumulation of flowing debris (Figure 3, left). Downstream, debris flows lose energy and stabilize, leaving typical cones of deposited sediments (Figure 1).

Fig. 3. Left: landscape carved by debris flow in the Pyrenees (capture from Google Earth of the Pyrenees. Data: Airbus.) Right: close-up view of a debris-flow channel in the Himalayas (©Dan Hobley, Creative Commons).

We propose a novel expression of debris flow erosion that accounts for these three different effects:

$$\mathcal{D} = -E_{th} - E_{df} + D_{df}, \tag{7}$$

where E_{th} corresponds to the amount of loosening rocks that effectively trigger a debris flow, E_{df} the terrain erosion from the abrasion by debris flows, and D_{df} the downstream deposition.

Debris flow starts from a slope failure, described by a Coulomb criterion [Iverson 1997]. Therefore, we model the triggering of debris flow with an equation similar to thermal erosion:

$$E_{th} = k_{th}(\|\nabla z\| - \tan\theta_c)^+, \tag{8}$$

where k_{th} is an erosion coefficient and $\theta_c \approx 30^\circ$ is a critical *talus angle*.

Unsealed rocks and soaked sediments merge to form a fluid mixture that flows downslope with a strong abrasion capacity (Figure 3, right). Following previous work in geomorphology [McGuire et al. 2022], we express this abrasion as a power law:

$$E_{df} = k'_{df} t_{df} \|\nabla z\|^{\alpha'} h^{\beta'} \Theta, \tag{9}$$

where k'_{df}, α' and β' are erosion constants, t_{df}, is the time during which a single debris flow event actively flows at a given point, $\|\nabla z\|$ is the terrain surface gradient (the slope), and h is the debris flow thickness. Friction at the bed interface slows the debris flows until they stabilize in gentle slopes. We model this effect and the accompanying reduction in erosive power with the threshold factor:

$$\Theta = \left(1 - \frac{\tau_y}{\tau}\right)^+, \tag{10}$$

where τ_y is the (constant) yield shear stress and the shear stress at the base of the flow is given by $\tau = \rho g h \|\nabla z\|$ with ρ and g the debris flow density and the gravitational acceleration.

An accurate computation of the debris flow thickness and duration would require a simulation of the flow dynamics and, therefore, strongly reduce the time domain. We found that the proposition of McGuire et al. [2022] to derive debris flow properties from the drainage area A leads to strong discontinuities in erosion, for instance, when a smooth ridge overlays a steep slope. Instead, we integrate the upstream mobilized material in a debris flow volumetric flow rate Q_{df}:

$$Q_{df} = -\iint_A \mathcal{D} \mathrm{d}x^2, \tag{11}$$

where A is the *upstream area*, and express the debris flow thickness h and duration t_{df} as a function of Q_{df} and the surface slope $\|\nabla z\|$. As McGuire et al. [2022], we express the debris flow duration as $t_{df} = M_{df}/Q_{df}$, where the debris flow volume M_{df} is given by a power law of Q_{df}, yielding a power-law relationship between t_{df} and Q_{df}. We also follow McGuire et al. [2022] and assume a power law between Q_{df} and the debris-flow width w_{df}, and introduce a debris flow velocity v_{df} proportional to $h^2 \|\nabla z\|$ [Rickenmann 1999]. Noting that Q_{df} is also the flux of debris flowing through the channel cross section $Q_{df} = h\, w_{df} v_{df}$, we deduce a power law between h and the product of Q_{df} and $\|\nabla z\|$. Substituting these power laws in Eqs. 9 and 11, we obtain:

$$E_{df} = k_{df} \|\nabla z\|^{\alpha} Q_{df}^{\beta} \Theta, \tag{12}$$

and

$$\Theta = \left(1 - k_\tau \|\nabla z\|^{-\alpha_\tau} Q_{df}^{-\beta_\tau}\right)^+. \tag{13}$$

Here, k_{df}, k_τ, α, β, α_τ and β_τ are user-controllable erosion coefficients. Note that we obtained these expressions by assuming a power law relationship between the debris flow width and the volumetric flow rate. This choice implies that the width of the debris flow is smaller than the cell size. A finer spatial and temporal discretization of the debris flow (*e.g.,* to accurately simulate individual debris flow events) would require a different formulation.

Eventually, the debris flow settles downslope when the threshold factor Θ (Eq. 13) approaches zero, quickly depositing the available rocks and sediments on the terrain:

$$D_{df} = \begin{cases} 0, & \text{if } \Theta = 0, \\ k_{dep,df}\, Q_{df}, & \text{otherwise.} \end{cases} \tag{14}$$

The condition of $\Theta = 0$ prevents our model from adding excessive piles of debris; a large deposition increases the slope, eventually increasing Θ. In practice, we do not set a value for $k_{dep,df}$: at the considered time scale, all available debris is deposited.

Note that our new formulation for debris flow erosion is structurally similar to the Stream Power Law with deposition (Eq. 1). In the next section, we propose a unified algorithm to solve both equations.

5 EROSION AND DEPOSITION ALGORITHM

We propose a new algorithm for solving erosion/deposition equations in the form:

$$\frac{\partial z}{\partial t} = -E(\|\nabla z\|, Q_w, Q_m) + D(Q_w, Q_m), \tag{15}$$

where Q_w is the upstream water discharge (Eq. 2), and Q_m the volumetric flow rate of upstream eroded material:

$$Q_m = -\iint_A \frac{\partial z}{\partial t} \mathrm{d}x^2, \tag{16}$$

a formulation that we find for both the fluvial sediment and the debris flow volumetric flow rates (Eqs. 3 and 11). Therefore, with the proper choice of functions E and D, Eq. 15 models both fluvial erosion (Eq. 1) and debris flow erosion (Eq. 7).

We discretize Eq. 15 in time with an explicit forward Euler scheme, which requires first computing the volumetric flow rates Q_w and Q_m and then evaluating erosion and deposition. We note that Eq. 16 requires the values of $\partial z/\partial t$ *upstream*. Therefore, there is no bidirectional dependency between distant cells of the terrain, and it is safe to use $\partial z/\partial t$ from the previous time step. This choice simplifies the solution of the integral equation induced by the coupling of Eqs. 15 and 16.

Propagation of volumetric flow rate. Let us take the example of the volumetric flow rate of water (water discharge Q_w); the same algorithm applies for Q_m, only changing the precipitation p by the rate of erosion $-\partial z/\partial t$ at the previous time step. We denote by $Q(i)$ the discrete counterpart of a variable Q, evaluated at the location of a cell i. The discharge is defined as an upstream integral (Eq. 2), which is commonly expressed recursively as a downstream accumulation [Braun and Willett 2013]. For a cell i, we define $R(i)$ as the set of *receiver* cells r strictly lower than i among the four neighbors of i, and conversely $G(i)$ as the set of *giver* cells g such that $i \in R(g)$. The discharge Q is then computed recursively:

$$Q_w(i) = p\, \Delta x^2 + \sum_{g \in G(i)} w^{g \to i} Q_w(g), \tag{17}$$

where $w^{g \to i}$ is the proportion of discharge moving from cell g to cell i, and Δx the cell size.

We reviewed different propositions from previous work for the computation of w. We observed that *multi-direction flow* [Holmgren 1994], where w is chosen proportionally to the elevation offsets tend to blur the erosion, while the *single-direction* choice [Cordonnier et al. 2016] which sets w to 0 for all receivers except for the one along

the steepest slope, leaves strongly axis-aligned erosion patterns. We propose an alternative *random single-direction* strategy, where we randomly choose one receiver r per cell i and assign it a weight of one against zero for the over receivers. The probability that r is selected is proportional to the elevation difference: $z(i) - z(r)$, which ensures that a trajectory following successive receivers will, on average, follow the steepest slope.

We follow Schott et al. [2023] and iteratively update $Q_w(i)$ in parallel on the GPU. While this would, in theory, require n iterations to fully accumulate the discharge where n is the maximum river length, we observe that using a single iteration per time step does not significantly change the resulting eroded terrain.

Erosion and deposition. Equipped with the different transport flow rates, we now focus on slope computation, which is the last missing component to compute erosion and deposition. Eq. 15 is hyperbolic and therefore requires an upwind spatial discretization scheme. Previous discretization of the stream power law [Braun and Willett 2013; Cordonnier et al. 2016; Schott et al. 2023] proposes a computation of the slope based on the elevation difference between a cell and its lowest neighbor, which tends to produce strongly axis-aligned slopes. Instead, we suggest a rotation-invariant discretization:

$$\nabla z = \frac{1}{\Delta x} \begin{pmatrix} (z(x,y) - z(x+\Delta x, y), z(x,y) - z(x-\Delta x, y))^+ \\ (z(x,y) - z(x, y+\Delta x), z(x,y) - z(x, y-\Delta x))^+ \end{pmatrix} \quad (18)$$

where Δx is the cell size.

When the deposition adds a significant amount of debris at a time step (especially debris deposition), the excess sediments can locally interrupt the flow path. This situation is a numerical artifact and could not appear in nature: if the flow path is interrupted, no sediment can be deposited downward. To prevent this issue,

we limit the total elevation of a cell after deposition to the average elevation of the upstream cells weighted proportionally to their sediment volume flow rate.

We modeled debris flow deposition with a large coefficent (Eq. 14). An interpretation of this formulation is that, during a time step Δt, the maximum height of deposited material d_{max} is limited by the available debris volume and by the constraint that the threshold factor Θ should remain null. These two constraints give:

$$d_{max} = \min\left(\frac{\Delta t}{\Delta x^2} Q_{df}, d_\Theta\right), \quad (19)$$

where d_Θ is obtained implicitly by solving $\Theta = 0$:

$$\|\nabla(z + d_\Theta)\|^2 = \left(k_\tau Q_{df}^{-\beta_\tau}\right)^{-2\alpha_\tau}. \quad (20)$$

We use the discretization of Eq. 18 to compute the surface gradient with the conservative approximation that d_Θ is null downstream, which yields a second-order equation for d_Θ (if the equation does not have any solution, we discretize the slope as toward the lowest neighbor only).

6 DEPRESSION ROUTING

Our algorithm for flow routing presented in Section 5 assumes that a path always exists from any cell to the bound of the domain. However, this assumption does not hold in the presence of *depressions* in the terrain, coming either from a noisy initial terrain, an irregular uplift, or converging deposition streams. These depressions, or local minima in the topography, are a common problem, for example, for the computation of discharge in a noisy terrain in hydrology [Wang et al. 2019]. *Depression routing* (Figure 4), therefore, seeks to add a connection between each local minimum and another cell, such as there exists a (cycle-free) path of minimum elevation between any position on the terrain and a bound. While several solutions exist on the CPU [Cordonnier et al. 2019; Scott and Dodgson 2021], to the best of our knowledge, the available parallel solutions [Barnes 2016] only focus on CPU parallel or distributed computing and are not suitable for the GPU.

Similar to the algorithm for the discharge (Section 5), we propose an iterative approximate GPU algorithm that *improves* the depression routing at each iteration, which therefore gives us the option for either a slow algorithm, where we iterate until convergence at each time-step of the simulation, but ensures an accurate flow routing, or a fast approximate algorithm, where we update the depression routing by one iteration for each simulation time step. The second option might yield inaccurate results, especially regarding sediment and discharge loss after sudden changes in the topography, such as the emergence of local minima, but progressively improves, provided that the terrain does not change significantly between iterations. Furthermore, this strategy does not slow the simulation as each iteration is fully parallelized.

Depression routing is challenging because when depressions connect, it is impossible to know *a priori* in which order one flows into another. Cordonnier et al. [2019] observed that the terrain could be segmented into independent *basins*, each one corresponding to the upstream area of a local minimum, and design a *basin graph* by connecting each pair of basins with an edge-weighted by the altitude of corresponding *pass* (lowest cell neighboring both basins). They compute a path out of the depressions as a minimum spanning tree embedded on this graph and rooted at the exterior bounds of the terrain. We use a similar structure on our GPU algorithm, which consists of two steps: 1) the identification of the basins and 2) the estimation of the per-depression outflow.

6.1 Basin Identification

We segment the terrain as a set of basins by storing the position of the corresponding local minimum in each cell, which is the one that would be found by flowing along the steepest slope. We use an iterative algorithm [Kotyra 2023] to propagate the cell position from bottom to top. For a cell i, we store the *receiver* $r(i)$ that was randomly selected among the lower neighbors in Section 5 to distribute the discharge, and as $lm(i)$ the local minimum corresponding to i. We compute lm recursively, starting with $lm_0(i) = 0$ if i is a bound of the terrain and $lm_0(i) = i$ otherwise, and iteratively refining it with $lm_{k+1}(i) = lm_k(r(i))$. This algorithm converges in n iterations when n is the maximum distance between a cell and the corresponding local minimum. In practice, we do not require the algorithm to converge as we only need to compute lm up to potential outflows of the depressions. In practice, we do a single iteration of refinement for lm per simulation time-step because the gradual variation of the terrain does not cause a significant change in the basin definition over time.

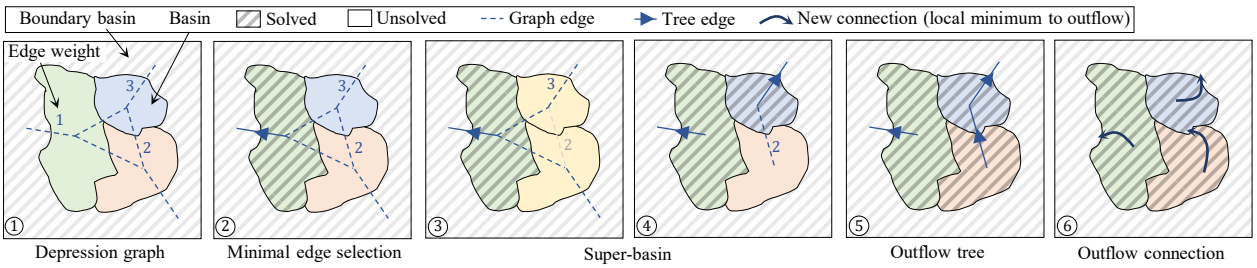
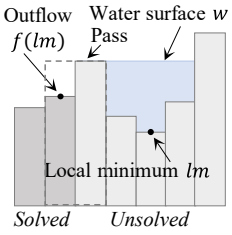


Fig. 4. Illustration of our depression routing algorithm. We start with a depression graph (1) that connects basins with edges weighted by the associated pass altitude (here 1, 2, and 3 meters, the other edges assumed to be higher). We initialized the boundary basins as solved and the internal basins as unsolved. Then, (2), we select the minimal edges (in parallel), remove them from the graph, and add them to the outflow tree. In (3), the lowest edge connects two unsolved basins. In this case, we merge both as a super-basin, which temporarily disables edge 2, and choose edge 3 as the next branch of the outflow tree (4). We mark the associated basin as solved, split the super-basin, and iterate. The final structure is a tree (5) (considering all boundary basins to be a single one), which we follow to transfer the discharge from the local minima to the selected outflows (6).

This assumption that the terrain slowly changes can be challenged in specific conditions, for instance, in the presence of a noisy uplift, which tends to generate many small depressions. We significantly reduce their impact by running the basin identification on a Gaussian-smoothed version of the terrain. Even after smoothing, small depressions occasionally appear from fluctuations in the simulation. They have little impact on erosion but induce short-lived basins that are hard to maintain through a single-step basin identification. If a new local minimum appears at cell i, we heuristically determine if it corresponds to a new depression if the distance between i and $lm(i)$ exceeds a user-given threshold. In which case, we set $lm(i) = i$; otherwise, we do not change $lm(i)$. This heuristic, together with the smoothing, does not have a visible impact on erosion but significantly stabilizes basin identification.

6.2 Computation of Depression Outflow



When basins are identified, our goal is now to find an *outflow* for each local minimum l, as the lowest cell $f(l)$ adjacent to the basin of l (or, equivalently, $lm(f(l)) \neq l$). However, this condition is insufficient, as the path out of several adjacent depressions could form a cycle.

Following Cordonnier et al. [2019], we formalize the depression routing problem with a depression graph, where each basin is a vertex, and each pair of adjacent basins form an edge (Figure 4, (1)). We neglect the river depth (assuming that the river surface is the same as the terrain surface), implying that each depression has a single outflow. Therefore, the set of paths out of depressions forms an algorithmic tree that covers the depression graph. We call this tree the *outflow tree* (Figure 4, (5)). In particular, the outflow tree minimizes the potential energy needed for the water to overflow from each basin. If we weigh each edge by such energy, the outflow tree is then the minimum spanning tree of the depression graph.

For any pair of basins, the potential energy needed for the water to overflow from one basin into the other is given by the lowest energy path that connects the local minimum of both basins, where the energy is proportional to the maximum altitude along the path. In practice, we do not need the multiplicative factor and directly assign the altitude as the path energy and, therefore, as the edge weight. This path necessarily traverses a pair of terrain cells such that each of the two cells belongs to a different basin. We call this pair of cells a *pass*. The pass between basins $B1$ and $B2$ is easily computed in parallel. The first cell of the pass is the lowest cell of $B1$ with a lower neighbor in $B2$, and the second cell is its direct neighbor in $B2$. Note that changing the order of $B1$ and $B2$ might give a different pass, in which case we choose the lowest of both. The elevation of the highest cell of the pass gives the corresponding edge weight.

From Boruvka's algorithm [Vineet et al. 2009], we know that any edge with minimum weight out of a vertex is a candidate for the minimum spanning tree, which motivates our iterative algorithm. We initialize all basins as *unsolved* except basins connected to the bound. Indeed, any local minima on the terrain boundary is a valid global outflow. Note that this choice can be altered by the user as long as they define at least one global outflow.

Then, at each iteration, in parallel, we compute the altitude of the pass between each pair of (solved-unsolved) basins. If for an unsolved basin U with local minimum l, the lowest pass is adjacent to a solved one S, we set as $f(l)$ the cell of the pass that is in S and mark U as *solved*. Then, we iterate until all lakes are solved (Figure 4, (2)). Note that a special case occurs if no minimum pass point to solved depressions *e.g.*, if two neighboring unsolved depressions share the same pass). In this case, we merge all connected basins as a *super-basin* (Figure 4, (3)) and find the outflow among the passes out of the super-basin (Figure 4, (4)). In the worst case, the complexity of this algorithm is proportional to the number of depressions. In contrast, in the best situation, all depressions can be routed in a single parallel pass. In practice, similarly to the identification of basins, we run one iteration at each simulation step. When the MST is fully computed, we keep the outflows for the discharge update (Figure 4, (6), Section 6.3) but reset all depressions to *unsolved* so that the connections are progressively updated in the subsequent iterations. We show in Figure 17 that after an initial number of iterations, this strategy does not result in a significant loss of discharge trapped in local minima.

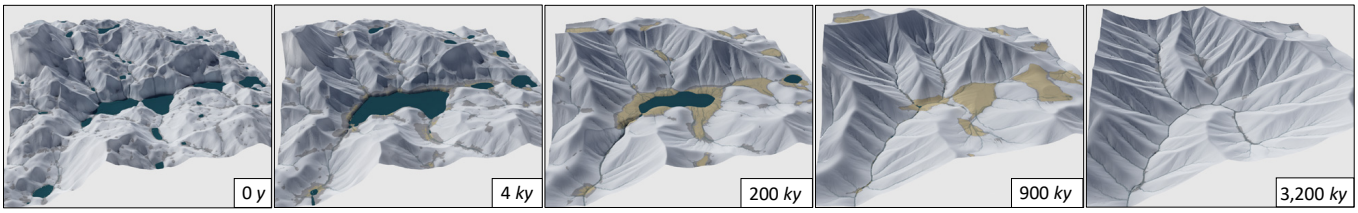


Fig. 5. Time lapse of simulation, from an initial noise-based terrain to a debris-flow eroded landscape. Note how the lakes fill, sediment, and how the sediments (in yellow) are progressively washed away when the landscape approaches steady-state.

Several parts require specific attention because of our iterative process. First, we allowed in Section 6.1 local minima l to point to another one $lm(l) \neq l$. In that case, we still need to define a path out of l and, therefore, set the outflow of l as $f(l) = f(lm(l))$. Furthermore, it is possible that, during the simulation, slopes change such that an outflow becomes part of the corresponding basin, eventually trapping the discharge and sediment in an infinite feedback loop. In that case, we update the outflow by parsing the cells upstream of $f(l)$ until we find one in the neighboring basins. We observed the new outflow is always in the direct neighborhood of $f(l)$.

6.3 Discharge and Lakes Update

We update the discharge computation to take advantage of the depression routing by transferring all the discharge present in local minima to the corresponding outflow. This also defines a *water surface* w given by the altitude of the outflow:

$$w(i) = \max\left(z(i), z(f(lm(i)))\right). \tag{21}$$

We use the water surface for rendering but also to deposit sediment and debris below water level, proportionally to the amount of available sediment:

$$s(i) = s(i) + (w(i) - z(i)) \min(1, S/W), \tag{22}$$

where W is the total water volume above $lm(i)$ and $S = \Delta t\, Q_s$ the total sediment volume. We handle debris similarly and transfer any excess sediment to the outflow. Note that all these operations are easily performed in parallel.

7 RESULTS AND VALIDATION

We implemented our method with PyTorch and PyTorch Scatter to optimize reduction operations in the depression routing algorithm. All results were generated on a desktop computer equipped with an Intel Xeon Gold CPU clocked at 2.10GHz, 128GB RAM, and an Nvidia RTX A6000 with 48GB of memory. We produced the 1024×1024 terrain for Figure 1, which required 1600 iterations in 32s (an average of 0.02s per iteration). We use Terragen (Planetside Software [2024]) for rendering, with a uniform color for each bedrock (light grey), sediment (yellow), debris (brownish dark grey), and water (dark blue) layer and without additional noise on the elevation or textures. While this requires ad hoc choices of realism of the depicted natural scenes, it allows us to control the output of our system precisely.

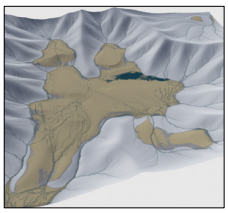
Here, we show some landscapes produced by our approach, validating that we observe the desired landforms. Then, we advocate for debris flow erosion by comparing it with thermal and fluvial erosion, and we validate our flow-routing approach. Unless stated otherwise, our results are terrains at 1024×1024 with $\Delta x = 8$ meters resolution eroded with parameters shown in Table 1. Our simulation is stable and without noticeable artifact with values of Δt ranging from 10 to 300 years. The visual variations in the results come from different initial topography and simulation lengths. The simulation code is available at: https://gitlab.inria.fr/landscapes/debrisflow.

Table 1. Parameters

Name	Value	Unit
Uplift rate: $\mathcal{U}$	0.0012	$\mathrm{m y}^{-1}$
Fluvial erosion: k_f	0.0007	$\mathrm{m}^{1-2m}\mathrm{y}^{-1}$
Fluvial deposition: k_d	0.01	
Fluvial exp.: m, n	0.4, 1	
Precipitation: p	1	$\mathrm{m y}^{-1}$
Hillslope: k_h	0.04	$\mathrm{m}^2\mathrm{y}^{-1}$
Debris-flow erosion: k_{df}	0.001	$\mathrm{m}^{1-\alpha-2\beta}\mathrm{y}^{\alpha-1}$
Debris-flow exp.: α, β	0.08, 1	
Debris-flow threshold: k_τ	0.3	$\mathrm{m}^{3\alpha_\tau}\mathrm{y}^{-\alpha}$
Debris-flow threshold exp.: α_τ, β_τ	0.25,1	
Thermal erosion: k_{th}	0.005	$\mathrm{m y}^{-1}$
Thermal critical angle: θ_c	37	degree

7.1 Debris-Flow Eroded Landscapes

We first present some eroded landscapes with our method, highlighting the main emerging landforms.



Large-scale landscape. Figure 5 shows the progressive erosion of a terrain, which starts from a purely procedural noise-based shape and is progressively uplifted and eroded. We observe that fluvial erosion is mostly active at the bottom of valleys during the first few thousand years, while debris flow erosion actively flattens upstream cliffs during the simulation period. Note how lakes progressively sediment and how lakes and sediments progressively disappear in uplifted regions, which is consistent with geological observations [Yuan et al. 2019]. Indeed, when the topography reaches a steady state, the balance between sedimentation and erosion should be negative everywhere to compensate for the uplift. The inset figure shows that if the uplift is disabled afterward, the valley refills with sediments.

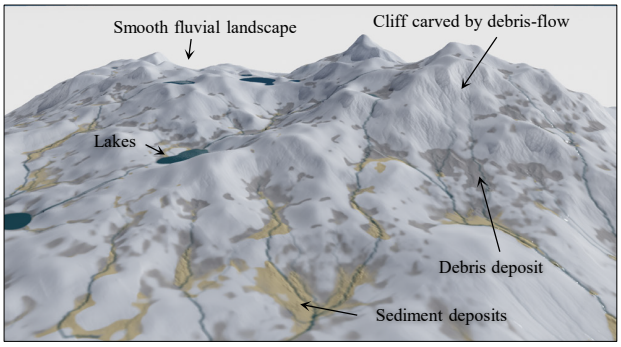


Fig. 6. An 8 × 8 km terrain eroded with our method to showcase the difference between a steeper debris flow eroded area on the right and a smooth region on the left eroded mostly by water.

Debris flow erosion patterns. Figure 1 shows a close-up view of a valley carved by debris flow, with specific debris flow erosion scars (1) and talus deposit (2) next to a smoother fluvial-eroded slope (3). The steep slopes eroded by debris flow tend to form sharp ridges (4), while the remaining lakes (5) are found by our depression routing algorithm, and they are then progressively filled with sediments. Figure 6 demonstrates an erosion of a landscape with gentler slopes to highlight the difference between the debris-flow eroded region on the right and the overall fluvial landscape on the left. We observe that sediments deposit in lower, gentler regions while debris stabilizes at the foot of the steepest slopes.

Effect of debris-flow on slopes. Debris-flow affects steep slopes, which we demonstrate in Figure 7. We model terrain with a progressively increasing slope from gentle (left) to steep (right) and show the resulting erosion patterns. We notice, in particular, that the steepest slopes tend to exhibit more diverse erosion directions.

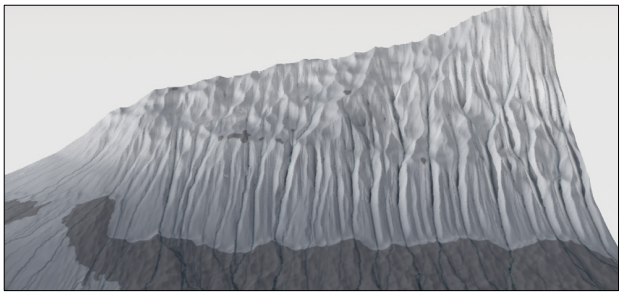


Fig. 7. The impact of debris erosion shown on different slopes by progressively varying the slope of terrain from left (gentle) to right (steep).

7.2 Ablation and Parameters

Hillslope erosion. First, we observe the importance of hillslope by changing the coefficient k_h. Figure 8 shows that hillslope controls the roughness of the erosion scars and the valley spacing.

Fluvial deposition. Deposition plays an important role in the appearance of the terrain. We vary the fluvial deposition parameter k_d in Figure 9 to show the impact of fluvial deposition on a large-scale terrain ($\Delta x = 16$ meters) where only a further half of the landscape is subject to uplift. We increase the sea level at rendering to highlight the deposition patterns. A low deposition constant has only a marginal effect. A large value, however, leads to the accumulation of significant deposits, shaping deposition fans at the outlet of the main mountain rivers and deploying deltas into the sea. Deposition also impacts the upper part of the mountain, where it opposes erosion and leads to steeper slopes.

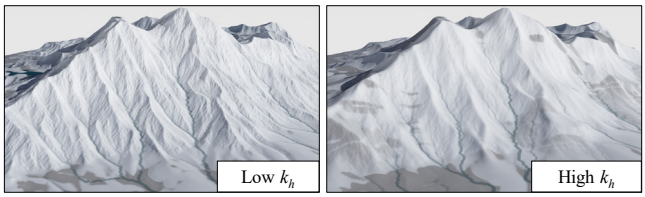


Fig. 8. We change the value of the coefficient k_h to highlight the effect of hillslope erosion. Higher values smooth the erosion patterns and increase valley spacing.

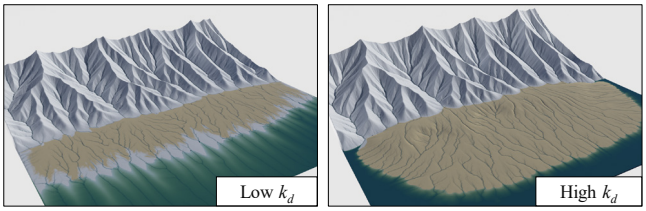


Fig. 9. Effect of fluvial deposition. Large values (right) yield steeper slopes in the uplifting region and fan-shaped deposits at the foot.

Debris deposition. While fluvial erosion yields deposits with gentle slopes, debris flow only occurs in steeper regions. Therefore, debris forms deposition patterns with strong angles and dramatically impacts the landscape. We can observe this behavior in Figure 10, where we compared a replication of Figure 1 with the same setup without debris deposition. The higher regions are similar, but the bottom of the valley fills with debris, resulting in V-shape patterns of constant slopes.

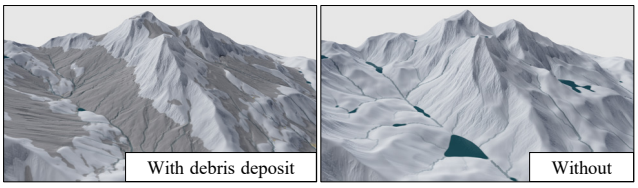


Fig. 10. We highlight the importance of debris deposition by comparing a replication of Figure 1, with (left) and without (right) debris deposition.

We further show in Figure 11 how the threshold parameter k_τ controls the slope of the deposits, a low value giving gentle slopes, while a high value tends to produce steeper deposits.

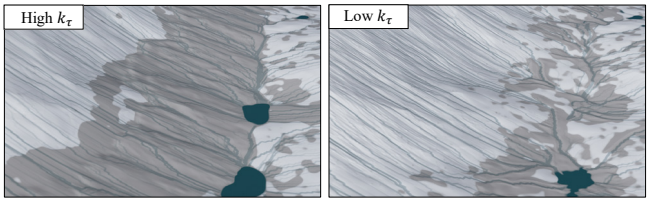


Fig. 11. The parameter k_τ controls the slope of the debris deposit, from steep with high k_τ to gentle with low k_τ.

Vegetation. We suggest using "vegetation" as a proxy control parameter, which mimics the smoothing effect of vegetation on the terrain. The vegetation parameter causes the displacement of some rocks and sediments and, in effect, increases hillslope erosion. However, it also shields the terrain from other erosion processes. We demonstrate this effect in Figure 12, where we introduce an altitude-controlled vegetation layer (green overlay) and show notably different erosion patterns below and above the vegetation limit.

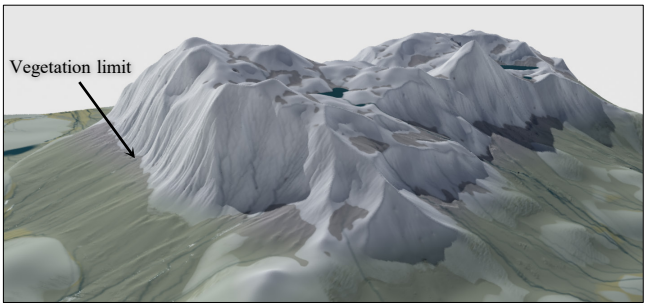


Fig. 12. We use an altitude-dependent vegetation layer (green overlay) to show how to control erosion roughness.

7.3 Comparisons

Comparison with real landscapes. We qualitatively compare our results with a real landscape in the French Alps captured on Google Earth (Figure 13, left). Our simulation (Figure 13, right) is able to capture similar landforms: deep erosion scars left by debris flow, debris deposits, and fluvial erosion patterns below.

In Figure 14, we compare the slope-drainage graph of a terrain generated with our model (orange dots) with one from real data from the San Gabriel Mountains, California, USA [McGuire et al. 2022] (blue line). Each orange dot corresponds to the mean slope of a bin of terrain cells with similar drainage (computed to total 100 logarithmically spaced bins). The blue line fits real data [McGuire et al. 2022], showing a close match between our slope profiles and real data and the slope-reducing impact of debris flow at low drainage. To highlight this point, we add a curve of the theoretical profile (green) of a valley eroded purely with fluvial erosion, which shows increasingly steeper slopes when the drainage area reduces. To compute the fluvial profile, we observe that changing the equation of the

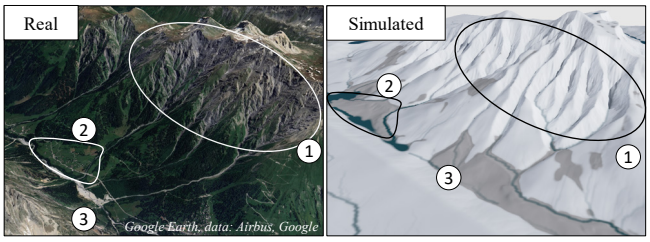


Fig. 13. Comparison between a real landscape in the French Alps (left, capture from Google Earth, data from Google and Airbus) and our method (right). We observe similar (1) deep scars left by debris flow, (2) debris deposits, and (3) fluvial erosion patterns.

blue curve from $S = k_1/(1+k_2A)$ (with S the slope) to $S = k_1/(k_2A)$ yields a steady-state solution of fluvial erosion.

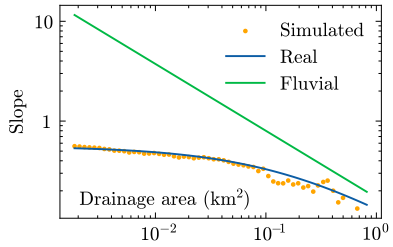


Fig. 14. Slopes calculated by our simulation (orange dots) tend to constant values with lower discharge areas, a feature attributed to debris flow. We compare simulated channel profiles with real terrain profiles from San Gabriel Mountain, California, USA [McGuire et al. 2022] (blue) and a theoretical fluvial profile (green).

Comparison with thermal erosion. We validate the results of our method by comparing them with thermal erosion. Figure 15 shows side by side comparison of a uniform slope with a small amount of added Perlin noise eroded with thermal (left) and debris flow erosion (right). Our method produces deep erosion scars, while thermal erosion keeps the slope uniform.

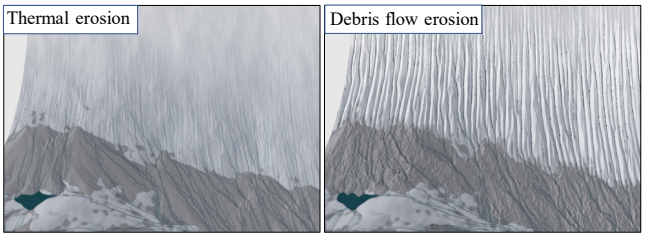


Fig. 15. Comparison of thermal erosion (left) with our method (right).

Comparison with fluvial erosion. Figure 16 demonstrates the necessity of debris flow erosion. While fluvial erosion alone is mostly smoothed by hillslope processes except for localized streams, debris flow erodes the steep slopes and provides more detailed structures.

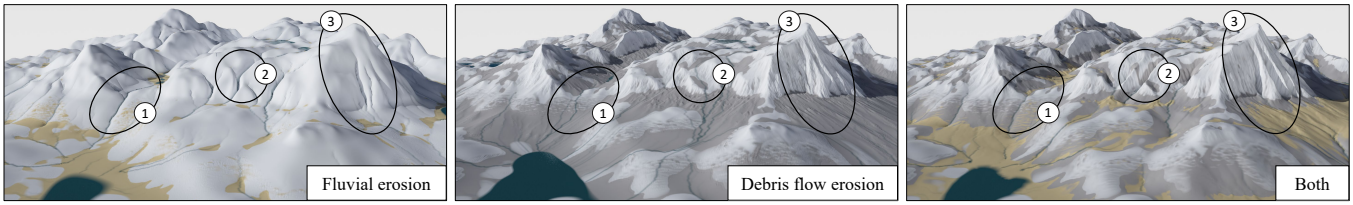


Fig. 16. Comparison of fluvial erosion only (left), debris flow erosion only (center), and combining both (right). Fluvial erosion only carves high drainage areas (1), while debris flow affects the steep slopes (3). The combination of both carves distinct features in high-drainage, high-slope areas (2).

7.4 Validation of Depression Routing

We first show in Figure 18 that depression routing is critical to correctly erode a terrain with depressions. Without depression routing, the discharge and sediments are trapped in local minima. In contrast, our algorithm, which approximates the exact solution, provides a new path for water and strongly erodes the outflow of the lakes. Figure 17 shows the impact of the number of iterations on the accuracy of our method by plotting the proportion of input discharge that exits the terrain. Our algorithm quickly converges to 100%, while no depression routing would trap 80% of the discharge. Finally, we compared our timings with an exact CPU implementation [Cordonnier et al. 2019]. On a 1024×1024 terrain, the CPU algorithm takes 500 ms, while a single step of our GPU implementation requires 1.8 ms. The 280× speedup of our algorithm is significant for the 1-step approximate version, which, interleaved with the simulation, progressively improves its accuracy. However, if an exact solution is needed, our algorithm requires 200 – 250 iterations, which results in performances on par with the CPU version.

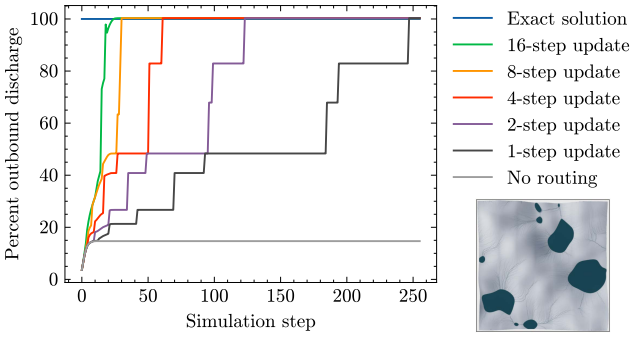


Fig. 17. We validate our depression routing algorithm by evaluating the amount of discharge that leaves the bound of a 1024×1024 terrain (bottom right), and that is therefore not trapped in a local minimum. We observe our approximate algorithm converges to the exact solution and routes 100% of discharge while this number drops to 20% if we do not handle depressions.

7.5 Limitations

Although the depression routing algorithm is fast, it comes with limitations of accuracy. Sudden changes in the terrain cause the lakes to be invalidated, requiring more iterations to converge. This behavior also occurs with the discharge propagation, preventing extreme parameters or high-time steps from being used. Furthermore,

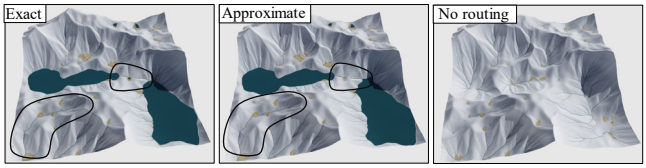


Fig. 18. We compare our approximate algorithm for depression routing with the exact version and no depression routing. We observe that connections established by our depression routing algorithm cause a high discharge on highlighted points and allow these points to be carved with erosion

we introduce a random single flow receiver to propagate discharge and sediment flux in Section 5 and find it superior. However, further analysis is required to understand better when each possible strategy should be preferred.

Finally, while physically based approaches ensure a global consistency that is hard for an artist to enforce by hand, more research is required to find intuitive handles that would allow for more direct and expressive user control.

8 CONCLUSION

We have introduced a new method for simulating erosion by debris flows. Our method is compatible with large-scale erosion models, such as fluvial erosion, and yields higher-resolution landscapes at the scale of the valley. We derived a new formulation for debris flow erosion from geomorphological observations, developed a unified algorithm to model erosion and deposition in both debris flow and fluvial processes, and proposed an iterative refining algorithm for depression routing. Our results adequately capture erosion scars left in cliffs by debris flow and foothill debris cones competing with fluvial erosion. We show that, compared to fluvial erosion, debris flow erosion enhances high-frequency patterns in eroded terrains. Finally, our implementation is fully parallelized on the GPU, which allows for fast terrain generation and erosion.

However, this efficiency comes at the cost of a loss in accuracy, with approximations that progressively correct over time but prevent the usage of large time steps or too strong parameters. In future work, we propose to explore new *exact* GPU solutions and couple them with more stable – implicit – time-stepping schemes, which would allow us further to extend the accessible range of scales for terrain simulation.

ACKNOWLEDGMENTS

This project was sponsored by the Agence Nationale de la Recherche project Invterra ANR-22-CE33-0012-01 and research and software donations from Adobe Inc. This project was also sponsored by USDA NIFA, Award #2023-68012-38992 grant "Promoting Economic Resilience and Sustainability of the Eastern U.S. Forests" to Benes. This work is based upon efforts supported by the EFFICACI grant, #NR233A750004G044, under USDA NRCS to Benes. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government or NRCS. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes, notwithstanding any copyright annotation therein.

REFERENCES

Richard Barnes. 2016. Parallel Priority-Flood depression filling for trillion cell digital elevation models on desktops or clusters. *Computers & Geosciences* 96 (2016), 56–68.

Bedrich Benes. 2007. Real-Time Erosion Using Shallow Water Simulation.. In *VRIPHYS*. 43–50.

Bedrich Benes and Rafael Forsbach. 2001. Layered Data Representation for Visual Simulation of Terrain Erosion. In *Proceedings of SCCG*, Vol. 25(4). IEEE Computer Society, 80–86.

Jean Braun and Malcolm Sambridge. 1997. Modelling landscape evolution on geological time scales: a new method based on irregular spatial discretization. *Basin Research* 9, 1 (1997), 27–52.

Jean Braun and Sean D. Willett. 2013. A very efficient O(n), implicit and parallel method to solve the stream power equation governing fluvial incision and landscape evolution. *Geomorphology* 180-181 (2013), 170–179.

Guillaume Cordonnier, Benoît Bovy, and Jean Braun. 2019. A versatile, linear complexity algorithm for flow routing in topographies with depressions. *Earth Surface Dynamics* 7, 2 (2019), 549–562.

Guillaume Cordonnier, Jean Braun, Marie-Paule Cani, Bedrich Benes, Eric Galin, Adrien Peytavie, and Eric Guérin. 2016. Large scale terrain generation from tectonic uplift and fluvial erosion. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 165–175.

Guillaume Cordonnier, Marie-Paule Cani, Bedrich Benes, Jean Braun, and Eric Galin. 2017. Sculpting mountains: Interactive terrain modeling based on subsurface geology. *IEEE transactions on visualization and computer graphics* 24, 5 (2017), 1756–1769.

Guillaume Cordonnier, Guillaume Jouvet, Adrien Peytavie, Jean Braun, Marie-Paule Cani, Bedrich Benes, Eric Galin, Eric Guérin, and James Gain. 2023. Forming Terrains by Glacial Erosion. *ACM Transactions on Graphics* 42, 4 (2023).

Eric Galin, Eric Guérin, Adrien Peytavie, Guillaume Cordonnier, Marie-Paule Cani, Bedrich Benes, and James Gain. 2019. A Review of Digital Terrain Modeling. *Computer Graphics Forum* 38, 2 (2019), 553–577. https://doi.org/10.1111/cgf.13657

Peter Holmgren. 1994. Multiple flow direction algorithms for runoff modelling in grid based elevation models: An empirical evaluation. *Hydrological Processes* 8, 4 (1994), 327–334.

K Hutter, B Svendsen, and Rickenmann. 1994. Debris flow modeling: A review. *Continuum mechanics and thermodynamics* 8 (1994), 1–35.

Richard M Iverson. 1997. The physics of debris flows. *Reviews of geophysics* 35, 3 (1997), 245–296.

Bartłomiej Kotyra. 2023. High-performance watershed delineation algorithm for GPU using CUDA and OpenMP. *Environmental Modelling & Software* 160 (2023), 105613.

Peter Krištof, Bedrich Benes, Jaroslav Křivánek, and Ondřej Šťava. 2009. Hydraulic Erosion Using Smoothed Particle Hydrodynamics. *Computer Graphics Forum (Proceedings of Eurographics 2009)* 28, 2 (mar 2009), 219–228.

Dimitri Lague and Philippe Davy. 2003. Constraints on the long-term colluvial erosion law by analyzing slope-area relationships at various tectonic uplift rates in the Siwaliks Hills (Nepal). *Journal of Geophysical Research: Solid Earth* 108, B2 (2003).

Luke A McGuire, Scott W McCoy, Odin Marc, William Struble, and Katherine R Barnhart. 2022. Steady-state forms of channel profiles shaped by debris-flow and fluvial processes. *Earth Surface Dynamics Discussions* 2022 (2022), 1–33.

F Kenton Musgrave, Craig E Kolb, and Robert S Mace. 1989. The synthesis and rendering of eroded fractal terrains. *ACM Siggraph Computer Graphics* 23, 3 (1989), 41–50.

Alexander B. Neely and Roman A. DiBiase. 2023. Sediment controls on the transition from debris flow to fluvial channels in steep mountain ranges. *Earth Surface Processes and Landforms* 48, 7 (2023), 1342–1361.

J Taylor Perron, James W Kirchner, and William E Dietrich. 2009. Formation of evenly spaced ridges and valleys. *Nature* 460, 7254 (2009), 502–505.

Dieter Rickenmann. 1999. Empirical relationships for debris flows. *Natural hazards* 19 (1999), 47–77.

Pascal Roudier, Bernard Peroche, and Michel Perrin. 1993. Landscapes Synthesis Achieved through Erosion and Deposition Process Simulation. *Computer Graphics Forum* 12, 3 (1993), 375–383.

Hugo Schott, Axel Paris, Lucie Fournier, Eric Guérin, and Eric Galin. 2023. Large-scale terrain authoring through interactive erosion simulation. *ACM Transactions on Graphics* 42 (2023), 15 pages.

Joshua J Scott and Neil A Dodgson. 2021. Example-based terrain synthesis with pit removal. *Computers & Graphics* 99 (2021), 43–53.

Planetside Software. 2024. Terragen 4. http://planetside.co.uk/.

Jonathan D. Stock and William E. Dietrich. 2006. Erosion of steepland valleys by debris flows. *GSA Bulletin* 118, 9-10 (09 2006), 1125–1148.

Tamotsu Takahashi. 1978. Mechanical characteristics of debris flow. *Journal of the Hydraulics Division* 104, 8 (1978), 1153–1169.

Tamotsu Takahashi. 1981. Debris flow. *Annual review of fluid mechanics* 13, 1 (1981), 57–77.

Juraj Vanek, Bedrich Benes, Adam Herout, and Ondrej Stava. 2011. Large-Scale Physics-Based Terrain Editing Using Adaptive Tiles on the GPU. *IEEE Computer Graphics and Applications* 31, 6 (2011), 35–44.

Vibhav Vineet, Pawan Harish, Suryakant Patidar, and PJ Narayanan. 2009. Fast minimum spanning tree for large graphs on the GPU. In *Proceedings of the Conference on High Performance Graphics 2009*. 167–171.

Yi-Jie Wang, Cheng-Zhi Qin, and A-Xing Zhu. 2019. Review on algorithms of dealing with depressions in grid DEM. *Annals of GIS* 25, 2 (2019), 83–97.

Kelin X Whipple and Gregory E Tucker. 1999. 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, B8 (1999), 17661–17674.

Sean Willett, Christopher Beaumont, and Philippe Fullsack. 1993. Mechanical model for the tectonics of doubly vergent compressional orogens. *Geology* 21, 4 (04 1993), 371–374.

Chris Wojtan, Mark Carlson, Peter J. Mucha, and Greg Turk. 2007. Animating corrosion and erosion. In *Proceedings of the Third Eurographics Conference on Natural Phenomena* (Prague, Czech Republic) *(NPH'07)*. Eurographics Association, Goslar, DEU, 15–22.

Xilin Xia, Kristine Thorkildsen Jarsve, Tom Dijkstra, Qiuhua Liang, Xingmin Meng, and Guan Chen. 2023. An integrated hydrodynamic model for runoff-generated debris flows with novel formulation of bed erosion and deposition. *Engineering Geology* 326 (2023), 107310.

Xiaoping P. Yuan, Jean Braun, Laure Guerit, Delphine Rouby, and Guillaume Cordonnier. 2019. A New Efficient Method to Solve the Stream Power Law Model Taking Into Account Sediment Deposition. *Journal of Geophysical Research: Earth Surface* 124, 6 (2019), 1346–1365.