REVIEWER 1

First, I would like to thank the reviewer for his useful comments. Below is my response to these comments.

General comments:

GC1: *I would like to hear how the author defends criticism that this model is not novel. eSCAPE v1 was published in the Journal of Open Source Software, how does v2 differ? What makes it require a whole new publication?*

Response: The main differences between v1 and v2 of eSCAPE are in the way v2 handles the marine deposition and in the implementation of the depression filling algorithm. In v2, a priority-flood + epsilon variant of the algorithm proposed by Barnes et al. (2014) is implemented. It prevents the formation of flat surfaces and allows for the determination of flow directions on all regions of the simulated landscape. The depression-less surface is then used to estimate depositional regions and to force marine deposition. An analyse of the differences between v1 and v2 on GitHub shows that there have been 36 commits over 12 files with 2,241 additions and 1,212 deletions. In addition, the first version published in JOSS (https://doi.org/10.21105/joss.00964) was a one-page summary that did not explain the details of the algorithms. This new publication describes in detail the physics and numerical approaches from eSCAPE, it also provides a series of hands-on examples that illustrate the code usage in different settings.

GC2: *Furthermore, and I ask this out of naivety, how does eSCAPE differ from Bad- lands? Is the difference significant? Overall this is my only major concern, and it is one that is potentially wrong.*

Response: There are many differences between Badlands and eSCAPE. First, the number of processes that can be simulated with eSCAPE is quite limited compared to Badlands. When considering the processes that both models simulate, the numerical approaches are completely different. Badlands is an explicit serial model able to simulate single flow direction river erosion/deposition. eSCAPE relies on an implicit iterative parallel approach able to evaluate multiple flow direction river processes. The approach in eSCAPE consists in solving a series of sparse matrix systems using the parallel library PETSc. In addition, eSCAPE can be used at global scale on a spherical mesh and relies on a different strategy to simulate depression filling (Planchon and Darboux 2001 for Badlands – Barnes 2014 for eSCAPE). In terms of outputs, one might find these two models similar, but they are really distinct when looking at the underlying algorithms and implementation strategies.

Minor comments:

MC1: T*he introduction way oversells the model. Yes, it can model global erosion and deposition using a set of rules, however, the model cannot capture lateral movement of the surface due to faulting. In fact, there is no faulting, which is arguably the major process that connects mantle convection to surface processes. This is a very challenging problem, and not one the author seeks to solve. However, much text is wasted on describing a vision of a global coupled model. This should be saved for a research proposal and not used here.*

Response: Following reviewer's comment, I have modified the introduction and re- moved the paragraph related to the coupling with geodynamic/lithospheric models as this is not essential to the paper and I definitely do not want to oversell the model, as pointed out in the introduction: "The model presented in this paper is a first step toward the development of a parallel global scale landscape evolution model."

***MC2:** Explain what the advance is in this model, how it advances on v1 and Badlands. What is eSCAPE v2 for?*

Response: See response to the general comments from the reviewer above (GC1 & GC2).

MC3: *Line 15: What was the reason for cherry-picking these citations, none of which date from the '80s?*

Response: Following the reviewer's comment, I have modified the text from the '80s to '90s. The choice of citations illustrates some of the LEM models that have been created over the years: Caesar (Coulthard), Cascade (Braun), Apero (Davy), Badlands (Salles) or Landlab (Hobley). In addition, these models represent different numerical approaches based on cellular automata, stream power law, or more standard flow hydrodynamics. They have also been developed to look at different spatial domains from river to catchment scale up to regional and continental extent.

MC4: *Line 28: What is the purpose of this paragraph? As it is, it is far too short to encompass how global mantle flow is expressed at the earth's surface.*

Response: Following the reviewer's comment, I have removed this paragraph from the introduction.

MC5: *Line 70: I thought the approach of Jean Braun was O(N) efficient, always? Is the author saying otherwise?*

Response: The approach from Braun is O(N) efficient but its parallel implementation relies on the number of outlets present in the simulation and therefore can become inefficient and scale poorly when the number of processors increases.

MC6: *Equation 2: The first line does not make sense. q_1 = b_1, not q_i = b_i*

Response: I have made the correction in the manuscript

MC7: *Line 127: "calibration" is out of place here.*

Response: I have deleted "calibration"

MC8: *Line 128: "evidence" should not get an "s", likewise "behaviour". There are other minor grammatical errors which I am sure will be corrected when copy edited.*

Response: I have removed the "s" from evidence and behaviour in the text.

MC9: *Equations 7 and 8: Here it is hard coded that n=1 and m=0.5. This is stated later in the manuscript, but this is potentially a major limitation of the model, as the recent study by Kwang & Parker (2017) suggests that "the choice m/n=0.5 yields a curiously unrealistic result: the predicted landscape is invariant to horizontal stretching".*

Response: From Kwang & Parker (2017), this unrealistic behaviour is found when hillslope diffusion is neglected. In eSCAPE, hillslope diffusion could be turned on and thus should help to limit this behaviour. In addition, it is worth mentioning that the effect observed by Kwang & Parker is made when accounting only for a single flow direction (D8) when computing flow and drainage area. eSCAPE allows to simulate multiple flow direction (MDF) and the curious observations from Kwang & Parker have not been reported in such case.

MC10*: Line 159: In this equation, the non-suspended sediment gets left behind, right? But the stream power law assumes instantaneous sediment transport. Therefore the two are incompatible? I am missing something here. Perhaps some additional explanation of how the model goes from erosion to deposition would help.*

Response: At line 160, I define Ff as the fraction of fine sediment that remains in suspension. Ff represents the volumetric fraction of bedrock that breaks into sediment small enough to be considered permanently in suspension and for which no further treatment of bed–water column interactions is needed. For bedrock that breaks only into sand and gravel fractions, Ff would be zero. Therefore, simulated bed deposits and transported sediment flux only include sediment coarse enough that it does not permanently stay in suspension. I have added the explanation above in the manuscript.

MC11: *Section 2.3: The "priority-flood" algorithm is non-physical, right? I wonder if it should not be done after the hillslope processes (diffusion), as this would smooth depressions and potentially fill them. Then the subsequent filling by fluvial deposition should occur?*

Response: The reviewer is right, the "priority-flood" algorithm is a non-physical process and can be done prior to fluvial deposition. It could potentially help in cases where depressions are made of only a single point (local pit) or really small in size because induced filling from hillslope processes only occurs over much longer temporal scale than river ones. I believe over time, as the model iterates over the main loop the order proposed by the reviewer and the implemented one will produce equivalent results.

MC12: *Section 2.5: Does marine deposition use a constant diffusion coefficient? Some marine deposition models vary this diffusion coefficient with water depth, to simulate wave and tide effects. I assume that this is not the case within eSCAPE?*

Response: The reviewer is right I only use a constant diffusion coefficient for marine deposition in eSCAPE and do not account for water depth dependent (non-linear) dif- fusion. This could potentially be a new feature for the next model version.

MC13: *Table 3: I think the marine parameters are missing from the table.*

Response: The only user-defined parameter required to simulate marine processes is the diffusion parameter sedimentK defined in table 3 and at line 288 page 13.

<u>Reproducibility:</u>

RC1: *The code is available, and I have successfully installed it. I have come across minor issues in running the code, due to my install of python and petsc, but this will be fixed before publication I am sure.*

Response: I have made some changes in the code to fix some of the issues encountered by the reviewer (https://github.com/Geodels/eSCAPE/issues/9). I have also added some documentation about the petsc installation (https://github.com/Geodels/eSCAPE/wiki/Dependency)

REVIEWER 2

Again, I would like to thank the second reviewer for his useful comments. Below I provide a point by point response. I have also attached the updated manuscript based on the two reviews.

Main comments:

MC1: *Although simulations at this scale are presented in the manuscript, the resolution (16 km) and the total number of mesh points (3 millions) used for these simulations remain somewhat coarse and small, respectively. The mesh size is not much greater than what could be processed nowadays at tractable computational costs using sequential model implementations. In this regard, it would be interesting (1) to see how eSCAPE performance does roughly compare with efficient, sequential implementation(s) of the same processes run on grids or meshes of a similar size (at least for the SFD / purely erosive case), and (2) to see how eSCAPE scales at much greater mesh sizes of, e.g., 10-100 millions of nodes (at least for the purely erosive case since applying pit resolving in serial might become computationally intractable at that scale). In my opinion, and this is my main concern, the manuscript could do a better job at showing when eSCAPE would become a compelling alternative to landscape evolution models based on sequential algorithms (graph traversal or other). The two suggestions above might help improving that.*

Response: I will also be pleased to make such comparisons. Yet this is a pretty difficult task as (1) it is pretty hard to find authors that reports their computation time for their codes and (2) I have never used any code using an implicit approach like eSCAPE. From testing with the other code that I have access to (*e.g. Badlands*), eSCAPE for purely erosive simulations is performing much faster for mesh above half a million points thanks to the implicit approach that allows longer time step to be used even in serial. Yet the comparison is not that valuable as the algorithm design is very different. If one sets the model time steps to some smaller values, then Badlands will be more efficient… In other words, it is pretty hard to compare the efficiency of these codes if we set similar parameters. I found that the only potential comparison could be with *fastscape* code based on the Braun & Willett (2013) algorithm. In their paper they reported for a model similar to the one presented in section 3.2.1 (Fig. 3) that for a 100 million of nodes their code took 2.7 seconds per time step on eight cores. I did a test using the same parameters as the ones presented in section 3.2.1 and found the performance results presented in the Table below. I don't think that these results should be added to the manuscript as a proper benchmark as I haven't been using the same type of processors nor the exact same simulation parameters as the one in Braun & Willett (2013).

Proc. #	Time step (s)
2	10.8
4	6.2
8	3.5
16	2.1

MC2: *Besides that, I have a minor comment about the method chosen in eSCAPE to control the convergence vs. divergence of flow paths (SFD vs. MFD). Is there any specific reason why manually setting the number of flow receivers is preferred over unconditionally partitioning the outgoing flow between all downslope neighbors and let the user control the SFD vs MDF behavior, e.g., by tuning the parameter(s) of the weight vs. slope relationship? While both approaches to this problem are arbitrary, the second one would allow finer control and has been studied more in depth (see, e.g., Qin et al. 2007, https://doi.org/10.1080/13658810601073240). Or perhaps a combination of the two approaches would allow even greater flexibility.*

Response: The algorithm actually uses an adaptive approach for determining the flow-partition between downslope neighbours as suggested by the reviewer. The description of this capability is found in the *MFDreceivers* function from the file 'fortran/functions.f90'. In my approach, the influence of local terrain on flow partition is modelled by a weight function which is based on local maximum downslope gradient. The manual setting of the number of flow receivers is there to improve the efficiency of the approach. From testing, it appears that having a maximum number of downstream nodes set to 3 gives results similar to a full MFD approach.

<u>Specific comments:</u>

MC1: *Line 69: Due to the issue (rightly pointed by the author in the following paragraph) of load balancing vs. the relative sizes of the catchments in the simulated domain, methods based on depth-first graph traversal may not scale at all in the worst case scenario of one single simulated catchment.*

Response: I agree with the reviewer's comment!

MC2: *Line 78: Note that approaches like the one described by Braun and Willet's (2013) can actually be easily modified to incorporate MFD algorithms. Unfortunately, no paper has been published yet on this.*

Response: Thanks for the information, I haven't seen the application of Braun and Willet's algorithm to the MFD case but it will be a great addition for sure!

MC3: *Fig 1a: I guess that the color map used for cell elevation values has been chosen so that it emphasizes dry (high) vs. wet (low) cells. Still, I doubt that the value of 4 meters has a special meaning, and a non-diverging color map would be more appropriate.*

Response: The color scale represents elevation in (m) for a simple example used to illustrate flow paths on the triangular irregular network. Maybe the confusion comes from the fact that in addition to the cell color I have also written the nodes number (ID) on top of the figure. I have added this information to the figure caption.

MC4: *Line 143: "m/y" units badly formatted.*

Response: Corrected

MC5: *Line 154: I might be missing something in the source code, but "scipy.sparse" is imported only in "surfprocplex.py" and it is not used further in that module. Maybe there is an inconsistency between the source code and the manuscript about how SciPy is used here?*

Response: The reviewer is right *SciPy sparse matrix* is only called in the surfprocplex.py file and is required by petsc4py library to solve the system described in eq. (7). I have modified the text to specify that I use scipy sparse matrices (e.g. csr_matrix) to efficiently load matrices into a petsc4py matrix.
Ref: https://bitbucket.org/petsc/petsc4py/issues/94/scatter-scipy-sparse-matrix-to-petsc-local

MC6: *Line 178: It might be worth also mentioning the O(N) depression resolving algorithm (not part of the "priority-flood" family of algorithms) that has been published very recently by Cordonnier et al. (2019, https://doi.org/10.5194/esurf-7-549-2019). Disclaimer: I'm co-author of this paper.*

Response: I have added the reference to Cordonnier et al. (2019) and this family of algorithms in the manuscript.

MC7: *Section 2.4: It might be worth adding a few words on how the depression areas are delineated and how the spillover nodes are retrieved using the priority-flood + epsilon filling algorithm. This is not obvious, at least to me and potentially to other readers as well.*

Response: To obtain the spillover nodes from the priority-flood + epsilon filling algorithm I use the approach proposed by Barnes (2017) in its parallel version (section 3.1 in the paper). I have added a reference to Barnes' work in the manuscript for this section: *"The spillover nodes are obtained using the method proposed by Barnes (2017) where in addition to depressions, the priority-flood approach labels watershed indices. Spillover nodes correspond to the lowest points connecting different* *watersheds. The updated elevation field is then used to compute the flow accumulation following the approach presented in section 2.1."*

MC8: *Line 380: Typo "be compare" -> "be compared".*

Response: Corrected

MC9: *Line 415: The Cordonnier et al.'s depression resolving algorithm cited here above is optimized specifically for use in landscape evolution models. Compared to the priority flood + epsilon variant of Barnes et al. (2014), it may drastically improve the overall performance when it is executed at every time step. That said, there is no parallel version yet and it works best when coupled with graph traversal algorithms.*

Response: I have added the reference to the approach proposed by Cordonnier et al. (2019) in the corresponding section.

eSCAPE: Regional to Global Scale Landscape Evolution Model v2.0

Tristan Salles[1]

[1]School of Geosciences, University of Sydney, Sydney, NSW, 2006, Australia

Correspondence: Tristan Salles (tristan.salles@sydney.edu.au)

Abstract. ~~This paper presents eSCAPE , an open-source~~ eSCAPE is a Python-based landscape evolution ~~framework that computes~~ model that simulates over geological time (1) ~~landscape dynamic~~ the dynamic of the landscape, (2) ~~sediment transport~~ the transport of sediment from source to sink, and (3) continental and marine sedimentary basins formation under different climatic and tectonic conditions. eSCAPE is open-source, cross-platform, distributed under the GPLv3 license and available on GitHub (escape-model.github.io). Simulated processes ~~relies on~~ rely on a simplified mathematical representation of landscape processes - the stream power and creep laws - to compute Earth's surface evolution by rivers and hillslope transport. The main difference with previous models is in the underlying numerical formulation of the mathematical equations. The approach is based on a series of implicit iterative algorithms defined in matrix form to calculate both drainage area from multiple flow directions and erosion/deposition processes. eSCAPE relies on PETSc parallel library to solve these matrix systems. Along with the ~~algorithms description~~ description of the algorithms, examples are provided and illustrate the model current capabilities and limitations. eSCAPE is the first landscape evolution model able to simulate processes at global scale and is primarily designed to address problems on large unstructured grids (several millions of nodes).

1 Introduction

~~Over the past decades, many numerical models have been proposed~~ Since the '90s, many software have been designed to estimate long-term catchment dynamic, drainage evolution as well as sedimentary basins formation in response to various mechanisms such as tectonic or climatic forcing (Braun and Sambridge, 1997; Coulthard et al., 2002; Davy and Lague, 2009; Simoes et al., 2010; Salles, 2016; Grieve et al., 2016b; Hobley et al., 2017). These models ~~combine empirical data and conceptual methods into~~ rely on a set of mathematical ~~equations that can be used to reconstruct landscape evolution and associated sediment fluxes (Tucker and Hancock, 2010; Shobe et al., 2017). They are currently used in many research fields such as hydrology, soil erosion, hillslope stability and geomorphology studies to cite a few.~~ and physical expressions that simulate sediment erosion, transport and deposition and can reproduce the first order complexity of Earth's surface geomorphological evolution (Tucker and Hancock, 2010; Shobe et al., 2017).

In most of these models, climatic and tectonic conditions are imposed and often consist in rather simple forcing such as uniform spatial precipitation and vertical displacements (uplift or subsidence) far from reflecting the complexity of the natural system. In addition such approaches are unable to properly explore potential feedback mechanisms between each of the Earth components. In fact, only a handful of these models are able to account more completely for the dynamics of the lithosphere and mantle, the role of sedimentation and provide a more quantitative representation of climate relative to its interactions with topography (such as orographic rain) ~~(Beaumont et al., 1992; Salles et al., 2011; Thieulot et al., 2014; Yang et al., 2015; Salles et al., 2017; Beucher et al., 20~~
(Beaumont et al., 1992; Salles et al., 2011; Bianchi et al., 2015; Thieulot et al., 2014; Yang et al., 2015; Salles et al., 2017; Beucher et al.,
. When made possible, it is often realised through the coupling of specialised numerical models involving the expertise of geodynamicists, geophysicists, Earth surface and atmospheric scientists.

~~Many advanced numerical models of tectonic processes constrained by geological and geophysical observations have been developed and global scale geodynamic models (Zhong et al., 2000; Moresi et al., 2003; Heister et al., 2017) have shown how mantle convection drives the motion of tectonic plates and dictates the long-term evolution of the Earth. Similarly progresses in the understanding of past, present, and future climates have been made by the development of mathematical models of the general circulation of a planetary atmosphere or ocean that simulate climate at an increasing level of detail (Dutkiewicz et al., 2016; Brown et al., 2018).~~

Yet, we are still missing a tool to evaluate global scale evolution of Earth surface and its interaction with the atmosphere, the hydrosphere, the tectonic and mantle dynamics. Such a tool will certainly provide new insights and help to better characterise many aspects of the Earth system ranging from the role of atmospheric circulation on physical denudation, from the influence of erosion and deposition of sediments on mantle convection, from the location and abundance of natural resources to the evolution of life.

The model presented in this paper is a first step toward the development of a parallel global scale landscape evolution model. It ~~provides a more direct and flexible way~~ allows to couple the Earth's surface with global climatic perturbations and geodynamic forces acting within the Earth's interior. Landscapes and sedimentary basins evolution in eSCAPE are driven by a series of standard stream power incision and diffusion laws (Howard et al., 1994; Tucker and Slingerland, 1997; Chen et al., 2014) designed to address problems from regional to global scales and over geological time (10^5-10^9 years). Due to the inherent assumptions made in the set of equations used, eSCAPE is not intended to estimate the evolution of individual fluvial channels but to quantify large scale and long term evolution of Earth's surface regions (Salles et al., 2017; Armitage, 2019). ~~It is worth mentioning that eSCAPE simulates sediment supply and routing from source to sink in a self-consistent manner. In other words, the erosion occurring in upstream catchments is linked to sedimentation on basin margins through sediment routing resulting from a combination of channelling and hillslope processes. Sediment supply to continental margins is dynamically determined and results from both allogenic causes (*e.g.* the interactions with tectonic and/or climatic forcing or eustatic variations) and autogenic changes like the ones induced on catchments physiography.~~

First, this paper presents the implicit, iterative approaches that are used to solve the multiple flow direction water routing and the erosion deposition processes (section 2). Then in section 3, I provide a list of all the parameters required to run the eSCAPE model and I discuss the input and output formats. In addition, three examples based on both generic and global scale experiments are described in detail and showcase the code main capabilities. Finally in section 4, I analyse the scalability of eSCAPE and discuss some of the limitations and future implementations that are necessary to improve the performance of the code on parallel architectures.

2 Modelled processes and algorithms

eSCAPE (Salles, 2018) is a parallel landscape evolution model, built to simulate landscapes and basins dynamic at various space and time scales over unstructured grids. The model accounts for river incision using stream power law, hillslope processes and sediment transport in land and marine environments. It can be forced with spatially and temporally varying tectonics (horizontal and vertical displacements) and climatic forces (temporal and spatial precipitation changes and sea-level fluctuations). eSCAPE is primarily written in Python with some functions in Fortran and takes advantage of PETSc solvers (Balay et al., 2012) over parallel computing architectures using MPI. In this section, I describe the simulated physical processes along with the algorithms that are used.

2.1 Implicit parallel flow discharge implementation

Flow accumulation (FA) calculations are core component of landscape evolution models as they are often used as proxy to estimate flow discharge, sediment load, river width, bedrock erosion as well as sediment deposition. Until recently conventional FA algorithms were serial and limited to small spatial problems (O'Callaghan and Mark, 1984; Mark, 1988). With ever growing high resolution digital elevation dataset, new methods based on parallel approaches have been proposed over the last decade. Due to the recursive nature of FA computation, graph traversal techniques are common in determining the upstream-summation and most approaches (Wallis et al., 2009; Wallace et al., 2010; Tarboton, 2013; Bellugi et al., 2011; Braun and Willett, 2013) are based on an initial ordering process followed by efficient priority-queue implementations with some variants such as the sub-basin acyclic graph partitioning method in Salles and Hardiman (2016) or the breadth-first traversal approaches proposed by Barnes (2019). Except for the approach proposed by Barnes (2019), the previous methods scale well as long as the number of processors used is modest but quickly deteriorates as inter-processors communication cost increases.

In addition, when using the aforementioned implementation strategies, several problems might arise in (1) load balancing, when catchments size greatly changes in the simulated domain or (2) handling very high resolutions where multiple processes are needed for a single catchment. In addition, most of these methods assume a single flow direction (SFD - Fig. 1a). This assumption makes the emergent flow network highly sensitive to the underlying mesh geometry and most dendritic shape of obtained stream networks is often an artefact of the surface triangulation. To reduce this effect, authors have proposed to consider not only the steepest downhill direction but also to represent other directions appropriately weighted by slope (multiple flow direction - MFD). Using MFD algorithms prevent locking of erosion pathways along a single direction and help to route flow over flat regions into multiple branches (Tucker and Hancock, 2010). Yet, graph traversal approaches cannot be easily modified to incorporate MFD algorithms as catchments are no longer strictly isolated in low slope areas and flow pathways often diverge (Fig. 1b).

Figure 1. (a) Schematic diagram showing flow paths when considering a triangular irregular network composed of 10 vertices (node IDs are given for each case). Cells (i.e. voronoi area defining the region of influence of each vertex) are coloured by elevation. Two cases are presented considering single flow direction (top sketch – SFD) and multiple flow direction (bottom sketch – MFD/D$_\infty$). White arrows indicate flow direction and their sizes vary in proportion to slope (not at scale). Nodes numbers correspond to the subscripts in equations 2 and 4. (b) Differences in calculated drainage area for a portion of South America from eSCAPE using the two flow direction methods.

To overcome these limitations, Richardson et al. (2014) proposed to use linear solvers. The approach consists in writing the FA calculation as a sparse matrix system of linear equations (Eddins, 2007; Schwanghart and Kuhn, 2010). It can take full advantage of purpose-built, efficient linear algebra routines including those provided by parallel libraries such as PETSc (Balay et al., 2012). eSCAPE computes the flow discharge (m^3/y) from FA and the net precipitation rate P using the parallel implicit drainage area (IDA) method proposed by Richardson et al. (2014) but adapted to unstructured grids (Fig. 1).

The flow discharge at node i (q_i) is determined as follows:

$$q_i = b_i + \sum_{d=1}^{N_d} q_d \tag{1}$$

where b_i is the local volume of water $\Omega_i P_i$ where Ω_i is the voronoi area and P_i the local precipitation value available for runoff during a given time step. N_d is the number of donors with a donor defined as a node that drains into i (as an example the donor of vertex 5 in the SFD sketch in Fig. 1a is 1). To find the donors of each node, the method consists in finding their receivers first. Then, the receivers of each donor is saved into a receiver matrix, noting that the nodes, which are local minima, are their own receivers. Finally the transpose of the matrix is used to get the donor matrix. When Eq. 1 is applied to all nodes and considering the MFD case presented in Fig. 1a, the following relations are obtained:

$$\begin{aligned}
q_1 &= b_1 \\
q_2 &= b_2 + q_1 w_{1,2} \\
q_3 &= b_3 + q_2 w_{2,3} + q_4 w_{4,3} \\
q_4 &= b_4 + q_1 w_{1,4} + q_2 w_{2,4} \\
q_5 &= b_5 + q_1 w_{1,5} + q_4 w_{4,5} \\
q_6 &= b_6 + q_4 w_{4,6} + q_5 w_{5,6} + q_7 w_{7,6} \\
q_7 &= b_7 + q_{10} w_{10,7} \\
q_8 &= b_8 + q_3 w_{3,8} + q_4 w_{4,8} + q_6 w_{6,8} + q_7 w_{7,8} + q_{10} w_{10,8} \\
q_9 &= b_9 + q_3 w_{3,9} + q_8 w_{8,9} + q_{10} w_{10,9}
\end{aligned} \tag{2}$$

The choice of weights $w_{m,n}$ depends on the number of flow directions that is used. The weights range between zero and one and sum to one for each node:

$$\sum_n w_{m,n} = 1 \tag{3}$$

In eSCAPE, the number of flow direction paths is user-defined and can vary from 1 (i.e. SFD) up to 12 (i.e. MFD) depending of the grid neighbourhood complexity. The weights are calculated based on the number of downslope neighbours and are proportional to the slope (Quinn et al., 1991; Tarboton, 1997; Richardson et al., 2014).

In matrix form the system defined in Eq. 2 is equivalent to **Wq=b** or:

$$\begin{bmatrix} 1 & & & & & & & & & \\ -w_{1,2} & 1 & & & & & & & & \\ & -w_{2,3} & 1 & -w_{4,3} & & & & & & \\ -w_{1,4} & -w_{2,4} & & 1 & & & & & & \\ -w_{1,5} & & & -w_{4,5} & 1 & & & & & \\ & & & -w_{4,6} & -w_{5,6} & 1 & -w_{7,6} & & & \\ & & & & & & 1 & & & -w_{10,7} \\ & & -w_{3,8} & -w_{4,8} & & -w_{6,8} & -w_{7,8} & 1 & & -w_{10,8} \\ & & -w_{3,9} & & & & & -w_{8,9} & 1 & -w_{10,9} \\ & & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \\ q_{10} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \\ b_9 \\ b_{10} \end{bmatrix} \tag{4}$$

the vector **q** corresponds to the unknown flow discharge (volume of water flowing on a given node per year) and the elements of **W** left blank are zeros.

As explained in Richardson et al. (2014), the above system is implicit as the flow discharge for a given vertex depends on its neighbours unknown flow discharge. The matrix **W** is sparse and is composed of diagonal terms set to unity (identity matrix) and off-diagonal terms corresponding to at most the immediate neighbours of each vertex (typically lower than 6 in constrained Delaunay triangulation).

In eSCAPE, this matrix is built in parallel using compressed sparse row matrix functionality available from SciPy (Jones et al., 2001). Once the matrix has been constructed, PETSc library is used to solve matrices and vectors across the decomposed domain (Balay et al., 2012). The performance of the IDA algorithm is strongly dependent on the choice of solver and preconditioner. In eSCAPE, the solution for **q** is obtained using the Richardson solver (Richardson, 1910) with block Jacobi preconditioning (*bjacobi*). This choice was made based on the convergence results from Richardson et al. (2014) but can be changed if better solver and preconditioner combinations are found. Iterative methods allow for an initial guess to be provided. When this initial guess is close to the solution, the number of iterations required for convergence dramatically decreases. I take advantage of this option in eSCAPE by using the flow discharge solution from the previous time step as an initial guess. This allows to decrease the number of iterations of the IDA solver as discharge often exhibits small change between successive time intervals.

2.2 Erosion and sediment transport

River incision, associated sediment transport and subsequent deposition are critical elements of landscape evolution models. Commonly these are defined based on either a transport-limited (Willgoose et al., 1991) or a detachment-limited (Howard et al., 1994) approach. On one hand, the transport-limited hypothesis assumes that rivers may be able to transport sediment up to a concentration threshold (often referred to as the stream transport capacity) linked to discharge, slope, sediment size, and channel form (channel depth/width ratio) and that an infinite supply of sediment is available for transport. On the other hand, the detachment-limited hypothesis supposes that erosion is not limited by a transport capacity but instead by the ability of rivers to remove material from the bed. Even though validations of each hypothesis have been conducted based on field studies ~~calibration~~ (Snyder et al., 2003; Tomkin et al., 2003; van der Beek and Bishop, 2003; Valla et al., 2010; Hobley et al., 2011) there are many ~~evidences~~ evidence suggesting that both transport and detachment limited ~~behaviours~~ behaviour take place simultaneously in natural systems and models accounting for transition between the two have been proposed in the past (Beaumont et al., 1992; Braun and Sambridge, 1997; Coulthard et al., 2002; Davy and Lague, 2009; Hodge and Hoey, 2012; Salles and Duclaux, 2015; Carretier et al., 2016; Turowski and Hodge, 2017; Lague, 2010; Shobe et al., 2017; Hobley et al., 2017; Salles et al., 2018). For simplicity, the approach proposed in this paper is similar to the initial version of eSCAPE (v1.0.0 - Salles (2018)) and is based on a standard form of the stream power law assuming detachment-limited only behaviour. In the future, a better representation of erosion and sediment transport could be added such as the SPACE approach proposed by Shobe et al. (2017).

As mentioned above and following Howard et al. (1994), I consider that sediment erosion rate is expressed using a stream power formulation function of river discharge and slope. The volumetric entrainment flux of sediment per unit bed area E is of the following form:

$$E = KQ^m S^n \tag{5}$$

where K is the sediment erodibility parameter, Q is the water discharge, S is the river slope. In eSCAPE, I incorporate local precipitation-dependent effects on erodibility (Murphy et al., 2016) and use the flow discharge defined in previous section $Q = PA$ to represent rainfall gradients effect on discharge. A is the flow accumulation and P the upstream annual precipitation rate. m and n are scaling exponents. In our model, K is user defined and the coefficients m and n are set to 0.5 and 1 respectively (Tucker and Hancock, 2010). E is in ~~m/y~~ m/y and therefore the erodibility dimension is $(\mathrm{m{\cdot}y})^{-0.5}$.

The entrainment rate of sediment (E) is approached by an implicit time integration and consists in formulating the stream power component in Eq. 5 in the following way:

$$\frac{\eta_i^{t+\Delta t} - \eta_i^t}{\Delta t} = -K\sqrt{Q_i}\frac{\eta_i^{t+\Delta t} - \eta_{rcv}^{t+\Delta t}}{\lambda_{i,rcv}}$$

where $\lambda_{i,rcv}$ is the length of the edges connecting the considered vertex to its receiver. Rearranging the above equation gives:

$$(1 + K_f)\eta_i^{t+\Delta t} - K_{f,i|rcv}\eta_{i,rcv}^{t+\Delta t} = \eta_i^t \tag{6}$$

with the coefficient $K_{f,i|rcv} = K\sqrt{Q_i}\Delta t/\lambda_{i,rcv}$. In matrix form the system defined in Eq. 6 is equivalent to: $\boldsymbol{\Gamma}\boldsymbol{\eta}^{\,t+\Delta t} = \boldsymbol{\eta}^{\,t}$. Using the case presented in Fig. 1a, the matrix system based on the receivers distribution is defined as:

$$\begin{bmatrix}
\gamma_{1,1} & -\gamma_{1,2} & & -\gamma_{1,4} & -\gamma_{1,5} & & & & & \\
& \gamma_{2,2} & -\gamma_{2,3} & -\gamma_{2,4} & & & & & & \\
& & \gamma_{3,3} & & & & & -\gamma_{3,8} & -\gamma_{3,9} & \\
& & -\gamma_{4,3} & \gamma_{4,4} & -\gamma_{4,5} & -\gamma_{4,6} & & -\gamma_{4,8} & & \\
& & & & \gamma_{5,5} & -\gamma_{5,6} & & & & \\
& & & & & \gamma_{6,6} & & -\gamma_{6,8} & & \\
& & & & & -\gamma_{7,6} & \gamma_{7,7} & -\gamma_{7,8} & & \\
& & & & & & & \gamma_{8,8} & -\gamma_{8,9} & \\
& & & & & & & & 1 & \\
& & & & & & -\gamma_{10,7} & -\gamma_{10,8} & -\gamma_{10,9} & \gamma_{10,10}
\end{bmatrix}
\begin{bmatrix}
\eta_1^{t+\Delta t} \\ \eta_2^{t+\Delta t} \\ \eta_3^{t+\Delta t} \\ \eta_4^{t+\Delta t} \\ \eta_5^{t+\Delta t} \\ \eta_6^{t+\Delta t} \\ \eta_7^{t+\Delta t} \\ \eta_8^{t+\Delta t} \\ \eta_9^{t+\Delta t} \\ \eta_{10}^{t+\Delta t}
\end{bmatrix}
=
\begin{bmatrix}
\eta_1^{t} \\ \eta_2^{t} \\ \eta_3^{t} \\ \eta_4^{t} \\ \eta_5^{t} \\ \eta_6^{t} \\ \eta_7^{t} \\ \eta_8^{t} \\ \eta_9^{t} \\ \eta_{10}^{t}
\end{bmatrix} \tag{7}$$

with

$$\begin{aligned}
\gamma_{i,j} &= w_{i,j} K_{f,i|j} \qquad i \neq j \\
\gamma_{1,1} &= 1 + \sum_{j=2,4,5} w_{1,j} K_{f,1|j} \\
\gamma_{2,2} &= 1 + \sum_{j=3,4} w_{2,j} K_{f,2|j} \\
\gamma_{3,3} &= 1 + \sum_{j=8,9} w_{3,j} K_{f,3|j} \\
\gamma_{4,4} &= 1 + \sum_{j=3,5,6,8} w_{4,j} K_{f,4|j} \\
\gamma_{5,5} &= 1 + w_{5,6} K_{f,5|6} \\
\gamma_{6,6} &= 1 + w_{6,8} K_{f,6|8} \\
\gamma_{7,7} &= 1 + \sum_{j=6,8} w_{7,j} K_{f,7|j} \\
\gamma_{8,8} &= 1 + w_{8,9} K_{f,8|9} \\
\gamma_{10,10} &= 1 + \sum_{j=7,8,9} w_{10,j} K_{f,10|j}
\end{aligned} \tag{8}$$

This system is implicit and the matrix is sparse. The SciPy compressed sparse row matrix functionality (Jones et al., 2001) is used to build $\boldsymbol{\Gamma}$ on local domains. The SciPy matrix format (*e.g.* csr_matrix) is efficiently loaded as a PETSc Python matrix and the Eq. 7 is then solved using Richardson solver with block Jacobi preconditioning (*bjacobi*) using an initial guess for the solution set to vertices elevation.

Once the entrainment rates have been obtained, the sediment flux moving out at every node Q_s^{out} equals the flux of sediment flowing in plus the local erosion rate. Q_s^{out} takes the following form:

$$Q_s^{out} = Q_s^{in} + (1 - F_f) E \Omega$$

Ω is the voronoi area of the considered vertex and F_f is the volumetric fraction of fine sediment ~~that remains~~ small enough to be considered permanently in suspension. As an example, in case where bedrock breaks only into sand and gravel fractions, F_f

would be zero. As a result, simulated deposits and transported sediment flux in the model only include sediment coarse enough that it does not permanently stay in suspension.

The solution of the above equation requires the calculation of the incoming sediment volume from upstream nodes Q_s^{in}. At node i, Eq. 8 is equivalent to:

$$q_{s,i} = e_i + \sum_{d=1}^{N_d} q_{s,d} \tag{9}$$

where $e_i = (1 - F_f)E_i\Omega_i$ and N_d the number of donors. Assuming that river sediment concentration is distributed in a similar way as the water discharge we can write a similar set of equalities as the ones in Eq. 2. Then a matrix system as proposed for the FA (Eq. 4) can be obtained. The new system is then solved using the PETSc solver and preconditioner previously defined.

2.3 Priority-flood depression filling

In most landscape evolution models, internally-draining regions (e.g., depressions and pits) are usually filled before the calculation of flow discharge and erosion-deposition rates. This ensures that all flows conveniently reach the coast or the boundary of the simulated domain. In models intended to simulate purely erosional features, such depressions are usually treated as transient features and often ignored. However, eSCAPE is designed to not only address erosion problems but also to simulate source-to-sink transfer and sedimentary basins formation and evolution in potentially complex tectonic settings. In such cases, depressions may be formed at different periods during runtime and may be filled or remain internally drained (e.g., endorheic basins) depending on the volume of sediment transported by upstream catchments.

Depression filling approaches have received some attention in recent years with the development of new and more efficient algorithms (Wang and Liu, 2006; Barnes et al., 2014; Zhou et al., 2016, 2017; Wei et al., 2018). These methods based on priority-flood offer a time complexity of the order of $O(Nlog(N))$ compared to older approaches such as the Jenson and Domingue (1988) ($O(N^2)$) or Planchon and Darboux (2002) ($O(N^{1.2})$) algorithms.

Priority-flood algorithms consist in finding the minimum elevation a cell needs to be raised to (e.g., spill elevation of a cell) to prevent downstream ascending path to occur. They rely on priority queue data structure used to efficiently find the lowest spill elevation in a grid. Depending on the chosen method, priority queue implementation approaches affect the time complexity of the algorithm (Barnes et al., 2014). In eSCAPE, the priority-flood + ϵ variant of the algorithm proposed in Barnes et al. (2014) is implemented. It provides a solution to remove automatically flat surfaces and it produces surfaces for which each cell has a defined gradient from which flow directions can be determined. Recently, Cordonnier et al. (2019) proposed a different potentially more efficient approach based on a O(N) depression resolving algorithm that explicit compute the flow paths through the construction of a graph connecting together all adjacent drainage basins.

In eSCAPE, this part of the algorithm is not parallelised and is performed on the master processor. It starts from the grid border vertices and processes vertices that are in their immediate neighbourhoods one by one in the ascending order of their spill elevations (Barnes et al., 2014). The initialisation step consists in pushing all the edges nodes onto a priority queue. The priority queue rearranges these nodes so that the ones with the lowest elevations in the queue are always processed first.

Figure 2. Illustration of the two cases that may arise depending on the volume of sediment entering an internally drained depression (panel a). The red line shows the limit of the depression at the minimal spillover elevation. b) The volume of sediment (V_s^{in}) is lower than the depression volume V_{pit}. In this case all sediments are deposited and no additional calculation is required. c) If $V_s^{in} \geq V_{pit}$, the depression is filled up to depression filling elevation (priority-flood + ϵ), the flow calculation needs to be recalculated and the excess sediment flux (Q_s^{ex}) is transported to downstream nodes.

To track nodes that have already been processed by the algorithm a Boolean array is used in which edge nodes (that are by definition at the correct elevation) are marked as solved. The next step consists in removing (i.e. popping) from the priority queue the first element (i.e. the lowest node). This node n is guarantee to have a non-ascending drainage path to the border of the domain. All non-processed neighbours (based on the Boolean array) from the popped node are then added to the priority queue. In the case where a neighbour k is at a lower elevation than n its elevation is raised to the elevation of n plus ϵ before being pushed to the queue. Once k has been added to the queue, it is marked as resolved in the Boolean array. In this basic implementation of the priority-flood algorithm, the process continues until the priority queue is empty (Barnes et al., 2014).

2.4 Depression filling and marine sedimentation

The filling algorithm presented above is used to calculate the volume of each depression at any time step. Once the volumes of these depressions are obtained, their subsequent filling is dependent of the sediment fluxes calculation defined in section 2.2 (Fig. 2a). In cases where the incoming sediment volume is lower than the depression volume (Fig. 2b), all sediments are deposited and the elevation at node i in the depression is increased by a thickness δ_i such that:

$$\delta_i = \Upsilon(\eta_i^f - \eta_i) \tag{10}$$

where η_i^f is the filling elevation of node i obtained with the priority-flood + ϵ algorithm and the ratio Υ is set to V_s^{in}/V_{pit}.
If the cumulative sediment volume transported by the rivers draining in a specific depression is above the volume of the depression ($V_s^{in} \geq V_{pit}$ - Fig. 2c) the elevation of each node i is increased to its filling elevation (η_i^f) and the excess sediment volume is allocated to the spillover node (Fig. 2c). The spillover nodes are obtained using the method proposed by Barnes (2017) where in addition to depressions, the priority-flood approach labels watershed indices. Spillover nodes correspond to the lowest points connecting different watersheds. The updated elevation field is then used to compute the flow accumulation following the approach presented in section 2.1. The sediment fluxes are initially set to zero except on the spillover nodes and using Eq. 9 the excess sediments are transported downstream until all sediments have been deposited in depressions, have entered the marine environment, or have moved out of the simulation domain.
In the marine realm, sedimentation computation follows a different approach to the one described above. First, the flow accumulation is computed using the filled elevation in both the aerial and marine domains and a maximum volumetric deposition rate ζ_i is calculated based on the depth of each marine node:

$$\zeta_i = 0.9(\eta_{sl} - \eta_i)\Omega_i/\Delta t$$

with η_{sl} the sea-level position. Using similar solver and preconditioner as the ones proposed for the flow discharge calculation, we solve implicitly a matrix system equivalent to the one in Eq. 4 with the same weight ($\boldsymbol{W}$) and a vector $\boldsymbol{b}$ equals to $q_{s,i} - \zeta_i$. From the solution, only positive sedimentation rates are initially kept and the sedimentation thicknesses for these nodes are set to $\zeta\Delta t$. Then remaining sediment fluxes on adjacent vertices are found by computing the sum of ζ and obtained sedimentation rates and by considering again only positive values.

2.5 Hillslope processes and marine top sediment layer diffusion

Hillslope processes are known to strongly influence catchment morphology and drainage density and several formulations of hillslope transport laws have been proposed (Culling, 1963; Tucker and Bras, 1998; Perron and Hamon, 2012; Howard et al., 1994; Fernandes and Dietrich, 1997; Roering et al., 1999, 2001). Most of these formulations are based on a mass conservation equation and with some exceptions such as CLICHE model (Bovy et al., 2016), these models assume that a layer of soil available for transport is always present (i.e. precluding case of bare exposed bedrock) and that dissolution and mass transport in solution can be neglected (Perron and Hamon, 2012).

Under such assumptions and via the Exner's law, the mass conservation equation widely applied in landscape modelling is of the form (Dietrich et al., 2003; Tucker and Hancock, 2010):

$$\frac{\partial \eta}{\partial t} = -\nabla \cdot q_{ds} \tag{11}$$

where q_{ds} is the volumetric soil flux of transportable sediment per unit width of the land surface. In its simplest form, q_{ds} obeys the Culling model (Culling, 1963) and hypothesises a proportional relationship to local hillslope gradient (i.e. $q_{ds} = -D\nabla\eta$, also referred to as the creep diffusion equation):

$$\frac{\partial \eta}{\partial t} = D\Delta\eta \tag{12}$$

in which D is the diffusion coefficient that encapsulates a variety of processes operating on the superficial soil layer. As an example, D may vary as a function of substrate, lithology, soil depth, climate and biological activity (Tucker et al., 2001; Tucker and Hancock, 2010). The creep law is found in many models such as GOLEM (Tucker and Slingerland, 2017), CHILD (Tucker and Slingerland, 1997), LANDLAB (Hobley et al., 2017) or Badlands (Salles and Hardiman, 2016; Salles et al., 2018), and in Willgoose et al. (1991), Fernandes and Dietrich (1997), Tucker and Slingerland (1997), Simpson and Schlunegger (2003). In eSCAPE, hillslope processes rely on this approximation even though field ~~evidences~~ evidence suggest that the creep approximation (Eq. 12) is only rarely appropriate (Roering et al., 1999; Tucker and Bradley, 2010; Foufoula-Georgiou et al., 2010; DiBiase et al., 2010; Larsen and Montgomery, 2012; Grieve et al., 2016a). In the future, a possible improvement could be based on the nonlinear hillslope transport equation incorporating a critical slope to model hillslope soil flux (Roering et al., 1999, 2001).

For a discrete element, considering a node i the implicit finite volume representation of Eq. 12 is:

$$\frac{\partial \eta_i}{\partial t} = \frac{\eta_i^{t+\Delta t} - \eta_i^t}{\Delta t} = D \sum_{j=1}^{N} \frac{\chi_{i,j}(\eta_j^{t+\Delta t} - \eta_i^{t+\Delta t})}{\Omega_i \lambda_{i,j}} \tag{13}$$

N is the number of neighbours surrounding node i, Ω_i is the voronoi area, $\lambda_{i,j}$ is the length of the edge connecting the considered nodes and $\chi_{i,j}$ is the length of voronoi face shared by nodes i and j. Applied to the entire domain, the equation above can be rewritten as a matrix system $\boldsymbol{Q\eta}^{\ t+\Delta t} = \boldsymbol{\eta}^{\ t}$ where $\boldsymbol{Q}$ is sparse. The matrix terms only depend on the diffusion coefficient D, the grid parameters and voronoi variables ($\chi_{i,j}$, $\lambda_{i,j}$, Ω_i). In eSCAPE, these parameters remain fixed during a model run and therefore $\boldsymbol{Q}$ needs to be created once at initialisation. At each iteration, hillslope induced changes in elevation $\boldsymbol{\eta}$ are then obtained in a similar way as for the solution of the other systems using PETSc Richardson solver and block Jacobi preconditioning.

In addition to hillslope processes, a second type of diffusion is available in eSCAPE and consists in distributing freshly deposited marine sediments in deeper regions. This process is the only one treated explicitly and in this case the length of the diffusion time step Δt_m must be less than a CFL factor to ensure numerical stability:

$$\Delta t_m < 0.1 \min_{i,j} \left(\lambda_{i,j}^2 / D_m\right) \tag{14}$$

where D_m is the diffusion coefficient for the newly deposited marine sediments. Even with a reasonable small time step, the Eq. 14 can produce incorrect results. Following Bovy et al. (2016), the following set of inequalities are also added:

$$\begin{aligned} \Delta t \textstyle\sum_j \chi_{i,j} q_{ms,ij}^{out} &\leq h_i \Omega_i \\ &\leq \alpha(\eta_i - \eta_m)\Omega_i \end{aligned} \tag{15}$$

where $q_{ms,ij}^{out}$ is the flux of sediment from the marine top layer leaving node i towards the downstream neighbours j, h_i is the depth of the marine top layer, η_m is the elevation associated to the highest downslope neighbour of i and α is a factor lower than 1. These inequalities are always satisfied if positive outgoing fluxes are scaled by a factor β given by:

$$\beta_i = \min\left(\frac{\Omega_i \min(h_i, \alpha(\eta_i - \eta_m))}{\Delta t \sum_j \chi_{i,j} q_{ms,ij}^{out}}, 1\right) \tag{16}$$

In eSCAPE, marine diffusion of freshly deposited sediment is performed explicitly using the CFL condition described in Eq. 14 and the restriction proposed in Eq. 16.

3 Usability and applications

In this section, I present the main files used to run eSCAPE and to visualise the generated outputs. I then illustrate the capability of the code using a series of 3 examples presenting two generic models and one global scale experiment.

3.1 Input parameters and visualisation

eSCAPE uses YAML syntax for its input file. YAML structure is done through indentation (one or more spaces) and sequence items are denoted by a dash. When parsing the input file, the code is searching for some specific keywords defined in tables 1, 2 and 3. Some parameters are optional and only need to be set when specific forces (table 2) or physical processes (table 3) are applied to a particular experiment.

Table 1. Input parameters relative to intial surface, temporal extent and output.

Parameters	*Definition*	*Default values*
name	Description of simulation - string	optional
domain	Definition of the simulated region	required
filename	TIN grid (vtk file) and elevation field - list	required
flowdir	Flow direction method integer between [1,12]	1
bc	Boundary conditions (choices: flat, fixed or slope)	slope
sphere	Set to 1 for spherical experiments	0
time	Simulation time definition all values are in years	required
start	Simulation start time	required
end	Simulation end time	required
tout	Simulation output interval time	required
dt	Simulation time step	required
output	Output folder	optional
dir	Directory name containing Hdf5, XMF and XDMF outputs	optional - Default name: output
makedir	Boolean is False: output folder with same name is deleted or	True
	Boolean is True: Keep previous folder adding a number	

Table 2. Input parameters relative to forcing conditions.

Parameters	*Definition*	*Default values*
sea	Sea-level forcing	optional
position	Relative sea-level position (m)	0.
curve	File containing 2 columns (time and sea-level position)	optional
climate	Sequence of precipitation events in m/yr	optional
start	Starting time of a given event in year	required if module turned on
uniform	Either an uniform value or	0.
map	A VTK map of spatial change in precipitation	
tectonic	Sequence of vertical tectonic events in m/yr	optional
start	Starting time of a given event in year	required if module turned on
step	Time step to apply tectonic time step in year	optional when **sphere**=1
end	Ending time of a given event in year	optional when **sphere**=1
uniform	Either an uniform value applied to all domain except edges or	0.
mapX	Displacement VTK maps along esch axis defined either	
mapY	as rate in m/yr if the **sphere** parameter is set to 0 or	
mapZ	as a distance in m if **sphere**=1	

Table 3. Input parameters relative to physical processes.

Parameters	*Definition*	*Default values*
sp_br	Stream power parameters for bedrock	required
Kbr	Bedrock erodibility ($\text{m}^{-0.5}\text{yr}^{-0.5}$)	$1.e^{-12}$
sp_dep	Deposition parameter definition	optional
Ff	Fraction of sediment in suspension [0.,1.]	0.
diffusion	Diffusion parameters declaration	optional
hillslopeK	Hillslope diffusion coefficient (m^2/yr)	required
sedimentK	Marine fresh sediment coefficient (m^2/yr)	10.

All the input parameters that are defined in external files like the initial surface, different precipitation or displacement maps are read from VTK files. These input files are defined on an irregular triangular grid (TIN). Examples on how to produce these files are provided in the eSCAPE demo repository on Github and Docker. The only exception is the sea-level file which is a two-columns CSV file containing on the first column the time in years and ordered in ascending order and on the second one the relative position of the sea-level in metres (**curve** in table 2).

The **domain** and **time** keywords (table 1) are required for any simulation. The flow direction method to be applied in a given simulation is specified with **flowdir** and takes an integer value ranging between 1 (for SFD) and 12 (for MFD/D_∞). On the edges of the domain three types of boundary conditions (**bc**) are available and applied to all edges *flat*, *fixed* or *slope*. The *flat* option assumes that all edges elevations are set to the elevations of their closest non-edge vertices, the *fixed* option is used when edges elevations need to remain at their initial positions during the model run and the *slope* option defines a slope based on the closest non-edge node average slope.

The **climate** and **tectonic** keywords (table 2) may be defined as a sequence of multiple forcing conditions each requiring a starting time (**start** in years) and either a constant value applied to the entire grid (**uniform**) or spatially varying values specified in a file (**map**).

Surface processes parameters (table 3) define the coefficients for the stream power law (**Kbr** is K in Eq. 5). It is worth noting that the coefficient m and n are fixed in this version of eSCAPE and take the value 0.5 and 1 respectively. The **diffusion** keyword defines both hillslope (creep law – Eq. 12 and **hillslopeK** is D in $q_{ds} = -D\nabla\eta$) and marine diffusion coefficients. The freshly deposited marine sediments are transported based on a diffusion coefficient **sedimentK** equivalent to D_m in $q_{ms} = -D_m\nabla\eta$ and used in Eq. 13 with the restriction proposed in Eq. 16.

The model outputs are located in the output folder (**dir** keyword – table 1) and consist of a time series file named eSCAPE.xdmf and two additional folders (h5 and xmf). The HDF5 files are wrote individually for each processors and the XMF files combine them together to show the global outputs. The XDMF file is the main entry point for visualising the outputs and should be sufficient for most users. The file can be opened with the Paraview software (Ahrens et al., 2014).

Figure 3. Resulting topographies of an initial flat squared surface (100 km side) after 100,000 years of uniform precipitation and linear uplift from west to east. Three simulations are performed in which the time step Δt is set to a) 10^4, b) 10^3 and c) 10^2 years. Panel d) presents the temporal change in mean elevation for the three cases. Differences between the runs are related to the transient nature of landscape evolution.

3.2 Examples

3.2.1 Analysing the influence of time step on eSCAPE runs

The first example illustrates the effect of increasing time step length on the resulting landscape evolution. The initial surface consists in a flat triangulated squared grid of 100 km side and approximately 100 m resolution containing $\simeq$1.3 million points. This surface is exposed to an uniform precipitation regime of 1 m/y and is uplifted linearly from its fixed western side to the eastern one that experiences an uplift of 5 mm/y (Fig. 3a). The proposed setting is similar to the one in Braun and Willett (2013) and the value of the bedrock erodibility parameter K is set to 2×10^{-4} in order to reach steady-state during the simulated 10^5 years. Under such conditions, the model is purely erosional and therefore neither the aerial and marine sedimentation nor the depression filling algorithm are considered. In addition hillslope processes are also turned off, meaning that this example only relies on the implicit parallel flow discharge and erosion equations defined in sections 2.1 and 2.2.

Three cases are presented after 10^5 years for different time steps Δt varying from 10^4 to 10^3 and 10^2 years in Figure 3a, 3b and 3c respectively, implying that the number of steps is 10, 100 and 1000. In both cases the implicit schemas converge for the chosen solver and preconditioner (*i.e.* Richardson with block Jacobi). The solutions for the mean landscape elevation (Fig. 3d) show that the landscape reaches steady state in all cases and overall the final elevations are in good agreement with a maximum elevation of 482 ± 3 m and a number of catchments n_c almost identical between models ($87 \leq n_c \leq 94$). Yet as the time step increases the differences between models increase over time. By the end of the simulation, the mean elevation difference between the case with Δt equals to 10^2 y and the one at 10^3 y is around 2.5% whereas the difference with a Δt of 10^4 y is above 30% (Fig. 3d). It illustrates the transient nature of the landscape and its strong dependence to antecedent morphologies. Even small changes on elevation could potentially trigger completely different landscape features. Compared to the explicit algorithm proposed for the drainage area computation in Braun and Willett (2013), the approach here relies on an implicit schema and produces a more stable solution for longer time scale. Yet time step limitations are still required to ensure a good representation of landscape features (*e.g.* knickpoint propagation) and care should be taken when choosing a given simulation time step.

As mentioned in section 2.1, the iterative linear solvers of the implicit methods for both flow accumulation and erosion use previous time step solution as an initial guess. In cases where the landscape does not change significantly between consecutive time steps, both the flow accumulation and erosion rates are likely to remain almost unchanged and the number of iterations required by the solver to reach convergence will be small. As an example if the drainage network remains the same between two iterations, the flow accumulation solver solution will be obtained immediately and the results given directly. It highlights a second implication of the choice of time step. Not only does the time step influences the final landscape morphology, it

Figure 4. Effect of flow routing algorithms on flow accumulation patterns and associated erosion. Panel a) presents the initial radially symmetric surface defined with a central, high region and a series of distal low-lying valleys. Resulting topographies of the south-west area after 100,000 years of evolution under uniform precipitation for the SFD and MFD algorithms are shown on the right hand-side. Patterns of flow accumulation after 20,000 and 50,000 years for the SFD, two neighbours and MFD approaches are presented in panel b) as well as estimated landscape erosion at the end of the simulation time interval are given in panel c).

also controls the model running time. In some cases, similar running times will be achieved with smaller time steps if solvers solutions are obtained in a reduced number of iterations.

3.2.2 Comparison of single and multiple flow direction algorithms

In this second example, I present a series of three experiments in which the flow routing calculations are based on one (SFD), two and multiple (MFD) flow direction approaches (Fig. 4). In eSCAPE, it is possible to use different flow-routing algorithms by specifying the number of directions (Fig. 1a and **flowdir** parameter in table 1) appropriately weighted by slope that rivers could potentially take when moving downhill.

For this example, the initial surface consists in a rotationally symmetric surface (Fig 4a) composed of valleys and ridges with lowest regions (at 0 m elevation) located on the edges of the domain and increasing to 1000 metres towards the center. The triangulated circular grid of 50 km radius is built with a resolution of approximately 200 m. The three experiments with varying water routing directions are ran for 100,000 years with a Δt of 1000 years under a 1 m/y uniform precipitation. In addition to stream incision (bedrock erodibility K set to 2×10^{-5}), hillslope processes are also accounted for using a diffusion coefficient D of 10^{-2} m^2/y.

After 20,000 years, the dendritic flow accumulation pattern observed on the surface for the SFD case (Fig. 4b) is analogue to many natural forms of drainage systems but is actually a numerical artefact and depends on the random locations of the nodes in the surface triangulation. By increasing the number of possible downstream directions, this sensitivity to the mesh discretisation is significantly reduced (as illustrated in Fig. 4b where a second direction is added). In addition, routing flow to more than one destination node allows a better representation of channels pathways divergence into multiple branches over flat regions (Tucker and Hancock, 2010).

Landscape evolution models tend to be highly dependent on grid resolution and this dependency is mostly related to the approach used to route water down the surface (Schoorl et al., 2000; Pelletier, 2004; Armitage, 2019). As discussed by Armitage (2019), enabling *node-to-node* MFD algorithm decreases the dependence of landscape features (*e.g.* valley spacing, branching of stream network, sediment flux etc) to grid resolution. As shown in Figure 4b and c, SFD algorithm leads to increase branching of valleys whereas the MFD approach, by promoting wider flow distribution, produces smoother topography where local carving of the landscape is reduced. Armitage (2019) also showed that when using models that operate at scale larger than river width resolution, *node-to-node* MFD algorithm creates landscape features that are not resolution dependent and that evolve closer to the ones observed in nature. Therefore it is recommended to use more than one downhill direction (**flowdir**)

Figure 5. eSCAPE global scale experiment of Earth morphological evolution over 200,000 years. Panel a) presents the initial elevation based on ETOPO1 dataset and forcing precipitation obtained from the WorldClim dataset. Panels b) and c) show the elevation and cumulative erosion-deposition resulting from the action of rivers and hillslope processes at different time intervals.

Figure 6. Similar to figure 5 from a different perspective.

in eSCAPE when looking at global and continental scale landscape evolution or for cases where multiple resolutions are considered within a given mesh.

3.2.3 Global scale simulation

The last example showcases a global scale experiment with eSCAPE. The simulation looks at the evolution of the Earth 200,000 years into the future starting with present day elevation and precipitation maps. The model is ran forward in time without changing the initial forcing conditions and is primarily used to highlight the capabilities of eSCAPE and does not represent any particular geological situation.

The elevation is obtained from the ETOPO1 1 arc-minute global relief model of Earth's surface that integrates land topography and ocean bathymetry (Amante and Eakins, 2009). For the rainfall, I summed all the WorldClim gridded climate monthly dataset to obtained a global yearly rainfall map with a spatial resolution of about 1 km^2 (Fick and Hijmans, 2017). From these dataset, I then built the initial surface and climate meshes at 16 km resolution consisting in approximately 3 millions points (Fig. 5a and 6a). The model inputs are temporally uniform, but any other climatic scenarios could have been chosen for illustration as well as tectonic conditions (both vertical: uplift and subsidence and horizontal: advection displacements).

In addition to these grids, the following parameters are chosen: **flowdir** is set to 5 (similar to the MFD flow routing approach), the time step Δt equals 500 years, bedrock erodibility K is 5×10^{-5}, the diffusion coefficient D equals 10^{-1}, the fraction of sediment in suspension **Ff** is 0.3 (F_f parameter in section 2.2 and defined in table 3) and the marine fresh sediment diffusion D_m is set to 5×10^5 (see section 2.5 and **sedimentK** parameter in table 3). The simulation took 2 hours to run on a cluster using 32 processors.

From these set of input parameters, eSCAPE can predict the global evolution of topography (Figures 5 and 6) and quantifies the associated volume and spatial distribution of sediments trapped in continental plains or transported into the marine realm. By recording eroded sediment transport over each drainage basin, eSCAPE provides estimation of sedimentary mass fluxes carried by major rivers into the ocean (section 2.2). As shown in Figures 5 and 6, the predicted locations of largest basin outlets match quite well with observations and many of the biggest simulated deltaic systems are related to sediment transported by some of the world's largest rivers ~~(Milliman and Syvitski, 1992; Syvitski et al., 2003)~~(Milliman and Syvitski, 1992; Syvitski et al., 2003; Li et al., 2
. The model can also be used to evaluate the evolution of drainage systems, the stability of continental flow directions, the exhumation history of major mountain ranges, the timing and geometry of sedimentary bodies formation (*e.g.* deltas or intra-

continental deposits) as well as basins stratigraphy. All these predictions can be directly compared to sedimentary (sediment budgets, paleogeographic maps, etc) or thermochronology data.

This simulation illustrates a global scale model of Earth's surface evolution. In case where paleo-climatic conditions are known then eSCAPE can in principle be used to perform quantitative analysis of different tectonic forcings with complex spatial and temporal variations. The results of these tests can then be ~~compare~~ compared with available geological records (such as denudation rates, paleotopographies, basins sedimentary thicknesses/volumes). In addition to climate and tectonic conditions, it is also possible to impose varying sea level fluctuations over geological times.

As such and even with the limited number of simulated processes, eSCAPE can be used to retrieved global sedimentary basins formation and their evolution based on temporal and spatial responses of both landscape and sediment fluxes to different sea level conditions and tectonic and precipitation regimes.

4 Performance analysis

The performance of the implicit flow accumulation, erosion and sediment transport algorithms is strongly dependent on the choice of solver and preconditioner. As shown in sections 2.1 and 2.2, the forms of the matrices are not symmetric or positive definite and in this case only a limited number of iterative solvers and preconditioners are suitable. From the extensive analysis provided in Richardson et al. (2014), the non-Krylov solver based on the Richardson method (Richardson, 1910) has been chosen in eSCAPE as it converged with the greatest number of preconditioners and exhibited superior scaling. For the preconditioner, several candidates (SOR, ILU, ASM...) are available (Saad, 2003) and I decided to use the block Jacobi preconditioner as it is one of the simplest methods and produces in combination with the Richardson solver good scalability (Richardson et al., 2014). Yet other combinations such as the Richardson solver with the Euclid preconditioner (*i.e.* HYPRE package, Falgout et al. (2012)) might exhibit better scalability in some cases.

The analysis of the profiling work realised in Fig. 7a suggests that for purely erosive models (similar to the ones presented in section 3.2.1) most of the computational time is spent solving the Richardson iterative method (*PETSc solver KSP*). From the graph on the right hand side of the panel (Fig. 7a), one can deduce that performance improvements are obtained when the problem size increases. However the scaling performance decreases when reaching 32/64 processors depending on the problem size. This does not agree with some of the conclusions from Richardson et al. (2014), where the scaling of the implicit drainage accumulation algorithm continues even for large numbers of processors (>192). To improve performance, I will be exploring two directions. First the problem might be related to the chosen solver and preconditioner combination and I will run new tests using the HYPRE package as discussed above. Secondly, the poor performance for larger processors might also be linked to issues related to either Python PETSc wrapper or installation problems and incompatibilities between some of the compilers and packages that I used. In the future, different software libraries and compilers versions from GNU and Intel will be tested and might help to improve the performance for increasing number of processors.

For experiments accounting for marine deposition and pit filling, a similar trend is found when comparing performance against processors number (right hand side in Fig. 7b). However the results from the profiling (left hand side in Fig. 7b) suggest that

Figure 7. Sunburst visualisation obtained from SnakeViz package showing the profiling results of multiple eSCAPE experiments. The analysis is performed for different numbers of processors (up to 256). On the left hand side, the fraction of time spent in each function is represented by the angular extent of the different arcs. On the right hand-side results of the computational runtime versus processors number over a series of time steps is given for experiments of different size. Panel a) presents the results for purely erosional simulations such as the ones presented in the first example (section 3.2.1). For panel b) eSCAPE is ran with all the processes turned on and uses a global scale experiments similar to the last example (section 3.2.3).

more than half of the computation time is now spent on non-PETSc work with the biggest proportion related to the pit filling function. In eSCAPE, the priority-flood algorithm is performed in serial (see section 2.3). This is the major limitation of the code as shown by the time spent in broadcasting the information from the master node to the other processors (MPI Bcast in Fig. 7b). To take advantage of parallel architectures, several authors (Wallis et al., 2009; Tesfa et al., 2011; Yıldırım et al., 2015; Zhou et al., 2017) have proposed partitioning implementation of depression filling algorithms. However, most of these methods require frequent interprocess communication and synchronisation. This becomes even more problematic in the case of eSCAPE where the depression filling algorithm needs to be performed at every time step (Barnes, 2019). Barnes (2017) presented an alternative to the aforementioned parallelisation methods that limits the number of communications. Yet this approach is not fully satisfactory as it only fills the depressions up to the spilling elevation but does not provide a way of implementing efficiently the ϵ variant of the algorithm proposed in (Barnes et al., 2014). Finding a strategy to perform a parallel version of the ϵ variant of the priority-flood algorithm ~~that could provide~~ or to efficiently fill the depressions (Cordonnier et al., 2019) while providing flow directions on flat surface will likely improve greatly the performance of eSCAPE.

5 Conclusions

In this paper, I describe eSCAPE, an open-source, Python-based software designed to simulate sediment transport, landscape dynamics and sedimentary basins evolution under the influence of climate, sea level and tectonics. In its current form, eSCAPE relies on the stream power and creep laws to simulate the physical processes acting on the Earth's surface. The main difference with other landscape evolution models relies on the formulation used to solve the system of equations. The approach builds upon the Implicit Drainage Area calculation from Richardson et al. (2014) and consists in a series of implicit iterative algorithms for calculating multiple flow direction and erosion deposition that written in matrix form. As a result, the obtained systems can be solved with widely available parallel linear solver packages such as PETSc.

Performance analysis shows good parallel scaling for small number of processors (under 64 processors as shown in section 4) but some work is required for larger numbers. The code profiling suggests that the main issue is in the inter-processes communications happening when broadcasting the pit filling information computed in serial by the master to the other processors. In the future, a parallel approach allowing depression filling and flow direction computation over flat regions will be critical to improve the overall performance of the code.

Examples are provided in the paper and available through the Docker container. They illustrate the extent of temporal and spatial scales that can be addressed using eSCAPE. As such this code is highly versatile and useful for geological applications related to source to sink problems at regional, continental and for the first time global scale. It is already possible to use eSCAPE to simulate global geological evolution of the Earth's landscape at about 1 km resolution providing accurate estimates of quantities such as large-scale erosion rates, sediment yields and sedimentary basins formation. In the future, the code will be coupled with atmospheric and geodynamic models to bridge the gap between local and global scales predictions of Earth past and future evolutions.

Code and data availability. The source code with examples (Jupyter Notebooks) is archived as a repository on Github as the release version v2.0 from Zenodo (doi:10.5281/zenodo.3239569). The code is licensed under the GNU General Public License v3.0. The easiest way to use eSCAPE is via our Docker container (searching for **Geodels escape-docker** on Kitematic) which is shipped with the complete list of dependencies and the case studies presented in this paper. Our wiki page provides useful documentation regarding installation and code usage. API documentation is available from the eSCAPE-API website.

Author contributions. T.S., development of the code, design of the experiment, output analysis, and manuscript writing.

Competing interests. The author declares that he has no conflict of interest.

Acknowledgements. The author acknowledge the Sydney Informatics Hub and the University of Sydney's high performance computing cluster Artemis for providing the high performance computing resources that have contributed to the research results reported within this paper. The author also acknowledge the technical assistance of David Kohn from the Sydney Informatics Hub which was supported by Project #3789 and from Artemis HPC Grand Challenge. I also thank John Armitage, Benoit Bovy and the journal editor for their comments that greatly improved the manuscript.

References

Ahrens, J., Jourdain, S., O'Leary, P., Patchett, J., Rogers, D. H., and Petersen, M.: An image-based approach to extreme scale in situ visualization and analysis, Proceedings of the International Conference for High Performance Computing, 2014.

Amante, C. and Eakins, B. W.: ETOPO1 1 Arc-Minute Global Relief Model: Procedures, Data Sources and Analysis., NOAA Technical Memorandum NESDIS NGDC-24, 19 pp, https://doi.org/[http://www.ngdc.noaa.gov/mgg/global/global.html], 2009.

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, 2019.

Balay, S., Brown, J., Buschelman, K., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H.: Argonne National Laboratory, PETSc, Web page. [Available at http://www.mcs.anl.gov/petsc.], 2012.

Barnes, R.: Parallel non-divergent flow accumulation for trillion cell digital elevation models on desktops or clusters, Environmental Modelling & Software, 92, 202–212, https://doi.org/10.1016/j.envsoft.2017.02.022, 2017.

Barnes, R.: Accelerating a fluvial incision and landscape evolution model with parallelism, Geomorphology, 330, 28–39, https://doi.org/10.1016/j.geomorph.2019.01.002, 2019.

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

Beaumont, C., Fullsack, P., and Hamilton, J.: Erosional control of active compressional orogens, in: Thrust Tectonics, edited by: McClay, K. R., Chapman Hall, New York, pp. 1–18, https://doi.org/10.1007/978-94-011-3066-0_1, 1992.

Bellugi, D., Dietrich, W. E., Stock, J., McKean, J., Kazian, B., and Hargrove, P.: Spatially explicit shallow landslide susceptibility mapping over large areas., in Fifth International Conference on Debris-flow Hazards Mitigation, Mechanics, Prediction and Assessment, edited by R. Genevois, D. L. Hamilton, and A. Prestininzi, Casa Editrice Universita La Sapienza, Rome, pp. 309–407, https://doi.org/10.4408/IJEGE.2011-03.B-045, 2011.

Beucher, R., Moresi, L., Giordani, J., Mansour, J., Sandiford, D., Farrington, R., Mondy, L., Mallard, C., Rey, P., Duclaux, G., Kaluza, O., Laik, A., and Moroon, S.: UWGeodynamics: A teaching and research tool for numerical geodynamic modelling., Journal of Open Source Software, 4, 1136, https://doi.org/10.21105/joss.01136, 2019.

Bianchi, V., Salles, T., Ghinassi, M., Billi, P., Dallanave, E., and Duclaux, G.: Numerical modeling of tectonically driven river dynamics and deposition in an upland incised valley, Geomorphology, 241, 353–370, https://doi.org/10.1016/j.geomorph.2015.04.007, 2015.

Bovy, B., Braun, J., and Demoulin, A.: Soil production and hillslope transport in mid-latitudes during the last glacial-interglacial cycle: a combined data and modelling approach in northern Ardennes, Earth Surface Processes and Landforms, 41, 1758–1775, https://doi.org/10.1002/esp.3993, 2016.

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, 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, 2013.

Brown, J. L., Hill, D. J., Dolan, A. M., Carnaval, A. C., and Haywood, A. M.: PaleoClim, high spatial resolution paleoclimate surfaces for global land areas, Scientific Data, 5, 180 254, 2018.

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

Chen, A., Darbon, J., and Morel, J.-M.: Landscape evolution models: a review of their fundamental equations., Geomorphology, 219, 68–86, 2014.

Cordonnier, G., Bovy, B., and Braun, J.: A versatile, linear complexity algorithm for flow routing in topographies with depressions, Earth Surface Dynamics, 7, 549–562, https://doi.org/10.5194/esurf-7-549-2019, 2019.

Coulthard, T. J., Macklin, M. G., and Kirkby, M. J.: A cellular model of Holocene upland river basin and alluvial fan evolution, Earth Surf. Proc. Land., 27, 269–288, https://doi.org/10.1002/esp.318, 2002.

Culling, W. E. H.: Soil creep and the development of hillside slopes, The Journal of Geology, 71, 127–161, 1963.

Davy, P. and Lague, D.: Fluvial erosion/transport equation of landscape evolution models revisited., J. Geophys. Res. - Earth, 114, https://doi.org/10.1029/2008JF001146, 2009.

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, 2010.

Dietrich, W. E., Bellugi, D., Sklar, L. S., Stock, J., Heimsath, A. M., and Roering, J. J.: Geomorphic transport laws for predicting landscape form and dynamics, in: Wilcock, P. R., Iverson, R. M. (Eds.), Prediction in Geomorphology, 135, 103–132, 2003.

Dutkiewicz, A., Müller, R. D., Hogg, A. M., and Spence, P.: Vigorous deep-sea currents cause global anomaly in sediment accumulation in the Southern Ocean, Geology, 44, 663–666, https://doi.org/10.1130/G38143.1, 2016.

Eddins, S.: Upslope area – Forming and solving the flow matrix, MathWorks, [Available at http://blogs.mathworks.com/steve/2007/08/07/upslope-area-flow-matrix/.], 2007.

Falgout, R., Kolev, T., Schroder, J., Vassilevski, P., and Yang, U. M.: Lawrence Livermore National Laboratory, HYPRE Package - Available at https://computation.llnl.gov/casc/hypre/software.html, 2012.

Fernandes, N. and Dietrich, W. E.: Hillslope evolution by diffusive processes: The timescale for equilibrium adjustments, Water Resources Reasearch, 33, 1307–1318, 1997.

Fick, S. E. and Hijmans, R. J.: WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas, International Journal of Climatology, 37, 4302–4315, https://doi.org/10.1002/joc.5086, 2017.

Foufoula-Georgiou, E., Ganti, V., and Dietrich, W. E.: A nonlocal theory of sediment transport on hillslopes, Journal of Geophysical Research: Earth Surface, 115, 2010.

Grieve, S., Mudd, S., and Hurst, M.: How long is a hillslope?, Earth Surf. Process. Landforms, 41, 1039–1054, https://doi.org/10.1002/esp.3884, 2016a.

Grieve, S., Mudd, S., Hurst, M., and Milodowski, D.: A nondimensional framework for exploring the relief structure of landscapes, Earth Surf. Dynam., 4, 309–325, https://doi.org/10.5194/esurf-4-309-2016, 2016b.

Heister, T., Dannberg, J., Gassmöller, R., and Bangerth, W.: High Accuracy Mantle Convection Simulation through Modern Numerical Methods. II: Realistic Models and Problems, Geophysical Journal International, 210, 833–851, https://doi.org/10.1093/gji/ggx195, https://doi.org/10.1093/gji/ggx195, 2017.

Hobley, D. E. J., Sinclar, H. D., Mudd, S. M., and Cowie, P. A.: Field calibration of sediment flux dependent river incision, J. Geophys. Res. - Earth, 116, https://doi.org/10.1029/2010JF001935, 2011.

Hobley, D. E. J., Adams, J. M., Nudurupati, S. S., Hutton, E. W. H., Gasparini, N. M., Istanbulluoglu, E., and Tucker, G. E.: Creative computing with Landlab: an open-source toolkit for building, coupling, and exploring two-dimensional numerical models of Earth-surface dynamics, Earth Surface Dynamics, 5, 21–46, https://doi.org/10.5194/esurf-5-21-2017, 2017.

Hodge, R. A. and Hoey, T. B.: Upscaling from grain-scale processes to alluviation in bedrock channels using a cellular automaton model, J. Geophys. Res., 117, https://doi.org/10.1029/2011JF002145, 2012.

Howard, A. D., Dietrich, W. E., and Seidl, M. A.: Modeling fluvial erosion on regional to continental scales, Journal of Geophysical Research: Solid Earth, 99, 13 971–13 986, 1994.

Jenson, S. and Domingue, J.: Extracting topographic structure from digital elevation data for geographic information system analysis, Photogramm. Eng. Remote Sens., 54, 1593–1600, 1988.

Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python, http://www.scipy.org/, 2001.

Lague, D.: Reduction of long-term bedrock incision efficiency by short-term alluvial cover intermittency, J. Geophys. Res., 115, https://doi.org/10.1029/2008JF001210, 2010.

Larsen, I. J. and Montgomery, D. R.: Landslide erosion coupled to tectonics and river incision, Nature Geoscience, 5, 468, 2012.

Li, F., Griffiths, C. M., Dyt, C. P., Weill, P., Feng, M., Salles, T., and Jenkins, C.: Multigrain seabed sediment transport modelling for the south-west Australian Shelf, Marine and Freshwater Research, 60, 774–785, https://doi.org/10.1071/MF08049, 2009.

Mark, D. M.: Network models in geomorphology, Modelling Geomorphological Systems, ed: M. G. Anderson, Wiley, N. Y., pp. 73–97, 1988.

Milliman, J. and Syvitski, J.: Geomorphic/tectonic control of sediment discharge to the ocean: the importance of small mountainous rivers, The Journal of Geology, 100, 525–544, 1992.

Moresi, L., Dufour, F., and Muhlhaus, H.: A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials, Journal of Computational Physics, 184, 476–497, https://doi.org/10.1016/S0021-9991(02)00031-1, 2003.

Murphy, B. P., Johnson, J. P. L., Gasparini, N. M., and Sklar, L. S.: Chemical weathering as a mechanism for the climatic control of bedrock river incision, Nature, 532, 223, 2016.

O'Callaghan, J. F. and Mark, D. M.: The extraction of drainage networks from digital elevation data, Computer Vision, Graphics, and Image Processing, 28, 323–344, 1984.

Pelletier, J. D.: Persistent drainage migration in a numerical landscape evolution model, Geophys. Res. Lett., 31, https://doi.org/10.1029/2004GL020802, 2004.

Perron, J. T. and Hamon, J. L.: Equilibrium form of horizontally retreating, soil-mantled hillslopes: Model development and application to a groundwater sapping landscape, J. Geophys. Res., 117, https://doi.org/10.1029/2011JF002139, 2012.

Planchon, O. and Darboux, F.: A fast, simple and versatile algorithm to fill the depressions of digital elevation models, Catena, 46, 159–176, https://doi.org/10.1016/S0341-8162(01)00164-3, 2002.

Quinn, P., Beven, K., Chevallier, P., and Planchon, O.: The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models, Hydrol. Processes, 5, 59–79, https://doi.org/10.1002/hyp.3360050106, 1991.

Richardson, A., Hill, C. N., and Perron, J. T.: IDA: An implicit, parallelizable method for calculating drainage area, Water Resources Research, 50, 4110–4130, https://doi.org/10.1002/2013WR014326, 2014.

Richardson, L.: On the approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam, Proc. R. Soc. London Ser. A, 83, 335–336, 1910.

Roering, J. J., Kirchner, J. W., Sklar, L. S., and Dietrich, W. E.: Hillslope evolution by nonlinear creep and landsliding: An experimental study, Geology, 29, 143—-146, 1999.

Roering, J. J., Kirchner, J. W., and Dietrich, W. E.: Hillslope evolution by nonlinear, slope-dependent transport: Steady state morphology and equilibrium adjustment timescales, Journal of Geophysical Research: Solid Earth, 106, 16 499–16 513, 2001.

Saad, Y.: Iterative Methods for Sparse Linear Systems, Soc. for Ind. and Appl. Math. - Philadelphia, Penn, 2003.

Salles, T.: Badlands: A parallel basin and landscape dynamics model, SoftwareX, 5, 195–202, 2016.

Salles, T.: eSCAPE: parallel global-scale landscape evolution model, Journal of Open Source Software, 3, 964, 2018.

Salles, T. and Duclaux, G.: Combined hillslope diffusion and sediment transport simulation applied to landscape dynamics modelling., Earth Surf. Process Landf., 40, 823–39, 2015.

Salles, T. and Hardiman, L.: Badlands: An open-source, flexible and parallel framework to study landscape dynamics, Computers & Geosciences, 91, 77–89, 2016.

Salles, T., Griffiths, C., Dyt, C., and Li, F.: Australian shelf sediment transport responses to climate change-driven ocean perturbations., Marine Geology, 282, 268–274, 2011.

Salles, T., Flament, N., and Müller, D.: Influence of mantle flow on the drainage of eastern Australia since the Jurassic Period, Geochemistry, Geophysics, Geosystems, 18, 280–305, 2017.

Salles, T., Ding, X., and Brocard, G.: pyBadlands: A framework to simulate sediment transport, landscape dynamics and basin stratigraphic evolution through space and time, PLOS ONE, 13, 1–24, https://doi.org/10.1371/journal.pone.0195557, 2018.

Schoorl, J. M., Sonneveld, M. P. W., and Veldkamp, A.: Three-dimensional landscape process modelling: the effect of DEM resolution, Earth Surf. Proc. Land., 25, 1025–1034, 2000.

Schwanghart, W. and Kuhn, N. J.: Topotoolbox: A set of Matlab functions for topographic analysis, Environ. Modell. Software, 25, 770–781, https://doi.org/10.1016/j.envsoft.2009.12.002, 2010.

Shobe, C., Tucker, G., and Barnhart, K.: The SPACE 1.0 model: a Landlab component for 2-D calculation of sediment transport, bedrock erosion, and landscape evolution., Geoscientific Model Development, 10, 4577–4604, https://doi.org/10.5194/gmd-10-4577-2017, 2017.

Simoes, M., Braun, J., and Bonnet, S.: Continental-scale erosion and transport laws: A new approach to quantitatively investigate macroscale landscapes and associated sediment fluxes over the geological past, Geochemistry, Geophysics, Geosystems, 11, https://doi.org/10.1029/2010GC003121, 2010.

Simpson, G. and Schlunegger, F.: Topographic evolution and morphology of surfaces evolving in response to coupled fluvial and hillslope sediment transport, J. Geophys. Res., 108, https://doi.org/10.1029/2002JB002162, 2003.

Snyder, N. P., Whipple, K. X., Tucker, G. E., and Merritts, D. J.: Importance of a stochastic distribution of floods and erosion thresholds in the bedrock river incision problem, Water Resources Research, 108, https://doi.org/10.1029/2001JB001655, 2003.

Syvitski, J., Peckham, S., Hilberman, R., and T., M.: Predicting the terrestrial flux of sediment to the global ocean: a planetary perspective, Sedimentary Geology, 162, 5–24, https://doi.org/10.1016/S0037-0738(03)00232-X, 2003.

Tarboton, D. G.: A new method for the determination of flow directions and upslope areas in grid digital elevation models, Water Resour. Res., 33, 309–319, https://doi.org/10.1029/96WR03137, 1997.

Tarboton, D. G.: Utah State University, TauDEM Web page., [Available at http://hydrology.usu.edu/taudem/taudem5.], 2013.

Tesfa, T., Tarboton, D., Watson, D., Schreuders, K., Baker, M., and Wallace, R.: Extraction of hydrological proximity measures from DEMs using parallel processing, Environ. Model. Softw., 26, 1696–1709, 2011.

Thieulot, C., Steer, P., and Huismans, R. S.: Three-dimensional numerical simulations of crustal systems undergoing orogeny and subjected to surface processes, Geochemistry, Geophysics, Geosystems, 15, 4936–4957, 2014.

Tomkin, J. H., Brandon, M. T., Pazzaglia, F. J., Barbour, J. R., and Willett, S. D.: Quantitative testing of bedrock incision models for the Clearwater River, NW Washington State, J. Geophys. Res., 108, https://doi.org/10.1029/2001JB000862, 2003.

Tucker, G. E. and Bradley, D. N.: Trouble with diffusion: Reassessing hillslope erosion laws with a particle-based model, Journal of Geophysical Research: Earth Surface, 115, 2010.

Tucker, G. E. and Bras, R. L.: Hillslope processes, drainage density, and landscape morphology, Water Resources Reasearch, 34, 2751–2764, 1998.

Tucker, G. E. and Hancock, G. R.: Modelling landscape evolution, Earth Surface Processes and Landforms, 35, 28–50, 2010.

Tucker, G. E. and Slingerland, R.: Drainage basin responses to climate change., Water Resources Research, 33, 2031–2047, 1997.

Tucker, G. E. and Slingerland, R. L.: Erosional dynamics, flexural isostasy, and long-lived escarpments: A numerical modeling study, J. Geophys. Res., 99, 12 229–12 243, https://doi.org/10.1029/94JB00320, 2017.

Tucker, G. E., Lancaster, S. T., Gasparini, N. M., Bras, R. L., and Rybarczyk, S. M.: An object-oriented framework for distributed hydrologic and geomorphic modeling using triangulated irregular networks, Computers & Geosciences, 27, 959–973, 2001.

Turowski, J. M. and Hodge, R.: A probabilistic framework for the cover effect in bedrock erosion, Earth Surf. Dynam., 5, 311–330, https://doi.org/10.5194/esurf-5-311-2017, 2017.

Valla, P. G., van der Beek, P. A., and Lague, D.: Fluvial incision into bedrock: insights from morphometric analysis and numerical modeling of gorges incising glacial hanging valleys (Western Alps, France), J. Geophys. Res. - Earth, 115, https://doi.org/10.1029/2008JF001079, 2010.

van der Beek, P. and Bishop, P.: Cenozoic river profile development in the Upper Lachlan catchment (SE Australia) as a test of quantitative fluvial incision models, J. Geophys. Res., 108, https://doi.org/10.1029/2002JB002125, 2003.

Wallace, R. M., Tarboton, D. G., Watson, D. W., Schreuders, K. A. T., and Tesfa, T. K.: Parallel algorithms for processing hydrologic properties from digital terrain., in Sixth International Conference on Geographic Information Science, edited by R. Purves, and R. Weibel, Zurich, Switzerland., 2010.

Wallis, C., Wallace, R. M., Tarboton, D. G., Watson, D. W., Schreuders, K. A. T., and Tesfa, T. K.: Hydrologic terrain processing using parallel computing., in 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, edited by R. S. Anderssen, R. D. Braddock and L. T. H. Newham., pp. 2540–2545, 2009.

Wang, L. and Liu, H.: An efficient method for identifying and filling surface depressions in digital elevation models for hydrologic analysis and modelling, International Journal of Geographical Information Science, 20, 193–213, https://doi.org/10.1080/13658810500433453, 2006.

Wei, H., Zhou, G., and Fu, S.: Efficient Priority-Flood depression filling in raster digital elevation models, International Journal of Digital Earth, p. 10.1080/17538947.2018.1429503, 2018.

Willgoose, G. R., Bras, R. L., and Rodriguez-Iturbe, I.: A physically based coupled network growth and hillslope evolution model: 1 - Theory, Water Resour. Res., 27, 1671–1684, https://doi.org/10.1029/91WR00935, 1991.

Yang, R., Willett, S. D., and Goren, L.: In situ low-relief landscape formation as a result of river network disruption, Nature, pp. 526–529, https://doi.org/10.1038/nature14354, 2015.

Yıldırım, A., Watson, D., Tarboton, D., and Wallace, R.: A virtual tile approach to raster-based calculations of large digital elevation models in a shared-memory system, Computers & Geosciences, 82, 78–88, 2015.

Zhong, S., Zuber, M., Moresi, L., and Gurnis, M.: Role of temperature-dependent viscosity and surface plates in spherical shell models of mantle convection, J. Geophys. Res. Solid Earth, 105, 11 063–11 082, https://doi.org/10.1029/2000JB900003, 2000.

Zhou, G., Sun, Z., and Fu, S.: An efficient variant of the priority-flood algorithm for filling depressions in raster digital elevation models, Computers & Geosciences, 90, 87–96, https://doi.org/10.1016/j.cageo.2016.02.021, 2016.

Zhou, G., Liu, X., Fu, S., and Sun., Z.: Parallel Identification and Filling of Depressions in Raster Digital Elevation Models, International Journal of Geographical Information Science, pp. 1–18, https://doi.org/10.1080/ 13658816.2016.1262954, 2017.