Authors' Response to Reviews of

CHONK 1.0: landscape evolution framework: cellular automata meets graph theory

Boris Gailleton, Luca Malatesta, Guillaume Cordonnier, Jean Braun
copernicus GMD, https://doi.org/10.5194/egusphere-2022-1394

RC: ***Reviewers' Comment***, AR: Authors' Response, ☐ Manuscript Text

1. Reviewer #1: Kerry Callaghan

1.1. Main comment

RC: ***In this paper, the authors present a new framework for landscape evolution simulation, dubbed CHONK 1.0, that combines aspects of both graph theory and cellular automata methods. CHONK 1.0 uses D8 flow directions to construct a directed acyclic graph for the landscape, then uses this to break the landscape into depressions that are linked in a binary tree structure. Both water and sediment fluxes into and across lakes are simulated. The authors show examples of several different cases that this framework could be applied to.***

AR: We thank Reviewer #1 for their thorough and constructive review of our work. Please find below our responses to the different comments and clarifications as well as related changes.

RC: ***I am glad to see more acknowledgment of the impacts of landscape complexity – specifically, closed lake depressions – in this work. I am excited to see the range of potential applications that the authors cover here. However, there are a few issues with this iteration of the paper, including: A lack of clarity around the differences between the methods presented here and the works by Barnes (2019, 2021). This is of particular importance given the focus of GMD on model development.***

AR: We agree with reviewer #1 that the manuscript in its original form did not convey the novelty we intended to. CHONK 1.0 provides a proof of concept of a design for landscapes evolution models. One of the goal is to explore the evolution through time of depression and in particular their filling by sediments and water. The true novelty our model brings is the integration of depressions and their exact topography/outlet pattern in a "classic" graph traversal where we calculate a topological order of cells from the most upstream ones to the most downstream one acknowledging depression system and a dedicated dynamic. To do so, we use a method extremely similar (and in part inspired) from Barnes et al., 2019 to determine a depression hierarchy and then modify our Directed Acyclic Graph topology to allow a single topological sorting. This is where our method diverges from Barnes et al., 2020: the Fill-Spill-Merge algorithm iteratively (but very efficiently) traverse the depression hierarchy to fill the depression with water while we only traverse our node graph once with all the flux our cells gather from precipitation/erosion/hillslopes/ .... The finality and numerical elegance are different, and we wish to remind that we seek a proof of concept for a modeling framework rather than describing an optimised algorithm to do so.

AR: We clarified these points in the manuscript (you can find the detail in section 1.2, below). In particular, we highlighted the differences in goals between Fill-Spill-Merge and CHONK. We also acknowledge more explicitly that our method is significantly less optimised than Barnes et al., 2019 and is not a better alternative for filling depression, but we use it to update a landscape-scaled multiple flow directed acyclic graph to provide a depression-aware sense of upstream/downstream. The purpose of depression filling in this contribution is to be integrated in a highly modular structure, at the cost of dedicated optimization but for the benefit of a broad range of applications requiring explicit tracking of fluxes and their role in landscape evolution.

RC: ***More information about model input data and general use is needed.***

AR: This point has also been raised by reviewer#2, we added a table containing all the simulation parameters used for the examples.

RC: ***There is a lack of clarity in the explanations in some of the application sections.***

AR: see bellow our responses to the line-by-lines comments.

1.2. Line by line comments

RC: ***Line 89: "Cellular automata models are reduced complexity models" followed by line 94: "Cellular automata methods are not restrained to reduced complexity models". My best guess here is that the authors meant that cellular automata models are often reduced complexity, but not always; but as it currently reads these are just two conflicting statements.***

AR: That is correct, we removed the negation and clarified the sentence.

> l.113-115 "Cellular automata methods can also include more sophisticated equations and processes and have been utilised for modelling elements of landscapes evolution element like water and sediment fluxes".

RC: ***Line 99: "One limitation of a purely cellular automata model is that cells are processed at the same time at each time step, which is not compatible with quantities that are informed by spatial integration like the downstream accumulation of drainage area or sediment flux" I found this statement unclear. Do you mean that all cells in a cellular automata model are processed at the same time? Or do you mean that the processing order is always the same?***

AR: We mean that there is no particular order at which the cells are processed within a time step, as opposed to models like fastscape or ours defining a topological order to propagate non-local signals such as drainage area. We clarified the sentence.

> l.117: It is important to note the cells are processed in no particular order and cannot propagate non-local fluxes like drainage area within a single timestep.

RC: ***Line 104: "every nodes on the grid is treated as a cell" and also line 176: "Each individual node of the matrix becomes a cell, noted i" Can you define a cell and a node and/or clarify the difference between them?***

AR: A node is a discretised location of a grid in graph theory technical terms and the cell is the same in the Cellular automata lexicon. We introduce the two terms in the previous subsections *Graph-based frameworks and methods* and *Cellular automata method*. We added references to the latters and clarified the sentences.

> l.132 Every nodes (i.e. every discretised location as described in section 2.1) on the grid is then treated as a cell only processed in the downstream direction.

RC: ***Lines 143-147: "It is worth noting that some algorithms have been specifically develop to explicitly process, calculate and fill depressions with arbitrarily given amount of water (e.g. L. Callaghan and D. Wickert,***

2019; Barnes et al., 2019, 2021). However, these methods are only designed to fill pits with water and would require significant amount of modifications to be utilised as cellular automata processor, or even to any other purpose than what they are designed for." I will note that I am an author on the papers cited here. I think there may be some misrepresentation or misunderstanding of these works.

AR: As stated in our main response, these are most likely link to the fact we developed that part of the model roughly at the time of publication of your work, we apology for the misrepresentations. Hopefully our many changes clarified both the differences in aims and in the technical details while acknowledging your work correctly. We rephrased the above paragraph:

> "Among these, the closest to our aim are the developments of y L. Callaghan and D. Wickert (2019); Barnes et al. (2019, 2021). They designed an efficient method to (i) identify, (ii) hierarchise and (iii) fill the depression with a particular focus on numerical efficiency. While we partially built our numerical method to manage lake on these works, there are a couple of differences, most of them related to our need to integrate the lake solver into a preexisting multiple flow graph of node connectivity. More detailed differences are outlined in the method section."

RC: ***Callaghan and Wickert (2019) is basically a reduced complexity cellular automata method already (see section 5.2 in that paper), though it is true that it is only set up to move water into pits in that paper.***

AR: We added this point to the manuscript. It is worth mentioning that Callaghan and Wickert (2019) is an iterative solver, as far as we understand, while we aim to only apply surface process on each cell once per time step in order to being able to track water and sediment fluxes.

> l.169 It is worth noting that the model developed by L. Callaghan and D. Wickert (2019) is a cellular automaton.

RC: ***Barnes et al (2019) describes a graph structure that can be used to link depressions in a landscape to one another (essentially, the same or very similar structure as used in this work). While the focus of subsequent work has been on filling these depressions with water, the actual data structure has other potential applications, including topographic modification and statistical analysis of landscapes. Barnes et al (2021) focuses on using Barnes et al (2019) to fill pits with water. However, it incorporates an option that would activate a cellular automata method: the option to include infiltration of water during downslope flow when moving water to fill depressions. This functionality already exists and, in contrast to the statement made in this work, it would not be that big of a leap to incorporate additional complexity into the cellular automaton portion.***

AR: We significantly reworked the section describing our "explicit" lake solver (renamed depression-aware simulations) and how it differs from both Barnes et al (2019) and Barnes et al (2021). We almost entirely rewrote section 3.3.2 (l.275 of the new manuscript).

We hope it reflects better these works and how our method resolving lake is "just" a heavily modified version to fit our prototype's very specific needs. While it would be entirely possible to base our whole graph-building on these methods and probably fit cellular automata methods in it, there are a couple of significant differences that made us opt for a full topographic graph method. The foremost goal of our prototype is to provide a flux-oriented cellular automata data-structure allowing, once the graph is constructed and the depression hierarchised, to process each cells only once from the most upstream to the most downstream. This point is really important because the work of Barnes et al. (2021), although significantly more computationally efficient, remains an iterative process where water is first moved downhill and then recursively navigate through the forest of depression tree to efficiently flood the landscape. It would break our goal of processing everything-at-once-and-move-on-the-next-cell (required to keep process interdependent or tracking capabilities for example). The second important difference is that Barnes et al (2019) and Barnes et al (2021) are designed for single flow direction, while we aim to be generically compatible with multiple flow algorithms. The difference are conceptually very small, but in the detail it opens a wide range of numerical rabbit hole (imagine for example a depression system directly connected at its outlet to two distinct other ones) which explains why we opted for designing our own version. Finally, we aim to provide the cellular automata structure on a topographic graph with different types of depression management (e.g. carving).

RC: ***Line 205: "Finally an algorithm takes advantage of the modified DAG to calculate the depression-aware topological order." Did you mean finally 'the' algorithm?***

AR: We rephrased the confusing sentence: l.245 "Finally we apply a topological sorting algorithm on the modified DAG to calculate the depression-aware topological order." (The said algorithm is detailed in the following paragraph).

RC: ***Line 248: "For example if the current cell is already labelled with another depression index in which case both are labelled as twins and this cell represent their tipping node" and Line 250: "If one of the checked neighbours has a lower elevation, the cell is labeled as outlet and this depression will be the top one if not later labeled as twin." I find these explanations to be unclear.***

AR: We rephrased the sentences: l.310 "If the neighbour is higher in elevation, it is placed in the queue for later processing or labelling. This process runs until the depression is complete. This happens if the current cell is already labelled as belonging to another depression in which case both are registered twins. Each twin records the given connecting cell as their tipping node (e.g. depression 2 and 3 on figure a). If one of the checked neighbours has a lower elevation, the cell is labeled as outlet and this depression is the top one of its tree - or at least as long as not labelled as twin by another priority queue."

RC: ***Line 258: "the minimum volume of a depression (0 if base depression)," How does this differ from the volume of the depression listed on the line above?***

AR: We rephrased both points. The volume total of a depression is its volume if filled to the top. The minimum volume represents the minimum volume of water required to "enter" a given depression. If the given depression is a base depression it is 0, but it start making sense in the presence of an intermediate/top depression where this volume represents the minimum volume required to fill all of its children.

> l.320: the minimum volume of a depression (0 if base depression, the minimum volume to fill all the children and "reach" the depression in the tree),

RC: ***Lines 255-263: "Our implementation only slightly differs from Barnes et al. (2019). The original algorithm is tuned for computing speed, we compute more information about each depression useful for the model: – the volume of the depression Vtotal, note that it includes the volume of their children if any, – the minimum volume of a depression (0 if base depression), – a depression level, which represents the maximum distance in the tree from a base depression. Each base depression is at level 0, and each parent's level is equal to the maximum level of their children plus 1, – the tipping node of the depression, which represents either the outlet of the whole subsystem, or the node joining two twins – the maximum elevation of the depression if filled." More clarity is needed around the difference between this implementation and Barnes et al (2019) including why and how the differences are important for this work and why a new framework was needed to achieve this. Note that Barnes et al (2019) does also compute depression volume, tipping node, maximum elevation, and other related information about the depressions. This structure is a major part of***

the work presented and so it is important to clarify this and to focus on the aspects of the work that are new.

AR: We significantly reworked the whole section 3.3.2 (l.275 of the new manuscript) and hopefully clarified all these points.

RC: ***Line 266: "based on matrices Evaporation rates" Here a possible input matrix, evaporation, is mentioned. It would be helpful if there were information regarding what inputs are required, or possible, to use CHONK 1.0.***

AR: We added a table detailing the different inputs used for the different scenarios (Table 1).

RC: ***Line 290: "Some of these properties, like erosion or water for example, are reinitialised at each time step while others like topography 290 or sediment thickness gets updated at each time step" It is not clear what the difference is between a reinitialisation and an update.***

AR: We removed that sentence, as it was both confusing and irrelevant (drainage-area based water fluxes are by definition recalculated at each time steps while sediment thickness kept in memory).

RC: ***Lines 292-297: "Three kinds of parameter inputs are currently available. First, external parameters which can be single values (e.g. dx, dy, dt), global arrays (e.g. 2D matrices of precipitation or uplift), spatially varying or even varying through time. Second, parameters that are label-dependent: a 2D matrix of labels defines discrete spatial areas and each label has a set of distinct parameters, for example different rock-type can be associated with different erodibility and diffusivity (Gailleton, 2021). And third parameters that are fully dynamic: they are interdependent of each other and define by a function rather than a given value. Example of the latter are detailed in section 4.4." I found it difficult to imagine which types of parameters might match up with which of these three classes listed. This is another place where information about the required and/or possible inputs would be helpful.***

AR: The newly added table 1 hopefully clarifies that point.

RC: ***Lines 325-346 discuss the method for filling lakes with water and computing the water elevation. Although I can see that the methods are not identical to those used in Barnes et al (2021), they are similar and the aim – to compute lake water levels – seems to be the same. Can you clarify how this differs from Barnes et al (2021) and why this was necessary to achieve the aims of the rest of the paper?***

AR: Hopefully the reworking of section 3.3.2 clarified this point.

RC: ***Line 476-478: "It quickly becomes disconnected from the foreland as the blue linessets as described below must be submitted to specialized repositories. Please consider providing at least preliminary links to such assets for the period of minimum elevation shows (fig. 5C)." I'm struggling to make sense of these sentences and wondering if they were in an incomplete state or a copy/paste gone wrong?***

AR: We are struggling too and reviewer #1 is probably right about a copy-pasting issue, we removed the bizarre part of that sentence.

RC: ***Page 18-23: These examples are great, but I did find myself wanting some more information. For example, what were the time steps used? What method was used to simulate the tectonic change? Are there more figures that might be informative? (for example, in application 1, is a graph of water flux out of the lake informative? Can a viewer learn something from seeing the result at more than one point in time?)***

AR: We added Table 1 with more details about the parameters. The simulations we ran use a very simple 2D variable uplift field, constant through time, approximating the surface deformation field across a normal fault with a set of exponential segments of flipped direction with 2/3 of the throw going into subsidence of the hanging wall, and 1/3 into uplift of the footwall. We also generated animations for application 1 and added them to the supplemental material of the paper. Note that it would not add much to the story for the other applications so we did not add them.

RC: ***Within this, pages 21-23: I found the explanations of these applications to be significantly less clear. I found section 4.3 particularly confusing, both in the text describing this and in Figure 7 which I found difficult to interpret. More explanation of the colours used and description of what is being viewed in each panel would go a long way.***

AR: We significantly detailed the caption of Figure 7, and to lesser extents the text, hopefully clarifying the section and the example.

RC: ***Line 545: "With hard tool enhancing incision, the area downstream of the harder 545 rocks lowers its base level which will propagate knickpoints up tributaries, regardless of their lithology" - (and the rest of this section) - How do you deal with mixed sediments, i.e. in the case where water moves over hard, then soft, then hard again? The sediments should be a mix of hard and soft. Are they assumed to mix evenly?***

AR: We indeed keep track of the proportion of sediments coming from both rock types. Numerically speaking, we use a sparse matrix to store that information the same way we do for tracking provenance. The K_{sed} is calculated from a weighted average of all the K from the different lithologies. We modified the text to reflect the latter and point to the relevent method section: l.590 "K_{sed} is a weighted average proportional to the content of each lithologies in the model. We store the proportion of each lithologies as detailed in section 3.4.1."

2. Reviewer #2: Sébastien Carretier

2.1. General comments

RC: ***This manuscript presents a new landform evolution model, CHONK, with three major advances, that of filling lakes dynamically, of taking into account the tool effect in the abrasion of river bedrock as a function of the lithology of the transported sediments and that of being able to trace several properties of the sources of eroded rocks upstream. This opens up many possibilities for source-to-sink studies. After several readings, however, I found it difficult to understand why this new model was a different philosophical approach to other LEMs. I have the impression that it is more a matter of assembling relevant elements from previous models to allow for sediment and water tracking rather than a real change in philosophy (cellular automata + graph theory). I think that part of my misunderstanding comes from a language that uses implicit formulas and shortcuts that, for me, do not simplify the discourse but rather make it more obscure. I provide an annotated pdf pointing to all these elements of language that I did not fully understand. I quote one here as an example to illustrate my misunderstanding (line 320): "When the implicit lake solver is activated, lakes are not processed differently than the rest of the landscape, but cells in- claulated areas affected by flow rerouting have reduced topographic gradient and less direct connection to the rest of the landscape, effectively simulating a "passive" landscape." What is "flow rerouting"? Why is the topographic gradient reduced? What is a passive landscape?***

RC: ***To justify this new model, the authors point out that the previous models deal separately with water discharge, lake resolution, sediment transport, etc. But here I do not see where the fundamental difference lies. LEMs that solve the topographic evolution explicitly (in terms of numerical scheme) start by propagating the water discharge, possibly by filling the lakes, then once the water flow field is known, these LEMs compute the erosion-sedimentation balance on each cell, in cascade (See review in Tucker and Hancock,***

2010, ESPL). The order in which the grid cells are processed can vary but is always from upstream to downstream, possibly distinguishing between catchments (as in Fastscape). The fact of classifying the cells in order of decreasing altitude, which was already present in the first models such as GOLEM or DRAINAL, determines a graph. The calculation of the water and sediment balance in these models follows the logic of cellular automata. There is non-locality in some of these models, associated with a transport length, so that these LEMs have long since combined graphs, cellular automata and the consideration of the upstream-downstream connection for water and sediment flows. Moreover, the only LEM to my knowledge that can really solves both water and sediment balances at the same time is EROS (now River.lab) with its purely Lagrangian approach using "precipitons". In short, while acknowledging the significant advances of this new model regarding lakes, the tool effect and source tracing (see also the Badlands model in Petit et al., Esurf 2023), I think the manuscript would benefit from a better explanation of the fundamental difference between the approach implemented in CHONK and other LEMs. This remark may seem unfair because a large introductory part of the manuscript is dedicated to justifying this difference. I suggest to give more precise examples in this part and to improve the discussion to explain why the numerical scheme of the existing models could not allow to perform the simulations presented in this manuscript. The necessary corrections are therefore essentially a rewriting of certain paragraphs.

RC: ***As this is a manuscript submitted in a journal that presents the algorithm of the models rather than their applications around a scientific question, I think a pseudo code of this model should be presented, to first order, with at least the order of operations. At present, if I were to recode CHONK, I am not sure that I have the main elements to do so in what is presented in this manuscript.***

RC: ***The simulations presented as illustrations are quite relevant but no parameter values are given. A table should be added. It would also be interesting to have some indication of the calculation time of these simulations.***

RC: ***I am available to interact with authors to give more details if needed.***

RC: ***See my specific comments in the annotated manuscript.***

RC: ***Good luck with the revision.***

RC: ***Sebastien Carretier***

AR: We thank Reviewer #2 for thoroughly reviewing our manuscript and highlighting the need of significant clarifications in our model description and especially on the main “modelling design” message we want to convey. Individually, the methods we use in our model prototype are not ground-breaking new ways to numerically model surface processing. We did not intend to sound like we were reinventing the wheel. Fundamentally, we assemble and modify existing algorithms of graph theory and cellular automata to fit specific LEMs purposes. However, and this is the main point of this manuscript, the goal is to define a modelling framework designed to tackle LEMs research questions involving interconnections of fluxes in complex landscapes in a **generic** way. In other words, a general methods to (i) combine different process laws affecting potentially the same fluxes (e.g. hillslopes/landslides/fluvial processes affecting sediment transport through landscapes), (ii) independently from the landscapes complexity (i.e. lakes) and (iii) if needed, interactions between the different processes. All of these without having to assume any specific process law or equation.

AR: The graph part takes care of the connectivity between nodes/cells, in the end providing a landscape-wise upstream/dowstream directionality including any nested potentially-endoreic system of depressions with their original topography if needed. The fundamental difference of CHONK’s way is that it is separated from any process laws. As reviewer #2 mentioned, most LEMs indeed define a *de facto* graph by sorting nodes by elevations (after eventually filling lakes with water a way or another) and transmit fluxes to downstream neighbours. But (i) this is not an actual use of graph theory algorithms or an explicit use of them and (ii) the local-minima and depressions are often only partially integrated (i.e. flow rerouted by filling/carving, systematically filled, or preprocessed before other surface processes hence not dynamically). There is nothing wrong with that, but it does not fit our goal to make the processing of lake generic and separate from whatever would fill it. A necessity if one wants to eventually e.g. simulate the drop in sediment flux at a lake outlet while water discharge is maintained. In contrast, we use modified and combined graph theory algorithms from Cordonnier et al., 2019 to reroute flow in local minima; Barnes et al., (2021) to characterise nested lake systems; and we compute topography-agnostic topological sorting in single and multiple flow directions using Braun and Willett (2013) and Anand (2020). The novelty in our method lies in the unique combinations of all of them, which require a lot of adaptations. As we intend to keep our method independent from process laws, we cannot systematically pre-fill our landscape with water and then process topological ordering: for example, one may need to only outflow a lake if enough sediment and/or water enters it in the given timestep. The topology within the depression depends therefore on everything that had happened in all the cells upstream of a depression. While more complex and probably not currently implemented in its most optimal form, our design ensures that the nodes and depressions can be characterised without any knowledge of what processes will affect them.

AR: The cellular-automata (modified from its *sensu stricto*) part of the model handles actual physical fluxes and surface processes. It ensures that all inter-connected processes are applied simultaneously and can communicate between each other if needed (e.g. tracking the full rock-type of the sediment flux and using this information to inform erosivity). The latter is not that common in LEMs which tend to calculate processes in series - i.e. Lanscapes-wise (not all of them as CIDRE for example process first the water fluxes but then the hillslope and fluvial processes on a node to node basis if I am not mistaken from the source code). Also, CHONK's design philosophy separates fluxes and processes, encouraging easier and generic interactions between them. This is illustrated by our basic tool effect which uses information about sediment provenance to affect fluvial incision. But again the goal is to provide the basis for a generic framework, where adding a new process affecting existing fluxes (e.g. an episodic landslide law) could be done flawlessly without hampering the other features (like tracking).

AR: All of this is not an easy task and requires a lot of work, experimentations and numerical design. This manuscript presents the overall idea in the form of a proof of concept with a working prototype. As stated in the manuscript, we suggest this is a first step toward an actual modelling framework but this requires a lot more work (for a more mature example see the evolution from CHILD to LANDLAB 1 and 2 and all the papers associated with their different parts). Reviewer #2 mentioned EROS/River.lab as being one of the only LEMs to incorporate sediment and water at the same time (we may suggest to add CAESAR-lisflood to the list). Theoretically, our modelling design idea (not the prototype) is applicable to any surface processes, no matter the timescale or the nature of the fluxes and could be used with similar laws than EROS (we are actually working on a graph-based version and presenting it at the 2023 EGU).

AR: Reviewer #1 and #2 comments helped us identify different sources of confusion in our initial manuscript. We had failed to clearly present our aim in the first version of this manuscript and we got lost in between the description of the prototype, technical details and of the design. We accordingly restructured and reworked a number of points:

- We significantly reworked the section introducing the general concepts of graph in LEMs, cellular automata and more importantly of the hybrid solution we present in the manuscript. Hopefully that clarifies (i) the goals of this prototype, (ii) the benefits of its specific design and (iii) that it is building on existing method to provide new possibilities.

- We clarified our terminology regarding local-minima/lakes. Our former "explicit vs implicit" solvers are respectively replaced by "overflowing vs rerouting" lake management in order not to collapse with the numerical methods' for PDEs terminology.
- We reworked the Discussion and conclusion section to reflect our new introductory sections.
- We added a Table with the parameters used in the different simulations.
- we added a subsection presenting a general numerical implementation

2.2. Line-by-line comments from the annotated PDF

RC: ***l.9 what does it mean?***

AR: We mean that independently from the process laws implemented in the model, computing all the processes in a cell ensures access to the definitive stage of any modelled flux at a cell level. For example, this allows us to not only track provenance of the fluxes (which can be done in post-processing) but also to use the tracking information to inform the process laws in the downstream cells (e.g. our "tool" effect). We rephrased into:

> l.8 "We demonstrate how the former ensures interoperability of the different fluxes (e.g. water, fluvial sediments, hillslope sediments) independently from the process-law implemented in the model".

RC: ***l.25 I do not understand. These LEMs do not ignore sediment transport at all.***

AR: By "ignored", we mean detachment-limited models like the stream power-law in its most basic form, which ignores sediment transport by implicitly considering its effect on bedrock incision. "largely" may be an overstatement but refers to the numerical manipulations used in the implicit formulations of Hergarten (2020) and Yuan et al. (2019). In the end, both compute sediment fluxes indirectly by multiple iterations through topographic changes (the three "sweeps" for Hergarten (2020) and the Gauss Siedle scheme from Yuan et al. (2019)). Sediment fluxes are not explicitly considered and integrating provenance, chronometry, or interaction with other processes is not possible. We rephrased:

> l.24 "For example, the evolution of surface topography over millions of years can be efficiently explored with erosion laws that only indirectly consider sediment transport in their numerical scheme (e.g., Yuan et al. 2019, Hergarten 2020) or even completely ignore them (Braun et al., 2013) to the benefit of numerical performances."

AR: l.26 typo corrected.

RC: ***l.34 this is not true that they are done independently. Erosion and sedimentation depend on the water discharge field in all the LEMs, some of them solve lakes to establish this field at each time step.***

AR: This is right and I believe we meant that numerically speaking, the different fluxes are processed serially. In long-term LEMs, fluvial erosion and sedimentation are systematically calculated after the entirety of water fluxes (where lakes are therefore considered as sediment pools or flat rivers) limiting the interactions between fluxes. This contrasts with the modelling design we suggest in this contribution where all the fluxes (or at least all the ones that need to inter-operate) should be solved at once for each cells in order to enable interoperability. We replaced the word "independently" with "serially".

RC: ***l.83 not true that it is not possible actually. Solving lakes dynamically is possible during a time step as done in CIDRE does that (although not published. . .)***

AR: This is correct, we were not clear (and we are not claiming to be the first interested by lakes in LEMs). Many LEMs do manage lake/local minima. However they usually do it in multiple steps where all the lakes of the grid are filled with water and then used as a potential sediment trap. However, for more sophisticated applications, all the fluxes need to be processed at once, whether the domain is a lake or not: e.g. if the outflow of a lake is a function of how much water and sediments it receives while applying a different process equations in the lake domain (for sediment deposition and evaporation). We had not clearly illustrated that dynamics with lakes, so we edited the text to clarify:

> l.100 e.g. calculating fluvial incision function of the nature of its upstream sediment input mixing all the processes and what could have been stored in potential lakes.

RC: ***l.84 feedbacks of what?***

AR: feedbacks between independently calculated processes (e.g. sediment fluxes from hillslopes and from fluvial processes). We rephrased (Note that this sentence has also moved following the paper restructuring).

> l.59 the exploration of dynamic feedback within the different fluxes transported from a cell to another

RC: ***l.85 - 87 (3 comments) unclear***

AR: We reworked most of the section 2.1 and hopefully clarified it.

RC: ***l.103 Above I only see one method, the cellular automata one. The graph method has not been presented yet***

AR: We reworked section 2.1, 2.2 and 2.3 to better reflect the definition of Graph-based vs Cellular-automata method and outline CHONK's one.

RC: ***l.121 Here I still do not understand the difference in philosophie with previous LEMs***

AR: We significantly reworked the whole section 2.3.

RC: ***l.132 Not entirely true. CASESAR has a specific algorithm but aiming at calculating the field of accumulated water discharge in order to solve the sediment transport field by taking into account the downstream propagation of sediment.***

AR: Reviewer #2 is right, we clarify our point as we meant it does not propagate sediment through the whole landscape **in a single timestep** (see l.154) like "long-term" LEMs.

RC: ***l.151 After a first reading I think I understand what you mean but this was not the case at the first reading.***

AR: We rewrote and detailed most of section 2. to clarify all of the points raised so far.

RC: ***l.156 So this is already done in some of the LEMs...***

AR: We clarified this point in our reworking of previous section. Filling lakes with sediments and treating them separately with evaporation has indeed been done with TISC (and to lesser extent other models). However, this has been done in a "static" way where the numerical implementation makes it extremely difficult to modify or test. Simple things like changing water routing (e.g. D infinity, multiple flow, ...) would require basically a re-coding of the whole model — where we aim, at terms, to make the incorporation of different inter-connected domains and processes more generic and replaceable with CHONK's method.

RC: ***l.160 Are you refering to CHONK or TISC here ?***

AR: Referring to TISC, we clarified.

> l.183 TISC's implementation however is not compatible with our design as water fluxes are calculated separately from the rest of the processes

RC: ***l.163 but this is actually done in a specific order (Fig. 3), as in many LEMs where elevations are sorted and treated in a decreasing order…***

AR: As now more detailed in section 2 and illustrated in Fig. 4 of the original manuscript, the topological order we calculate (can) incorporate the depressions' exact topography. The relationship between absolute elevation and node ordering is therefore broken and we use dome graph manipulation to define a new one. Depressions' geometry is preprocessed in the graph building phase without filling them. It is only when all the cells entering a lake are processed that we actually fill (or not) the lake with sediments and water and transfer the remaining fluxes downstream.

RC: ***l.174 what is it ?***

AR: We mean that we provide a sense of upstream/downstream that takes into account the exact topography of lakes. We rephrased.

> l.215 The first step consists in building a graph of connectivity on the landscape in order to determine a processing order for the cells that takes into account the topography of endorheic basins.

RC: ***l.185 format citation***

AR: done

RC: ***l.192 give an example***

AR: We reworked most of this paragraph now starting section 3.3.2 l.276.

RC: ***l.196 As in all LEMs with an explicit numerical schema (e.g. CIDRE). I miss something.***

AR: The sentence was not claiming this is a novel way to do it, but also does not really add to the point so we removed it from the manuscript.

RC: ***l.198 What is it ?***

AR: We rephrased most of that paragraph now starting section 3.3.2 l.276.

RC: ***l.200 Do you mean lakes ?***

AR: We replaced by "lake and depression systems". A local minima and associated area can be a lake, an endorheic basin, a (numerically) flat surface or simply few pixels of noise.

RC: ***l.207 what are these scenarios ? Change these terms, at the first reading I thought they refered to the numerical scheme. I understand now that it refered to respecting the water volume budget or nor.***

AR: We understand the confusion and adapted the terminology in the whole manuscript, as stated in a previous response.

RC: ***l.215 Do you mean multiple flow ?***

AR: Braun and Willett (2013) present a single flow stack order, and we present a multiple flow equivalent.

RC: ***l.221 Do you mean that Cordonnier et al. is based on a steepest descent hypothesis ?***

AR: Indeed! we clarified the sentence.

RC: ***l.222 Steepest or Multiple Flow ?***

AR: Steepest, we clarified:

> l.262 We only modify it to be compatible with multiple flow directions.

RC: ***l.226 How ?***

AR: as stated in the text, we use Algorithm 3 from Cordonnier et al. (2018). We added a sentence summarising it:

> l.268 (see algorithm 3 in Cordonnier et al. (2018) which inverts the node-to-node steepest descent connections from the sill to the pit).

RC: ***l.305 No sure I understand that***

AR: The (semi)-implicit numerical schemes mentioned there requires multiple graph traversal: at least one in the downstream direction to calculate drainage area and one in the upstream direction to calculate the erosion field. Adding sediments (e.g. Hergarten (2020) or Yuan et al., 2019) add even more iterations - meaning that integrating upstream information to affect processes is not directly possible. We clarified:

> l.362 For example, solving stream power-like equations implicitly necessitate multiple graph traversals in the upstream and downstream directions therefore limiting the amount of upstream information a cell can use in the processes (e.g. Braun and Willett, 2013; Campforts et al., 2017; Hergarten, 2020).

RC: ***l.306 which model ?***

AR: We clarified:

> l.363: or example, solving stream power-like equations implicitly necessitate multiple graph traversals in the upstream and downstream directions therefore limiting the amount of upstream information a cell can use in the processes (e.g. Braun and Willett, 2013; Campforts et al., 2017; Hergarten, 2020). However they are not fully incompatible: one could imagine calculating a "static" erosion field with one of these implicit scheme and post process them using the cellular automata method to integrate upstream information (e.g. provenance), only sacrificing the dynamic adjustment capabilities.

RC: ***l.308 What does it mean exactly?***

AR: We removed the confusing sentence.

RC: ***l.320 I do not understand.***

AR: We rephrased. Our "implicit lake solver" reroutes flow from the pits to the rest of the landscape, meaning some of the slope will be negative and we recast them to a numerically insignificant value where erosion is nearly null and deposition enhanced - what we called a "passive" landscapes.

> l.377 When the solver for passive lakes is activated, cells in the area affected by the local minima are processed like any other cells. However, flow is rerouted from the pit cell to the lake outlet and this effectively reduces topographic gradient, enhances deposition processes, and reduces erosion processes.

RC: ***l.324 What is it different from others LEMs solving lakes during a time step ?***

AR: We rephrased the whole paragraph (starting l. 380) to clarify the latter. The difference is that most LEMs first fill the lakes with water and then then either process the lake as unconditional trap for sediments or reduced-gradient areas. In our case we separate totally the preprocessing of the depression from its filling by actual fluxes and processes. When computing the graph, we conserve the original topography within the lake and "only" precompute (i) what volume of material it can receive and (ii) in what order lakes should be filled. Its only when the pit cells for an entire depression system (including nested depressions) that we actually fill the lake with water/sediments and transmit the excess to the downstream cells (or not).

RC: ***l.450 Give the parameter values of these simulation, cell size etc. A table would help.***

AR: We added a parameter table (Table 1) with all the values used for the simulations.

RC: ***l.461 quels sont les dimensions, paramètres de ces simulations ?***

AR: Nous les avons ajoutés dans le tableau mentionné ci-dessus.

RC: ***l.478 ?***

AR: Probably a copy-pasting issue, we removed the sentence that appeared to be from EGU general assembly website?

RC: ***l.537 that's really new***

AR: Thanks! We believe a lot of processes could benefit from more dynamic integration. For example, we considered testing dynamic transfer of sediment flux from hillslopes to fluvial within a single timestep. The current numerical implementation is rather experimental due to all the trial-and-errors and makes the addition of new features tedious, so we preferred to switch our effort toward demonstrating the benefits of the overall method (this publication) and toward the development of the next iteration of this model with an emphasis on numerical performances and flexibility (in prep.).

RC: ***l.542 what are the parameters values of Equation 14 in this simulation ? (s, Ksed etc...)***

AR: See the new Table 1.

RC: ***l.553 You have to better convince why previous models are not cellular-automata (I think they are). See Tucker and Hancock (2010).***

AR: We hopefully clarified in our main response that we suggest here that the coupling of Cellular-Automata and graph theory, following the design of separating landscapes' topology from the processes/fluxes capitalised within cellular automaton. The novelty is not in the cellular automaton itself, but more its integration with the graphs. We clarified the discussion and conclusions in order to reflect that point.