Dear Stefan,

Thank you for your constructive replies. As a foreword to these comments, I wish to emphasize that my remarks are essentially minor compared to the overall scope of the paper, and I am not requesting significant modifications. If my response is somewhat lengthy, it reflects my interest in the discussion rather than any disapproval.

Below are a few additional details in response to your comments. I look forward to seeing the revised version.

We would, of course, have written it in a different way as a research paper, and it would not be a huge problem to extend the introduction. In turn, I still feel that there should be a clear difference between research papers and short communications (or technical notes in other journals). And to be honest, I am not good in telling stories. The idea behind this paper was never to motivate anyone to start research on coupling landform evolution models with geodynamic convection models since some (or several) groups have started working on this topic intensely. They will explain quite well in their (upcoming) papers why considering coupled models is important. We just want to leave a message for them if they ever encounter problems with rivers in their models.

To be fair, I find the technical motivations adequate in their current form (arguing otherwise would be somewhat hypocritical given my own publications). My comment was intended for the broader readership; readers of ESurf may be less acquainted with the numerical details and would benefit from broader contextualization, which could be achieved without significantly expanding the text. But ultimately, this is more of an editorial decision than a reviewer's.

You understood it correctly, which means that the paper somehow conveys its message. However, it is neither a matter of timescale nor related to the explicit treatment of the flow directions or to the "non-diffusive" D8 scheme. Changes in flow direction of major rivers are rare under uplift. So a simple fixed-point iteration would even converge and make the flow directions formally implicit. And a multiple flow direction scheme would not make the rivers climb up the walls of the valleys just because the valley is slightly asymmetric. This would only have a notable effect in flat areas and potentially on hillslopes. The problem is inherent to the type of model, so on mapping rivers as linear objects on a discrete grid. It is the second fundamental weakness of this type of model, behind the still unsolved and serious scaling problem (dependence of the result on the spatial resolution) when coupling fluvial erosion with diffusion. The Eulerian treatment cannot keep track of gradual horizontal shift; erosion is erasing all information on this rapidly. However, I am not entirely sure whether I got your point fully. Are you just not fully convinced that the problem exists or would you suggest a way to fix it? In principle, you could disregard horizontal movement until it accumulates to one unit of grid spacing, then adjust the elevation instantaneously, recompute the flow directions and finally erode. I did not test this because it looks a bit weird compared to a Lagrangian scheme and I guess that it would end in a mess.

Small-scale tectonics is complicated and represented in landform evolution models only in very specific ways to my knowledge. So let us better talk about large-scale geodynamics. I would request that a coupled geodynamic-landform evolution model must at least be able to translate and to rotate an island or a continent without destroying the topography.

I think that once having an efficient model in which valley floors and rivers consist of multiple grid points (and perhaps includes lateral erosion) would solve several problems (see above), but it would have to be shown that they are really compatible with Eulerian schemes. Anyway, we cannot tell those who are actually working on coupled models to stop and wait for such a model.

I have grouped my comments to provide a consolidated response. To be clear, I am convinced the numerical problem exists, and that the presented approach offers a valuable direction toward improving the treatment of horizontal tectonics in coupled LEM-geodynamic models. I also believe, however, that the problem may be broader in scope than currently framed. You addressed most of my points correctly; what follows is a rephrasing and synthesis in light of your responses.

On the technical side first. I was wondering whether, in a theoretical setting where flow routing would be perfectly implicit (tiny time steps, MFD, etc.), an Eulerian treatment of fluvial erosion could better follow horizontal advection by gradually shifting drainage area to neighbouring pixels — effectively relocating erosion in the direction of the motion and producing results closer to the Lagrangian scheme. This is what I meant by timescale, and my question is more precisely: *is the weak coupling between fluvial processes and uplift not part of the problem as well?* This does not diminish the advantage of the Lagrangian approach, which clearly improves things on the numerical side — but the physical framework may carry an inherent limitation that is worth acknowledging.

On a broader geomorphological point. Beyond the numerical issue, I was suggesting this could also reflect a fundamental weakness of Stream Power Law-like models, which focus on purely vertical erosion and deposition and enforce static river locations by design. This connects to my other main point about missing physical processes — lateral erosion and deposition, river mobility within an actual valley, short-term stochastic events, and multiple flow directions — all of which could allow rivers to better respond to horizontal motions. I am not expecting these processes to be added in a revised version, as they are clearly beyond the scope of this paper, but they could warrant a mention in the contextualization or as opening points in the conclusion, if the short communication format allows for it.

Neither of these points undermines the merit of developing this method, nor do they require major additions to the study — which is precisely why I recommended only minor revisions. They rather raise the broader question of whether SPL-like models are entirely adequate in settings with significant horizontal processes, tectonic or otherwise, as you also noted for hillslope diffusion.

I am also not entirely sure about the direction of your idea. To my experience, you can destroy your results easily be defining bad boundary conditions, but not fix inherent problems by defining specific boundary conditions. Here the northern and southern boundaries are defined by zero elevation, while outflow is allowed along the entire edge (as usual). Do you think about allowing outflow only at a few points and letting this points "jump" by one grid spacing after a certain time? This will generate dams and the model will fight against this by generating boundary-parallel channel segments. Without having tested it numerically, I am sure that this will not solve the problem.

You answered my question. I was concerned the boundary conditions where fixed outlet points or moving with the horizontal advection.

How do you see that it is not realistic? And if so, I would not be sure that a model with a better process representation would be better in such a situation.

The rivers appear severely stretched, which was the core of my concern. However, having never worked extensively in settings with significant horizontal tectonics, I note that Goren et al. (2015, *Geology*) show a comparable example that resembles your results — so this may well be physically reasonable. I will therefore rephrase my question: is the model shown in Figure 9 at steady-state, or is it an arbitrary snapshot? If it is transient, what does steady-state the look like ? If it is not, what happens as the simulation continues — does the network keep stretching indefinitely, or does it converge toward a stable configuration? (these questions make more sense for the remeshed version probably).

Each point has a variable x- and y-coordinate and distances and cell sizes (and cell edges for diffusion in oceans, but not ret released) are computed accordingly (introduced in OpenLEM version 45). I guess that this is "operating on the same irregular grid" in your comment. However, I am not entirely sure about the alternatives you mention. Projecting back in the sense of interpolation would not bring any progress compared to the Eulerian approach. The other option would be not letting the LEM grid know that it has been deformed. In the shear example (Fig. 9), this would be the version "Lagrangian with frozen flow pattern" as there would be not motivation to change any flow direction. Not really good, but clearly better than the Eulerian approach.

OK that makes sense, I guess I was confused with the remeshing. To make sure I understand, the Lagrangian approach is fully Lagragian and the remeshing does not involve interpolating back to Eulerian?

Finally, the reference list is composed of >30% of self-citation, on topics that have been researched by others as well. The manuscript would benefit from a more diverse bibliography with independent sources to strengthen the literature review.

I personally find 30% somewhat high for a scientific study, though I acknowledge this may reflect a personal opinion or an early-career bias on my part rather than an objective standard — this is ultimately an editorial decision more than a reviewer's. I am not fully up to date with the literature on horizontal deformation and landscape evolution, but here are a few suggestions: Goren et al. 2015 (doi:10.1130/G36841.1), Castelltort et al., 2012 (doi:10.1038/ngeo1582).

Researchers with a very deep background in numerics (not very likely in geomorphology) would probably recognize from the the words in my 2001 PRL paper that there cannot be a serious difference between the two approaches. Otherwise, we would need my 2002 book "Self-organized Criticality in Earth Systems" (one more unwanted self-citation) to recognize that:

First, I want to emphasize that this point, which I raised, is genuinely minor and was mainly about phrasing. It bears no impact on the overall quality of this contribution or on the review outcome — though it has sparked an interesting discussion, so I will carry on. As I mentioned, and as a numerical geomorphologist, I recognize that the schemes are very similar, and your 2001 paper deserves more credit for providing the first implicit solution to the stream power law — something I was not aware of before this review.

I will try to be as objective as possible and base my assessment solely on what has been published. That said, I should disclose that I have worked with Jean Braun and Fastscape, albeit some years ago. With that in mind, I think it is important to distinguish two aspects: the implicit scheme itself, and its numerical implementation.

(1) My nonlinear version was implemented only for n = 2 by solving a quadratic equation. So restricted to n = 2, but in turn more efficient than the Newton scheme used by Braun and Willett.

Yes, actually marginally slower than the n=1 case (no NR and no pow) but faster in any other case, to be thorough.

(2) Recursive function calls are the "natural" way to solve such a system (where either the donors or the flow target has to be treated prior to the respective points). Each recursive algorithm can be converted into a "list-based" classical scheme, which is what Braun and Willett did. Employing graph theory it just overkill in this context. "My" recursive implementation is a bit faster, while the list implementation of the stack is a bit more memory-efficient under some conditions and helpful for very old programming languages such as FORTRAN77 that do not allow recursion.

Let me clarify a few points. I took some shortcuts with the phrase "use of graph theory" in reference to Braun and Willett — as both implementations are based on it. I maintain the use of the term: there is nothing exotic here, it is simply the generic name for algorithms operating on graph data structures, which is precisely what river networks are. Your recursive function is a Depth First Search (DFS) graph traversal — the recursion is just an implementation pattern and could be replaced in C++ by a LIFO-based `std::stack` with equivalent behavior and performance. The "list-based" approach is a topological sort, ordering the nodes of a Directed Acyclic Graph (DAG) from the most upstream to the most downstream — or the reverse. In OpenLEM 45, the REDSEQ option appears to compute this using Kahn's algorithm: a straightforward BFS from downstream to upstream when DOWN is not activated (since all in-degrees are 1), and the full in-degree computation for the upstream-to-downstream ordering. Finally, your `fillLakes2` routine, if I am reading it correctly, is essentially the Priority Flood algorithm from Barnes (2014), which is Dijkstra-like. All of these are cornerstones of graph theory — or, if you prefer to reserve that term for the mathematical discipline, of graph algorithms.

Both Braun and Willett (2013) and OpenLEM 45 use a DFS traversal. OpenLEM 45, as far as I can tell, recomputes the DFS at each traversal step — for instance, recomputing fluxes and erosion on each SHAREDSP pass. Fastscape, by contrast, computes the topological order (the stack) once, stores it, and reuses it across subsequent operations. This is a significant distinction: beyond performance, many frameworks — TopotoolBox, LSDTopoTools, and LANDLAB, among others — adopt this approach of computing the stack once and then applying a wide range of geomorphological routines in sequence (watershed labelling, river network extraction, flow accumulation, etc.). These works cite Braun and Willett precisely for this contribution, not for the implicit solver. The value lies in accessibility: many geomorphologists lack a strong computer science background and would struggle to adapt generic algorithms such as Kahn's to their specific applications. Braun and Willett, without reinventing the concept of graphs, describe these algorithms with sufficient detail and geomorphological context to make them usable for non-specialists.

On memory efficiency, I would argue it depends on whether the donor array is stored, compacted, or dynamically queried — for instance, whether a neighbor j of node i is identified as having i as its sole receiver dynamically or if it is precomputed in an array. One could even argue that OpenLEM is more memory-efficient in this respect, precisely because it does not store the topological order.

(3) Of course, both schemes are O(n).

Agreed. Rerunning the DFS does not affect asymptotic complexity. However, there is a break-even point at which the overhead of precomputing the topological order becomes worthwhile, depending on how many traversals are required.

(4) Braun and Willett indeed mentioned that completely separated catchments can be treated in parallel, which I found too trivial to be mentioned. And if I was allowed to tell the story behind: I was disappointed at first that Jean Braun did not reply to my mail. But after talking in person, everything was fine because it was clear that he really did not know that almost everything was already there 12 years earlier. Finally, I was a bit disappointed again because he never kept his promise to cite my work.

I acknowledge this may be frustrating, given that the implicit solver for the SPL has become a cornerstone of landscape evolution modelling. The best I can do is to cite it myself if I publish on SPL solvers in the future. Returning to the review: I will not hold up publication over this — if you feel "popular" is the right word, I will defer to your judgment. I simply found the phrasing slightly subjective.

True, but only relevant in quite flat areas or at hillslopes (see above).

Yes.

l. 46: Add a reference to Howard and Kerby (1983) for the SPIM.

To my knowledge it is.

Right, but as mentioned in the introduction, Benjamin Campforts has demonstrated that knickpoints are preserved better with a suitable shock-capturing scheme, which would also improve the behavior for horizontal movement parallel to the rivers. So I would see knickpoints not as a clear argument against the Eulerian approach.

Fair point.

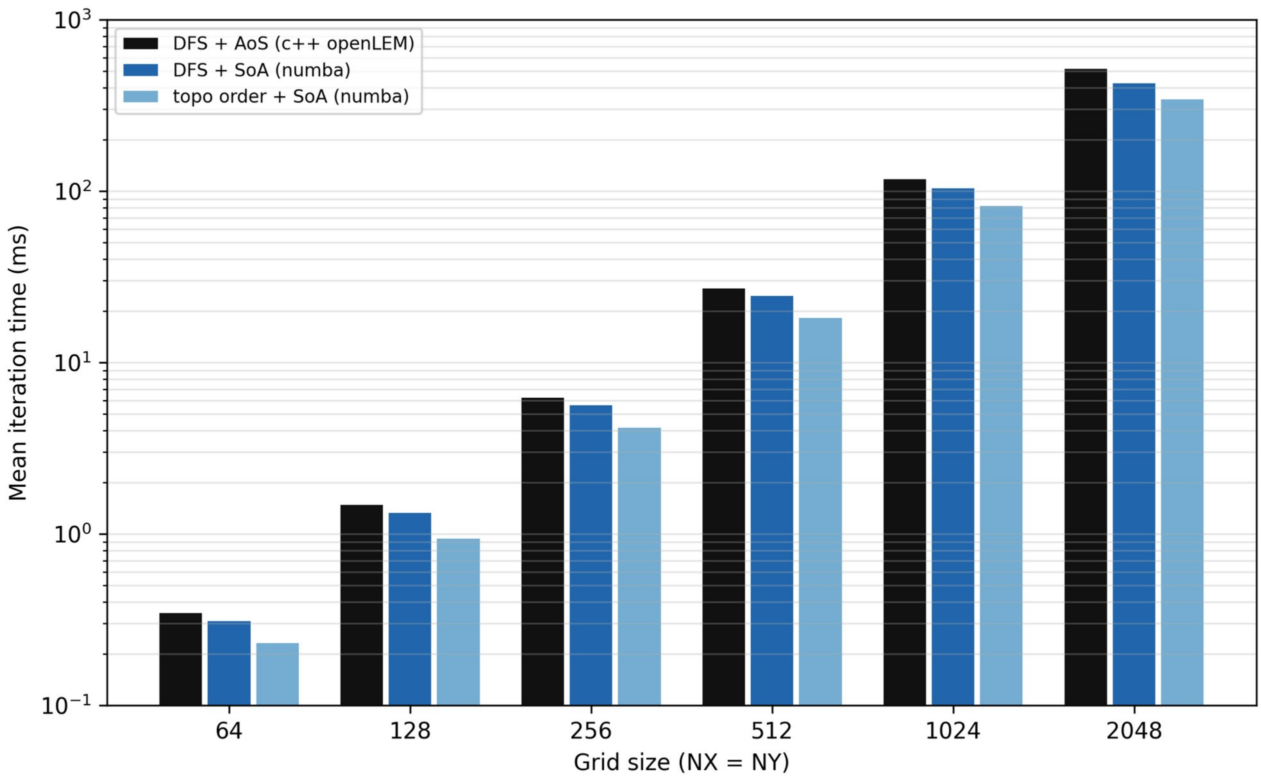
Perhaps a matter of the wording, but focus should have been on "while keeping the conceptual simplicity and the numerical efficiency of the SPIM."

Yes, to my interpretation the wording provides the feeling having sediment transport is a key component specifically.

There is no doubt that it is also possible to implement my scheme in an inefficient way. I think it is also possible to construct scenarios where the simple fixed-point iteration around the explicit treatment of the sediment fluxes by Yuan et al. (2019) is a bit faster than my implicit scheme. In my other applications, the main aspect is that my scheme keeps its performance even for a completely transport-limited model, but this is not relevant here. Anyway, we will find another excuse for using OpenLEM.

I do not think there is any need to justify the use of OpenLEM — it is a well-established model. To be fair, this contribution could use a standalone code and would still be worth publishing, as that is not the main message. I tend to be triggered (in a positive sense) by claims of numerical efficiency, as these depend on many factors beyond the base numerical scheme. That said, I was actually wrong in my assessment of the Yuan et al. (2019) implementation, and your scheme is indeed unconditionally faster as long as $G > 0$. A quick algorithmic analysis confirms this: the shared SP requires two traversals — one top-down to accumulate fluxes based on h at t, and one bottom-up to implicitly propagate changes in dh and correct the sediment flux. Yuan requires at least two traversals as well — one initial flux accumulation and one for the first Gauss-Seidel iteration, with each additional iteration requiring two further traversals. Both schemes have comparable arithmetic per traversal so as long as the GS requires more than 1 iteration it is theoretically slower. So I was wrong, and I apologize — though the picture becomes more complex for $n \neq 1$.

That said, and purely out of professional pride, numerical efficiency still depends heavily on implementation. To make the point concretely — and this is entirely machine-specific — replacing the Array of Structures layout (AoS; in OpenLEM, the Node class aggregates all variables) with a Structure of Arrays (SoA) and precomputing the topological order improves cache locality and memory throughput, yielding a speedup of around 1.5× on my machine. I compiled OpenLEM with `g++ 12 -O3 -march=native` and the SHAREDSP option (RECSEQ segfaulted for reasons I have not yet identified). It is worth noting that the algorithm is theoretically O(n), though irregular memory access patterns introduce cache effects in practice that the asymptotic analysis does not capture. See the figure bellow.



Here it is the same, but not necessarily in future applications involving numerical convection models. And a single adaptive dt for all points would not make much sense because it would only capture the fastest point well. Even for the rotation of a rigid body, it would not help much. Theoretically, we could hold back movement individually for each point and individually in the x- and y-direction until it reaches dx. However, I think that this will generate dams and violate the "mass balance". And at one point we should ask: Given that the present version of Fastscape indeed cannot handle Lagrangian coordinates, would it really make sense to build a new world around Fastscape?

Agreed, and I acknowledge that tracking and advecting horizontal motion would introduce its own stability challenges. I should note that I refer to Fastscape frequently throughout this review simply because it is the tool I am most familiar with (and as you said is a “popular” one) — I am not suggesting it should be treated as the reference implementation. Similarly, just as the strength of the SPL lies in its efficiency and simplicity, Eulerian grids are inherently simpler and more generic to work with than Lagrangian irregular grids — and that is legitimate to consider when suggesting the migration to a lagrangian scheme.

*Fig. 2 is only the reference without erosion to illustrate that the central scheme and the Lax-Wendroff scheme would not be too bad without fluvial erosion. Here v*dt is 3 orders of magnitude smaller than dx, which means that the error from dx dominates over the error from dt. In this sense, it happens no matter the dt value, except for the very specific case of uniform translation into the x-direction at the CFL limit. And about grid spacing: Finally, it is a nondimensaional combination of velocity, erodibility and grid spacing that defines the behavior. However, we can just switch between rivers just not moving and destroying topography and not get rid of all problems easily.*

Thank you for the clarification.

Best regards and thanks,

Boris