Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2019-126-RC2, 2019

Geoscientific Model Development Discussions

Interactive comment on "eSCAPE: Regional to Global Scale Landscape Evolution Model v2.0" *by* Tristan Salles

Benoit Bovy (Referee)

bbovy@gfz-potsdam.de

Received and published: 14 August 2019

General comments:

This paper presents the eSCAPE landscape evolution model. It is overall well written and nicely illustrated. eSCAPE features are explained in a consistent order and each with a sufficient level of detail. The manuscript contains references to comprehensive material available online that will help users installing, running eSCAPE and reproduce the experiments presented in this paper.

The author clearly highlights the various benefits of using a matrix-based approach for computing flow accumulation and sediment transport in both continental and marine environments. The most outstanding benefits are, in my opinion, the implicit scheme

used for computing drainage area, the horizontal scalability (distributed computing) and the ability to re-use powerful libraries like PETSc. This approach is not new, but eSCAPE represents to my knowledge the first effort towards integrating it into a user-friendly landscape evolution model that has the minimal set of features required for applications at global scale.

In the examples section, the author carefully shows the implications of the choice of the time step, even in the context of implicit numerical schemes used for most of the model components, which is very much appreciated.

eSCAPE is designed to handle very large grids and aims at simulating landscape evolution at regional to global scales. The author states that this is the first landscape evolution model able to simulate processes at global scale.

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.

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.

Specific comments:

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.

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.

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.

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

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?

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). Dis-

claimer: I'm co-author of this paper.

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.

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

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.

Figure 7: Very nice figure that captures detailed profiling results at a glance!

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2019-126, 2019.