Dear Boris,

so many interesting things to discuss! And of course, all your ideas to bring Eulerian schemes back into play are important. As mentioned, the reasoning is still just that I cannot imagine that any Eulerian scheme would work. So I end up with the result that I already thought about ideas similar to yours and that they will probably not work, but it is important to think about these ideas. And I really enjoy the discussion.

Just about the Lagrangian scheme: It is quite simple for the shear example considered here, and there is nothing like an interpolation anywhere. And remeshing is just reconnecting each point to its 8 neighbors, but neither inserting nor deleting points. So just good enough for the example considered here, but there is some way to go towards an efficient Lagrangian LEM.

I also did not expect that anyone would implement "my" numerical scheme and find ways to improve performance compared to OpenLEM (although machine-dependent, as you say). I never tested "array of structs" vs. separate arrays. While the 2001 version and the version in the Robl et al. papers used separate arrays, I found the "array of structs" more elegant when starting with OpenLEM and never questioned this choice. For coupling with other models or integration into bigger frameworks it is, of course, not as good as I thought. So it is interesting to know that the separate arrays could also be more efficient.

Concerning the list vs. recursion question, I thought that I found the "best" mixture some time ago (the REDREC option you mentioned). Allowing recursion, but memorizing the order of nodes to reduce the amount of recursive steps. Not as good as I thought so far. The bottleneck may be that I still have to iterate over all neighbors to check whether all donors have already been treated. It should be possible to improve it, and we will see whether it catches up with the list implementation.

And of course, my statement about overkill in the context of graph theory was not to criticize that you mentioned graph theory. Everything you wrote in this context is correct, and it is graph theory. It was rather that the downward sweep needed to sum up the catchment sizes is more complicated than the upward sweep in the erosion model. And I think that something like this must already have been included in the flow-accumulation functions of GIS and in other LEMs before 2013. So I felt that Braun and Willett (2013) expanded an almost trivial aspect in an unnecessarily complicated framework. Anyway, I think we will end up at something like "extended and made popular by Braun and Willett (2013)", which allows to read between the lines that the extension was not really big in my opinion.

Finally, about the potential steady state under ongoing shear. The attached figures are Figs. 8 and 9 from the manuscript, but for a longer time span. Figure 8 indeed suggests that some kind of steady state is approached in the two versions with remeshing. But unfortunately, I did not realize initially that the periodic boundary conditions could be a problem under strong shear. There is one long river that drains the northern part and one that drains the southern part, which interacts with itself concerning the topography. While these rivers still follow the shear, tributaries are more feather-like, indicating reorganization events at different times. Quite interesting, but we still cannot say what is correct. I guess that SPIM-like models overestimate the (undoubtely existing) persistence of drainage patterns in mountain regions, but we will probably not know before "better" models have become available.

Best regards,
Stefan

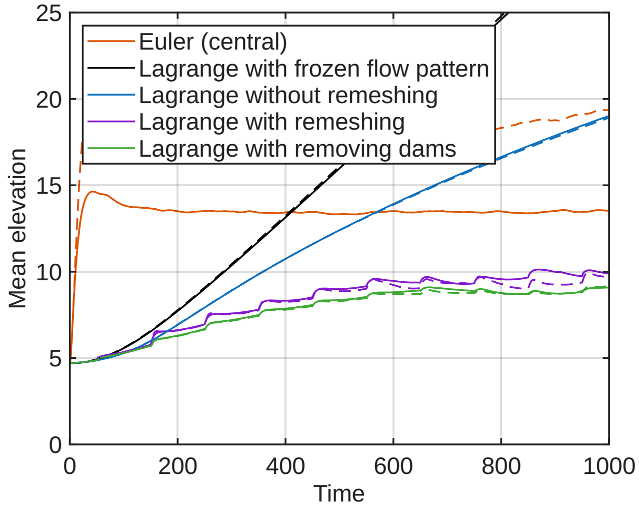

Figure 8: Mean elevation for different advection schemes. Solid lines refer to the shared stream power model and dashed lines to the SPIM.

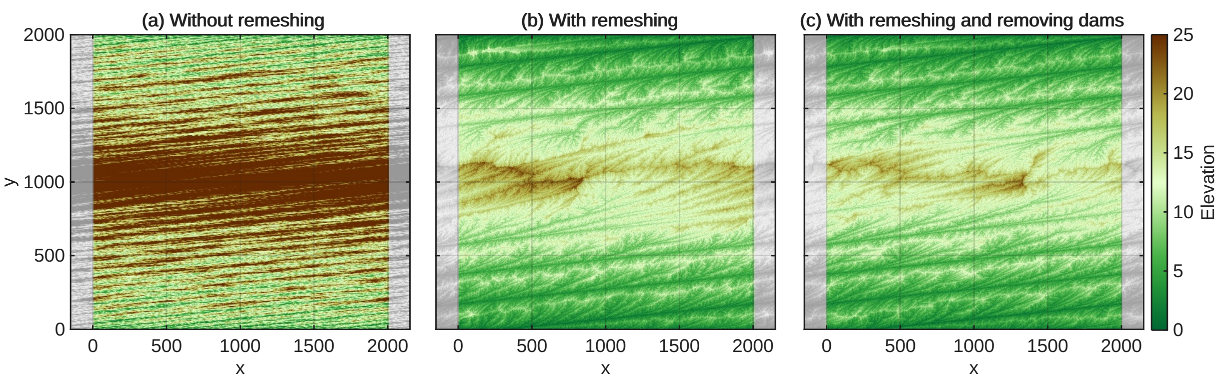

Figure 9: Topographies at $t = 1000$ ($\gamma = 10$) with and without remeshing.