Earth Surf. Dynam. Discuss.,
doi:10.5194/esurf-2016-45-AC3, 2016

Interactive comment on "Creative computing with Landlab: an open-source toolkit for building, coupling, and exploring two-dimensional numerical models of Earth-surface dynamics" *by* Daniel E. J. Hobley et al.

Daniel E. J. Hobley et al.

hobleyd@cardiff.ac.uk

Received and published: 24 November 2016

Dear Andy,

Thanks for taking the time to complete such a thorough review. As requested by the AE, please find below our replies to your comments. We've pasted each comment you made, then replied to it, and as necessary then copied in any modifications made in the text. We hope that these replies address your concerns to your satisfaction.

Thanks again,

Dan (on behalf of all the authors)

Summary

Hobley and co-authors present Landlab, a toolkit that is poised to significantly change the face of Earth-surface modeling. I expect their framework to easily build and integrate models of the evolving Earth's surface to enable the community to improve – possibly rapidly – the state of our knowledge and theory of Earth-surface dynamics. On a personal note, I have followed the development of Landlab since its inception, and offer my enthusiastic support of the publication of this article that makes this comprehensive modeling infrastructure easy to follow in a way that mirrors the clean and well-written code base.

I recommend that the article be accepted for publication after some very minor technical corrections. As the article is, in my opinion, ready for publication, this review contains both technical notes and copyedits as a result of a very thorough read.

Main text

General: Many geomorphic models are run in 1 dimension. Is Landlab capable of solving 1-D problems? As I understand from my reading, it is designed for 2-D solutions, but it would be helpful to have a comment on this to clarify.

ν Added to grid section: "Although Landlab grids are inherently two-dimensional, in many cases it is nonetheless possible to create an effectively one-dimensional simulation by creating a 3-by-N regular grid and closing the nodes along the top and bottom edges (see Section 3.1.4). Three-dimensional grids are not possible in Landlab at this time, though may be supported in a future release."

Page 4, line 9: I second Wolfgang Schwanghart's comment regarding open-source modules written in closed-source languages. Here, the algorithm may be known even if it cannot be run without the closed-source core programming language. ν We've simplified this section by removing reference to these 'open-source-but-on-a-closed-source-framework' type pieces of software, as the level of detail required to do the subject justice seemed a bit much for what is ultimately a tangent to the manuscript.

Page 6, line 9: Strong agreement regarding rapid prototyping. Based on reading the article, and having never used Landlab before (though having some familiarity with it), I was able to assemble a simple program for a class exercise in 1-2 hours. ν DH: Glad to hear we're achieving our targets in this area. . .!

Page 9, lines 6-10: Easy to note that this is standard right-hand-rule, if you think this is helpful. ν DH: Added: ". . .(i.e., the right-hand rule). . .."

Page 10, "get" and "create" methods: I have seen "get" commands, such as "mg.get_grid_ydimension", that are exposed to the end user and have no underscore. So I must guess that there are two types of "get" commands, or... ? ν DH: Good catch. There are nine of these, and they are stragglers from before Landlab's internal style standardization. All have direct modernized equivalents, and they should be expunged in the final release of Landlab that goes alongside this paper.

Page 11, section 3.1.4: More a comment: the single set of boundaries implies to me that you have intended the model for discretization of only first and second derivatives. ν As you note, all current components operate only with first and second derivatives. However, it would be conceptually straightforward to create more boundary condition types as and when the need arises to accommodate higher order derivatives.

Page 12, line 16: Missing a noun between "ID" and "of". ν I don't think so; but nonetheless this comment indicates this phrasing isn't super clear. I've changed it a little to try to improve it: "By indexing these arrays with the IDs of element subsets, the values at specific locations and on each element type can be recovered."

Page 12, line 24: "Pairwise Transition Automata" appears, on Google and Google Scholar, only in this article, and its meaning is unknown to me. Could you elaborate? ν Modified to: 'For instance, pairwise transition automata (Narteau et al., 2001; 2009)

represent the states of cells on a grid as paired "doublets", with rules prescribed to govern the rates of transition between each doublet type. These are readily implemented in Landlab by mapping the pair states onto the links of a Landlab grid, and representing the corresponding automaton cell states at grid nodes (Tucker et al., 2016).'

Page 13, line 20: What is "syntactic sugar"? ν We now explicitly define this (rather delightful) piece of computer science jargon: "Landlab offers some degree of "syntactic sugar" for its field name interface – i.e., the field interface is made more user-friendly by the addition of more readable grid properties to query the fields at each element type, rather than requiring the user to access the both dictionaries directly. For instance, grid.at_node['my_field_name'] is equivalent to grid['node']['my_field_name']. In addition, Landlab also provides convenient shortcuts to create new fields of ones (grid.add_ones), zeros (grid.add_zeros), and from existing data (grid.add_field)."

Page 13, line 28: Missing hyphen for steady-state (modifies "solutions") ν Added, and similarly at ln 29.

Page 15, line 18: Suggest replacing "However" with "While these are focused on Earth's surface" (or "While these are Earth-surface focused", if you like some dislike using the vestigial English genitive in scientific writing. ν We've settled on: "Although these existing components are largely Earth-surface focused, we emphasize that Landlab permits modeling of the evolution of almost any two-dimensional system that lends itself to description by discretized systems of differential equations or cellular automaton rules."

Page 16, line 4: Having not thought about it before, it is not clear to me how you would combine a set of Jacobian matrices to, presumably, simultaneously solve an arbitrary number of coupled processes. It is probably just my ignorance, but I would like to see a bit of explanation and/or a reference. ν Given this remains entirely speculative material about future work, we've decided to remove this (admittedly pretty unclear) paragraph entirely, rather than invest the space needed to actually do this idea justice.

Page 16, line 9: changes → change; the word “data” is plural. ν Done

Page 17, line 19: Are “known correct solutions” all analytical solutions, or do you include solutions that go beyond this limited class of solutions? And/or, is “solutions” meant more generally – as in, is this also just making sure that general functions that you have programmed that may not be generating mathematical solutions are functioning as expected? As I type, I’m starting to think it must be the latter... ν Yep, the latter. I think this is strictly correct as written, and bringing up analytical solutions specifically implies that the unit tests are written only for the final solutions to differential equations in the components, which they are not. The grid is heavily tested too, for example. On this basis I don’t intend to make changes here.

Page 18, line 26: uplifting → uplift ν For maximum clarity: “. . .or other similar scenario with a radially symmetric uplift field.”

Page 20, line 15: More general question: Your method implies central differencing. How do you ensure that numerical diffusion is minimized and/or is not large enough to be important? ν This is indeed a central differencing method, as is widely deployed in handling diffusion problems in gridded models. However, our understanding is that numerical diffusion should only become an issue in advection-diffusion problems, or alternatively in upwinding schemes. This component as described here is neither, so a full discussion of the general issues around stability would seem a bit out of place at this point in the text (see also our reply to Campforts on numerical stability).

Page 21, line 8: Would this be the depth-integrated flux, as “flux” generally means [quantity/[area time]], so volumetric flux is L/t and therefore depth-integrated would become L2/t? Of course, Earth-system modelers use “flux” more liberally, but in a mathematics (and fluid mechanics), this is the use that I know, and I think it’s best to be precise with language – especially considering how much you are in general. ν Indeed. We have added this wording.

Page 21, line 11: Again, more general question: if you are using the CFL condition, I

am guessing that this is referring to something that is Euler forward. How would you choose time steps internally while using some of your implicit methods? ν I’m not sure which implicit methods you’re referring to here. As emphasized in section 3.3.2, each component handles its own timestepping internally, and independently from all other components (see also Fig. 6). Thus in this specific case, the LinearDiffuser component is using a timestep governed by the CFL condition, while a putative separate implicit component would be using its own (longer?) internal timestep as governed by some other stability condition. The “coupling timestep” set by the user (sensu 3.3.2) is different again. I note that issues of timestep stability also came up in Campfort’s comments, which resulted in some new text (now on p. 15) described in reply to him. Hopefully this text will also clarify things in relation to your comment here.

Page 22, Equation 9. Note $\geq$ is possible ($\geq$) in LaTeX. There are also options to use curly brackets to make the conditional part of the equation clearer; see http://tex.stackexchange.com/questions/47170/ how-to-write-conditional-equations-with-one-sided-curly-brackets. ν This is not a LaTeX manuscript! This symbology was chosen largely as it mirrors that in Python. We’d absolutely be happy for the manuscript to be later typeset by ESurf like this. . .

Page 22, line 22. Near m/n, you may mention that this is the channel concavity. Perhaps, “m/n, i.e., channel concavity, ≈ 0.5. ν Added: “. . .interpreted from channel concavities of natural rivers at apparent topographic steady state.”

Page 22, line 26. Remove “possible” (repeated later) ν Done

Page 23, line 2: Remove “performs as order-n, and as expected”: redundant. You could keep “as expected” if you like. ν We disagree that this is redundant; we wish to demonstrate that the component is performing as it should in terms of its scaling. We’ve left this.

Page 23, lines 3-4: “Broadly linearly” sounds strange: two dimension terms, the first metaphorical. How about “approximately”, or even just no modifier (it is so close to linear)? ν Modified to “close-to-linearly”

Page 23, lines 17-19: “based on...2001b).” is a repeat of text at the start of section 5.2 (p. 22, l. 2-6). ν We’ve deleted this text the first time it appears.

Page 23, line 20: “storm depth” to “stormwater depth” (or something like this)? ν “. . .storm water depth. . .”

Page 23, lines 24-25: “looped upon” to “called within the loop”, and remove “until...cease”. If I understand you here, this is a simpler way of writing this. ν Now “in other words, the code block below the generator will repeat with fresh values for each iteration until the total time is elapsed, at which point the loop will cease”

Page 23, line 28: “and where it only rains” to “with rainfall occurring” for better parallel structure ν Done

Page 25, line 5: remove comma after “region” ν Done

Page 25, line 6: remove “using” after “by”: unnecessary ν Done

Page 25, line 6: did you just write that stormwater depth is given by a gamma function? Also, “storm depth” should be turned into something like “stormwater depth” that makes more sense, per the above comment. ν Yeah, this was ambiguous as written. The durations are exponentially distributed, but the depths are gamma because they combine exponentials. The latter section is rephrased lightly to clarify. Changed also to “storm water depth”.

Page 25, line 9: Does Bras’ model incorporate latitude, integration over a day, etc.? And is it simple incident radiation (i.e. no diffuse or reflected)? I think that the latter is true from what you have written, but not sure about the former. ν This section has been expanded to address this explicitly: “The Radiation component calculates daily average extra-terrestrial and clear-sky shortwave radiation incident on a flat surface, based on latitude and day of the year (ASCE-EWRI, 2005). This component also calculates daily radiation ratio, defined as the ratio of cosine of solar angle of incidence for the true

sloped surface to that for a flat surface (Bras, 1990). The Radiation component does not explicitly calculate diffused and reflected radiation. The PotentialEvapotranspiration component uses the radiation ratio to calculate spatial net radiation using daily maximum and minimum temperature, and potential evapotranspiration (ASCE-EWRI, 2005; Zhou et al., 2013).” Note that we also had a reference manager issue with the reference that was formerly listed as “Snyder, 2005”; this should in fact have been rendered as “ASCE-EWRI, 2005” (a technical report chapter), as it now is.

Page 25, line 21: “and also” → “and” ν Done

Page 25, line 29: “shrubs cluster as they...” ν Much better! Done.

Page 26, line 12: “shallow” is ambiguous since you are writing about flow depth and slopes. I would write “low-slope” instead. Also, did you mean a different word than “urban”? ν Good call; did this. Also no, urban is right. The Bates algorithm was originally derived for use in city planning.

Page 26, line 29: “a uniform rainfall rate”. Also, this is by filling the pits with water, I presume. ν I’m unclear what this comment is in reference to, sorry. The section around here seems OK to me, so I’m leaving it.

Page 28, line 4: You haven’t mentioned visualization except for one reference to ParaView that you didn’t follow up and a passing reference to matplotlib. Not actually a problem, but a note. ν This material was excluded mainly for reasons of manuscript length, and I think that’s a decision we stand by. Lots more information about plotting and visualization can be found through the LL website.

Tables Table 1b: What about raster D8? ν This gets at an aspect of Landlab’s architecture which we have specifically not covered in this paper, because it is still fluid in the code: although it is possible to build rasters and rectilinear grids with diagonal node connections in version 1, the details of this are not exposed to the user at this stage, and may yet be modified. Essentially, Landlab doesn’t view “diagonals” as true links (since they don't honour the graph theory rules that underlie Landlab's grid); any time they are invoked, they must be invoked separately from the true, orthogonal links. Since this functionality isn't user-facing, we've deliberately not covered it in this manuscript. I note that this was slightly ambiguous when we referred to "diagonal links" in table 2, so I've modified that description to now read "diagonal connections between nodes" (they aren't links!) This should make everything technically correct.

Table 2: Should rectilinear have rectangular cells, not "quasi-rectangular"? ν Not necessarily. This category also permits topologically warped rectangles (e.g., as in a global projection), should the user so desire.

Table 4: More of a comment: based on all boundaries being inactive, a 1D model in Landlab would have to have 3 × (N + 2) nodes. So it really isn't optimized for 1D, it seems. ν Certainly not optimized, no, but still perfectly capable. Landlab doesn't actually perform component calculations on inactive links in most cases, so it's only really an issue of overhead in grid setup. During the iteration of a run, it would run pretty efficiently in most cases.

Table 5: A nice set of modules... but at risk of sounding ungrateful of the large amount of work that this was, it would be really great to see a depositional components for Landlab appear... in case this isn't obvious, this is not a criticism of the paper, but more a comment on where this work could lead. ν Yep yep, this is very much on our minds (see also comments to Campfort's review).

Figures

Figure 3: Please ask copyeditors to ensure that this appears on the same page as Table 1. ν Noted.

Figure 4: Are patches numbered in order of min(max(surrounding nodes or links))? (Question inspired by (b)(ii).) ν Nope. Patches are ordered by centroid position (i.e., by corner), according to the same scheme as all the other element types as described in

the main text. It's just a little unclear from this particular figure – which really focuses on the link directions – exactly where all the centroids are for these patches! I did in fact have to confirm this was right for myself, as it looks a little shonky.

Figure 7: Include line numbers. Also, flux → discharge, per our discussions? In addition, for this and the other figures with code, I am not sure if Âż> and ... are needed or are distracting... they are more the latter for me, I guess. But your decision in the end on style. ν In this case, we think "unit flux" is probably the right terminology, as it's a discharge per unit width of the face. We think we will retain the »>'s here and elsewhere, as this style explicitly echoes the "doctesting" style in Python that users will see widely through Landlab's documentation. We think it would be good to introduce them to it here, rather than adopt a different style here and another in the online documentation.

Figure 8: You use both "base level" and "baselevel"; check for consistency (2x/each). ν We've committed to "base level". We also noticed that we had inconsistencies between "time step" and "timestep", which have now been unified as "timestep".

Figure 9: Why not combine this with Figure 8, if they make the same plot? ν We think having them separate is less confusing, as they appear separately in the text, and it would be hard to explain in the caption alone why both were being shown.

Figure 10: Please tell us how many cells you computed over to help readers gauge their own compute times. ν I'm unsure how this is different to what we are already showing here. LL generally regards nodes as the primary element type, so the figure gives speeds for numbers of nodes. Given these are simply nxn grids, users could easily compute numbers of nodes for themselves from the information given (e.g., ncells = (sqrt(nnodes)-2.)**2 for a raster). In most of these cases, it makes basically no difference whether we count nodes or cells.

Figure 14: (b) Vertical axis: PFT is more precise than plant type? Although this is clear from everything else. Legend: shown in parts (b) and (c) of this figure. ν We've chosen to leave this, as it seemed very minor.

Supplement

No concerns.

Testing

During the review process, I took advantage of the fact that I am teaching a geomorphology class to test Landlab. In our modeling lab, the students, most of whom have little experience in programming, were all able to install and run Landlab. In some cases, differences between computers and Python versions caused a little confusion, but the code itself worked well. The students are currently working using it with a simple coupled channel–hillslope model to (1) find relationships between hillslope diffusivity, the stream power coefficient, and the drainage area of the hillslope–channel transition, (2) investigate which factors lead to changes in the time it takes to reach steady-state conditions in a landscape, and (3) explore the effects of changing grids, uplift patterns, stream-power exponents, and more on the model results. The overall reception was very positive in this mixed undergraduate–graduate class, with students realizing through the model how real landscapes form and evolve. The code that I used, modified from one of those supplied in the paper, is attached below. I intend to teach the lab again. One issue that I had while running this code on Linux within iPython is that the plotting tools would not release control of the shell to me (i.e. after plotting, I could not type anything more into the terminal, even after closing all of the graphical windows). This of course is not an issue to preclude publication, but I will add it as a ticket on GitHub.

ν Thanks for this feedback, it's very valuable to us. Please do submit a ticket, though past experience of plotting issues suggests to us that this will be an issue arising from matplotlib's idiosyncrasies rather than an actual issue with Landlab. Nonetheless, we'll take a look.

Interactive comment on Earth Surf. Dynam. Discuss., doi:10.5194/esurf-2016-45, 2016.