Response to reviewers

Comments from the peer reviewers are typeset in black, our response is typeset in blue and changes made to the manuscript are indented.

Community comment #1

- Comment

  This manuscript presents TopoToolbox 3, a re-designed set of software components that provide tools for both topographic analysis and modelling.

  The manuscript illustrates well how the development of TopoToolbox 3 follows the principles of Findability, Accessibility, Interoperability, and Reusability (FAIR) for research software. For example, section 2.3.4 describes how it integrates with Landlab, another software widely used for landscape evolution modelling.

  I'd like to emphasize that developing research software according to the FAIR principles is a significant undertaking (API and architecture design, documentation, testing and continuous-integration, packaging, reviews, etc.) that has the potential of greatly benefiting an entire community. I believe that TopoToolbox 3 will be no exception for the Geomorphology community. Such effort merits recognition and credit, and I sincerely hope that TopoToolbox 3 maintainers and contributors will get all the credit they deserve!

  I'd also like to note that besides TopoToolbox 3 (and Landlab), there exists other stacks of software that share similar broad — more or less ambitious — objectives and features, that have been developed using the same FAIR principles and that together have been benefiting the same community. This includes, among others, Fastscape (`https://github.com/fastscape-lem`) and LSDTopoTools (`https://github.com/LSDtopotools`). The manuscript could certainly be more effective at mentioning such relevant projects, perhaps with some brief sentences on how they compare with TopoToolbox 3 and/or how they might be integrated with each other, conceding that an exhaustive review or thorough comparison is well outside of the scope here. Sadly, though, this appears to have been completely overlooked in the current version of the manuscript (e.g., no citation found for any of the aforementioned projects). Improving this aspect in the manuscript would be useful, much appreciated and — mostly — fair.

- Response

  We have added some citations to geomorphology software packages in this paragraph:

  > TopoToolbox's dependence on MATLAB (The MathWorks Inc., 2022) has been critical to its success. MATLAB provides an integrated development environment, an interactive visualization system and its own extensive documentation. Users can download MATLAB and TopoToolbox and immediately begin using the software.

However, while TopoToolbox itself is open source and freely available, it requires a license for MATLAB as well as several additional toolboxes that provide both essential functionality (Image Processing and Mapping Toolboxes) and important additional functions (Optimization Toolbox). Many other software tools for quantitative geomorphology also exist from topographic analysis packages similar to TopoToolbox such as Whitebox (Lindsay, 2016) and LSDTopoTools (Mudd et al., 2019), to landscape evolution modeling frameworks such as Landlab (Hobley et al., 2017; Hutton et al., 2020; Barnhart et al., 2020) and Fastscape (Benoit Bovy and Lange, 2023), to standalone analytical tools such as TopoMetricUncertainty (Smith et al., 2019), pyTopoComplexity (Lai et al., 2025) or CliffDelineaTool (Swirad and Young, 2022), to complete GIS platforms like ArcGIS and QGIS (Dawson et al., 2026) and programming languages like Python, R and Julia, and these represent only a fraction of the many software tools that researchers use to study the Earth's surface (Grohmann et al., 2026). Each of these tools provides a unique combination of analyses and user interfaces that may serve a user better than TopoToolbox, but if a user of one of these tools wishes to access an analysis available in TopoToolbox or if a TopoToolbox user wishes to integrate one of these tools into their workflow, they can face challenges interfacing between the different computational environments.

Reviewer #1

- Comment

  I am glad to see this work. That's surely valuable. TopoToolbox has already been widely used in a wide variety of fields including but not limited to tectonic geomorphology, geology, hydrology, glaciology and even ecology. The previous versions are based on Matlab platform, which may lack the extension with GIS software and landscape evolution modelling packages. This new version developed by Kearney et al. is available for Python, an open-source platform that is supported by e.g. QGIS, landlab . . . . There is no doubt that the development is at the state of art and will meet the evolving needs of the geoscience community.

  I only propose two minor suggestions. 1) What is and how to calculate chi, because this software is widely used and not every reader is familiar with this metric. 2) A shematic showing the function and work flow of TopoToolbox, may be I missed this part.

  This manuscript is well written and no grammar errors.

- Response

  We have added more details about the computation of χ to section 2.3.1 as follows.

  > χ maps provide a graphical illustration of the potential for dynamic reorganization of river basins (Willett et al., 2014). They display the value of the χ metric (Perron and Royden, 2013) at points within the stream network. χ is a coordinate transformation that arises when solving the steady-state stream power equation by separation of variables. It is obtained by integrating upstream the inverse of the contributing area (Eq. 2)

$$\chi = \int_{x_b}^{x} \left(\frac{A_0}{A(x)} \right)^{\frac{m}{n}} \, \mathrm{d}x \tag{1}$$

where A_0 is a reference area, $A(x)$ is the contributing area at each point x in the stream network, which begins at a location corresponding to base level x_b. and m and n are the parameters of the stream power model. When the along-stream coordinate is transformed to χ, stream profiles that solve the steady-state stream power model appear as straight lines whose slope is controlled by the ratio of uplift rate to erodibility.

We have modified Figure 1 to illustrate a hypothetical workflow similar to the χ maps. We have also modified Figure 2 so that it illustrates the different portions of the χ maps workflow.

Reviewer #2

- Comment

  Over the last 15 years, TopoToolbox has become a valuable and widely used open-source package for topographic analysis. This paper presents a major new release, which adds a Python interface and includes a major restructuring of the software to allow it to support multiple languages. To my mind, this is a very nice contribution. The new capabilities substantially broaden the scope of TopoToolbox and its interoperability with other packages and across languages. The development of a Python interface is especially significant because it allows users to do combined analyses with TopoToolbox and an expanding collection of other Python-language scientific packages, without having to switch between languages or needing a Matlab license. At the same time, retaining the original Matlab interface provides backward compatibility and serves the large Matlab user base in geoscience. Coding the core routines in a shared C library seems like an effective strategy for enabling cross-language capability.

  The examples presented in the manuscript nicely illustrate its capabilities. I especially appreciate the thoughtful and comprehensive approach to software testing and quality control. The manuscript is well written, and the figures clearly designed. Pointers to supporting code allow readers to try out the examples for themselves. The authors might also consider saying more about what community involvement looks like in practice, how the software is managed and governed, and how contribution is encouraged and supported (beyond the technical practices of continuous integration and pull requests). I realize, however, that this may go beyond the scope of this paper.

  One potential area for improvement would be to explain how code reviews are handled. The manuscript notes that community contributions are encouraged, and that they are done via pull requests. But apart from a passing mention around line 422, nothing further is said about code reviews: who does them, what criteria are used, etc. Adding a few sentences about this would help give readers a sense of what might be involved in making contributions.

- Response

  We have added the following regarding code reviews to section 4.1:

> Once a contributor makes a pull request, a TopoToolbox maintainer works with them to ensure that the pull request passes the automated test suite, that it successfully addresses any pertinent bug or issue reports, that it comes with sufficient tests and documentation to enable users and other contributors to make use of the code, and that it meets the stylistic guidelines of the repository, which are documented in each repository's contribution guidelines. There is currently no formal process to become a TopoToolbox maintainer who has permission to merge pull requests, and contributors who have made contributions to TopoToolbox repositories and who express a willingness to help new contributors through this process can be given this permission by existing maintainers.

- Comment

  Overall, in my view this manuscript is a great example of how geoscience research software should be presented in journal articles. I recommend publication after minor revisions.

  Comments keyed to text:

  37-49 I appreciate this list of examples of use of TT combined with other software.

  115-119 use of contiguous arrays rather than objects is an interesting strategy, and one that is similar to the approach used by the Basic Model Interface, where data handling is kept simple in order to accommodate languages like C and Fortran.

  149-155 I'm not totally clear on the platform compatibility issue here. Any compiled library will need to be compiled specifically for a given hardware, so it is only portable in the sense that compilation is possible across different architectures. Is there something special about data-tiling that restricts it to only certain platforms and not others? And why would this not also be an issue for the hardware acceleration frameworks, mentioned? For example, if a particular acceleration requires an NVIDIA GPU, how is that portable? Obviously, this isn't a big deal here, since the point of the discussion is just to note possible future solutions to this problem, but I'm guessing that a little editing in this paragraph could help make the issue clearer for readers.

- Response

  Upon further reflection, we agree that describing this as a "platform compatibility" or "portability" issue confuses matters. We have removed that language and replaced this paragraph with the following, which is more specific about why the large-scale solutions don't work in libtopotoolbox.

  > libtopotoolbox currently focuses on in-memory processing of data to make as few assumptions as possible about where a user's data comes from. A DEM loaded into memory in MATLAB or in Python using rasterio or created by a landscape evolution model can all be passed to libtopotoolbox functions. The user decides how and when to load their data into memory. This design does make it challenging to operate on DEMs too large to load entirely into memory. Existing solutions for processing large DEMs require control over the input and output of data tiles and communication across the boundaries of tiles (e.g. Arge et al., 2003; Barnes, 2016, 2017). Implementing these algorithms in libtopotoolbox would require users to access their data through libtopotoolbox instead of the libraries they are already using, which is contrary to the design goal for libtopotoolbox to operate on as wide

a variety of data sources as possible. Such solutions could, however, be implemented in TopoToolbox's higher-level packages using libraries like dask (Rocklin, 2015) or MATLAB's Parallel Computing Toolbox to orchestrate I/O tasks and libtopotoolbox to perform computations on each tile. An alternative that is also being considered is to reimplement the libtopotoolbox interface using hardware acceleration frameworks such as the Parallel Computing Toolbox or Taichi Lang, PyTorch and TensorFlow in Python that provide access to resources like multiprocessors and GPUs. In this case, the high-level interface would remain the same, but users could switch between between the portable libtopotoolbox backend and high-performance backends as needed. Work is ongoing to leverage these libraries to provide an easy-to-use, high-performance interface for analyzing large datasets.

- Comment

  158-164 Is the automated test suite run on multiple platforms and OS versions?

- Response

  The following sentence has been added here:

  > The test suites are run on recent versions of Linux, Windows and macOS, and the Python and MATLAB package repositories run their test suites with supported versions of Python (3.10 and newer) and MATLAB (R2023b and newer).

- Comment

  170 Is snapshot testing the same as what is sometimes called regression testing?

- Response

  We have modified this section to clarify that snapshot testing is one way to do regression testing:

  > Regression tests ensure that changes introduced in TopoToolbox 3 do not substantially affect the behavior of the software. However, the desired behavior of many of the functions of TopoToolbox is not always clearly specified, especially around problems such as the handling of missing data or boundary conditions. We use characterization tests (Feathers, 2004, p. 186) to document the existing behavior of TopoToolbox functions by running a set of real and simulated DEMs through the functions in the MATLAB package and storing the result.

- Comment

  198 metamorphic testing is new to me, and the example helps make it understandable. The authors might consider adding a second example.

- Response We have added an example of a noninvariant metamorphic relation satisfied by the gradient function:

  > An example of a noninvariant metamorphic relation obeyed by the gradient function is that the gradient should be proportional to the inverse of the cell size. A test of this relation constructs two `Grid` objects with identical data but different cell sizes,

h_1 and h_2, passes them to the gradient function to obtain two `Grid` objects, g_1 and g_2, and verifies that the ratio of the gradients is approximately the inverse ratio of the cell sizes ($g_1/g_2 = h_2/h_1$) for all pixels with nonzero gradient. A violation of this metamorphic relation indicates that the gradient implementation does not use the supplied cell size correctly.

- Comment

  214 I tested the chi-map Jupyter notebook, and was able to run it without much difficulty. The only obstacles were (1) I had to install GDAL separately using conda (after trying a pip install and then getting problems related to a missing config file) before successfully pip-installing topotoolbox, and (2) I didn't see an obvious way to download the notebook from the gallery, so I used view source, which works for the notebook but then the auxiliary data file is missing. Consider providing a link to the full set of notebooks and data files for the paper in the Figure 2 caption.

- Response

  Direct links to the examples and a link to the gallery GitHub repository have been included. The auxiliary data file needed by the Python script has been added to the GitHub repository as well, so running the example from a fresh clone of the repository should work.

  Complete working examples can be found in the TopoToolbox gallery (MATLAB: `https://topotoolbox.github.io/gallery/notebooks/matlab/alpine-chimaps/alpine_chimaps.html` and Python: `https://topotoolbox.github.io/gallery/notebooks/python/alpine-chimaps/alpine-chimaps.html`), and the code and necessary auxiliary data to run the examples can also be obtained from the gallery GitHub repository (`https://github.com/TopoToolbox/gallery`).

- Comment

  216 missing 'from'

  219 missing comma

- Response

  This section has been modified somewhat in response to Reviewer #1, but both typographical errors have been corrected. The new content reads as follows:

  χ maps provide a graphical illustration of the potential for dynamic reorganization of river basins (Willett et al., 2014). They display the value of the χ metric (Perron and Royden, 2013) at points within the stream network. χ is a coordinate transformation that arises when solving the steady-state stream power equation by separation of variables. It is obtained by integrating upstream the inverse of the contributing area (Eq. 2)

  $$\chi = \int_{x_b}^{x} \left(\frac{A_0}{A(x)} \right)^{\frac{m}{n}} \mathrm{d}x \tag{2}$$

  where A_0 is a reference area, $A(x)$ is the contributing area at each point x in the stream network, which begins at a location corresponding to base level x_b. and

m and n are the parameters of the stream power model. When the along-stream coordinate is transformed to χ, stream profiles that solve the steady-state stream power model appear as straight lines whose slope is controlled by the ratio of uplift rate to erodibility.

To conduct a χ analysis in TopoToolbox, the basic workflow is to 1) load a raster containing elevation data, 2) route flow over the topography to determine drainage directions and drainage areas for each cell in the elevation data, 3) identify the stream network from these drainage directions, and 4) perform the upstream integration of drainage area to compute χ.

- Comment

  226 Why the reprojection? Do some algorithms require projected horizontal ground units?

- Response

  The following sentences have been added to clarify why we reproject here.

  > While many of the algorithms implemented in TopoToolbox can work properly in geographic coordinate systems, their implementations have often assumed that a projected coordinate system is used, and we recommend projecting DEMs when possible to avoid problems. Work is ongoing to ensure functions work in geographic coordinate systems and to document when a function assumes a particular coordinate system.

- Comment

  254 This bug is a nice example of the value of metamorphic testing - bravo to the authors for their openness in showing it.

  270 This is great example of interoperability!

  315 a little more explanation would be helpful here. Is the idea that one package projects the data while the other uses geographic coordinates? Or they both project but with different projections?

- Response

  The two packages project into the same coordinate system, but the resulting DEMs are not identical. We have reworded this paragraph to make this clearer. More discussion of this problem is provided in section 4.2.

  > This discrepancy is due to differences in how the two packages transform rasters into the chosen coordinate system. While the `GRIDobj.project` function and the `GridObject.reproject` function are called with the same coordinate reference system and horizontal resolution, the resulting DEMs are not identical. If this reprojection step is bypassed by first transforming the OpenTopography data externally into a projected coordinate system and then loading the projected data into both packages, the computed χ values are identical.

- Comment

  322-324 Although I think I understand the gist here, it's not totally clear to me what 'complementary' means in this context, or why the lack of overlap implies signals beyond 1D. Consider re-wording.

- Response

  We have reworded this couple of sentences:

  > The patterns of width and shear stress illustrated by the swaths in Fig. 6c do not systematically covary, suggesting that the combination of the two provides more information than either alone. The variability in shear stress across the swath illustrates the value of 2D hydrodynamics in capturing signals beyond along-profile variations such as those that can be analyzed with the 1D methods provided by TopoToolbox's `Stream` objects.

- Comment

  362 Not clear what 'original Landlab example' refers to here.

- Response

  A citation has been added here to clarify that this is the notebook by Andy Wickert and Nicole Gasparini in the Landlab documentation.

  > The original example from the Landlab documentation does not use the `DepressionFinderAndRouter` (Wickert and Gasparini, 2026), but it is included to make better comparisons with pytopotoolbox, which currently always resolves sinks when routing flow.

  We have also changed how we introduce the notebook in section 2.3.4 to make it more obvious that this is part of the Landlab documentation:

  > We have created an example that uses TopoToolbox 3 in these two roles within a basic landscape evolution model based on an example available in the Landlab documentation (Wickert and Gasparini, 2026).

- Comment

  404 Great point - needs to happen!

  419 The sentence's subject should really be the developers, not the software itself.

- Response

  Corrected. The sentence now reads

  > We do not practice strict test-driven development (Beck, 2003) where failing tests are added before implementing any new code.

- Comment

  427-9 This sentence confused me at first because it suggests that the hardware and/or OS differences are to blame. But then the next paragraph suggests that it's not this at all but rather seems to be different behavior of rasterio and the MappingToolbox.

- Response

  We have added a more concrete definition of "platform" here to make it clear that by platform we mean not just the hardware and operating system, but also the programming languages and libraries. We have also taken out the reference to the Landlab example as a platform-dependent problem, since differences in that example are really due to using different algorithms. This paragraph now reads:

  > As TopoToolbox 3 has grown beyond MATLAB, we have encountered the problem of maintaining consistency across different computational platforms, the combinations of hardware, operating systems, programming languages, libraries and user code that make up a software application (Bartelheimer et al., 2022). Ideally, the results of an analysis conducted with TopoToolbox 3 would be the same regardless of whether a user conducted the analysis in MATLAB on a Windows PC or in Python on a Mac as the scientific validity of an analysis is undermined if its results depend on the platform used to run it. The χ maps example (Fig. 5) show that, unfortunately, there can be significant differences between the results of analyses implemented on different platforms.

- Comment 449 This reads as if apologizing for a limitation, but I would argue that this is absolutely appropriate. Understanding when/how/where models do or do not approximate the real world is the domain of science, not software development. The key point is that the software enables the science. Further, creating and exploring new models (whether models of processes or of terrain or whatever) is itself a key part of the scientific method. So I don't see the 'unvalidated' nature of software like this as a problem, as long as the scientific status of various algorithms and models (demonstrably accurate vs. provisional best-guess vs. rough approximation with big uncertainties vs. untested speculation, etc.) is clear in the literature and documentation.

- Response

  We agree completely here, and we do not see this as a limitation of TopoToolbox or its testing strategy. We instead want to emphasize to users that automated testing does not absolve them of the scientific duty to validate that algorithms and models apply to their particular problems and to potential contributors to TopoToolbox that they should be thinking about automated tests primarily as ways to verify their software rather than validate it. We have reorganized this paragraph:

  > TopoToolbox's automated test suites are tools for software development, not scientific investigation. They are *verification* tests, which check that the software faithfully implements its specifications, not *validation* tests, which check that the model implemented by the software faithfully represents the real world (Kelly and Sanders, 2008). The automated tests do not replace the essential scientific work of validating the algorithms implemented by TopoToolbox. Because the DEM is only an approximation of the real landscape (Fisher and Tate, 2006; Purinton and Bookhagen, 2017; Schwanghart and Scherler, 2017; Voigtländer et al., 2024), there will always be a limit to how well tests such as those we have illustrated here can validate the assumptions of the algorithms implemented in the software. We instead advocate careful comparison of the outputs of TopoToolbox to independent

data sets and to the outputs of other software and models. Because TopoToolbox 3 can be integrated more tightly with other software, it is now much easier to make these comparisons as illustrated in the Landlab example above. Validating the theoretical behavior of knickpoint migration and comparing the output of the Landlab `FlowAccumulator` with that of the TopoToolbox `TT3FlowRouter` were both essential for correctly implementing the `TT3FlowRouter`. At the same time, the test cases contributed to TopoToolbox do not need to demonstrate scientific validity. An algorithm intended for large-scale DEM analysis might require large datasets and complicated analyses to validate it properly, tests which are hard to develop and costly to run. Property-based tests with small, randomly generated DEMs can verify that the implementation in TopoToolbox works as expected with a lower computational expense and a smaller software development and maintenance burden.

Reviewer #3

- Comment

  Summary:

  This manuscript introduces readers to TopoToolbox3, which is a significant advance over the previous version of the open-source software, in that the fundamental architecture has changed, allowing a new Python-based interface and easier integration with other software. The manuscript is exceptionally well-written with only a few typos. Despite the complexity of the topic, particularly for those not familiar with the software, the platforms, or with coding in general, it is still accessible and clear. I have a few suggestions for inclusions and clarifications, but overall, this is an important manuscript and should be published with minor revisions. I offer a few general comments for the authors to consider below, followed by a few line-by-line and figure comments.

  General comments:

  I find manuscripts like this necessarily awkward – a new tool or database clearly requires a formal, peer-reviewed introduction to the community, along with an establishment of the framework and testing protocols and provision a few useful examples, as is done nicely here. But the authors aren't presenting and interpretating new results, so the traditional background-methods-results-discussion format doesn't work. To that end, the authors might consider reorganizing the methods and results sections and removing these headings. The format could be 2. Architecture and design decisions, 3. Testing and quality assurance, 4. Practical evaluation of TopoToolbox3 (including both methods and results for the X maps, GraphFlood, metamorphic testing, and integration with Landlab), then 5. Discussion.

- Response

  The manuscript was initially organized similarly to this suggestion, and we reorganized it to the current format on the advice of the associate editor. We can see arguments in favor of either of the structures and would appreciate more feedback from the editor before reverting this change.

- Comment

More on organization, I found that the X maps and metamorphic testing where more clearly assessments, whereas the integration with Landlab and GraphFlood were more about showcasing new capabilities. I would reorder the presentation to follow this progression.

- Response

  The examples have been reorganized as suggested, and we have added a summary of the examples to the introduction of this section:

  > The first example demonstrates how functionality existing in the previous version TopoToolbox has been adapted to TopoToolbox 3 by presenting MATLAB and Python implementations of χ maps and evaluating the consistency of results obtained with the two packages. The second example illustrates how a testing strategy known as metamorphic testing was able to catch a subtle bug in libtopotoolbox. The third example presents GraphFlood(Gailleton et al., 2024), a recently developed hydrodynamic model that has been ported to TopoToolbox 3. Finally, the fourth example demonstrates a two-way coupling between TopoToolbox 3 and Landlab that uses TopoToolbox both to implement a process component of a landscape evolution model and to analyze the model's output.

- Comment

  More on the awkwardness of trying to cover all angles in a paper like this – I think that the manuscript serves existing TopoToolbox users well. However, I suspect that many potential new users will start with reading this manuscript but will find it overwhelming and not end up finding their way to the how-to-guides and tutorials, or older publications. I would suggest adding a subheading to the introduction on scope, perhaps starting around and including the material at line 67, adding text to that section on what is NOT in scope, including that this manuscript is not intended as a how-to-guide or starting point, but rather an update on an existing tool. The authors can then include appropriate links for where people SHOULD start. Yes, this appears later in the manuscript, but highlighting it in the introduction will, I think, be useful.

- Response

  We have added a few sentences to the mentioned paragraph:

  > This manuscript focuses on the design decisions and software development practices that make TopoToolbox 3 possible. While we do present practical examples of TopoToolbox 3's use for scientific analyses, this is not intended as a tutorial or instructional guide on TopoToolbox 3, and some understanding of the underlying algorithms and the geoscientific reasoning behind them is assumed. The online documentation available at `https://topotoolbox.github.io` and especially the examples gallery at `https://topotoolbox.github.io/gallery` contain more detailed instructional material, including fully worked versions of the examples presented herein, and we encourage users and potential users of TopoToolbox to look there for guidance and inspiration.

- Comment One pretty significant omission in the manuscript is a consideration of AI. It will be used, that's just the new reality – by new users trying to figure out how to get started, or for

coding or trouble shooting. I would like to see the author's perspective on how to realistically and productively use AI in conjunction with TopoToolbox. It would also be helpful to have the authors including tips for working with AI and test which AI platforms (Claude, CoPilot, ChatGPT) work best.

- Response

  We appreciate that many users will use large language models to interact with TopoToolbox, and some of us have used various models and tools to work with and contribute code to TopoToolbox. We do not, however, have a comprehensive understanding of how well different LLMs work with TopoToolbox or what users can do to improve their performance on TopoToolbox-related tasks. It would be pure speculation on our part to include details on using LLMs with TopoToolbox in this manuscript, and any such speculations are likely to be out-of-date by the time a user encounters this manuscript. We do believe that the existing tests and documentation can help users make use of LLMs more effectively, and we hope that continuing to improve tests and documentation in collaboration with the community will benefit all users, including those using LLMs.

- Comment

  One aspect I think many users are looking for is better integration between TopoToolbox and ArcGIS or QGIS. This gets teased in lines 35-36, and briefly addressed in line 62, but it does not appear later in the manuscript. The authors make it clear that a QGIS integration is in the works, and obviously this manuscript doesn't need to wait for that to get finished. What I would suggest is that the authors add a future work section to the end of the discussion that summarizes ongoing work, including GIS integrations, so that readers who might have missed these brief mentions early in the manuscript won't be left hanging. This is a good opportunity in general for the authors to summarize what other work is planned or already underway, and what is yet to be started but would be good someday. Get people excited about TopoToolbox4, or making their own contributions down the line. #+end$_{\text{quote}}$

- Response

  A clearer description of the opportunities for contributions to TopoToolbox has been added to the conclusion, where there was previously only an exhortation to make contributions:

  > Development of TopoToolbox is ongoing, and we encourage the community to participate in this process. Collaboration occurs through the GitHub issue trackers for each package, which list outstanding tasks and through which contributors can discuss new features and changes to existing ones. As of this writing, development efforts focus on achieving complete feature parity between the Python and MATLAB packages, building out the R and QGIS integrations, prototyping high-performance implementations of TopoToolbox analyses, and expanding and improving the documentation. Counterparts of the `DIVIDEobj` and `SWATHobj`, further elaboration of the point pattern on stream networks object, `PPS`, and more sophisticated and user-friendly visualization tools are excellent opportunities to contribute to the Python package. The R package and integration for QGIS are in early stages of development, and we welcome users of R or QGIS to help guide the design of those tools. Implementations of algorithms that take advantage of hardware acceleration and parallel computing can enable analysis at larger scales, higher resolutions and greater physical fidelity. Documentation and examples of analyses that can be added to the

TopoToolbox Gallery are particularly impactful contributions from the community as they showcase the broad range of applications of TopoToolbox.Above all, TopoToolbox aims to establish a community of practice in quantitative geomorphology: user involvement in the future development of TopoToolbox is crucial to ensure that TopoToolbox evolves to solve the problems that users encounter in their study of the Earth's surface.

- Comment

  Finally, that community contributions piece felt quite unclear to me. This is addressed in lines 416-423, but there it feels like the process is vague and no responsibility is assigned. Which tests serves as a shared repository? The tests are automated? Who integrates them (continuously)? How do contributors make changes? How do they test their changes? How exactly to the tests coevolve with the code? What happens when it is unclear how code should be tested or what properties need to be preserved? Who does the careful code review? A clearer workflow would be helpful and might be more encouraging to community members who are keen to contribute. Perhaps an example could be given.

- Response

  We have added some more detail on the code review process to Section 4.1 (on FAIR principles) and on the automated testing process to Section 2.2 (on Testing and quality assurance). following the suggestion of Reviewer 2. See the responses to Reviewer 2 for the exact changes that have been made.

- Comment

  Specific comments:

  Figure 1: This figure could be cleaned up a little and the color coding could be made more intuitive. For example, perhaps blue for the libtopotoolbox box, and green for pybind11 and MEX, indicating visually that they communicate between the library and MATLAB/Python (by blending blue and green, get it?). I'd love to see different colors for TopoToolbox3 (and a darker shade for the MATLAB related packages) compared to the Python side of the figure (and associated packages). Would it be possible to add workflow arrows or more annotations directly on the figure?

- Response

  This figure has been made into a workflow diagram that illustrates the three main objects and places the various libraries where they get used. The Python and MATLAB sides are clearly delineated. The color-coding has been removed entirely.

- Comment

  Line 153: I caught this example here, but there are likely others. Not all users will automatically know which functions or packages go with MATLAB vs. Python. Is the Parallel Computing Toolbox a MATLAB thing? If so, add "in MATLAB" after "Toolbox." I also found line 145 a difficult to read for similar reasons.

- Response

  We have made the change on line 153 and changed line 145 to

A DEM loaded into memory in MATLAB using the Mapping Toolbox or in Python using rasterio . . .

- Comment

  Line 216: There's something not quite right here. Add "from" between "computed" and "the"?

- Response

  This paragraph has been rewritten following the comments of Reviewer 1, so this error has been removed.

- Comment

  Figure 5: The mismatch between Python- and MATLAB-based X results goes away completely if the raster is reprojected outside of TopoToolbox, right? I didn't get that from the figure and figure capture alone. This information could be added to the caption, or a second figure with the X analysis run on the externally reprojected raster could be added. I guess I just was looking for more advice on how to deal with problems and the authors do seems to have advice here, but it gets lost if one isn't paying enough attention. So, add it to the figure. #+end$_{\text{quote}}$

- Response

  The figure caption has been updated as shown below. Details are also presented in the final paragraph of Section 3.1 and further discussion is in the sixth paragraph of section 4.2 (the Discussion section on Quality assurance).

  > Comparison of χ profiles along the main trunks of the Rhine and Danube computed in both Python and MATLAB using the code in Fig. 2. The χ profiles by Python are shown with the blue solid line and by MATLAB with the dashed orange lines. The discrepancy between the two packages arises from differences in the coordinate transformation applied by Python and MATLAB in Step 2 of Fig. 2. If the DEM is first projected into a coordinate reference system in MATLAB, saved to disk and loaded into Python and then analyzed using Steps 3-6 of Fig. 2, the discrepancy goes away, and the χ profile is identical to the one computed entirely in MATLAB (the dashed orange lines).

- Comment

  Line 337: I think there should be a comma between "ties" and "the"

- Response

  Changed as suggested.

- Comment

  Figure 7: The authors could add counterclockwise and clockwise (right?) arrows to c and d respectively to illustrate more clearly how tie-breaking is done differently.

- Response

  Arrows have been added to the figure and the caption updated.

- Comment

  Line 470: Add a comma after "C".

- Response

  Changed as suggested

Reviewer #4

- Comment

  In this manuscript Kearney et al. present a new version of TopoToolbox, TopoToolbox 3. TopoToolbox has been widely used in quantitative geomorphology for over a decade, and this latest version extends the toolbox into Python, alongside the original MATLAB, and the development of other programming languages. The manuscript provides an overview of the architecture of the software, followed by some examples comparing outputs from Matlab and Python, and some examples of the enhanced interoperability with other Python software such as Landlab. I somewhat agree with Reviewer 3 that the manuscript is awkward, in that it does not present any new findings or interpretations, but requires peer review. I wonder whether the manuscript may have been better suited to a short communication, as the previous version was, or to a more software specific journal (GMD). Having said that, at this stage I don't think it would be appropriate to either significantly modify, or move the manuscript elsewhere. Overall, the manuscript is well written and organised and could be published with some minor revisions.

- Response

  We are, of course, receptive to further guidance from the editorial board on the suitability of the current manuscript to ESurf.

- Comment

  I just have one general comment and a couple of line comments.

  General Comment

  I appreciate that Python has the advantage of not requiring the MATLAB licence, as described on lines 30-35, and has the advantage of being able to be integrated more easily with other programming languages compared to MATLAB. However, is there a sustainability issue with the Python version, in that it is dependent on a number of libraries for which versions of the library may update or become deprecated over time. Do the authors foresee this being an issue down the line, in a way that it is less of an issue for MATLAB users? A small comment on this trade-off between Python and MATLAB might make a welcome addition to the discussion.

- Response

  This is a very important point. We have added a couple of paragraphs to the discussion section on platform dependence:

  > Both the Python and MATLAB versions of TopoToolbox depend on external libraries for some of their functionality. These dependencies can pose a risk to the future stability of both TopoToolbox and to the reproducibility of scientific results generated with TopoToolbox because the libraries may change their behavior.

While this risk is clearly visible in the Python package, where its dependencies are open source packages developed by the community, it does also affect the MATLAB package. One of the major updates required to the MATLAB package during the transition to TopoToolbox 3 involved how georeferencing information is maintained in the `GRIDobj`, `FLOWobj` and `STREAMobj`, which was necessitated by changes in MATLAB's Mapping Toolbox.

In thinking about these risks, we can roughly categorize TopoToolbox's external dependencies into two categories: interface dependencies and computational dependencies. Interface dependencies load information into and export information from TopoToolbox. Libraries like rasterio and the Mapping Toolbox read and write data from storage, and visualization libraries like matplotlib are used to communicate the results of analyses. In principle, changes to interface dependencies should not change the results of an analysis, but only the process by which that analysis is obtained. If matplotlib ceased to exist, the plotting functionality of pytopotoolbox would no longer work, but alternative plotting libraries could still be used with the same `GridObjects`, `FlowObjects` and `StreamObjects` as before, or the data could be exported for visualization in another system. Computational dependencies perform components of the topographic analyses. Libraries like Python's numpy and scipy and MATLAB's Image Processing Toolbox are used primarily as computational dependencies. The reprojection functionality of rasterio and the Mapping Toolbox illustrates that libraries that primarily serve as interface dependencies can also be computational dependencies when they manipulate data while loading it. Changes to computational dependencies may change the results of an analysis, and therefore must be handled with care.

- Comment

  Line Comments:

  Line 216-217: Sentence missing a word - Chi is computed using/ with?

- Response

  This paragraph has been modified on the request of Reviewer #1, so this error has disappeared. See the response to Reviewer #1 for the complete changes.

- Comment

  Line 237: I didn't realise you could use plotdz to plot a chi plot, but I realise now it is very simple using the 'distance', chi, parameter name/value pair for plotdz (at least in matlab). Maybe this was already said in a blog post, but this could be added to the github documentation of plotdz to make it obvious it can be used for chi plots? Very very minor detail and not necessary to this manuscript, but could be useful!

- Response

  The distance argument is documented in the pytopotoolbox online documentation (`https://topotoolbox.github.io/pytopotoolbox/_autosummary/topotoolbox.StreamObject.html#topotoolbox.StreamObject.plotdz`) and in the MATLAB help text for plotdz, but it is not especially clear how to use it. There are also gallery examples (such as the Roan Plateau knickpoint analysis `https://topotoolbox.github.io/gallery/notebooks/python/roan-plateau/roan-plateau.html`) that use custom distances. We can

add some more documentation/examples with plotdz and custom distances and make these examples more visible from the documentation.

- Comment

  Figure 4. There is a straight river at around 4280 East and 2720 west that appears to have different chi values. I think this may be an optical illusion to do with the different coloured basemaps – the matlab has a darker hillshade? Could the basemaps be made identical? Or is this difficult/not possible to achieve due to discrepancies in plotting between python and matlab?

- Response

  The chi values along that stream channel are actually different between MATLAB and Python. This is an effect of the reprojection discrepancy – it just happens to be clearly visible there. We have modified the color scheme to make the two plots more similar – differences in how the hillshade rendering works in Python and MATLAB make it difficult to make them identical.

- Comment

  Section 3.4 and Figure 8 – The chi plots don't collapse perfectly in chi space, which they should do on a perfectly simulated landscape. Do you know whether this is caused by the pixel size, numerical diffusion, or the inclusion of a hillslope term?

- Response

  This seems controlled largely by the high hillslope diffusion coefficient (0.05 m^2yr^{-1}). This was chosen to match the Landlab documentation example upon which this model was based. We have added some text to highlight this.

  > As the balance of erosion against uplift approaches steady state (Fig. 8b), the stream profiles collapse towards one another in χ space. They do not collapse perfectly indicating that a steady state between erosion and uplift has not been fully reached by two million years, which is affected by the relatively high hillslope diffusion coefficient used here (0.05 $m^2 \cdot yr^{-1}$).

Conclusions

References

Arge, L., Chase, J. S., Halpin, P., Toma, L., Vitter, J. S., Urban, D., and Wickremesinghe, R.: Efficient Flow Computation on Massive Grid Terrain Datasets, GeoInformatica, 7, 283–313, https://doi.org/10.1023/a:1025526421410, 2003.

Barnes, R.: Parallel Priority-Flood depression filling for trillion cell digital elevation models on desktops or clusters, Computers & Geosciences, 96, 56–68, https://doi.org/https://doi.org/10.1016/j.cageo.2016.07.001, 2016.

Barnes, R.: Parallel non-divergent flow accumulation for trillion cell digital elevation models on desktops or clusters, Environmental Modelling & Software, 92, 202–212, https://doi.org/10.1016/j.envsoft.2017.02.022, 2017.

Barnhart, K. R., Hutton, E. W. H., Tucker, G. E., Gasparini, N. M., Istanbulluoglu, E., Hobley, D. E. J., Lyons, N. J., Mouchene, M., Nudurupati, S. S., Adams, J. M., and Bandaragoda, C.: Short communication: Landlab v2.0: a software package for Earth surface dynamics, Earth Surface Dynamics, 8, 379–397, https://doi.org/10.5194/esurf-8-379-2020, 2020.

Bartelheimer, C., zur Heiden, P., Lüttenberg, H., and Beverungen, D.: Systematizing the lexicon of platforms in information systems: a data-driven study, Electronic Markets, 32, 375–396, 2022.

Beck, K.: Test-Driven Development: by example, Addison-Wesley, 2003.

Benoit Bovy and Lange, R.: fastscape-lem/fastscape: Release v0.1.0, https://doi.org/10.5281/ZENODO.8375653, 2023.

Dawson, N., Fischer, J., Kuhn, M., Pasotti, A., Rouzaud, D., mhugent, Bruy, A., Sutton, T., Dobias, M., Pellerin, M., Rouault, E., Olaya, V., Blottiere, P., Macho, W., Cabieces, J., Blazek, R., Bartoletti, L., Sherman, G., Sant-anna, H., Woodrow, N., signedav, Natsis, S., rldhont, Felder, J., Shaffer, L., Belgacem, N., Santilli, S., Larosa, S., Mani, S., and Jurgiel, B.: qgis/QGIS: 4.2.2, https://doi.org/10.5281/zenodo.22144220, 2026.

Feathers, M. C.: Working effectively with legacy code, Pearson, 2004.

Fisher, P. F. and Tate, N. J.: Causes and consequences of error in digital elevation models, Progress in Physical Geography: Earth and Environment, 30, 467–489, https://doi.org/10.1191/0309133306pp492ra, 2006.

Gailleton, B., Steer, P., Davy, P., Schwanghart, W., and Bernard, T.: GraphFlood 1.0: an efficient algorithm to approximate 2D hydrodynamics for landscape evolution models, Earth Surface Dynamics, 12, 1295–1313, https://doi.org/10.5194/esurf-12-1295-2024, 2024.

Grohmann, C. H., Lecours, V., and Reuter, H. I.: Overview of software packages used in geomorphometry, in: Geomorphometry, edited by Reuter, H. I., Grohmann, C. H., and Lecours, V., vol. 37 of *Developments in Soil Science*, pp. 371–383, Elsevier, https://doi.org/https://doi.org/10.1016/B978-0-44-333376-7.00021-5, 2026.

Hobley, D. E. J., Adams, J. M., Nudurupati, S. S., Hutton, E. W. H., Gasparini, N. M., Istanbulluoglu, E., and Tucker, G. E.: Creative computing with Landlab: an open-source toolkit for building, coupling, and exploring two-dimensional numerical models of Earth-surface dynamics, Earth Surface Dynamics, 5, 21–46, https://doi.org/10.5194/esurf-5-21-2017, 2017.

Hutton, E., Barnhart, K., Hobley, D., Tucker, G., Nudurupati, S. S., Adams, J., Gasparini, N. M., Shobe, C., Strauch, R., Knuth, J., margauxmouchene, Lyons, N., DavidLitwin, Glade, R., Giuseppecipolla95, Manaster, A., alangston, Thyng, K., and Rengers, F.: landlab/landlab: Mrs. Weasley, https://doi.org/10.5281/zenodo.3776837, 2020.

Kelly, D. and Sanders, R.: The challenge of testing scientific software, in: Conference for the Association of Software Testing, 2008, 2008.

Lai, L. S.-H., Booth, A. M., Duvall, A. R., and Herzig, E.: Short communication: Multiscale topographic complexity analysis with pyTopoComplexity, Earth Surface Dynamics, 13, 417–435, https://doi.org/10.5194/esurf-13-417-2025, 2025.

Lindsay, J.: Whitebox GAT: A case study in geomorphometric analysis, Computers & Geosciences, 95, 75–84, https://doi.org/https://doi.org/10.1016/j.cageo.2016.07.003, 2016.

Mudd, S. M., Clubb, F. J., Grieve, S. W. D., Milodowski, D. T., Hurst, M. D., Gailleton, B., and Valters, D. A.: LSDTopoTools2, https://doi.org/10.5281/zenodo.3245041, 2019.

Perron, J. T. and Royden, L.: An integral approach to bedrock river profile analysis, Earth Surface Processes and Landforms, 38, 570–576, https://doi.org/https://doi.org/10.1002/esp.3302, 2013.

Purinton, B. and Bookhagen, B.: Validation of digital elevation models (DEMs) and comparison of geomorphic metrics on the southern Central Andean Plateau, Earth Surface Dynamics, 5, 211–237, https://doi.org/10.5194/esurf-5-211-2017, 2017.

Rocklin, M.: Dask: parallel computations with blocked algorithms and task scheduling, in: Proceedings of the 14th Python in Science Conference, 2015.

Schwanghart, W. and Scherler, D.: Bumps in river profiles: uncertainty assessment and smoothing using quantile regression techniques, Earth Surface Dynamics, 5, 821–839, https://doi.org/10.5194/esurf-5-821-2017, 2017.

Smith, T., Rheinwalt, A., and Bookhagen, B.: TopoMetricUncertainty - Calculating Topographic Metric Uncertainty and Optimal Grid Resolution, https://doi.org/10.5880/fidgeo.2019.017, 2019.

Swirad, Z. M. and Young, A. P.: CliffDelineaTool v1.2.0: an algorithm for identifying coastal cliff base and top positions, Geoscientific Model Development, 15, 1499–1512, https://doi.org/10.5194/gmd-15-1499-2022, 2022.

The MathWorks Inc.: MATLAB, `https://www.mathworks.com`, 2022.

Voigtländer, A., Rheinwalt, A., and Tofelde, S.: Quantifying Earth's Topography: Steeper and Larger Than Projected in Digital Terrain Models, Geophysical Research Letters, 51, e2024GL109 517, https://doi.org/https://doi.org/10.1029/2024GL109517, e2024GL109517 2024GL109517, 2024.

Wickert, A. and Gasparini, N.: Modeling Hillslopes and Channels with Landlab, `https://landlab.csdms.io/teaching/geomorphology_exercises/drainage_density_notebooks/drainage_density_class_notebook.html`, accessed 2026-03-05, 2026.

Willett, S. D., McCoy, S. W., Perron, J. T., Goren, L., and Chen, C.-Y.: Dynamic Reorganization of River Basins, Science, 343, 1248 765, https://doi.org/10.1126/science.1248765, 2014.