Journal of Visualization and Interaction
JoVI review snapshot created at Feb 02, 2026 13:05.

Issue #14 (open): [REVIEW] Review 2

@RaphaelWimmer on Aug 17, 2025 10:12: [opened]

@RaphaelWimmer on Aug 17, 2025 10:12:

Conflicts of interest

- ☒ I declare that I have no known conflicts of interest with the authors.

Reviewed version

c09c15d

Review

This submission presents an approach for rendering and optionally running N-body simulations (i.e. simulations of thousands of particles that are affected by gravity) as WASM code within a browser.

My usual criteria in a review are: a scientific paper should be original, easy to understand, correct, reproducible, and relevant for researchers or practitioners.

Originality

I have no doubt that this submission is original work created by the author (just for completeness' sake). It builds upon earlier work by the same author. The new contribution is clearly outlined. I am not aware of other published works describing very similar ideas. While WASM has been around for some time, it seems to be rarely used for visualizations.

Writing and Illustrations

Overall, the paper is well written and easy to follow. There are still a few typos (see below).

The figures, especially the live simulation, help understanding key concepts. However, I would suggest also embedding a screenshot of the simulation (e.g., side-by-side) as a fallback option for e.g. PDF output. (1)

My main concern with the writing is that it does not go into sufficient detail, as discussed below. At times, the writing sounds more like a promotional text or blog post than a scientific paper. I would suggest toning down a few claims and/or supporting them a little bit better. (2)

I found this sentence confusing: *"A user can start an interactive visualization of a simulation within a Jupyter notebook by simply calling the .widget() function on any simulation object."* My understanding is that the Python bindings usually call native code compiled for the desktop platform. Where does the WASM code for the widget come from? (3)

Correctness

The author has compiled multiple examples to WASM and embedded them in the official REBOUND documentation. I have tried out a few of them and can confirm that they work as intended. I have not

tried out the hybrid mode so far but assume that it works as described.

However, the paper makes several claims that should be supported by empirical evidence. Most importantly: *"Although the performance is not (yet) on par with native desktop implementations, it comes very close and is sufficient for most applications."*

I would have expected at least some simple performance metrics in this paper (e.g. frames per second for the three different modes for two simulations (one with complex visualization, one without) running on the same hardware. Such a comparison would not only strengthen the authors argument - it would be very helpful for readers trying to find out whether (hybrid) WASM rendering might support their own use cases. (4)

As the author describes multiple advantages of the WASM approach, I would also have expected a short discussion of the potential drawbacks of the WASM approach compared to traditional JS frameworks. For example, WASM probably makes it harder for the user to inspect the code and for the browser to add accessibility features. If I understand correctly, text that is rendered via WebGL can not be selected by the user. (5)

Reproducibility

The full source code of REBOUND is available. The Makefiles in the `examples` directory contain the specific calls to emscripten. This makes it possible for others to implement their own solutions based on this code.

However, to the best of my knowledge, the REBOUND GitHub repo is not part of the submission. As repos may change, move, and disappear, I would strongly suggest adding a copy of the (relevant) code as supplementary files. (6)

I also would have expected a high-level overview of how REBOUND supports all three modes. Ideally, the paper alone should contain enough information for me to write my own implementation. At the moment, there is nearly no information on how to compile the same code to a native executable and WASM. For example, `display.c` contains some `#IFDEF emscripten` preprocessor directives that e.g., select different GLSL versions depending on output. This seems to be necessary for some reason - but it is never mentioned in the paper. I think that readers might benefit from a list of changes they need to consider. (7)

Relevance

Many of the issues mentioned above directly affect the practical relevance of this submission. In general, I think that the approach described here is quite interesting for developers who want to bring native, OpenGL-based applications to the web. However, in its current form, the submission very much focuses on REBOUND and gives very little general guidance. It demonstrates convincingly *that* it is possible to build native/web/hybrid visualizations - but it does not really explain *how* to do this in practice.

I think that its practical relevance might be significantly improved by providing a performance comparison and a more concrete description of how support for the three modes is implemented in the code base.

Minor issues

- "entry barriers" -> "barriers to entry" (Abstract)
- "for example (Williams et al. 2022) describe" -> for example Williams et al. (2022) describe" (though that might be an issue with the template?).
- "one can run a simulations" â†’ "simulation"
- " Figure 1 show" â†’ "shows"
- "~[REBOUND also comes in the form of a pre-compiled Python package in which case no compiler is needed.]" should probably be a footnote?!
- "this does not required" -> "require"
- "integrated with the N-body code REBOUND." - what does this mean? Any way to write it more clearly?
- The authors claim that a screenshot of Figure 2 takes up 200 kB. Obviously, this depends on the resolution - on my computer the screenshot is 63 kB in size. I'd suggest adding "at full-HD resolution" or similar.

- "a cross-platform graphics or user interface (UI) library such as QT, Unity, OpenGL, or Vulkan" - I'd argue that none of these are really "libraries". OpenGL and especially Vulkan are rather APIs, and that Qt and especially Unity are rather toolkits or frameworks. Maybe rephrase as "cross-platform libraries, frameworks, or APIs, such as ..."

Openness/Transparency

(already discussed in the review; I would suggest adding the code as supplementary files)

Submission categories

- ☐ Registered Report
- ☐ Replication Study
- ☐ Empirical Research - Quantitative
- ☐ Empirical Research - Qualitative
- ☒ Systems or design research
- ☐ Commentary
- ☐ Systematic Literature Review

Suggested outcome

Major revisions: this paper requires substantial improvements that I will need to re-review to decide whether or not to endorse it.

Requested changes

Suggestions mentioned in the review:

1. Also embed a static image preview of the iframe contents for Figure 2.
2. Rephrase assumptions that are not supported by further evidence, e.g., "it is likely" â†’ we find it likely".
3. Explain in a little bit more detail how the Jupyter Widget works.
4. Quantitatively compare performance of the three different modes.
5. Discuss potential drawbacks of the WASM approach.
6. Add code as supplementary files.
7. Describe the most relevant changes required for the WebAssembly/WebGL output in more detail.

I think that especially issues 4 and 7 need to be addressed in order to increase the practical relevance of the submission.

ORCID

https://orcid.org/0000-0001-5162-5113

@RaphaelWimmer on Aug 17, 2025 10:14:

(I'd like to apologize for the late response, especially given the amount of changes I suggest.)

@floe on Aug 18, 2025 06:26:

Many thanks for the review, @RaphaelWimmer!

@hannorein, off the top of my head, I feel like none of the requested changes are too extensive - I'm aware that this paper has been in the loop for a very long time, but I still think it would be worth addressing these? What do you think?

@hannorein on Sep 06, 2025 21:48:

I just wanted to give a short update: I very much appreciate @RaphaelWimmer's comments and suggestions! Thank you.

I am currently working on the revisions but will need a bit more time as some of them are a bit more substantial.

@hannorein on Sep 17, 2025 14:13:

Hello @floe and @RaphaelWimmer!

Thank you again for the helpful comments and suggestions. I have now implemented all the requested changes. Summarized below. I hope this message counts as a "resubmission", but let me know if you want me to do anything else.

Hanno

Requested changes:

1. **Also embed a static image preview of the iframe contents for Figure 2.** I have added a static image for when the paper is rendered to a non-HTML format. In that case a link is provided to the HTML version in the static figure's caption.
2. **Rephrase assumptions that are not supported by further evidence, e.g., "it is likely" â†’ we find it likely".** I have tried to adjust this throughout the paper.
3. **Explain in a little bit more detail how the Jupyter Widget works.** I have expanded this. See Section 6. But it really is that simple (~2 lines of code).
4. **Quantitatively compare performance of the three different modes.** I have done some performance tests, added them to the paper, and included some discussion. See Section 4.
5. **Discuss potential drawbacks of the WASM approach.** I have added a list of drawbacks at the end of Section 3.
6. **Add code as supplementary files.** I have added links to the Zenodo repository of REBOUND. This stores snapshots of the public releases, hopefully in a more permanent and scientific way (each release comes with a DOI) than GitHub.
7. **Describe the most relevant changes required for the WebAssembly/WebGL output in more detail.** I have added a new section (Section 7) on the changes made to the existing OpenGL code. I have written it as a step-by-step guide to help others who might want to follow the ideas outlines in this paper.

All the minor issues have been dealt with as well.

@hannorein on Oct 14, 2025 14:15:

Hello @floe and @RaphaelWimmer. It's been a couple of weeks, so I thought I'd quickly ping you. What are the next steps? Is there anything else you would like me to change? Thanks!

@floe on Oct 15, 2025 11:32:

Hello @hannorein - from my point of view, this is completed. I'll leave the issue open, though, as it will be easier to find alongside the published article later.

@floe on Oct 15, 2025 11:33:

P.S. We are still missing a third reviewer, unfortunately, as nobody from the astronomy community that I've contacted ever replied to me ðŸ˜• I will probably fill in as final reviewer myself, as a stopgap measure ðŸ¤·

@hannorein on Oct 15, 2025 12:45:

Thanks. I can send you suggestions, but those would probably be biased towards me.

@floe on Oct 18, 2025 09:01:

@hannorein as long as you're not in a conflict of interest with those persons (as per our guidelines), then suggestions are very welcome!

@hannorein on Oct 18, 2025 13:01:

@floe Sent you a few suggestions by e-mail!