Journal of Visualization and Interaction

JoVI review snapshot created at Feb 02, 2026 13:05.

Issue #12 (open): [REVIEW] 0-dependency REBOUND visualizations

@sharponlooker on May 07, 2025 19:08: [opened]

@sharponlooker on May 07, 2025 19:08:

Conflicts of interest

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

Reviewed version

6d40ae2

Review

This paper's claims are spot-on regarding visualization issues with dependency complexity and lifecycle management. Its web standards-based proposed solution is extremely adequate for solving the mentioned issues and beyond, with good chances for it to be extrapolated to additional use cases within the N-body simulation field as well as many other scientific visualizations.

As a longtime follower of, and occasional user of REBOUND, I have personally confronted some of the dependency issues presented in the paper. Every action taken at the package level to reduce this complexity improves the user experience.

In order to validate the methods and claims, I reran my most common use case for REBOUND, simple customized Solar system orbital plots, and it was exciting to be able to fully experience the 3D-visualization in a short while, with just a few hurdles unrelated to REBOUND as such. For the sake of completeness: my test was a "hybrid mode" run of a rendering of Sednoid orbits, the simulation and web server threads running on a Linux docker instance running ipython, the web browser rendering happening on an MS Windows host.

I have only a few comments about the paper, and except for number 1, they can be considered minor or improvement suggestions and should be weighed against affecting the paper's conciseness:

1 - I understand the reasoning behind the homebrew minimalist webserver and the benefits it provides, but it should still be considered a possible attack vector, especially in the "hybrid mode" where it may be being exposed on any kind of network. I believe it would suffice with warnings to users that this component would need to be secured in use cases and setups that may exceed the scope expected by the author(s).

2 - I would like to posit that the claim that this is a zero-dependency solution is reliant on WebAssembly truly remaining as a unique standard in the long run and its continued adoption and support by browsers. For instance, this manifests itself in the list of advantages at the end of the Introduction section, where the last bullet (future-proof) uses the word "likely".

3 - In section 2, after the list of operation modes, the following paragraph mentions the use of a C compiler in a context that makes it sound like a mandatory step for every use case. As suggested in later sections, the availability of a Python interface makes that a non-issue in other situations.

4 - Both in section 3 ("we can reuse our existing C code that uses OpenGL for rendering **with minimal changes_** ***") and section 6 ("We were able to reuse _*** **the majority** of the rendering code") hint at some

tweaks that had to be applied to existing codebases. Is it possible to have a footnote or short appendix summarizing what kind of changes were needed?

5 - It would be interesting to have a brief paragraph about a roadmap or future planned improvements, if there are any already, for the paper's proposed solution. Even perhaps a commentary on whether the solution already applies to REBOUND+ (if I have understood it right, REBOUND+ augments simulations with non-gravitational forces).

Very minor issues & typos

6 - Figure 1, I think the operating modes should be labeled in the graphic itself as well, so that the figure is self-contained even without the caption.

7 - Figure 1, it may be personal taste, but the user/client layer is usually at the top and for me the REBOUND simulation thread would better appear like a backend engine at the bottom. I would not mind if the whole figure is inverted upside down.

8 - Typo section 1, bullet 1, "tailor" should be "tailored".

9 - Typo section 2, above fig 2, "show case" should be "showcase".

10 - Typo section 4, third paragraph, "back end" should be "backend".

11 - Typo section 4, last paragraph, "facility" should be "facilitate".

12 - Section 5, "with python interface" would read better as "with a python interface".

Openness/Transparency

No issues

Submission categories

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

Suggested outcome

Endorse: I am willing to endorse this paper, with at most minor copyediting.

Requested changes

Issue 1, 6 and 8-12

ORCID

No response

@floe on
Jun 06, 2025 09:47:

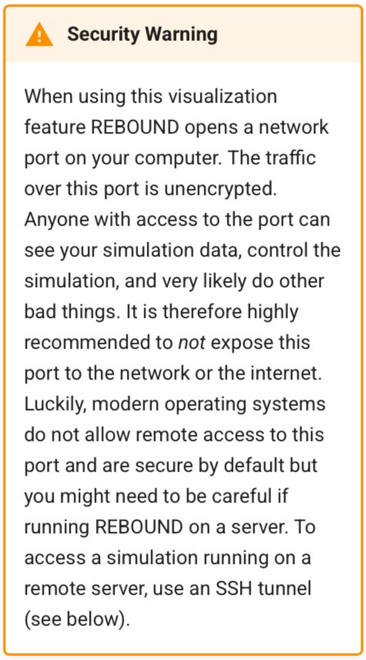
Many thanks for your review, @sharponlooker ! Once the other reviews are in, I will compile a meta-review (but I can already invite @hannorein to have a look at your comments and see if they can be easily addressed, most of them probably can).

@hannorein on
Jul 14, 2025 14:44:

Thank you @sharponlooker for the nice review! I have addressed all the issues you brought up. Specifically:

> 1 - I understand the reasoning behind the homebrewn minimalist webserver and the benefits it provides, but it should still be considered a possible attack vector, especially in the "hybrid mode" where it may be being exposed on any kind of network. I believe it would suffice with warnings to users that this component would need to be secured in use cases and setups that may exceed the scope expected by the author(s).

There was already a warning in the documentation. But I've made it more prominent with a yellow annotation:



⚠ Security Warning

When using this visualization feature REBOUND opens a network port on your computer. The traffic over this port is unencrypted. Anyone with access to the port can see your simulation data, control the simulation, and very likely do other bad things. It is therefore highly recommended to *not* expose this port to the network or the internet. Luckily, modern operating systems do not allow remote access to this port and are secure by default but you might need to be careful if running REBOUND on a server. To access a simulation running on a remote server, use an SSH tunnel (see below).

I've also added a "not secure" warning to the console output whenever a port has been opened.

In practice, I don't think this is an issue. These days, almost all operating systems block access to any host other than localhost by default.

> 2 - I would like to posit that the claim that this is a zero-dependency solution is reliant on WebAssembly truly remaining as a unique standard in the long run and its continued adoption and support by browsers. For instance, this manifests itself in the list of advantages at the end of the Introduction section, where the last bullet (future-proof) uses the word "likely".

I have added a footnote clarifying that a WebAssembly compiler is needed.

> 3 - In section 2, after the list of operation modes, the following paragraph mentions the use of a C compiler in a context that makes it sound like a mandatory step for every use case. As suggested in later sections, the availability of a Python interface makes that a non-issue in other situations.

I have added another footnote clarifying that a compiler is not needed for the precompiled python package.

> 4 - Both in section 3 ("we can reuse our existing C code that uses OpenGL for rendering with minimal changes") and section 6 ("We were able to reuse the majority of the rendering code") hint at some tweaks that had to be applied to existing codebases. Is it possible to have a footnote or short appendix summarizing what kind of changes were needed?

Good point. The main things I had to change was the run loop (i.e. how and when new frames are requested) and how text is rendered (it makes sense to not render text via OpenGL but let the browser do it). I've added some details.

> 5 - It would be interesting to have a brief paragraph about a roadmap or future planned improvements, if there are any already, for the paper's proposed solution. Even perhaps a commentary on whether the solution already applies to REBOUND+ (if I have understood it right, REBOUND+ augments simulations with non-gravitational forces).

Also a good point. The main feature I would like to see added in the future is a port of the visualization to pyodide. This way one could run everything from within the browser interactively. I've added a section at the end discussion this and some of the challenges associated with it.

> Very minor issues & typos 6 - Figure 1, I think the operating modes should be labeled in the graphic itself as well, so that the figure is self-contained even without the caption.

This has been changed.

> 7 - Figure 1, it may be personal taste, but the user/client layer is usually at the top and for me the REBOUND simulation thread would better appear like a backend engine at the bottom. I would not mind if the whole figure is inverted upside down.

I have decided not to flip the chart.

All the other typos have been corrected.

@floe, please let me know if there is anything else I should do to address the review.

Thank you to both of you!

@sharponlooker on
Jul 31, 2025 18:17:

Sorry for the delay, your reply came in the midst of my vacation trip ;-) I am satisfied with the changes and additions and endorse the paper in its current version. I suggested a minor legibility improvement to one of the changes in the form of a pull request.

@hannorein on
Sep 06, 2025 21:30:

Completed with merge of #13.

@hannorein on
Sep 06, 2025 21:30:

[closed]

@floe on
Oct 29, 2025 14:52:

Reopening to keep reviews visible on main page.