



EDITORIAL

*Reviews and responses for*

**impunity: Enforcing Physical Unit Consistency at Definition Time in Python**

**Authors:** Antoine Chevrot and Xavier Olive 

**Reviewers:** Enrico Spinielli , Junzi Sun 

**Editor:** Martin Strohmeier 

DOI for the original paper: <https://doi.org/10.59490/joas.2023.7071>

**1. Review - round 1**

**1.1 Reviewer 1**

The paper addresses a highly significant topic and proposes an elegant, non-intrusive, and efficient solution. As detailed in the "Related Works" section, other researchers have explored the challenge of unit measurements in various languages, times, and manners. The paper is sound, well-structured, and engaging.

One point that requires clarification, in my opinion, is the statement on line 87 about the risk of floating-point precision errors. I fail to see why this issue is specific to the "template-based method" rather than a general concern in all computations.

Now, for a few typo and word changes:

\* line 111: Suppose ==> Given

\* line 164: assignation ==> assignment

Lastly, some aesthetic (not crucial, but nice-to-have) observations: the paper inconsistently uses monotype and normal font to indicate various libraries and objects (e.g., impunity, mupy, pylance, Annotated). I recommend using monotype font consistently for Python or other libraries and relevant components/objects.

**1.2 Reviewer 2**

This paper explains an open-source Python library that performs unit conversion with minimal overhead. The paper provides a comprehensive description of the background, implementation, and examples of the impunity Python library. It is evident that the authors possess deep expertise in Python and its abstract syntax tree. The paper is clear and easy to read.

The following are my minor suggestions to enhance readability:

\* I suggest revising the title to "Impunity: Enforcing Physical Unit Consistency in Python Efficiently."

\* Keywords: please add "Python."

- \* Elaborate a little more on the relevance to aviation research in Section 1.
- \* The code between lines 86 and 87 overflows.
- \* At the beginning of line 250, an extra dash appears, which also seems to be a URL. Please verify. (By the way, this might be a JOAS template issue.)
- \* To better distinguish between text and code/figures, please add shading to the code blocks. (I plan to update the JOAS template for future papers.)
- \* When mentioning the impunity library in the text, ensure that it appears as `impunity`, like on line 224. This improves readability.
- \* Regarding visuals, please consider arranging Figures 2 and 4 together.

Overall, the paper is well-crafted. I particularly appreciate the clean coding style showcased on GitHub, which can inspire future researchers.

I recommend a minor revision of the paper.

## 2. Response - round 1

Thank you very much for your comments.

- \* Regarding the aviation-specific issue, I plan to cite the Air Canada Boeing 747 crash-landing, which occurred due to a miscalculation of fuel loaded onto the aircraft. The error partially stemmed from the recent adoption of the metric system by the company ([https://en.wikipedia.org/wiki/Gimli\\_Glider](https://en.wikipedia.org/wiki/Gimli_Glider)). Though somewhat dated, the example remains pertinent.
- \* About the floating-point precision error, I propose to modify lines 84-86 from:

"The tricky part .... follows:"

to:

"A notable caveat exclusive to the template-based conversion method lies in its susceptibility to floating-point precision errors arising from unnecessary conversions to the SI system. This contrasts with other methods that employ compatibility checks to avoid such conversions when unnecessary."

Would this modification suffice for clarification?

All other comments are addressed.

### 2.1 Follow-up

I believe the suggested changes for lines 84-86 effectively clarify the text. As for the aviation-related citation, I have a preference for "on-board" examples but am satisfied with your proposal; the pool of relevant examples is indeed limited.

Thank you for your suggestion.