The Journal of Open Source Software

PyUncertainNumber for uncertainty propagation: more than just probability arithmetic

Yu Chen [1¶], **Scott Ferson** [1], and **Edoardo Patelli** [2]

1 Institute for Risk and Uncertainty, University of Liverpool, UK **2** Centre for Intelligent Infrastructure, University of Strathclyde ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

The forces on stars, galaxies, and dark matter under external gravitational fields lead to the dynamical evolution of structures in the universe. The orbits of these bodies are therefore key to understanding the formation, history, and future state of galaxies. The field of "galactic dynamics," which aims to model the gravitating components of galaxies to study their structure and evolution, is now well-established, commonly taught, and frequently used in astronomy. Aside from toy problems and demonstrations, the majority of problems require efficient numerical tools, many of which require the same base code (e.g., for performing numerical orbit integration).

Statement of need

`Gala` is an Astropy-affiliated Python package for galactic dynamics. Python enables wrapping low-level languages (e.g., C) for speed without losing flexibility or ease-of-use in the user-interface. The API for `Gala` was designed to provide a class-based and user-friendly interface to fast (C or Cython-optimized) implementations of common operations such as gravitational potential and force evaluation, orbit integration, dynamical transformations, and chaos indicators for nonlinear dynamics. `Gala` also relies heavily on and interfaces well with the implementations of physical units and astronomical coordinate systems in the `Astropy` package (Astropy Collaboration, 2013) (`astropy.units` and `astropy.coordinates`).

`Gala` was designed to be used by both astronomical researchers and by students in courses on gravitational dynamics or astronomy. It has already been used in a number of scientific publications (Pearson et al., 2017) and has also been used in graduate courses on Galactic dynamics to, e.g., provide interactive visualizations of textbook material (Binney & Tremaine, 2008). The combination of speed, design, and support for Astropy functionality in `Gala` will enable exciting scientific explorations of forthcoming data releases from the *Gaia* mission (Gaia Collaboration, 2016) by students and experts alike.

`pyuncertainnumber` enables rigorous uncertainty analysis for real-world situations of mixed uncertainties and partial knowledge. Aleatoric and epistemic uncertainties are recognised and treated appropriately in characterisation and propagation.

Uncertainty arithmetic is underpinned by probability bounds analysis. While it has the potential to automatically compile a non-deterministic subroutines via primitives such as intervals or uncertain numbers, its usages face several challenges.

Besides the issues of xx such as dependency problems, one notable challenge is that code accessibility is often not guaranteed. Also, the lack of capability one the main reasons restricting the adoption of xxx in practice.

The Journal of Open Source Software

pyuncertainnumber addresses that by enabling non-intrusive capability. How to work with black-box models? This capability significantly boost its versatility for scientific computations by interfacing with many engineering softwares.

Interval propagation in a non-intrusive manner

Interval analysis has the advantages of providing rigorous enclosures of the solutions to problems, especially for engineering problems subject to epistemic uncertainty, such as modelling system paramters due to lack-of-knowledge or characterising measurement incertitude. It is evident that computational tasks requiring complex numerical solutions of intervals are non-intrusive (i.e. the source code is not accessable). Besides, it shoule be noted even for cystal boxes (i.e. source code is accessible), naive interval arithmetic still faces challlenges such as the infamous interval dependency issue. Though it may be mitigated through mathematical rearrangements in some cases, it will be challenging for most of the cases.

Generally, the interval propagation problem can be cast as an optimisation problem where the minimum and maximum are sought via a function mapping. The functio, for example g in Eq.(xx), is not necessarily monotonic or linear and may well be a black-box model. Hence, for black box models the optimisation can only be solved via gradient-free optimisation techniques.

$$Y = g(I_{x1}, I_{x2}, ..., I_{xn}) \tag{1}$$

$$Y_m in, Y_m ax \tag{2}$$

where $I_{x1}, I_{x2}, ..., I_{xn}$ are intervals.

pyuncertainnumber provides a series of non-intrusive methodologies of varying applicability. It should be noted that there is generally a trade-off between applicability and efficiency. But with more knowledge about the characteristics of the underlying function, one can accordinly dispatch an efficient method. For example, whem monotonicity is known one can use vertex methods which 2_n.

Table 1: Several methods for interval propagation

Method	End-points	Subinterval reconstitution	Cauthy-Deviate method	Bayesian optimisation	Genetic algorithm
Assumption Result	monotonicity	heavy computation	linearity and gradient required	No	No

As shown in **??**, tabulation of xxx given a black box model.

Mixed uncertainty propagation for black-box models

Most realistic situation bla bla. Imprecise world bla bla. After faithful characterisation, the ability to propagate is the key in many critical engineering applications.

Dependency structures bla bla. It has been echoed in the engineering applications and also the NASA challenge.

Sampling methods play a significant role in xxx

Double Monte Carlo

Interval Monte Carlo…

Propagation of p-boxes via surrogate models

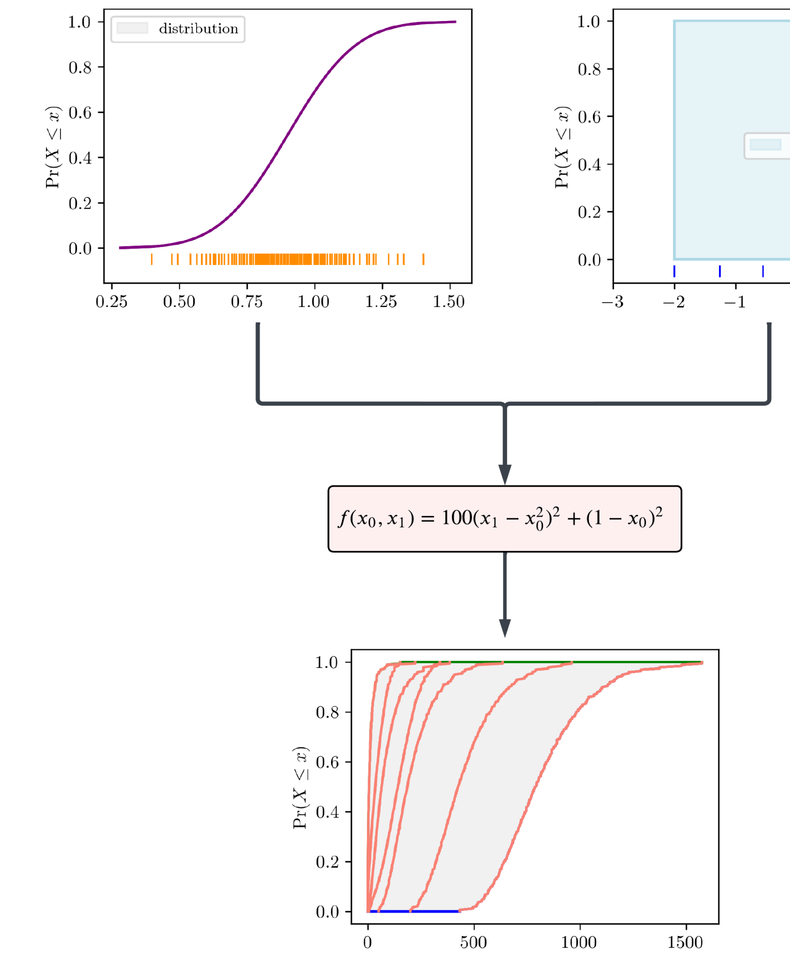
Citations

Citations to entries in paper.bib should be in rMarkdown format.

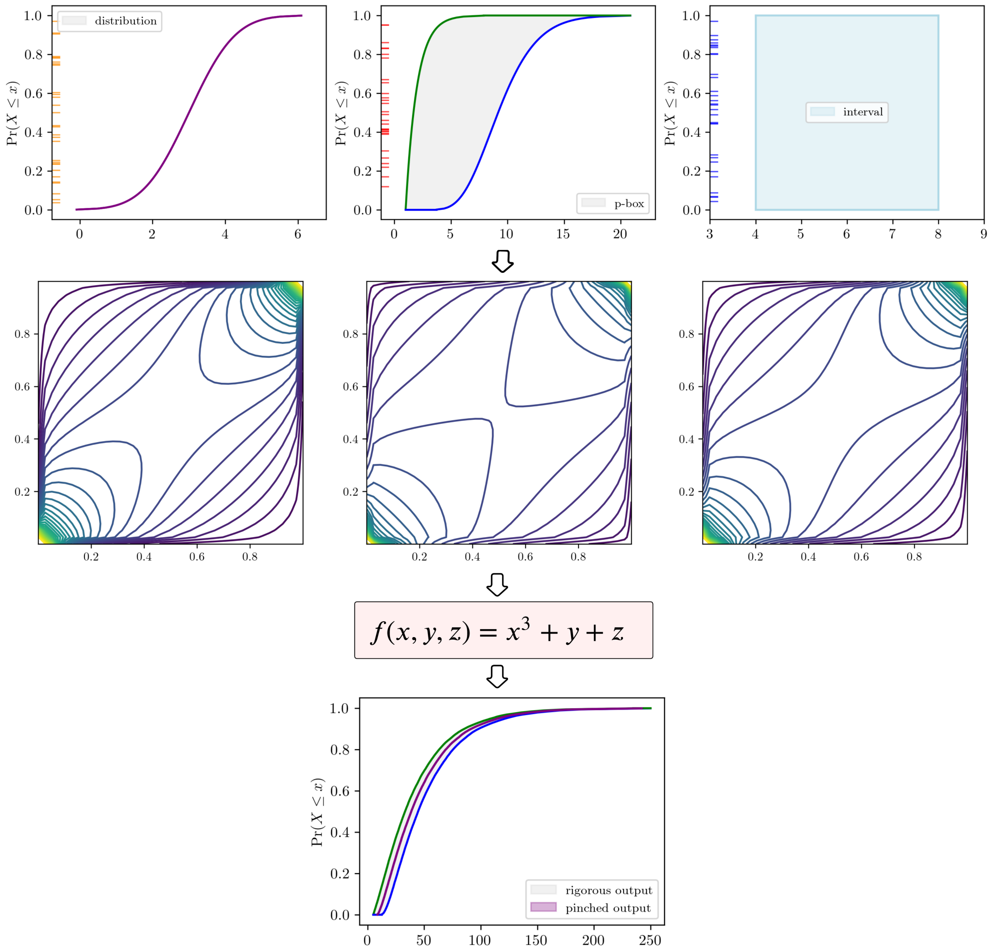
If you want to cite a software repository URL (e.g. something on GitHub without a preferred citation) then you can do it with the example BibTeX entry below for Smith et al. (2020).

For a quick reference, the following citation commands can be used: - `@author:2001` -> "Author et al. (2001)" - `[@author:2001]` -> "(Author et al., 2001)" - `[@author1:2001; @author2:2001]` -> "(Author1 et al., 2001; Author2 et al., 2002)"

Figures



Figures can be included like this:
?? illustrates the *nested Monte Carlo* method.



?? illustrates the *interval Monte Carlo* method.

Conclusion

Significance: this provides compatability as interfacing with many engineering applications. boost its usage.

Acknowledgements

We acknowledge contributions from Brigitta Sipocz, Syrtis Major, and Semyeong Oh, and support from Kathryn Johnston during the genesis of this project.

References

Astropy Collaboration. (2013). Astropy: A community Python package for astronomy. *Astronomy and Astrophysics*, *558*. https://doi.org/10.1051/0004-6361/201322068

Binney, J., & Tremaine, S. (2008). *Galactic Dynamics: Second Edition*. Princeton University Press. http://adsabs.harvard.edu/abs/2008gady.book.....B

Gaia Collaboration. (2016). The Gaia mission. *Astronomy and Astrophysics*, *595*. https://doi.org/10.1051/0004-6361/201629272

Pearson, S., Price-Whelan, A. M., & Johnston, K. V. (2017). Gaps in Globular Cluster Streams: Pal 5 and the Galactic Bar. *ArXiv e-Prints*. http://adsabs.harvard.edu/abs/2017arXiv170304627P

Smith, A. M., Thaney, K., & Hahnel, M. (2020). Fidgit: An ungodly union of GitHub and figshare. In *GitHub repository*. GitHub. https://github.com/arfon/fidgit