The Journal of Open Source Software

PyUncertainNumber for uncertainty propagation: more than 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Published: unpublished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

Scientific computations or simulations play a central role in quantifying the performance, reliability, and safety of complex engineered systems. However, these analyses are complicated by the various sources of uncertainties inherent in the computational pipeline. Underestimation may lead to suboptimal performance outside the most common scenarios while overestimation, on the other hand, may lead to over-engineered systems and significant waste of resources. To ensure that complex engineered systems can be operated reliably and robustly, even during rare and extreme environment conditions, a comprehensive analysis is required. The analysis should be comprehensive in two senses: (i) all of the possible sources of uncertainty must be identified and represented using approaite mathematical construct; (ii) that rigorously account for mixed or mixture of various types of uncertainties. Challenges include xxx, code accessbility, tools to conduct the analysis. By xxx, `pyuncertainnumber` bla bla.. non-intrusively.

Statement of need

A comprehensive uncertainty framework for scientific computation involves a mathematical model, through which various input uncertainties are propagated to estimate the uncertainty of an unknown quantity of interest. In real-world applications, these input uncertainties are commonly manifested as mixed uncertainties, e.g. probability boxes (p-boxes) which effectively represents a set of distributionbs, combining both the aleatory and epistemic uncertainty in one structure, or a mixture of uncertainties suggesting, for instance, a vector of inputs parameters of aleatory (e.g. probability distributions), epistemic (e.g. intervals), and mixed nature (e.g. probability boxes).

Probability bounds analysis is one of the expressive frameworks proposed to manage uncertainties in an imprecise setting. Packages have been developed to facilitate the calculations of uncertain quantities, such as interval arithmetic (Angelis, 2022) and probability arithemetic (A. Gray et al., 2021; N. Gray et al., 2022). Collectively, they can be referred to as *uncertainty arithmetic* which straightforwardly computes the response provided the performance function.

While it has the potential to automatically compile non-deterministic subroutines via uncertain primitives, its usages face several challenges. Besides the known issues such as dependency problems, one significant challenge is that code accessibility is often not guaranteed and hence unable to proceed. This would largely restrict the adoption of mixed uncertainty calculations in engineering practice.

`pyuncertainnumber` addresses that by enabling non-intrusive capability. That is, generic black-box models can be propagated with (that fancy word) various types of uncertainty. 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 accessiable). 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

Mixed uncertainty problem is the most realistic situation bla bla. It first requires faithful characterisation of uncertainty given the empirical information, and the approaach to rigorously progate them. Imprecise world bla bla. After faithful characterisation, the ability to propagate is the key in many critical engineering applications.

$$Y = f(\mathbf{u}; C) \tag{3}$$

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

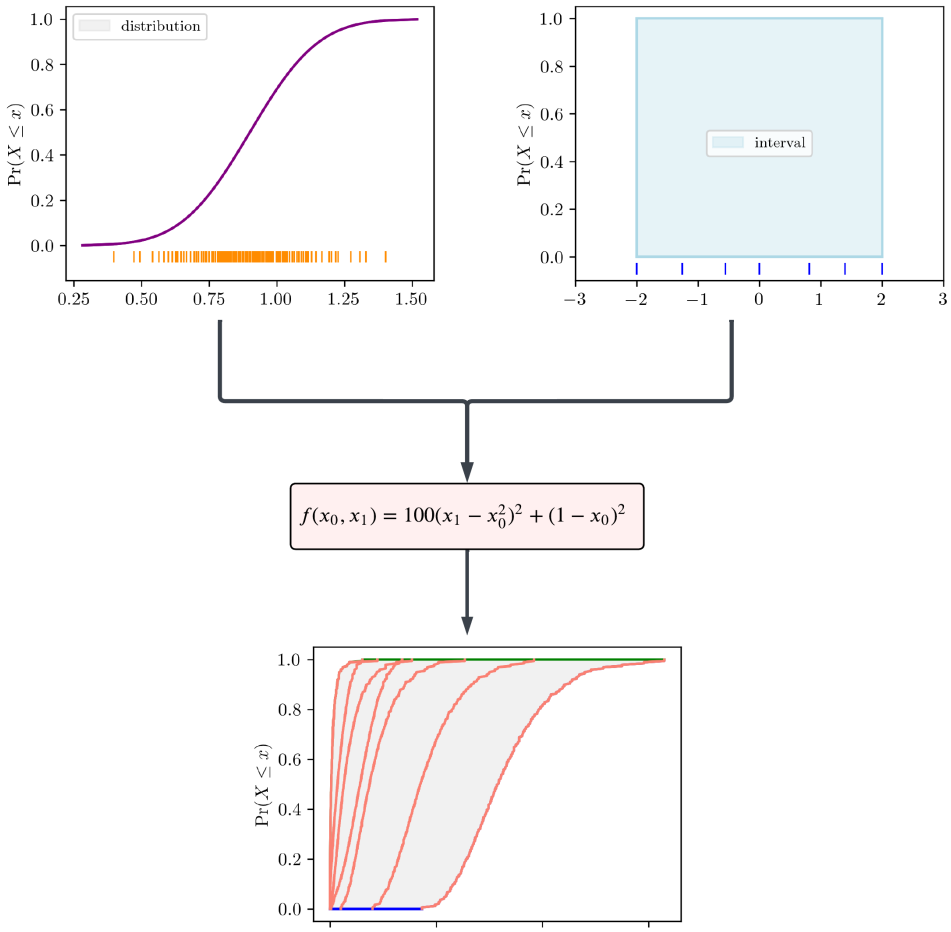
Sampling methods play a significant role in xxx

The Journal of Open Source Software

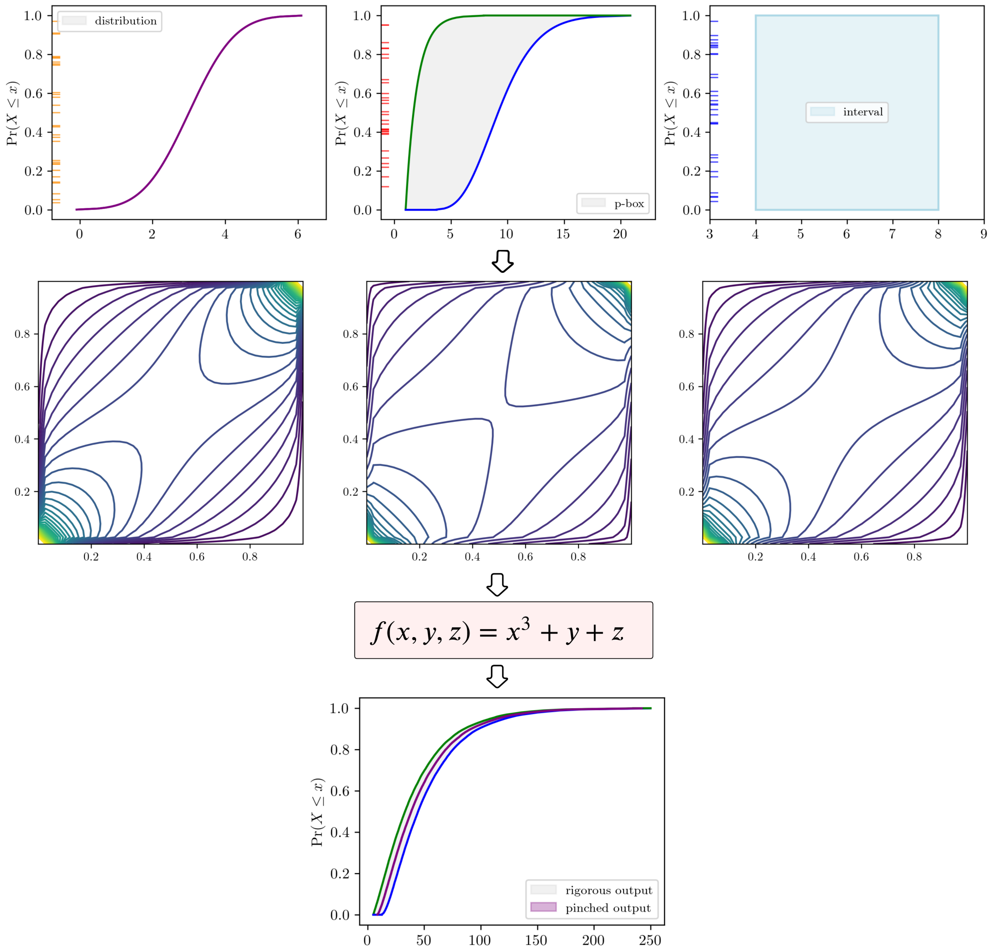
Double Monte Carlo

Interval Monte Carlo...

Figures can be included like this:



?? illustrates the *nested Monte Carlo* method.



?? illustrates the *interval Monte Carlo* method.

Conclusion

`pyuncertainnumber` enables rigorous uncertainty analysis for real-world situations of mixed uncertainties and partial knowledge. Significance: this provides compatability as interfacing with many engineering applications. boost its usage.

Acknowledgements

The work leading to these results received funding through the UK project Development of Advanced Wing Solutions 2 (DAWS2). The DAWS2 project is supported by the Aerospace Technology Institute (ATI) Programme, a joint government and industry investment to maintain and grow the UK's competitive position in civil aerospace design and manufacture. The programme, delivered through a partnership between ATI, Department for Business and Trade (DBT) and Innovate UK, addresses technology, capability and supply chain challenges.

References

Angelis, M. de. (2022). *Intervals* (Version v0.1). Zenodo. https://doi.org/10.5281/zenodo.6205624

Gray, A., Ferson, S., & Patelli, E. (2021). ProbabilityBoundsAnalysis. Jl: Arithmetic with sets of distributions. *Proceedings of JuliaCon*, *1*, 1.

Gray, N., Ferson, S., De Angelis, M., Gray, A., & Oliveira, F. B. de. (2022). Probability bounds analysis for python. *Software Impacts*, *12*, 100246. https://doi.org/https://doi.org/10.1016/j.simpa.2022.100246