

¹ PyUncertainNumber for uncertainty propagation with ² black-box models: beyond probabilistic arithmetic

³ **Yu Chen**  ¹¶, **Scott Ferson**  ¹, and **Edoardo Patelli**  ²

⁴ ¹ Institute for Risk and Uncertainty, University of Liverpool, UK  ² Centre for Intelligent Infrastructure, University of Strathclyde ¶ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Open Journals](#) ↗

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))

⁶ Summary

⁷ Scientific calculations and simulations are complicated by various uncertainties inherent in the computational pipeline. Comprehensive analysis requires uncertainties be represented using mathematical constructs that distinguish variability from lack of knowledge and combine them rigorously in calculations. The new Python library `pyuncertainnumber` enables such analysis by computing guaranteed bounds on functions of uncertain variables given only partial empirical knowledge. It supports both intrusive methods that are efficient for projection through explicitly defined mathematical models and non-intrusive methods that can be used with black-box models that associate outputs with given inputs but whose internal workings are not known.

¹⁵ 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 (QoI). Real-world complex systems (physical or engineered) of industrial significance typically involves input parameters subject to uncertainties of various nature ([Oberkampf et al., 2004](#); [Smith, 2024](#)). These input uncertainties often appear as a mixture of parameters subject to variability (i.e. aleatory uncertainty, typically represented by probability distributions), or lack of knowledge (i.e. epistemic uncertainty, often represented by intervals), or a combination of both (i.e. mixed uncertainty, often represented by probability boxes (p-boxes)), which effectively represent a set of distributions and thereby capture both aleatory and epistemic uncertainty within a unified structure.

²⁶ Probability bounds analysis ([Ferson, 2001](#); [Springer, 1979](#); [Williamson & Downs, 1990](#)) is one of the expressive frameworks proposed to manage uncertainties in an imprecise setting ([Beer et al., 2013](#)). Software packages have been developed to facilitate the calculations of uncertain quantities, such as *interval arithmetic* ([Angelis, 2022](#)) and *probabilistic arithmetic* ([A. Gray et al., 2021](#); [N. Gray et al., 2022](#)). Collectively, they can be referred to as *uncertainty arithmetic* ([Chen & Ferson, 2025](#)), which provides a straightforward way to compute outcomes from algebraic expressions involving uncertain variables.

³³ While these uncertainty-representing frameworks have the potential to automatically compile non-deterministic subroutines via uncertain primitives, their use faces several challenges. A major one is that code accessibility¹ of the simulation models or software (e.g. finite-element or computational-fluid-dynamics models) employed to describe the behavior of the system is often not guaranteed, rendering intrusive² use of the above-mentioned uncertainty-representing frameworks infeasible. Often, these models are treated as black-box models in practice. Consequently, there is a critical need for software tools capable of performing mixed-uncertainty

¹access to the source code.

²requiring access or modification of the source code of a computational model.

⁴⁰ calculations on black-box models in real-world applications of computational engineering and
⁴¹ physics.
⁴² `pyuncertainnumber` addresses this need by providing the non-intrusive³ capability designed to
⁴³ allow generic black-box models to be rigorously propagated in the face of mixed uncertainties.
⁴⁴ This capability significantly boosts versatility for scientific computations through interfacing
⁴⁵ with many engineering software whose internal workings are not known.

⁴⁶ Interval propagation in a non-intrusive manner

⁴⁷ Interval analysis ([Moore et al., 2009](#)) features the advantages of providing rigorous enclosures of
⁴⁸ the solutions to problems, especially for engineering problems subject to epistemic uncertainty,
⁴⁹ such as modelling system parameters due to lack-of-knowledge or characterising measurement
⁵⁰ incertitude. Naive interval arithmetic typically faces difficulties such as the infamous [interval](#)
⁵¹ [dependency](#) issue. Though it may be mitigated through arithmetic rearrangements in some
⁵² simple cases, it still can be challenging for models of most complex systems, and impossible
⁵³ for black-box models. The critical issue remains the accessibility of code.

⁵⁴ Generally, the interval propagation problem can be cast as an optimisation problem where the
⁵⁵ minimum and maximum of the response are sought given a function mapping. Consider the
⁵⁶ function, for example f in [Equation 1](#), which is not necessarily monotonic or linear and may
⁵⁷ well be a black-box deterministic model for a generic system.

$$Y = f(I_{x1}, I_{x2}, \dots, I_{xn}) \quad (1)$$

⁵⁸ where $\mathbf{I} = [\underline{\mathbf{I}}, \bar{\mathbf{I}}] = [I_{x1}, I_{x2}, \dots, I_{xn}]^T$ represents the vector of interval-valued inputs. For
⁵⁹ black-box models the optimisation can generally be solved via gradient-free optimisation
⁶⁰ techniques, as shown below:

$$\underline{Y} = \min_{\mathbf{I} \leq \mathbf{I} \leq \bar{\mathbf{I}}} [f(\mathbf{I})]; \quad \bar{Y} = \max_{\mathbf{I} \leq \mathbf{I} \leq \bar{\mathbf{I}}} [f(\mathbf{I})]$$

⁶¹ `pyuncertainnumber` provides a series of non-intrusive methodologies of varying applicability.
⁶² It should be noted that there is generally a trade-off between applicability and computational
⁶³ efficiency. With additional knowledge pertaining the characteristics of the underlying function,
⁶⁴ one can accordingly dispatch an efficient method. For example, if the function is known to be
⁶⁵ monotone, one may employ the vertex method ([Dong & Shah, 1987](#)) which requires only 2^n
⁶⁶ model evaluations and is therefore substantially more efficient than a brute-force grid sampling
⁶⁷ scheme. It should be noted that the accuracy of these methods varies, and a common guideline
⁶⁸ is that increasing the number of model evaluations generally leads to a better estimate of the
⁶⁹ bound but at the cost of computational burden. A summary of applicability is tabulated in
⁷⁰ [Table 1](#); see [Chen & Ferson \(2025\)](#) for additional details of their advantages and disadvantages.

Table 1: Supported methods for non-intrusive interval propagation.

Category	Assumption	Results of Example A
Vertex (Endpoints)	monotonicity	[13.0, 148.0]
Subinterval reconstitution	monotonicity in subintervals	[13.0, 148.0]
Cauchy-deviate method	linearity and gradient required	[-11.7, 100.67]
Bayesian optimisation	general applicability	[13.0, 148.0]

³operating on the model as a black box, without accessing or modifying its internals.

Category	Assumption	Results of Example A
Genetic algorithm	general applicability	[13.0,147.8]

71 To better demonstrate the non-intrusive capability, two numerical examples, as displayed
 72 below in [Figure 1](#), are provided where they are treated as black-box models. [Table 1](#) lists the
 73 response interval of $f_a([1, 5], [7, 13], [5, 10])$ for respective methods. A reference ground truth
 74 is [13.0, 148.0].

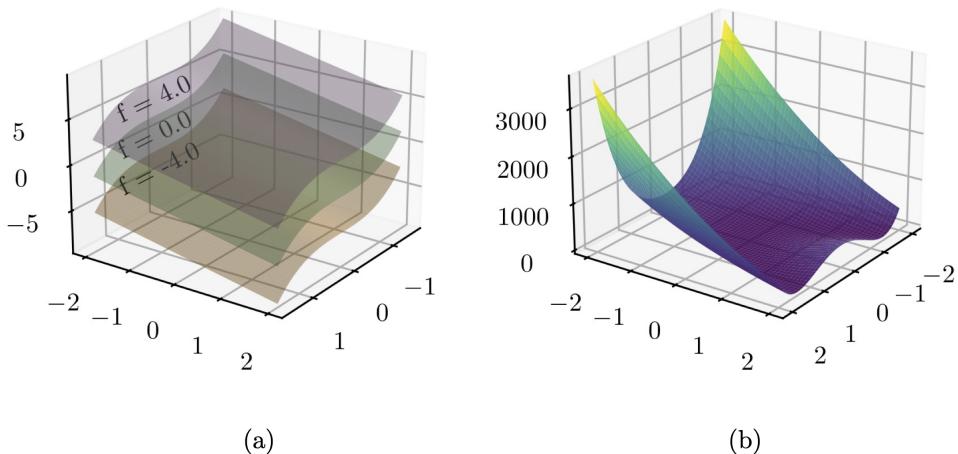


Figure 1: Exemplar functions as black-box models. (a) Example A: $f_a(x, y, z) = x^3 + y + z$, three level sets of $f_b = 4$, $f_b = 0$, $f_b = -4$ are shown; (b) Example B: $f_b(x, y) = 100(x - y^2)^2 + (1 - x)^2$.

75 Mixed uncertainty propagation for black-box models

76 Real world complex systems (physical or engineered) of industrial significance typically in-
 77 volves parameters subject to uncertainties of various nature ([Oberkampf et al., 2004](#)). It
 78 requires faithful characterisation of these uncertainties given the empirical information, and
 79 the approach to rigorously propagate them. Due to the fact that empirical information is
 80 often sparse or scarce or conflicting, even the uncertainty characterisation for one parameter
 81 could be of mixed nature, for example one may be confident about the distributional family
 82 but uncertain about its shape parameters, or when there exists multiple expert opinion of
 83 different credibility regarding its elicitation. Commonly, real systems expect a high-dimensional
 84 input which effectively represents a mixture of aleatory, epistemic, and mixed uncertainties, as
 85 symbolified below:

$$Y = f(\mathbf{u}; C) \quad (2)$$

86 where $\mathbf{u} \in \mathbb{R}^n$ denotes the collection of n uncertain inputs and C denotes intervariable
 87 dependency structure.

88 When both aleatory and epistemic uncertainties are present in \mathbf{u} , a *nested (double) Monte*
 89 *Carlo* approach can be used for deterministically models without confounding the two distinct
 90 types of uncertainty. As illustrated in [Figure 2](#) with the exemplar function B (see [Figure 1](#)),
 91 Latin-hypercube samples are first drawn from the epistemic interval, conditioned on which
 92 aleatory samples are drawn from the aleatoric probability distributions. Propagate these samples,
 93 which are visually denoted as rug ticks alongside the abscissa, through the computational

94 model results in an ensemble of CDF (cumulative distribution function) of the QoI whereby a
 95 final p-box is obtained as the envelope. Each CDF (orange color) corresponds to an epistemic
 96 sample.

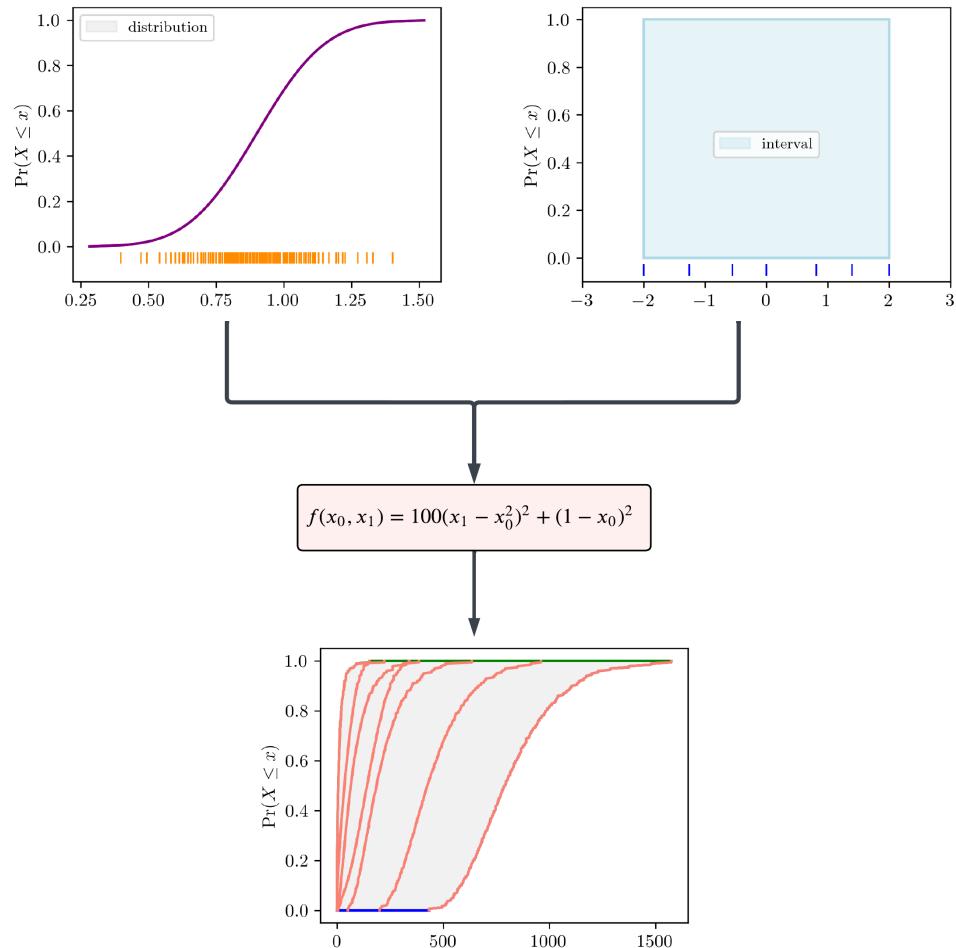


Figure 2: Workflow of the Double Monte Carlo method.

97 To scale to a more realistic setting, Figure 3 illustrates the workflow of *interval Monte Carlo*
 98 method using the exemplar function A where a mixture of aleatory, epistemic, and mixed
 99 uncertainty parameters are present, plus a certain copula is specified denoting the dependency
 100 structure. Correlated samples in the uniform space from the copula, visually denoted as rug
 101 ticks alongside the probability axis, are converted to physical space through alpha-cuts. Interval
 102 propagation (see the last section) then does the heavy lifting in which scalar values can be
 103 considered as degenerate intervals. As a result, the response QoI in Figure 3 is then obtained
 104 as a p-box shown in gray. In contrast, a pinched response, obtained from propagating pinched
 105 input variables (e.g. a p-box is pinched into a distribution and an interval is pinched into a
 106 scalar), is also shown as a comparison. Importantly, the pinched result being enclosed in the
 107 p-box manifests a critical feature of probability bounds analysis which yields results that are
 108 guaranteed to enclose all possible distributions of the output so long as the input p-boxes were
 109 all sure to enclose their respective distributions.

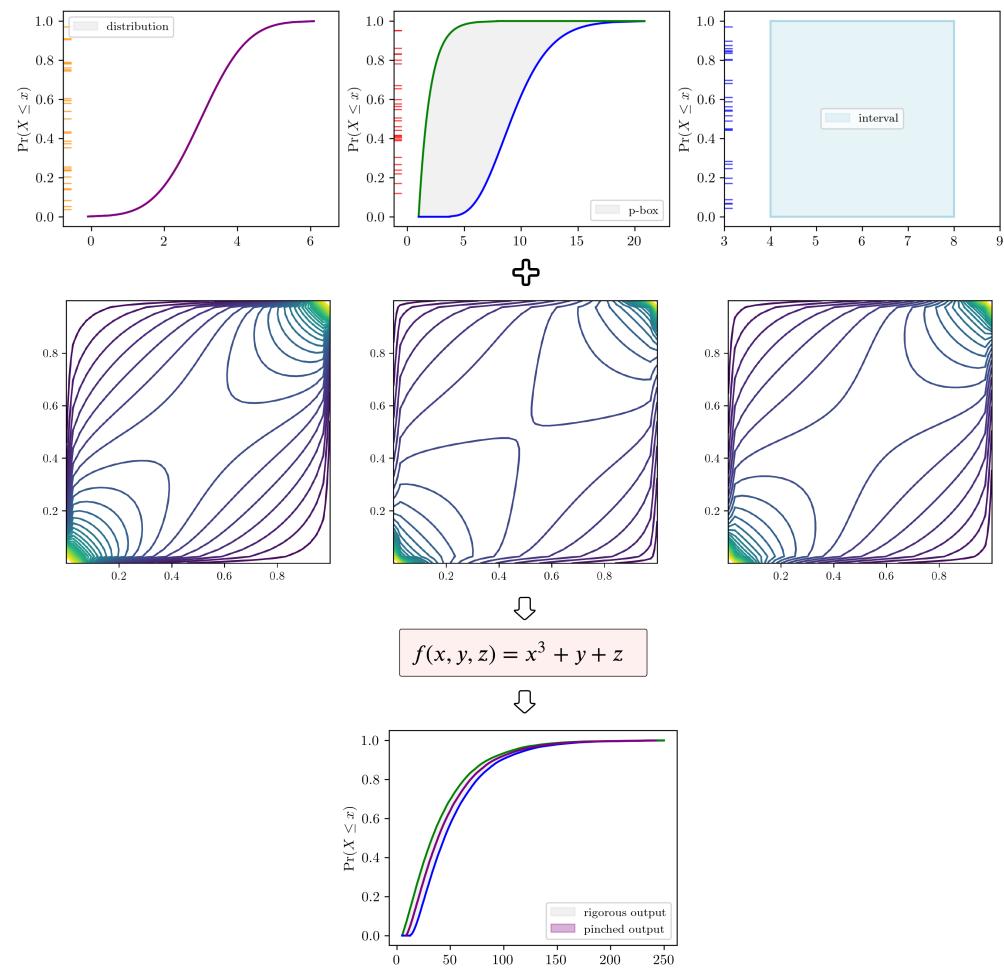


Figure 3: Workflow of the Interval Monte Carlo method. The first row shows the marginal of the inputs and the second row shows the bivariate copula density.

110

Conclusion

111

It is evident that many computational tasks in engineering and physics rely on complex numerical methods in which the model functions are black boxes. This makes uncertainty arithmetic difficult to manage because the model code is inaccessible. By providing methods supporting generic black box models, pyuncertainnumber enables rigorous uncertainty analysis for real-world scenarios of mixed uncertainties and partial knowledge. This non-intrusive capability allows pyuncertainnumber to interface with multiple software environments, offering extensive compatibility with engineering applications and diverse computational workflows.

112

113

114

115

116

117

118

119

120

121

122

123

124

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.

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.

125 References

- 126 Angelis, M. de. (2022). *Intervals* (Version v0.1). Zenodo. <https://doi.org/10.5281/zenodo.6205624>
- 127
- 128 Beer, M., Ferson, S., & Kreinovich, V. (2013). Imprecise probabilities in engineering analyses. *Mechanical Systems and Signal Processing*, 37(1-2), 4–29.
- 129
- 130 Chen, Y., & Ferson, S. (2025). Imprecise uncertainty management with uncertain numbers to facilitate trustworthy computations. *SciPy Proceedings*. <https://doi.org/10.25080/ahrt5264>
- 131
- 132
- 133 Dong, W., & Shah, H. C. (1987). Vertex method for computing functions of fuzzy variables. *Fuzzy Sets and Systems*, 24(1), 65–78.
- 134
- 135 Ferson, S. (2001). Probability bounds analysis solves the problem of incomplete specification in probabilistic risk and safety assessments. In *Risk-based decisionmaking in water resources IX* (pp. 173–188).
- 136
- 137
- 138 Gray, A., Ferson, S., & Patelli, E. (2021). ProbabilityBoundsAnalysis. Jl: Arithmetic with sets of distributions. *Proceedings of JuliaCon*, 1, 1.
- 139
- 140 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>
- 141
- 142
- 143 Moore, R. E., Kearfott, R. B., & Cloud, M. J. (2009). *Introduction to interval analysis*. SIAM.
- 144 Oberkampf, W. L., Trucano, T. G., & Hirsch, C. (2004). Verification, validation, and predictive capability in computational engineering and physics. *Appl. Mech. Rev.*, 57(5), 345–384.
- 145
- 146 Smith, R. C. (2024). *Uncertainty quantification: Theory, implementation, and applications*. SIAM.
- 147
- 148 Springer, M. D. (1979). *The algebra of random variables*.
- 149 Williamson, R. C., & Downs, T. (1990). Probabilistic arithmetic. I. Numerical methods for calculating convolutions and dependency bounds. *International Journal of Approximate Reasoning*, 4(2), 89–158.
- 150
- 151