Journal Pre-proof

Probability bounds analysis for Python

Nicholas Gray, Scott Ferson, Marco De Angelis, Ander Gray, Francis Baumont de Oliveira

PII: S2665-9638(22)00015-X
DOI: https://doi.org/10.1016/j.simpa.2022.100246
Reference: SIMPA 100246

To appear in: *Software Impacts*

Received date : 8 December 2021
Revised date : 17 January 2022
Accepted date : 24 January 2022

Please cite this article as: N. Gray, S. Ferson, M. De Angelis et al., Probability bounds analysis for Python, *Software Impacts* (2022), doi: https://doi.org/10.1016/j.simpa.2022.100246.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Journal Pre-proof

Probability Bounds Analysis for Python

Nicholas Gray, Scott Ferson, Marco De Angelis, Ander Gray and Francis Baumont de Oliveira
Institute for Risk and Uncertainty, University of Liverpool, Liverpool, United Kingdom, L69 7ZX. nickgray@liverpool.ac.uk

Abstract
Probability bounds analysis (PBA) is a collection of mathematical methods generalising interval analysis and probability theory. PBA can be utilised for uncertainty quantification for both aleatory and epistemic uncertainty across a wide range of scientific fields. PBA is most useful when information about variables is only partially known and can be used without requiring untenable assumptions to be made about parameter values, distribution shapes or dependence between variables. This paper introduces a PBA library for the Python programming language.

Keywords
Probability Bounds Analysis, Probability Boxes, P-boxes, Intervals, Uncertainty Quantification

Journal Pre-proof

Journal Pre-proof

Code metadata

Nr.	Code metadata description	
C1	Current code version	*v0.12*
C2	Permanent link to code/repository used for this code version	https://github.com/Institute-for-Risk-and-Uncertainty/pba-for-python
C3	Permanent link to Reproducible Capsule	https://codeocean.com/capsule/8485409
C4	Legal Code License	MIT License
C5	Code versioning system used	Git/GitHub
C6	Software code languages, tools, and services used	Python
C7	Compilation requirements, operating environments & dependencies	Python $\geq$ 3.7, NumPy $\geq$ 1.21.1, SciPy $\geq$ 1.7.0, Matplotlib $\geq$ 3.3.2
C8	If available Link to developer documentation/manual	https://pba-for-python.readthedocs.io
C9	Support email for questions	nickgray@liverpool.ac.uk

1. Introduction

Two types of uncertainty, *aleatory* and *epistemic*, appear in the numerical calculations essential to science and engineering. Aleatory uncertainty arises from the natural variability in dynamical environments and material properties, errors in manufacturing processes or inconsistencies in the realisation of systems. Aleatory uncertainty cannot be reduced by empirical effort. Epistemic uncertainty is caused by measurement imperfections or a lack of understanding about the underlying physics or biology of a system. This could be due to not knowing the full specification of a system in the early phases of engineering design or simplifying the mathematics of a simulation to save computational resources.

Probability bounds analysis (PBA) is a tool that can be used to compute with both types of uncertainties without requiring often untenable assumptions to be made about the parameters involved in calculations and any subsequent dependencies between them. Probability bounds analysis has many applications across diverse disciplines ranging from aerospace engineering [23] to conservation biology [12]. The Wikipedia page lists many applications to various scientific problems[1]. It is particularly popular when undertaking risk or reliability analyses when data is not perfectly known [24, 6, 5]. PBA objects and methods can also be used within machine learning techniques [15, 25, 26].

In this paper, we discuss the fundamental components of PBA, intervals and p-boxes, and how calculations are performed with them within PBA for Python. We make use of SciPy [29], NumPy [16] and Matplotlib [18] in order to define, store, display and perform calculations with p-boxes and intervals within PBA.

2. Probability Bounds Analysis

There are two main objects used for PBA, intervals and probability boxes (p-boxes). An interval is a value that is imprecisely known even though it may be fixed and unchanging, or perhaps an uncertain number representing values obeying an unknown distribution prescribed only by a specified range [4, 11, 17, 19, 20]. Intervals allow for epistemic uncertainty to be propagated through calculations.

A p-box is a generalisation of intervals and probability distributions in a single structure that allows the propagation of both epistemic and aleatory uncertainty through calculations in a rigorous way. A p-box can be considered as interval bounds on a probability distribution [8, 9, 10]. Within PBA it is convenient to think of a probability distribution as a special case of a p-box with precise inputs. Calculations performed with p-boxes yield results that are guaranteed to enclose all possible distributions of the output variable if the input p-boxes were also sure to enclose their respective distributions. The results may be best-possible if only valid distributions are enclosed within the p-box, although the output p-box may also contain distributions that could not arise under any dependence between the two input distributions. This property allows them to be used for automatic verification of computer codes [10, 21].

2.1 Intervals

An unknown real number x can be represented by an interval $[\underline{x}, \overline{x}]$, where $\underline{x} \leq x \leq \overline{x}$. This implies that the precise value of x can be any number within $\underline{x} \leq x \leq \overline{x}$. Intervals do not make any futher assumptions about which values within the range are more or less likely than other values.

[1] https://en.wikipedia.org/wiki/Applications_of_p-boxes_and_probability_bounds_analysis

Journal Pre-proof

Within the context of probability bounds analysis, it is useful to consider intervals as the set of all possible distributions that lie within the endpoints of the interval, this definition is discussed further in Section 2.3.

In PBA intervals can be defined by setting the left and right edges of the interval. If $a = [\underline{a}, \overline{a}]$ and $b = [\underline{b}, \overline{b}]$ are intervals, then the following arithmetic operations can currently be performed in PBA:

- **Addition**
$$a + b = [\underline{a} + \underline{b}, \overline{a} + \overline{b}] \tag{1}$$
- **Subtraction**
$$a - b = [\underline{a} - \overline{b}, \overline{a} - \underline{b}] \tag{2}$$
- **Multiplication**
$$a * b = \left[\min\left(\underline{a} * \underline{b}, \underline{a} * \overline{b}, \overline{a} * \underline{b}, \overline{a} * \overline{b}\right), \max\left(\underline{a} * \underline{b}, \underline{a} * \overline{b}, \overline{a} * \underline{b}, \overline{a} * \overline{b}\right)\right] \tag{3}$$
- **Division**
$$a/b = \left[\min\left(\underline{a}/\underline{b}, \underline{a}/\overline{b}, \overline{a}/\underline{b}, \overline{a}/\overline{b}\right), \max\left(\underline{a}/\underline{b}, \underline{a}/\overline{b}, \overline{a}/\underline{b}, \overline{a}/\overline{b}\right)\right] \tag{4}$$

If $0 \in b$ then a/b returns a division-by-zero error. If there is dependence between two intervals then PBA allows for this dependence to be included within the calculation. For intervals, perfect and opposite dependence calculations are possible. Perfect dependence between a and b implies that larger values of a correspond to larger values of b. In this scenario the arithmetic operations become

$$a \circ b = \left[\underline{a} \circ \underline{b}, \overline{a} \circ \overline{b}\right] \tag{5}$$

where $\circ \in (+, -, *, /)$. Whereas, under opposite dependence smaller values of a imply larger values of b, meaning that the arithmetic operations become

$$a \circ b = \left[\underline{a} \circ \overline{b}, \overline{a} \circ \underline{b}\right]. \tag{6}$$

An interval can be propagated through a function producing an interval output, $f([\underline{x}, \overline{x}]) = [\underline{y}, \overline{y}]$ where $\underline{y}$ is the minimum possible value of $f(x)$ for all $x \in [\underline{x}, \overline{x}]$ and $\overline{y}$ is the maximum possible value. This calculation is simple for monotonic functions. For instance, increasing monotonicity implies that the end points of the input interval correspond to the end points of the output interval, i.e.

$$f([\underline{a}, \overline{a}]) = [f(\underline{a}), f(\overline{a})]. \tag{7}$$

For more general functions alternative strategies are needed to insure correct calculations.

Comparison operations can be performed on intervals, however, the uncertainty associated with the interval leads to uncertainty in the Boolean operations. For example, if a decision relies on some value x being less than 1, when we know the value of x accurately then it is easy to make such a comparison. However, if there is some uncertainty about the value of x then this comparison may not be so easy. The comparison becomes

$$x < 1 = \begin{cases} 1 & \text{if } \overline{x} < 1 \\ 0 & \text{if } \underline{x} \geq 1 \\ [0,1] & \text{otherwise} \end{cases} \tag{8}$$

with 0 and 1 denoting false and true respectively, and [0,1] being the Boolean equivalent of "I don't know". We can call [0,1] the *dunno* interval. Similarly,

$$x > 1 = \begin{cases} 1 & \text{if } \underline{x} > 1 \\ 0 & \text{if } \overline{x} \leq 1 \\ [0,1] & \text{otherwise.} \end{cases} \tag{9}$$

For intervals it is often impossible to say whether an interval is equal to a value,

$$x == 1 = \begin{cases} [0,1] & \text{if } 1 \in x \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Two intervals can also be compared to each other. For intervals $x = [\underline{x}, \overline{x}]$ and $y = [\underline{y}, \overline{y}]$, then

$$x < y = \begin{cases} 1 & \text{if } \overline{x} < \underline{y} \\ 0 & \text{if } \underline{x} \geq \overline{y} \\ [0,1] & \text{otherwise} \end{cases} \tag{11}$$

Journal Pre-proof

and

$$x > y = \begin{cases} 0 & \text{if } \overline{x} \leq \underline{y} \\ 1 & \text{if } \underline{x} > \overline{y} \\ [0,1] & \text{otherwise} \end{cases} \tag{12}$$

This implies that we cannot say whether an uncertain value characterised by an interval is larger or smaller than another unless the interval is entirely greater or less than the other interval. For the equality comparison,

$$x == y = \begin{cases} [0,1] & \text{if } x \cup y \neq \varnothing \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

it is never possible to say that one value is equal to another. We can introduce a new Boolean operator (===) to test for whether two uncertain numbers are equivalent in form,

$$x === y = \begin{cases} 1 & \text{if } \underline{x} = \underline{y} \text{ and } \overline{x} = \overline{y} \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

The $[0, 1]$ interval can be converted into a true Boolean using operators such as always or sometimes

$$\text{always}\left([0,1]\right) = 0 \tag{15a}$$
$$\text{sometimes}\left([0,1]\right) = 1 \tag{15b}$$

so that we can get

$$\text{always}\left(x < y\right) = \begin{cases} 1 & \overline{x} < \underline{y} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

$$\text{sometimes}\left(x < y\right) = \begin{cases} 1 & \underline{x} < \overline{y} \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

2.2 Probability Distributions and Probability Boxes

A probability distribution is a mathematical function that gives the probabilities of occurrence for different possible values of a variable. Probability boxes (p-boxes) represent interval bounds on probability distributions. The simplest kind of p-box can be expressed mathematically as

$$\mathcal{F}(x) = [\underline{F}(x), \overline{F}(x)],\ \underline{F}(x) \geq \overline{F}(x)\ \forall x \in \mathbb{R} \tag{18}$$

where $\underline{F}(x)$ is the function that defines the left bound of the p-box and $\overline{F}(x)$ defines the right bound of the p-box. In PBA the left and right bounds are each stored as a NumPy array containing the percent point function (the inverse of the cumulative distribution function) for N evenly spaced values between 0 and 1, where N is the number of steps in the p-box. P-boxes can be defined using all the probability distributions that are available through SciPy's statistics library. Figure 1a shows a p-box that defined by a normal distribution with $\mu = [-1, 1]$ and $\sigma = [0.5, 1.5]$.

Naturally, precise probability distributions can be defined in PBA by defining a p-box with precise inputs. This means that within probability bounds analysis probability distributions are considered a special case of a p-box with zero width. Resultantly, all methodology that applies to p-boxes can also be applied to probability distributions. Figure 1b shows a standard normal distribution ($\mu = 0$, $\sigma = 1$).

Distribution-free p-boxes can also be generated when the underlying distribution is unknown but parameters such as the mean, variance or minimum/maximum bounds are known. Such p-boxes make no assumption about the shape of the distribution and instead return bounds expressing all possible distributions that are valid given the known information. Such p-boxes can be constructed making use of Chebyshev, Markov and Cantelli inequalities from probability theory. A p-box defined by $min = -3$,$max = 3$, $\mu = [0, 1]$ and $\sigma = 1$ is shown in Figure 1c.

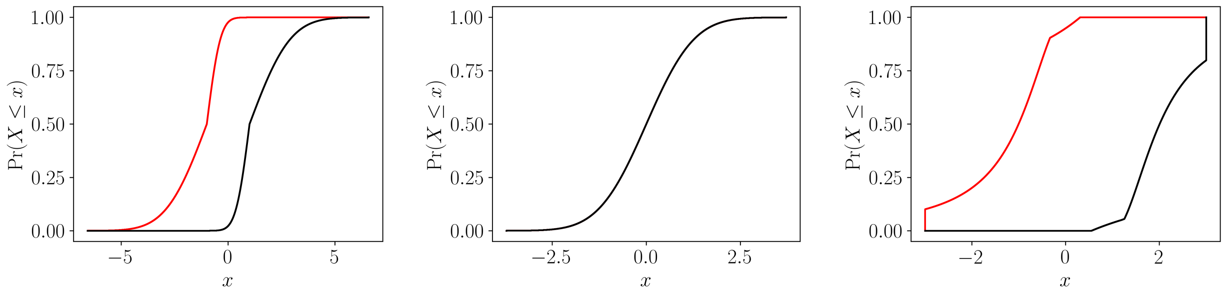
As with intervals, standard arithmetic operations can be performed on p-boxes (and therefore probability distributions which are special cases of p-boxes). For two p-boxes $\mathcal{A}(x) = [\underline{A}(x), \overline{A}(x)]$ and $\mathcal{B}(x) = [\underline{B}(x), \overline{B}(x)]$,

$$\mathcal{C}(x) = \mathcal{A}(x) \circ \mathcal{B}(x) = [\underline{C(x)}, \overline{C(x)}] \tag{19}$$

where

$$\underline{C(z)} = \inf_{z = x \circ y} \left[\min\left(\underline{A(x)} \circ \underline{B(y)}, 1\right)\right] \tag{20a}$$
$$\overline{C(z)} = \sup_{z = x \circ y} \left[\max\left(\overline{A(x)} \circ \overline{B(y)} - 1, 0\right)\right] \tag{20b}$$

Journal Pre-proof



(a) Normal distribution with $\mu = [-1, 1]$, $\sigma = [0.5, 1.5]$. (b) Standard normal distribution with $\mu = 0$, $\sigma = 1$. (c) Distribution-free p-box defined by $min = -3$,$max = 3$,$\mu = [0, 1]$ and $\sigma = 1$

Figure 1: Probability distributions and probability boxes.

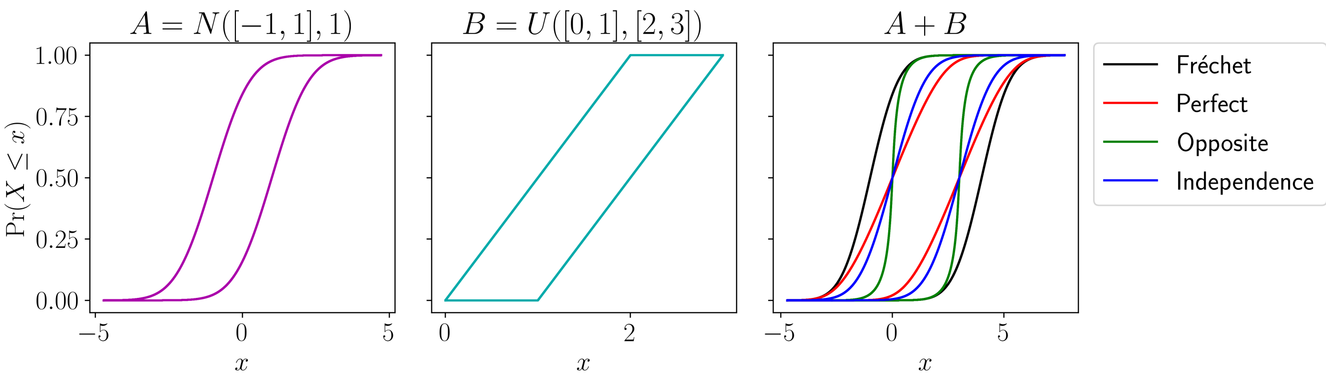


Figure 2: Adding together two p-boxes with different dependencies.

if $\circ \in [+, \times]$, or

$$\underline{C(z)} = 1 + \inf_{z=x\circ y}\left[\min\left(\underline{A(x)} \circ \overline{B(y)}, 0\right)\right] \tag{21a}$$

$$\overline{C(z)} = \sup_{z=x\circ y}\left[\max\left(\overline{A(x)} \circ \underline{B(y)}, 0\right)\right] \tag{21b}$$

if $\circ \in [-, \div]$. If $0 \in \mathcal{B}$ then the division returns a error [9, p. 89].

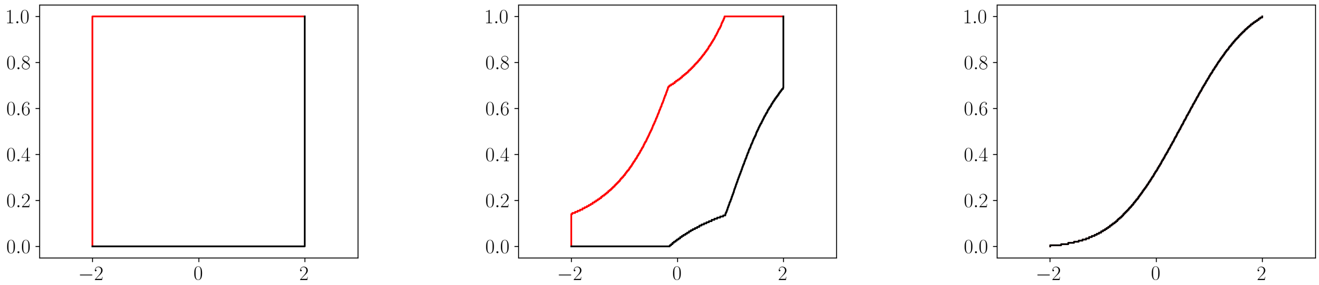
Knowledge of what the dependence is between the two p-boxes can reduce the amount of uncertainty present within the output p-box. Figure 2 shows the result of adding a normal p-box $A = \mathrm{N}([-1, 1], 1)$ to a uniform p-box $B = \mathrm{U}([0, 1], [2, 3])$, with different dependencies between A and B. When the dependence between A and B is unknown, the operation defined in equations 19–21 yields the most general bounds guaranteed to enclose the true distribution of $A+B$ which are called the Fréchet bounds. As depicted in Figure 2, the Fréchet bounds enclose all the other dependencies. Perfect (or comonotonic) dependence is where there is a perfect positive relationship between the two variables, with the highest possible correlation coefficient. Opposite (or countermonotonic) dependence creates a perfect negative relationship between the two variables with the lowest possible correlation coefficient. Independence is where there is no dependence between the two variables. It should not be assumed that variables are independent unless this is known because wrongly assuming independence can lead to incorrectly reducing the amount of uncertainty and understating tail risks.

2.3 Comparison between objects

As mentioned within Section 2.1, intervals can be considered as the set of all possible distributions that lie between the endpoints of the interval. This feature implies that interval objects can be converted into p-boxes by transforming the interval into a box-shaped p-box, such an object is shown in Figure 3a, this property means that arithmetic can be performed between p-boxes and intervals by casting the interval as a p-box when performing the calculation. Conversely, many unary operations that can be performed on intervals can be performed on p-boxes. This can be done by slicing the p-box into intervals, performing the operation before sorting and recombining the intervals back into a p-box.

Within PBA an interval can be considered as the most basic object, for example, if all we know about variable x is that its value lies between -2 and 2 then all we can say is that $x = [-2, 2]$, this is shown in Figure 3a. If more information about

Journal Pre-proof



(a) Probablistic representation of the Interval [-2,2]

(b) Distribution free p-box with: $min = -2$, $max = 2$, $\mu = 0.5$, $\sigma = 1$

(c) Truncated normal distribution with: $min = -2$, $max = 2$, $\mu = 0.5$, $\sigma = 1$

Figure 3: Comparing different PBA objects as the amount of information about x increases

the variable is known then the uncertainty can be reduced, for instance, if we know that x has mean 0.5 and standard deviation 1 then we can instead use a distribution-free p-box to model the uncertainty. Such an object can be seen in Figure 3b. Finally, if we know that x follows a truncated normal distribution then we can model x as shown in Figure 3c. As calculations with all of these objects can be performed using PBA, analysts can compute with what they know rather than making assumptions that may be unjustified.

3. Example

The attitude of a spacecraft is the direction in which it points. It is often important to control the attitude of a spacecraft; solar panels need to be pointed to the sun, communication antennas need to be pointed at the earth or scientific instruments need to point at the correct target. Attitude can be controlled through reaction wheels which can provide angular momentum to the spacecraft to point it in the desired direction.

The choice of how powerful a reaction wheel needs to be in depends on the torque needed to change the attitude of the spacecraft. The torque required depends on the moment of inertia of the spacecraft. The moment of inertia depends on the size of the spacecraft's solar panels which impacts the power available to the reaction wheels which impacts the torque available and so on. Therefore whilst there is uncertainty about the design of the spacecraft it is useful to make calculations using imprecise numbers. There are also additional uncertainties to consider such as the fact that solar radiation is not constant.

The equations of motion that determine the required angular momentum from the reaction wheel to change the attitude of a spacecraft within 1 dimension are as follows:

$$h = \tau_{tot} \times \Delta t_{orbit} \tag{22}$$

$$\tau_{tot} = \tau_{slew} + \tau_{dist} \tag{23}$$

$$\tau_{slew} = \frac{4\theta_{slew}}{\Delta t_{slew}^2} I \tag{24}$$

$$\tau_{dist} = \tau_g + \tau_{sp} + \tau_m + \tau_a \tag{25}$$

$$\tau_g = \frac{3\mu}{2(R_E + H)^3} \left| I_{max} + I_{min} \right| \sin(2\theta) \tag{26}$$

$$\tau_{sp} = L_{sp} \frac{F_S}{c} A_s (1 + q) \cos(i) \tag{27}$$

$$\tau_m = \frac{2MD\mu_0}{(R_E + H)^3} \tag{28}$$

$$\tau_a = \frac{1}{2} L_a \rho C_d A V^2 \tag{29}$$

$$V = \sqrt{\frac{m}{R_E + H}} \tag{30}$$

Table 1 gives definitions and values for all variables within these equations.

PBA for Python can be used to perform the calculation using the uncertainty expressed about the variables. The full calculation is available through the linked Code Ocean repository. Figure 4 shows the final step in the calculation (Equation 22). The resultant p-box can be used to make decisions about the requirements of the reaction wheels.

4. Impact Overview

Before the creation of this library, there was not a PBA library for Python. Although versions did exist for Risk Calc [7], MATLAB [2], R [3] and Julia [1, 13], as Python is one of the most popular programming languages [27, 28], especially

Journal Pre-proof

Symbol	Variable	Type	Value	Unit
h	Required angular momentum		Calculated	N m s
τ_{tot}	Total required torque		Calculated	N m
τ_{slew}	Slewing torque		Calculated	N m
τ_a	Torque due to atomspheric resistance		Calculated	N m
τ_{sp}	Torque due to solar radiation pressure		Calculated	N m
τ_g	Torque due to gravitational gradient		Calculated	N m
V	Velocity of spacecraft		Calculated	m s^{-1}
C_d	Drag coefficient	p-box	min=2, max = 4, mean=3.13	unitless
L_a	Aerodynamic drag torque moment	p-box	min=0, max=3.75, mean=0.25	m
L_{sp}	Solar radiation torque moment	p-box	min=0, max=3.75, mean=0.25	m
D	Residual dipole	interval	[0,1]	A m^2
i	Sun incidence angle	interval	[0,90]	degrees
ρ	Atmospheric density	interval	$[3.96 \times 10^{-12}, 9.9 \times 10^{-11}]$	kg m^3
θ	Major moment axis deviation from nadir	interval	[10,19]	degrees
q	Surface reflectivity	interval	[0.1,0.99]	unitless
I_{min}	Minimum moment of inertia	point	4655	kg m^2
I_{max}	Maximum moment of inertia	point	7315	kg m^2
m	Earth gravity constant	point	3.98×10^{14}	m^{3}s^{-2}
A	Area in the direction of flight	point	3.752	m^2
R_E	Earth radius	point	6378.14	km
H	Orbit altitude	point	340	km
F_S	Average solar flux	point	1367	W m^{-2}
q_{slew}	Maximum slewing angle	point	38	degrees
c	Light speed	point	2.9979×10^8	m s^{-1}
M	Earth magnetic moment	point	7.96×10^{22}	A m^2
Δt_{slew}	Minimum maneuver time	point	760	s
A_s	Area reflecting solar radiation	point	3.75^2	m^2
Δt_{orbit}	Quarter orbit period	point	1370	s
μ_0	Permiability of free space	point	$4\pi \times 10^{-7}$	N A^{-2}

Table 1: Definitions and values for Equations 22–30

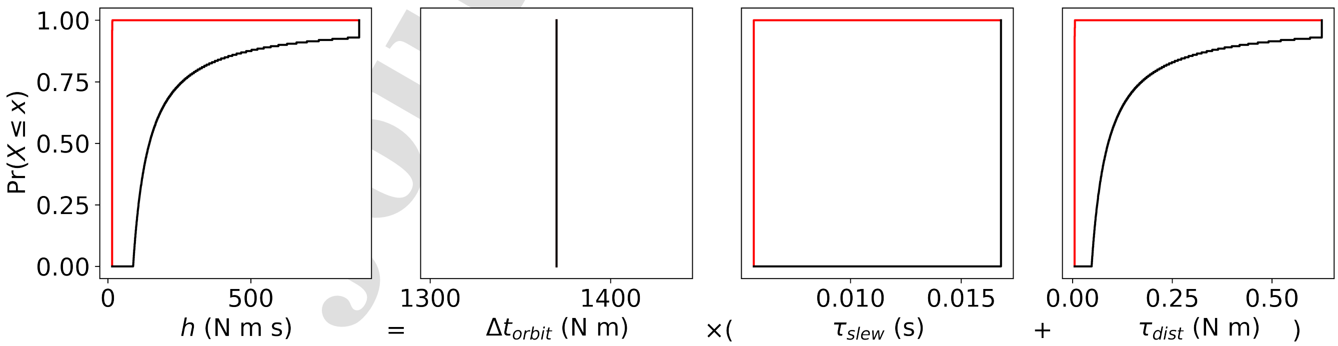


Figure 4: Calculation of Equation 22 using PBA.

Journal Pre-proof

within the field of scientific computing, many scientists and engineers who prefer to programme using Python were unable to make use of the powerful methodology and many advantages of probability bounds analysis. The creation of PBA for Python expands the reach of this methodology so that it can be applied to other disciplines and sectors.

5. Research Areas

Probability bounds analysis has many possible applications as discussed in the introduction. The authors are aware of the following work that uses PBA for Python in the following fields:

- Agricultural economics – Calculating financial cash flows and risk of insolvency for indoor farming businesses in the absence of data [22],
- Medical diagnosis – Propagating uncertainty through Bayes' rule to calculate whether a patient has a disease based upon their incomplete answers to a symptom questionnaire,
- Logistic regression – Generalising logistic regression models for use with possibly imprecise data and unknown status outcomes [15], and
- Automated uncertainty quantification – Translating Python code into uncertainty-aware code that can full take account of uncertainties in parameters and inputs [14].

Acknowledgements

The authors would like to acknowledge the support of Jason O'Rawe, Bright Uchenna Oparaji and George Dale in helping with the development of PBA. They would also like to thank Alex Wimbush and Dominic Calleja for using PBA in their projects thus providing invaluable feature requests and error reporting whilst the code was in its infancy.

This work has been funded by the Engineering and Physical Science Research Council (EPSRC) through the programme grant "Digital twins for improved dynamic design", EP/R006768/1.

References

[1] Probability Bounds Analysis for Julia. https://github.com/AnderGray/ProbabilityBoundsAnalysis.jl.

[2] Probability Bounds Analysis for MATLAB. https://github.com/Institute-for-Risk-and-Uncertainty/pba-for-matlab.

[3] Probability Bounds Analysis for R. https://github.com/ScottFerson/pba.r.

[4] Götz Alefeld and Günter Mayer. Interval analysis: Theory and applications. *Journal of Computational and Applied Mathematics*, 121(1):421–464, 2000.

[5] Michael Beer, Scott Ferson, and Vladik Kreinovich. Imprecise probabilities in engineering analyses. *Mechanical Systems and Signal Processing*, 37(1-2):4–29, 2013.

[6] Luis G. Crespo, Sean P. Kenny, and Daniel P. Giesy. Reliability analysis of polynomial systems subject to p-box uncertainties. *Mechanical Systems and Signal Processing*, 37(1-2):121–136, May 2013.

[7] Scott Ferson. *RAMAS Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers*. Lewis Publishers, Boca Raton, Florida, USA, 2002. https://books.google.co.uk/books?id=tKz7UZRs0CEC.

[8] Scott Ferson, Vladik Kreinovich, Lev Ginzburg, Davis S Myers, and Kari Sentz. Constructing Probability Boxes and Dempster-Shafer Structures. Technical Report January, Sandia National Laboratories, Albuquerque, NM, United States, 2003.

[9] Scott Ferson, Roger B Nelsen, Janos Hajagos, Daniel J Berleant, Jianzhong Zhang, W Troy Tucker, Lev R Ginzburg, and William L Oberkampf. Dependence in probabilistic modeling, Dempster-Shafer theory, and probability bounds analysis. Technical Report 19094, Sandia National Laboratories, Albuquerque, NM, USA, 2004.

[10] Scott Ferson and William L. Oberkampf. Validation of imprecise probability models. *International Journal of Reliability and Safety*, 3(1/2/3):3–3, 2009.

[11] Federica Gioia, Carlo N Lauro, and Napoli Federico. Basic statistical methods for interval data. *Statistica Applicata*, 17:1–29, 2005.

Journal Pre-proof

[12] Lloyd Goldwasser, Scott Ferson, and Lev Ginzburg. Variability and Measurement Error in Extinction Risk Analysis: The Northern Spotted Owl on the Olympic Peninsula. In *Quantitative Methods for Conservation Biology*, pages 169–187. Springer-Verlag, New York, 2000.

[13] Ander Gray, Scott Ferson, and Edoardo Patelli. ProbabilityBoundsAnalysis.jl: Arithmetic with sets of distributions. page 12, 2021.

[14] Nicholas Gray, Marco De Angelis, and Scott Ferson. The Creation of Puffin, the Automatic Uncertainty Compiler. *arXiv:2110.10153 [cs, stat]*, October 2021. http://arxiv.org/abs/2110.10153.

[15] Nicholas Gray and Scott Ferson. Logistic Regression Through the Veil of Imprecise Data. *arXiv:2106.00492 [stat]*, June 2021. http://arxiv.org/abs/2106.00492.

[16] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[17] Timothy Hickey, Qun Ju, and Maarten H Van Emden. Interval Arithmetic: From Principles to Implementation. *Journal of the ACM*, 48(5):1038–1068, 2001.

[18] John D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[19] IEEE. *1788-2015 – IEEE Standard for Interval Arithmetic.* 2015.

[20] Ramon E Moore, R Baker Kearfott, and Michael J Cloud. *Introduction to Interval Analysis*, volume 110. Society for Industrial and Applied Mathematics, Philadelphia, USA, 2009.

[21] William L Oberkampf and Scott Ferson. Model Validation under Both Aleatory and Epistemic Uncertainty. *New York*, (i):1–26, 2007.

[22] Francis Baumont de Oliveira Oliveira, Scott Ferson, Ron Dyer, Jens Thomas, Paul Myers, and Nicholas Gray. How high is high enough? Assessing financial risk for vertical farms. *In Prep.*, 2021.

[23] Christopher J. Roy and Michael S. Balch. A holistic approach to uncertainty quantification with application to supersonic nozzle thrust. *International Journal for Uncertainty Quantification*, 2(4):363–381, 2012.

[24] Daniel J. Rozell and Sheldon J. Reaven. Water Pollution Risk Associated with Natural Gas Extraction from the Marcellus Shale: Marcellus Shale Water Pollution Risk. *Risk Analysis*, 32(8):1382–1393, August 2012.

[25] Jonathan Sadeghi, M. de Angelis, and Edoardo Patelli. Efficient training of interval Neural Networks for imprecise training data. *Neural Networks*, 118:338–351, 2019.

[26] Jonathan Cyrus Sadeghi. *Uncertainty Modelling for Scarce and Imprecise Data in Engineering Applications.* PhD thesis, University of Liverpool, 2020.

[27] Stack Overflow. 2021 Developer Survery. https://insights.stackoverflow.com/survey/2021#technology-most-popular-technologies, May 2021.

[28] TIOBE - The Software Quality Company. TIOBE Index. https://www.tiobe.com/tiobe-index/, January 2022.

[29] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, SciPy 1.0 Contributors, Aditya Vijaykumar, Alessandro Pietro Bardelli, Alex Rothberg, Andreas Hilboll, Andreas Kloeckner, Anthony Scopatz, Antony Lee, Ariel Rokem, C. Nathan Woods, Chad Fulton, Charles Masson, Christian Häggström, Clark Fitzgerald, David A. Nicholson, David R. Hagen, Dmitrii V. Pasechnik, Emanuele Olivetti, Eric Martin, Eric Wieser, Fabrice Silva, Felix Lenders, Florian Wilhelm, G. Young, Gavin A. Price, Gert-Ludwig Ingold, Gregory E. Allen, Gregory R. Lee, Hervé Audren, Irvin Probst, Jörg P. Dietrich, Jacob Silterra, James T Webber, Janko Slavič, Joel Nothman, Johannes Buchner, Johannes Kulick, Johannes L. Schönberger, José Vinícius de Miranda Cardoso, Joscha Reimer, Joseph Harrington, Juan Luis Cano Rodríguez, Juan Nunez-Iglesias, Justin Kuczynski, Kevin Tritz, Martin Thoma, Matthew Newville, Matthias Kümmerer, Maximilian Bolingbroke, Michael

Tartre, Mikhail Pak, Nathaniel J. Smith, Nikolai Nowaczyk, Nikolay Shebanov, Oleksandr Pavlyk, Per A. Brodtkorb, Perry Lee, Robert T. McGibbon, Roman Feldbauer, Sam Lewis, Sam Tygier, Scott Sievert, Sebastiano Vigna, Stefan Peterson, Surhud More, Tadeusz Pudlik, Takuya Oshima, Thomas J. Pingel, Thomas P. Robitaille, Thomas Spura, Thouis R. Jones, Tim Cera, Tim Leslie, Tiziano Zito, Tom Krauss, Utkarsh Upadhyay, Yaroslav O. Halchenko, and Yoshiki Vázquez-Baeza. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020.

Journal Pre-proof

- In this paper we introduce a Probability Bounds Analysis (PBA) Library for Python
- PBA is a collection of mathematical methods generalising interval analysis and probability theory.
- The PBA library contains class definitions for intervals and p-boxes as well as key functions to enable their use within calculations
- The library is open source and available through both GitHub and pypi

Journal Pre-proof

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author.

Signed by all authors as follows:

Nicholas Gray, Scott Ferson, Marco De Angelis, Ander Gray, Francis Baumont de Oliveira
2nd December 2021

Journal Pre-proof