Towards a Framework for Non-intrusive Uncertainty Propagation in the Preliminary Design of Aircraft Systems

Ioanna Ioannou[*], Peter O. Hristov[†] and Scott Ferson[‡]
University of Liverpool, Liverpool, L69 3GH

Hau Kit Yong[§], Robert Marsh[¶], Edmar A. Silva[‖] and András Sóbester[**]
University of Southampton, Southampton, UK, SO16 7QF

Uncertainty is present throughout the aircraft design process, and is of key importance during the preliminary design stage. At the same time there is often insufficient understanding of the governing process to warrant a detailed description of an uncertainty model, which often leads to neglecting the issue altogether. In this paper we present the foundation of a framework for propagating uncertainty through black-box computer models organically integrated into highly flexible parametric geometry models. We form this foundation by focusing on the interval as a building block of uncertain numbers. We compare the reliability and efficiency of the uncertainty representation provided by various interval propagation methods by applying them to an early-stage geometric model of an electric motor mount.

Nomenclature

d	=	number of input dimensions
D	=	number of Michell lattice beams
e_x	=	error defining an interval in central form
E	=	material Young's modulus
F	=	load applied on a single Michell lattice
f	=	function representing a computer model
L	=	overall length of Michell lattice
M	=	mass of a single Michell lattice
m	=	number of divisions or samples for interval propagation
n	=	number of code evaluations
R_{in}	=	inner to outer beam ratio
R_{in_2}	=	second inner to outer beam ratio
R_{out}	=	outer beam width to lattice length ratio
x^I	=	an interval-valued variable x
$\underline{x}$	=	lower bound of an interval-valued variable x
$\overline{x}$	=	upper bound of an interval-valued variable x
ν	=	material Poisson's ratio
ρ	=	material density
σ_y	=	material yield stress

[*] Research Associate , Institute for Risk and Uncertainty

[†] Research Associate, Institute for Risk and Uncertainty

[‡] Professor, Institute for Risk and Uncertainty

[§] Research Fellow, Faculty of Engineering and Physical Sciences

[¶] Senior Research Fellow, Faculty of Engineering and Physical Sciences

[‖] Senior Research Fellow, Faculty of Engineering and Physical Sciences

[**] Professor, Faculty of Engineering and Physical Sciences, Senior Member AIAA

I. Introduction

A. Background and motivation

Uncertainty is present in all engineering design processes and is often an implicit driver behind a number of decisions concerning engineered systems. Uncertainty is abundant in aircraft design, during the early stages of subsystem development, but also where resilience must be built into the system [1]. It has been demonstrated that the careful and methodical quantification of uncertainty can lead to a more complete exploration of the design space, and therefore can result in aircraft with superior performance and higher reliability [2]. Diligent uncertainty quantification at every stage of the design process can reduce the occurrence of surprising events and the need for potentially costly redesigns at later stages of the aircraft life cycle [3]. Moreover, the use on non-deterministic design strategies allows designers to better understand the behaviour of the system they are dealing with and provide actionable solutions to potential problems. It seems, therefore, imperative that prudent uncertainty quantification (UQ) accompanies the design effort from early on.

Despite its significance, uncertainty is often treated in one of two inadequate ways. One way is to simply ignore it. The most common reasons for ignoring uncertainty include the inability to model it precisely, the perception that including uncertainty will take up too much by way of valuable computational resources, or that fully accounting for uncertainty will dilute results and render them useless to the decision-making process. Alternatively, uncertainty is incorporated in the form of rigid safety factors and margins, often leading to over-engineered systems. Aircraft are particularly sensitive to this problem, due to the delicate balance between performance and safety that must be maintained throughout their design and operation. Finding this balance using a deterministic set of rules is particularly difficult in the preliminary design stage, and becomes next to impossible as the number of parameters to consider grows.

The aversion to UQ is exacerbated by the ongoing debate in the UQ community about the best way to deal with different kinds of uncertainty. There is a general consensus among researchers and engineers that the two major types of uncertainty are epistemic and aleatory uncertainty. Sources of *epistemic* uncertainty in design include doubt about the value of fixed, but unknown parameters, ignorance about the details of the mechanisms and processes involved, and other imperfections in scientific understanding. Once the design begins to converge, the uncertainty starts to develop an *aleatory*, or stochastic, component as well. This is usually in the form of manufacturing and assembly tolerances, but could also enter the design through naturally varying environmental conditions. Even though it may sometimes be difficult to distinguish the two types of uncertainty, the fact that they seek to express fundamentally different aspects of design has been agreed upon [4]. However, there is no consensus on whether epistemic and aleatory uncertainty should be treated differently to one another, with some researchers maintaining that probability theory can be applied to both types (e.g., [5]), while others holding the view that alternative approaches are needed to deal with different kinds of uncertainty(e.g., [6, 7]). Likewise, in aeronautical design, the separation is acknowledged, but seldom effected in practice; see e.g., [8–11], cf. [12].

In the framework whose foundations are presented in this paper, we advocate a clear separation between the kinds of uncertainty, not only at the theoretical level, but also during practical engineering work. Our view is that the correct characterisation of uncertainty will not require the designer to make untenable assumptions about evidence or precision, thereby allowing them to only use information they are comfortable with. Moreover, instead of lumping all uncertainty together, this distinction allows the designer to develop a clearer appreciation of the uncertainty they can reduce (epistemic) and that which is outside of their control and needs to be mitigated instead (aleatory). Finally, maintaining the uncertainty distinction throughout the design life cycle, allows the designer to use the appropriate methods to propagate each type, optimising their computational budget.

In this study, we focus on the preliminary aircraft design stage, which is strongly affected by uncertainty in the geometry of components and subsystems. Geometric features determine, to a large degree, the performance, handling and reliability of aircraft. They are strongly coupled among each other, and with non-geometric attributes, often in complex ways, such that changing one feature may result in changes to a potentially large number of seemingly unrelated characteristics. The importance of dealing with geometric uncertainty is evidenced by the number of studies conducted in the field. These include the effect of uncertain geometry on the aerodynamic properties of airfoils [13–16] and wing flow modifiers [17, 18]; on the aeroelastic response of wing structures [19, 20]; on the propulsive efficiency of power plants [21, 22] and in full aircraft configurations [23].

B. A procedural approach to geometry-centric design

Nature's way of building (and optimizing) complicated geometry is by encoding it in the structure of a DNA molecule. This then serves as a recipe for the step-by-step construction of each copy of the organism. The clichéd metaphor of DNA being the "blueprint of life" is thus, incidentally, nonsense – a DNA molecule is not, in any way, a representation of the geometry itself, rather of the steps required to construct it [24].

Procedural geometry generation (geometry defined as code) – the CAD modelling technique we adopt here – is an artificial analogue of nature's recipe-based approach and it has a number of advantages compared to the more traditional 'CAD as a digital version of a drawing board', blueprint-like approach. In [24] we provided a detailed run-down of these; here we shall merely highlight those most relevant to the work we are reporting on:

- it facilitates "don't repeat yourself", "single source of truth", "separation of concerns", and other software engineering principles that are far harder to incorporate in more traditional "point and click" CAD environments,
- it facilitates version control and automated continuous integration (that is, unit testing and integration testing integrated directly into the version control system),
- it allows a much more favourable balancing of the fundamental trade-off at the heart of every parametric geometry, that between model flexibility (the size of the design space), conciseness (the dimensionality of the design space), and robustness (the proportion of the design space within which the geometry makes physical sense),
- it allows the direct weaving of surrogate models and other forms of machine learning into the fabric of the geometry model itself, giving intelligent "self design" capabilities,
- and finally, **most pertinently to the subject of this study**, it facilitates uncertainty propagation in a much more direct, organic, and computationally efficient way than conventional geometry modelling approaches.

The particular flavour of procedural geometry modelling we are adopting here hybridises a code-based approach (at the level of geometric primitives) and a visual programming paradigm for incorporating "engineering level" rationale into the parametric model. The visual (or graphical) programming approach, where the designer lays out abstractions of geometric objects on a "canvas", connecting them together with data pipelines in a manner akin to an electronic circuit, is particularly conducive to a UQ view of the world, allowing for intuitive sensitivity analysis "on the fly". This is as simple as inserting potentiometer-style components into the 'circuit' of the geometry, associated with input values. Turning or sliding these then allows immediate "one-factor-at-a-time" sensitivity studies of the sort one might engage in naturally when faced with the dashboard of an unknown machine. The unknown machine here is the (often) black box type model of the performance of the engineering artefact being designed, which offers a "first pass‘’ assessment of sensitivities even when the underlying mathematics is sufficiently complex to defy analytical understanding. In Section III.A we shall present our design case study based on these principles.

C. Challenges in uncertainty propagation

A key challenge faced at each stage of the design process is how best to express the uncertainty in the input variables. It is difficult and often questionable to assign precise models of uncertainty at early stages of the design process. This is because precise descriptions of uncertainty rely on a set of parameters, which is usually larger than the one used to describe the design itself and may, in turn, be affected by a considerable amount of uncertainty. Moreover, the typically accepted probabilistic description of uncertainty requires the specification of a shape of a probability distribution about the design parameters. Subsequent determination of parameters and potential inferences are conditional on this shape. It is common, at early stages to assign a uniform probability distribution to parameters, for whose shape we want to express ignorance. Even though the uniform distribution is the maximum entropy distribution over a given interval [25], it hardly expresses ignorance. On the contrary, it states that every outcome on that interval is equally likely, a statement for which there may be no evidence in early-stage design.

In an attempt to avoid an overly ambitious specification of input uncertainty, many engineers turn to *worst-case analysis* [26], which seeks to determine the values of system parameters that would result in the worst possible outcome. This risk-based strategy has been a pillar of engineering for decades, because it seeks to identify and minimise high-consequence events. In fact, worst-case analysis forms the foundation and justification for the use of safety factors and safety margins. However, worst-case analysis implicitly assumes that the engineer knows which combination of inputs results in the true worst case performance or safety. With many black-box models the engineer may not have a sufficient understanding of the physics, or implementation, or it may simply be that the performance function of the model is too complex to reliably identify input combinations that yield the global worst case. In essence, worst-case analysis is a deterministic solution to deal with design uncertainty, and as such it is sensitive to the specification of input values. In addition, when choosing a worst case, the engineer may be excessively conservative, choosing a particular scenario that is possible, but extremely unlikely, or even beyond the capacity of the design and not worth considering.

Unlike worst-case analysis, interval analysis [27, 28] is a formal, non-deterministic method, which uses the principles of interval arithmetic to propagate uncertainty through computational models. Intervals express uncertainty about the parameter x as either closed ranges of real values, $x \in [\underline{x}, \overline{x}]$, or as a nominal value plus or minus some error, $x \pm e_x$. The method is rigorous, because it guarantees that if input intervals enclose the true value of the associated parameter, than the true value of the output will be contained in its corresponding interval. In the absence of repeated uncertain variables in the mathematical expressions contained in the model, the resulting interval provides the tightest possible bounds around the answer, given the input uncertainty[*]. Because interval ranges enclose the parameter, interval analysis is equivalent to performing worst-case and best-case scenario analysis simultaneously. Interval analysis can be used to propagate any type of uncertainty, but it is typically applied in problems dealing with epistemic uncertainty. The relatively small amount of information required and the guarantees provided by interval analysis, make it well-suited to propagating uncertainty and conducting sensitivity studies during preliminary aircraft design. Moreover, intervals are fundamental in the construction and analysis of more general uncertain numbers, such as fuzzy sets [29] and probability boxes [30], which in turn provide the means to correctly propagate probability distribution through models, using samples of finite size.

In its original form, interval analysis requires access to the source code of the computational model, making it an intrusive uncertainty propagation method [31]. This fact presents a problem for the application of interval analysis because the mathematical expressions of the majority of industrially applicable models are inaccessible to the user. That is to say, the code is a black box. In this study, we focus on the propagation of interval uncertainty through just such black-box models. We make use of the algorithmic nature of procedural geometry generation to explore several different strategies for the propagation of interval uncertainty from geometric parameters to the mass of a motor mount for an electric aircraft. We discuss the advantages and disadvantages of each method and outline a general guidance for the selection of techniques for the propagation of interval uncertainty through black-box models.

The remainder of the study is organised as follows. Section II introduces the applied interval propagation methods in more detail and discusses their implementation. Section III.A describes the theory and parameterisation of Michell lattices, used as a case study for interval uncertainty propagation. Section III.B presents results from using each of the methods to propagate uncertainty in the geometric properties of the Michell lattice motor mount to its mass. This section also provides a discussion on the applicability of each method, in both the selected case and in general. The study finishes with concluding remarks in Section IV.

II. Non-intrusive propagation methods

Interval arithmetic, which forms the basis for interval analysis, redefines the four main arithmetic operations (addition, subtraction, multiplication and division), for two interval valued operands, x^I and y^I, as follows:

$$\begin{aligned}
x^I + y^I &= [\underline{x} + \underline{y}, \overline{x} + \overline{y}] \\
x^I - y^I &= [\underline{x} - \overline{y}, \overline{x} - \underline{y}] \\
x^I \times y^I &= [\min(\underline{xy}, \overline{x}\underline{y}, \underline{x}\overline{y}, \overline{xy}), \max(\underline{xy}, \overline{x}\underline{y}, \underline{x}\overline{y}, \overline{xy})] \\
x^I / y^I &= x \times [1/\overline{y}, 1/\underline{y}] \text{ as long as } 0 \notin y^I
\end{aligned} \tag{1}$$

Unary operations, such as exponentiation, logarithmic and trigonometric functions are also defined, but their result depends on the interval argument and is not detailed here (for a complete exposition on interval arithmetic, see [27]). Despite the fact that interval analysis is rigorous, to use it, each expression in the computer model must be decomposed into elementary arithmetic operations and evaluated, something that is commonly not feasible. For these cases, there is a plethora of non-intrusive approaches that can be adopted to propagate the uncertainty through a black-box model. In this study we review and compare the most commonly used ones. A common feature among all of these approaches is that they decompose intervals into a collection of real numbers that can readily be processed by any black-box model [32]. As such, these methods provide an approximation to the output interval, whose quality depends on the characteristics and complexity of the model itself.

In this section, we present a brief description of the adopted approaches. We compare the methods based on features important for reliable uncertainty propagation. The first two features we care about are how *rigorous* each method is and the *assumptions* that must be made in order for this rigor to hold. We use a "traffic-light system" to provide the reader with a feeling about the relative ranking of methods based on those characteristics. We remind the reader that for true black-box models, one cannot provide mathematical guarantees as to the quality of the propagation result, but the

[*] If the same uncertain variables occur multiple times in the expression of the model, the resulting interval may be wider than needed. This problem is known as the 'repeated variable problem'

comparison serves as a useful guide in selecting a method. The third and final point we include in the methods' analysis is their *computational cost*, expressed as a function of the number of code evaluations. A summary of the results is provided in Table 1

Table 1 Summary of key characteristics of the uncertainty propagation methods reviewed here. The cost for each method is expressed in terms of the number of samples or partitions, m, chosen by the engineer and the dimensionality of the problem, d. Rigor scores are based on the provision that the assumptions about the function are met.

Method	Underlying Assumptions	Cost	Rigor
Vertex propagation	Monotonic over intervals	2^d	High
Subinterval reconstitution	Monotonic over subintervals	$(m+1)^d$	High
Monte Carlo methods	None	m	Low
Cauchy deviates	Linear over intervals	m	Medium
Evolutionary optimisation	Problem-dependent	$\leq m$	Medium

A. Vertex propagation

The vertex propagation method [33] is a straightforward way to project intervals through the code, by projecting a number of input combinations given by the Cartesian product of the interval bounds. This results in a total of $n = 2^d$ evaluations, where d is the number of interval-valued input parameters. In the case of two intervals, x^I and y^I, the code, $f(\cdot)$ must be evaluated four times at $f(\underline{x}, \underline{y})$, $f(\underline{x}, \overline{y})$, $f(\overline{x}, \underline{y})$, $f(\overline{x}, \overline{y})$.

Despite its simplicity, this method suffers from two major disadvantages. The first problem is the exponential growth in computational complexity with respect to the number of parameters. For $d = 10$, a modest dimensionality by today's standards, vertex propagation requires $n = 1024$ evaluations and doubling the number of parameters to a somewhat more realistic figure results in $n = 1\,048\,576$, too many even for codes of moderate running times. The second problem with the method is that it gives bounds containing the true answer, i.e. it is rigorous, only when the function is monotonic. For many complex codes this is not the case, or at the very least cannot be readily established as a fact.

B. Subinterval reconstitution

To accommodate the common presence of non-monotonic trends, the input intervals can be partitioned into smaller subintervals, which can then be propagated through the model using vertex propagation and the output interval reassembled [34]. The logic behind this method is that even though the code may not be monotonic over the full width of the interval, it will likely behave monotonically on a smaller interval, provided the underlying function is not pathologically rough (i.e. it is not nowhere differentiable). Thus, output intervals for even highly non-linear functions can be computed to arbitrary precision, provided there is a sufficient computational budget.

The major disadvantage of this method is the exponential growth of evaluations required to compute output intervals, which is at least as high as that associated with the vertex propagation method. The reason is that for every interval there will be at least two and realistically many more points to compute the Cartesian product over. For example, a problem with four interval variables, partitioned into four subintervals each (that is, having five vertices per dimension) will require $n = 625$ evaluations. In general, the number of evaluations required is equal to $n = (m+1)^d$, where m is the number of subintervals.

One redeeming feature for the method is that there is no requirement to select the same number of subintervals for all parameters. This becomes very useful when the modeller has some knowledge about the behaviour of the function in some dimensions. In this way it is possible to allocate computational budget where it matters.

C. Monte Carlo methods

The Monte Carlo method, named after the home city of chance, has been developed for cases in which input uncertainty described by precise distributions needs to be propagated through a mathematical expression. The distributions are sampled and the scalars are propagated through the model. In this way, Monte Carlo can compute the empirical probability distribution for the model output, expressed either as a histogram (empirical PDF), or as an empirical CDF. In general, the cost of the method is not directly dependent on the number of dimensions, but instead implicitly scales with the *complexity* of the function.

Overall, Monte Carlo methods are general methods for approximate uncertainty propagation. Due to their relative simplicity, they remain popular for propagating uncertain numbers through black-box models. However, the engineer must remember that all results are based on a finite sample and can thus provide no guarantee about their reliability, unless repeated resampling is employed.

Even though Monte Carlo and other sampling methods are best suited to propagate the uncertainty of inputs only affected by aleatory uncertainty, their logic can be used for approximate propagation of epistemic uncertainty, too. We must emphasise here, that using Monte-Carlo-like approaches to propagate epistemic uncertainty will in fact have the effect of turning the intervals into uniform distributions. However, in contrast to the usual treatment described in Section I.C, here we are not concerned with the full distribution of the output, because it would be based on the untenable assumption of equal likelihood of outcomes in the range. Instead the minimum and maximum value of the resulting output values are computed and used as bounds on the output interval. Because the probability of sampling from the precise locations of the true bounds of the functions, using random sampling is 0, Monte Carlo methods provide an *inner* approximation to the true interval. For this reason, the associated solutions are generally considered to be of low rigor, but for the benefit of relaxing all assumptions about the computer model. In general, these methods underestimate the width of the output interval, and this underestimation error is found to increase with the dimensionality of the problem [35]. Moreover, often times output bounds may correspond to the bounds of one or more of the inputs, as discussed in Section II.A. However, these bounds will almost surely not be included in the Monte Carlo sample. A straightforward way to remedy this is to explicitly include the endpoints of the input intervals in the sample as proposed in the vertex propagation method (see Section III.B for the effect of this modification).

There has been a variety of sampling methods based on the logic of Monte Carlo. For the purposes of the current paper we are going to mention the two most commonly used ones.

1. Classical Monte Carlo

Classical Monte Carlo simulation [36] relies on drawing independent, pseudo-random samples from a precise probability distributions in m locations (the so-called independent and identically distributed (i.i.d.) samples). The popularity of classical Monte Carlo simulation can be attributed to the ease of its implementation and its natural applicability to propagate uncertain input parameters characterised by precise probability distribution. Moreover, most modern computational tools offer a built-in capability to generate pseudo-random numbers from the standard uniform distribution, $\mathcal{U}(0, 1)$. This can be rescaled to any other uniform distribution with bounds a and b, $\mathcal{U}(a, b)$. To sample distributions different from uniform, many tools offer an implementation of the *probability integral transform*, which transforms uniform pseudo-random numbers through the inverse cumulative (quantile) distribution function of the required distribution [see e.g., 37]. If the quantile function is not available in closed-form, samples can be generated through more advanced techniques; see e.g., [38, 39].

One of the main disadvantages of classical Monte Carlo is due to its reliance on random sampling across the entire support of the distribution. This results in some regions containing clusters of samples, while others, typically around the distribution tails, remaining under-explored .

2. Latin hypercube sampling

Latin hypercube sampling (LHS) [40] attempts to address the main disadvantage of classical Monte Carlo by first partitioning the range of the input probability distributions into m intervals and then randomly selecting a single point from within each interval. The samples are permuted, such that the set fills the space optimally, according to a prescribed criterion (usually maximin or minimax [41]).

D. Cauchy-deviate method

If there is evidence to believe that the investigated function is reasonably linear (that is ignoring second- and higher-order terms in a hypothetical Taylor expansion will not lead to significant errors), another way to propagate epistemic uncertainty through black-box models is to use the method of Cauchy deviates [42].

This method works by generating m random samples from d Cauchy distributions, one for each uncertain number and computing the width of the output interval through appropriate scaling of these samples [43]. Because the Cauchy distribution is unbounded, after scaling them, all samples will lie on the $d - 1$-dimensional hyperplane bounds of the input space. This allows the method to provide more accurate results than Monte Carlo, for cases where the linearity assumption holds. This assumption, however, raises two questions:

- "If we can assume the model is linear, why can not we use the vertex propagation method, which is rigorous in this case?"
- "Is not the model supposed to be black box? How can a linearity assumption be justified?"

The vertex method indeed provides rigorous interval propagation for linear or monotonic models. However, as stated in Section II.A, its cost is exponential in the number of dimensions. This is not true for the Cauchy-deviate method, whose computational cost is constant in d. It is thus up to the engineer to decide on the computational budget and design the propagation work.

To answer the second question, the mathematical surface of the typical engineering model is not pathologically rough. This means that, if the input interval is narrow enough, the model will behave approximately linearly. This makes it possible to combine partitioning methods, such as those used in subinterval reconstitution, with Cauchy deviates to control the computational expense of uncertainty propagation.

E. Optimisation methods

Interval input uncertainty can also be propagated through black-box models by solving two global constrained optimisation problems for the bounds of the output interval, as:

$$\begin{aligned} y^I &= \left[\underline{y}, \overline{y}\right] \\ &= \left[\min_{x_1^I,\ldots,x_d^I} f\left(x_1^I,\ldots,x_d^I\right), \max_{x_1^I,\ldots,x_d^I} f\left(x_1^I,\ldots,x_d^I\right)\right] \end{aligned} \tag{2}$$

Unlike random sampling methods, optimisation techniques query the model at points chosen from within the input intervals based on some fitness function to inform where to sample next. A detailed taxonomy of optimisation methods can be consulted, for example in [44, 45]. In this study, only evolutionary algorithms for global optimisation are considered, due to their general suitability to black-box applications. This large family of algorithms is inspired by the process of reproduction through natural selection. As such, evolutionary algorithm share common characteristics with sampling methods, with the difference that they are concentrated on finding a global optimum instead of generating the empirical distribution of the quantity of interest. Their key differences when compared to other optimisation methods is that for each iteration they initialise, vary and select a population of candidate solutions. Moreover, the selection of the population for each iteration is based on random processes. Evolutionary algorithms are generic methods and can be used for a variety of demanding complex models, including those with high-dimensional input spaces, highly nonlinear interaction between model parameters, non-continuous parameter space (e.g., an integer or logical discrete parameter), and multi-modal objective function with many local maxima or minima. Examples of evolutionary algorithms are the genetic algorithm, particle swarm optimisation, and ant colony optimisation, among others.

Although these methods reliably produce better bounds than other sample-based methods, such as those based on Monte Carlo simulation, the resulting intervals are not guaranteed to be correct. Moreover, evolutionary optimisation methods sample a large number of candidate solutions for each iteration. This means that depending on the computational cost of the model, they can be prohibitively expensive.

III. A case study in propulsion system engineering

In this section, we demonstrate the non-intrusive uncertainty propagation methods in the context of the preliminary structural design of an aircraft propulsion system. The centrepiece of the design problem we tackle is the requirement for a lightweight mounting structure designed to support an electric motor to the main spar of a wing. Despite the best intentions of the design team to achieve minimum weight for the structure, considerable uncertainty surrounds its main sizing parameters, giving rise to the risk of producing an lightweight structure which may require substantial changes later on in the design process. A description of the model is presented first, followed by the discussion of the results.

A. The model

In line with the procedural geometry modelling approach, described in Section I.B, we construct the model of the motor mount through a hybrid of visual and code-based programming. The computational infrastructure at

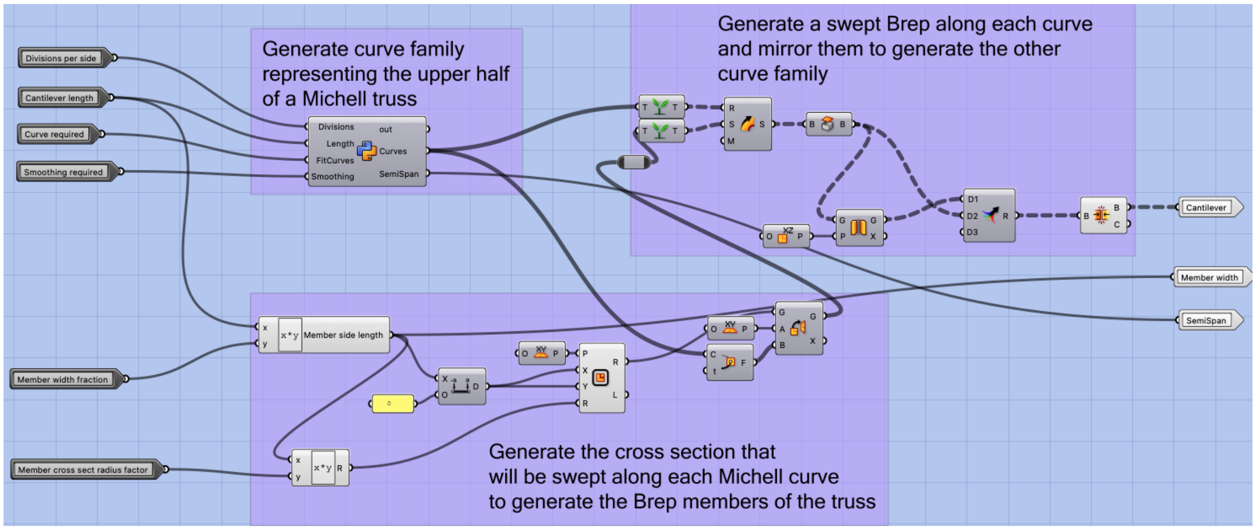
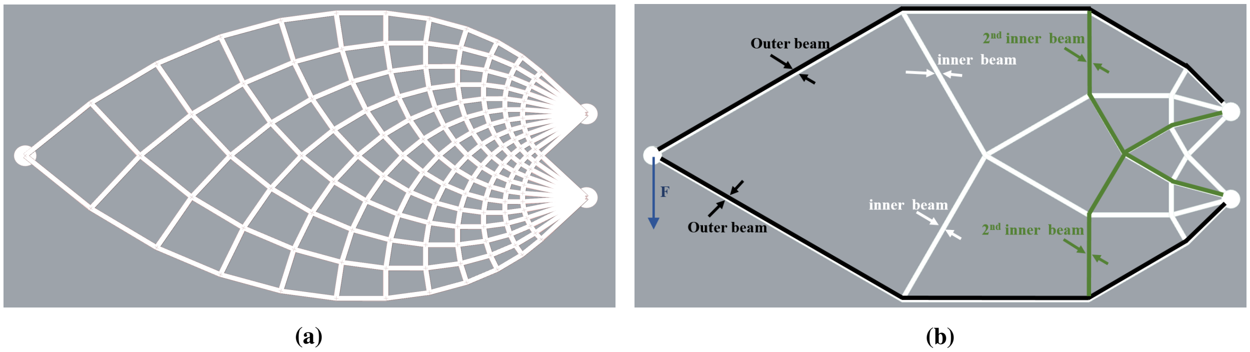

Divisions per side
Cantilever length
Curve required
Smoothing required
Generate curve family representing the upper half of a Michell truss
Divisions
Length
FitCurves
Smoothing
out
Curves
SemiSpan
Generate a swept Brep along each curve and mirror them to generate the other curve family
Cantilever
Member width
SemiSpan
Member width fraction
x*y
Member side length
Member cross sect radius factor
x*y
Generate the cross section that will be swept along each Michell curve to generate the Brep members of the truss

Fig. 1 Design canvas showing the components of the visual program that defines the parametric geometry at the heart of the case study described here.



(a)

(b)

Fig. 2 The Michell lattice used to construct the motor mount. (a) An example Michell lattice. (b) Geometric parameterisation of a lattice with 4 divisions.

the foundation of the demonstrator is the openNURBS® library, run through Rhino® 7 via its Grasshopper® visual programming interface [46], illustrated in Fig. 1.

The motor mount is a "cage", each of whose four sides is a Michell lattice, such as the one shown in Fig. 2(a). The complex geometry of each lattice can be created with relative ease through the principles of programmatic geometry generation, where topological features can be implemented as variables just as easily as physical dimensions. In particular, the geometry of each lattice is created by the algorithm proposed by Chan [47], which implements the mechanical principles developed by Michell [48].

In this case study, the mount is made out of stainless steel, with nominal values for density, $\rho = 7853\,\mathrm{kg/m^3}$, Poisson's ratio, $\nu = 0.29$, Young's modulus, $E = 205\,\mathrm{GPa}$ and yield stress, $\sigma_y = 1\,\mathrm{GPa}$. The overall length of the motor mount is $L = 1.9\,\mathrm{m}$. We assume these values are not subject to uncertainty. The four sides of the motor mount are identical. The topology of each Michell lattice is described by four geometric parameters. With respect to Fig. 2(b), these are the number of divisions, D, making up each beam (continuous line) of the lattice; the ratio, R_{out}, between the width of the outer beams (black) and the overall length of the lattice, L; the ratio between the width of the inner beams of the lattice (white), excluding the second beams, and the width of the outer beams, R_{in}; and the ratio between the width of the second inner beams (green) and the width of the outer beams, R_{in_2}. The width of the second inner beams is defined separately because they experience the highest axial stresses for many of the topologies considered. The lattice is subject to a downward force F, which is also uncertain due to uncertainty in the mass of the power plant and aircraft loading factor. The depth of the lattice is set by a maximum stress and deflection constraints under the influence of

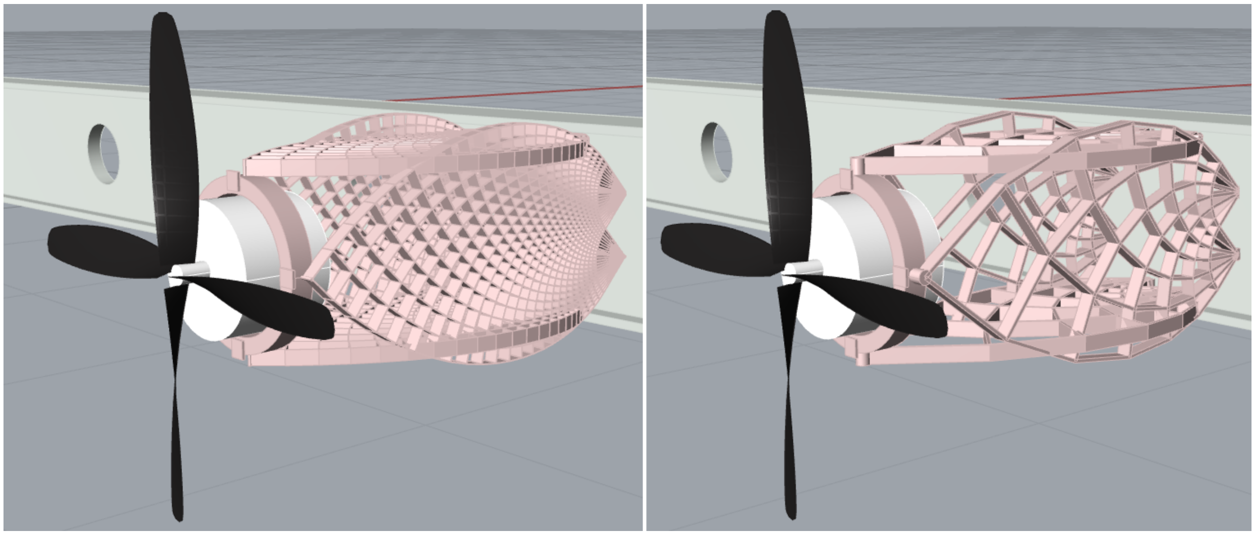


Fig. 3 Two instances of the parametric motor mount at the centre of the design case study presented in this study.

F. The beam depth calculation is based on structural analysis performed with the depth initially set equal to the outer beams width, i.e. the outer beams having a square cross section. Based on structural simulation results the minimum required depth is calculated for compliance with the maximum allowed beam deflection of 50 mm and the material yield strength. Figure 3 shows two instances of the complete parametric model, illustrating the impact of some of the geometric input parameters on the resulting topology of the lattices. The panel on the left shows a high-density lattice with 24 divisions and thin beams, whilst the panel on the right depicts a lattice with 7 divisions and thicker beams.

To reduce the computational cost of the model, linear static beam analysis is performed on a single vertical lattice, using the parametric structural tool Karamba3D [49]. Isolating a single lattice allows for the engine system forces and torques to be translated into a resultant in-plane loading. This, in turn, makes possible the use of beam analysis tools, such as Karamba3D. Karamba3D is fully embedded in the parametric design environment of Grasshopper. The reliability of its calculations was verified by comparing the predicted deflection and maximum stress, for a range of parameters, to those produced by the high fidelity finite element structural software ABAQUS [50]. As shown in Fig. 4, the one-dimensional beam simulations performed with Karamba3D, which run in milliseconds, produce results matching those from ABAQUS, for both displacement and stress, to within levels acceptable for this case study. Because of this, no multi-fidelity model management strategy is required [51].

B. Analyses and Results

Having created a simple and reliable Michell lattice model, the next step is to quantify the uncertainty around its structural mass, given the uncertainty in the geometric density of the lattice, the cross sectional geometries and the applied load. To identify suitable methods to propagate the uncertainty in the input to the structural mass, the characterisation of the uncertainty in the input parameters is required. The uncertain input parameters in this case study will eventually have constant scalar values, but at the preliminary stage of the design process, these values are not known. This can be either because we would like to keep the design open for as long as feasibly possible (known as volitional uncertainty), or because we do not have sufficient evidence to determine these values . Both situations lead to the presence of epistemic uncertainty which is likely to be reduced at later stages of the design process as more information becomes available. The best and, often, most intuitive way for engineers to express this type of uncertainty is to use their judgement and provide intervals which are likely to include the exact value of each input parameter. To construct an interval, all engineers need to do, is provide a minimum and maximum value for each input parameter without making any assumption regarding the frequency of the values in these bounds.

In the light of the above, the geometric density of the lattice is determined by the number of beams which for this study is assumed to be in the range $D = [4, 24]$. The cross-sectional geometry is determined by the width of each beam and its depth. The width of the outer beams is estimated as the fraction of the motor mount's length, and is assumed to

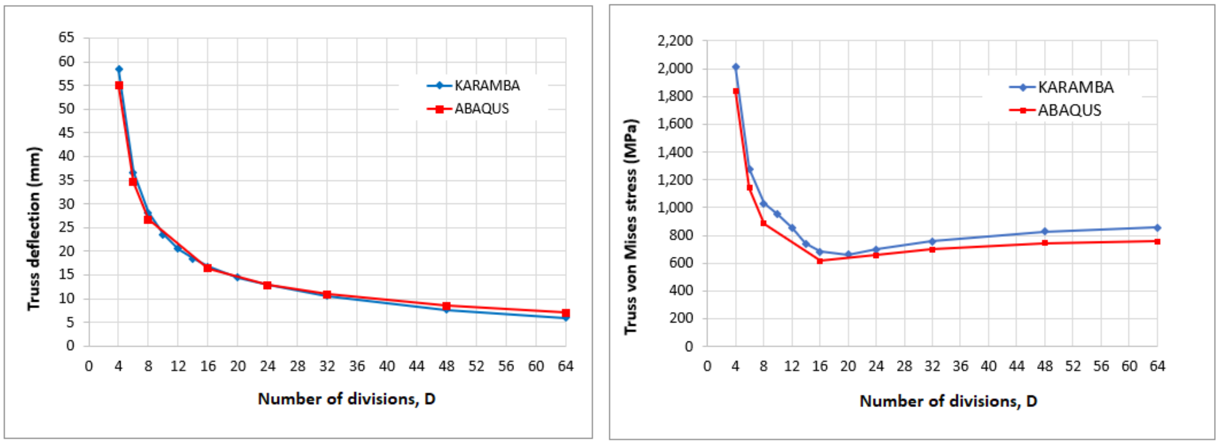


Fig. 4 Results from the Karamba3D structural beam analysis, compared to high-fidelity ABAQUS simulations, assuming identical square cross-sections for all beams with $R_{out} = 0.0020$ and load $F = 10\,\text{kN}$. Results from the fast, inexpensive model provide sufficiently accurate results for this study.

range between $R_{out} = [0.002, 0.008]$. The width of each beam is assumed constant throughout its length but is allowed to vary between the beams, highlighted in Fig. 2(b), as follows. The width of the inner beams over the width of the main outer beams is allowed to vary in the range $R_{in} = [0.25, 1.00]$. The width of the second inner beams over the width of the main outer beam is also allowed to range in $R_{in_2} = [0.25, 1.00]$. Finally, maximum load applied to the structure is assumed to vary between $F = [15, 25]$kN.

The uncertainty in the five input parameters is propagated through the Michell lattice model to estimate the uncertainty around the mass (in kg) of the lattice. All six propagation methods, presented in Section II, are applied here to demonstrate their performance. The evolutionary optimisation is performed using the Galapagos plug-in [52]. With regard to the remaining propagation methods, the sample of input parameters is first generated by code written in Python and the iterations are carried out using the Anemone plug-in [53]. The results for the studied methods are compared in Table 2 along with their associated computational cost. It should be reiterated here that all results can be considered approximate, as the methods are applied to a black-box model.

Overall, evolutionary optimisation yields the widest intervals of structural mass. This outcome, is not surprising, since optimisation approaches to interval propagation actively search for the bounds of the output interval. The additional width however, comes at a relatively high computational cost, when compared to other methods, as this method requires two optimisation rounds with a total of over 7000 iterations. Fig. 5(a)-(b) depicts the realisations of the Michell truss having the minimum and maximum structural mass, obtained via evolutionary optimisation.

By contrast, the Cauchy-deviate method, with 3000 samples yields the narrowest interval for structural mass. Interestingly, the latter interval includes a negative lower bound, which, of course, does not make any physical sense. This counter-intuitive outcome can be attributed to the violation of the Cauchy deviates' key assumption that the model is linear over the input intervals and for this reason the results from this method are unlikely to improve by increasing the number or iterations. As seen in Table 2, this method is the only one whose results are not associated with particular input combinations. This is an artifact of the numerical root-finding step of the algorithm [42] and serves as a reminder for its purely mathematical nature. In this regard, the Cauchy-deviate method is a true epistemic uncertainty propagation method, as it simply reports the width of the interval at the output of a function, when all we know are the bounds of the multidimensional box containing its input.

All the remaining propagation methods produce intervals contained in the one computed via evolutionary optimisation. In particular, vertex propagation is the fastest method requiring only 32 iterations for the studied problem with five uncertain input parameters. Similar to Cauchy deviates, this method is also based on a bold assumption, specifically that the studied model behaves monotonically. Table III.B shows that it performs much better than the Cauchy-deviate method but produces a narrower interval than evolutionary optimisation. Compared to the latter, it can be noted that vertex propagation notably overestimates the minimum structural mass by 11% and underestimates its maximum value by 4%. A small increase in the width of the structural mass intervals can be achieved by further dividing the intervals into four sub-intervals and perform sub-intervals reconstitution requiring 3125 iterations. For this method the overestimation of the minimum structural mass reduces to 5% compared to its counterpart obtained by evolutionary optimisation. Fig. 5(c)-(d) depicts the realisations of the Michell truss corresponding to the minimum and maximum mass for this method. The use of higher number of divisions could further improve the results, but will also increase the computational cost. For example, the division in five sub-intervals instead of four will require 7776 iterations, a cost comparable to that of evolutionary optimisation.

With regard to Monte Carlo methods, classical Monte Carlo with 5000 iterations appears to yield significantly narrower intervals for the structural mass than evolutionary optimisation. Compared to the latter method, classical Monte Carlo overestimates the minimum mass by 9% and underestimate the maximum mass by 25%. This is not surprising, as classical Monte Carlo is known to perform poorly, underestimating the uncertainty, when propagating intervals. A Latin hypercube sampling has also been applied with 5000 iterations and yielded a somewhat wider interval than classical Monte Carlo. Compared to evolutionary optimisation, Latin hypercube overestimates the minimum mass by 5% and underestimate the maximum by 21%. This minor improvement is expected given that Latin hypercube sampling performs better than Monte Carlo in evenly distributing samples from the uniform distributions. Nonetheless, both methods are associated with a high computational cost and substantially underestimate the maximum structural mass.

To remedy this, the samples of each Monte Carlo method are combined with the endpoints used in vertex propagation, and the minimum and maximum of the structural mass is reassessed. This way, a caveat of the Monte Carlo methods (see section II.C), namely that the samples do not include the bounds of one or more of the inputs, is mitigated. The combination of the two methods widens the intervals of structural mass. The combination of Latin hypercube sampling with vertex propagation yields intervals marginally wider than the ones produced by sub-interval reconstitution. It is interesting to note that the combination of classical Monte Carlo with vertex propagation produces intervals that are only marginally wider than those obtained via vertex propagation.

Jointly these results allow us to infer two main characteristics about the behaviour of the mass of the Michell lattice. The first one is that the function is non-monotonic, as evidenced by the narrower intervals produced by methods that rely on the monotonicity assumption (vertex propagation, Cauchy-deviates), when compared to those that do not (evolutionary optimisation). The second one is that the non-linear behaviour of significance to mass extrema occurs close to the edges of the input hyper-box, as seen by the narrower intervals produced by methods which only sample the interior of the input box (Monte Carlo, LHS), as compared to those that include its edges (e.g., vertex propagation, vertex propagation and sampling). Naturally, during aircraft design work, the engineer will likely not be able to afford to test out all possible methods and, so, engineering judgement must be used to reason about the best method to apply, considering all the engineer knows about the model. The use of intervals as an uncertainty propagation primitive is ultimately about practical conservatism and so is the selection of the most suitable propagation method.

This last point may raise the somewhat obvious question of using surrogate models to alleviate the computational burden of interval propagation methods. Indeed such a case is considered in [32]. In this study we have deliberately neglected the use of surrogate models in order to emphasise the original problem. We note that surrogate models are a powerful method for increasing the efficiency of general UQ, but must be used with due caution and in a way which does not compromise the effort put in characterising the uncertainty elsewhere in the design process. A subsequent study will focus on the use of and uncertainty quantification for surrogate models for epistemic uncertainty propagation.

IV. Conclusions

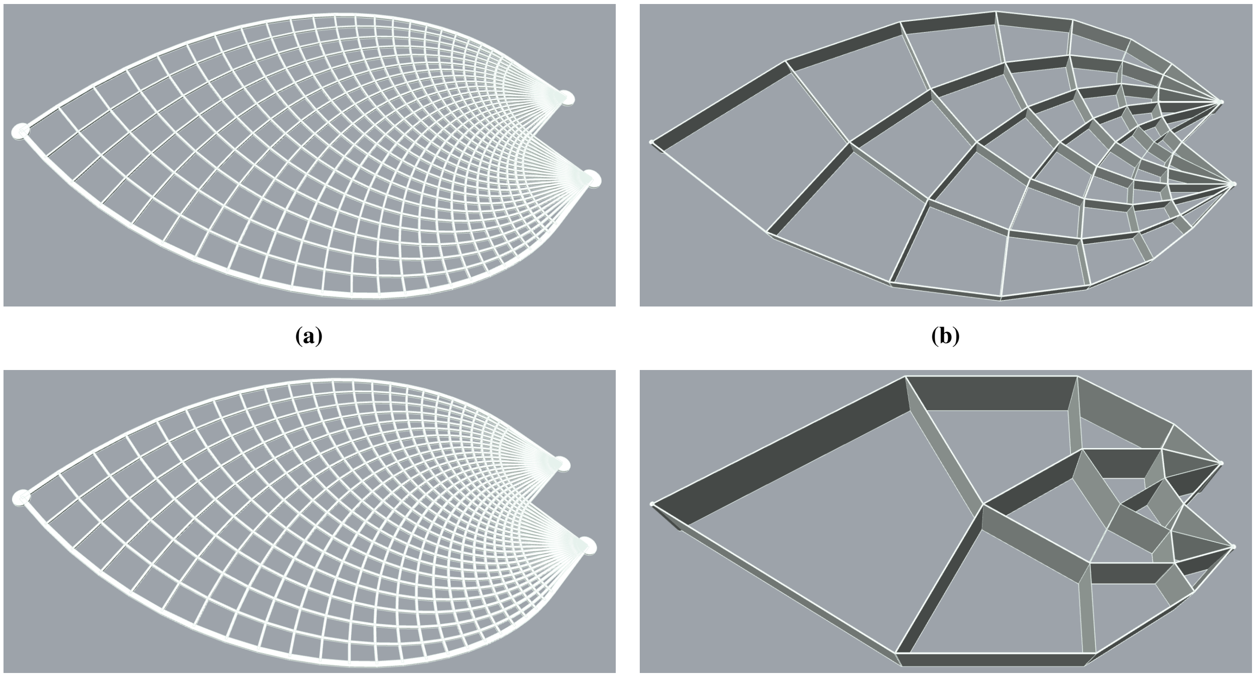
In this study, we presented the foundations of a framework which can be used to propagate the epistemic uncertainty prevalent in preliminary and early subsystems design, through black-box models. This type of uncertainty is best expressed in terms of intervals and should, therefore, be propagated as intervals. Five different propagation methods were applied to a procedurally-generated geometric model of an electric motor support structure.

The use of multiple uncertainty propagation methods allowed us to explore the theoretical assumptions upon which each method is constructed. Evolutionary optimisation was found to be the method which provided the widest interval around the mass of the support structure. This is good news as optimisation is a generic technique well-understood and widely used by aircraft designers. However, the evolutionary optimisation is the most computationally expensive technique and eve so, there is no guarantee as to the reliability of the computed mass intervals. Coming up with guaranteed solutions for the propagation of uncertainty through black-box models is very much an open problem.

The simplified model adopted in this study meant that the computational cost was manageable. However, more research is needed in order to reduce the computational cost through the use of surrogate models and make the method applicable to a larger class of problems irrespective of their computational complexity.

Table 2 Intervals of the Michell lattice structural mass produced by the adopted methods and the input values $(D, R_{out}, R_{in}, R_{in_2}, F)$ corresponding to the minimum and maximum mass. The intervals are ranked from the highest to lower width. Note that the negative lower bound on mass, computed via the Cauchy-deviate method reflects a potential failure of the linearity assumption of the algorithm.

Method	n	M^I(kg)	**Input ($\underline{M}$)**	**Input ($\overline{M}$)**
Evolutionary optimisation	$7,257$	$[1.7, 18.7]$	$(24, 0.0072, 0.34, 0.47, 15.0)$	$(7, 0.0020, 1.00, 0.25, 25.0)$
Latin hypercube + vertex	$5,032$	$[1.8, 18.0]$	$(24, 0.0073, 0.31, 0.42, 15.6)$	$(4, 0.0020, 1.00, 0.25, 25.0)$
Sub-intervals reconstitution	$3,125$	$[1.8, 18.0]$	$(24, 0.0080, 0.44, 0.44, 15.0)$	$(4, 0.0020, 1.00, 0.25, 25.0)$
Monte Carlo + vertex	$5,032$	$[1.9, 18.0]$	$(23, 0.0074, 0.45, 0.65, 15.4)$	$(4, 0.0020, 1.00, 0.25, 25.0)$
Vertex propagation	32	$[1.9, 18.0]$	$(24, 0.0080, 0.25, 0.25, 15.0)$	$(4, 0.0020, 1.00, 0.25, 25.0)$
Latin hypercube	$5,000$	$[1.8, 14.8]$	$(24, 0.0073, 0.31, 0.42, 15.6)$	$(7, 0.0021, 1.00, 0.33, 24.9)$
Monte Carlo	$5,000$	$[1.9, 14.0]$	$(23, 0.0074, 0.45, 0.65, 15.4)$	$(7, 0.0029, 0.97, 0.31, 23.3)$
Cauchy deviates	$3,000$	$[-0.3, 8.2]$	-	-



(a)

(b)

(c)

(d)

Fig. 5 Michell lattices corresponding to the minimum (left column) and maximum (right column) mass obtained by sub-intervals reconstitution (top row) and evolutionary optimisation (bottom row).

Acknowledgments

The Development of Advanced Wing Solutions (DAWS) 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 the ATI, Department for Business, Energy & Industrial Strategy (BEIS) and Innovate UK, addresses technology, capability and supply chain challenges.

References

[1] Yodo, N., and Wang, P., "Resilience Allocation for Early Stage Design of Complex Engineered Systems," *Journal of Mechanical Design*, Vol. 138, No. 9, 2016.

[2] Padulo, M., "Computational engineering design under uncertainty: an aircraft conceptual design perspective," Ph.D. thesis, School of Engineering, Cranfield University, 2009.

[3] Gray, A., Wimbush, A., de Angelis, M., Hristov, P., Calleja, D., Miralles-Dolz, E., and Rocchetta, R., "From inference to design: A comprehensive framework for uncertainty quantification in engineering with limited information," *Mechanical Systems and Signal Processing*, Vol. 165, 2022, p. 108210.

[4] Bianchi, D. D., Sêcco, N., and Silvestre, F., "A framework for enhanced decision-making in aircraft conceptual design optimisation under uncertainty," *The Aeronautical Journal*, Vol. 125, No. 1287, 2021, p. 777–806.

[5] O'Hagan, A., and Oakley, J. E., "Probability is perfect, but we can't elicit it perfectly," *Reliability Engineering & System Safety*, Vol. 85, No. 1, 2004, pp. 239–248.

[6] Ferson, S., and Ginzburg, L. R., "Different methods are needed to propagate ignorance and variability," *Reliability Engineering and System Safety*, Vol. 54, No. 2, 1996, pp. 133–144.

[7] Der Kiureghian, A., and Ditlevsen, O., "Aleatory or epistemic? Does it matter?" *Structural Safety*, Vol. 31, No. 2, 2009, pp. 105–112.

[8] Bashkirov, I., Chernyshev, S., and Veresnikov, G., "Parametric synthesis optimization models for high speed transport aerodynamic design to comply with flight safety and low environmental impact requirements," *Acta Astronautica*, 2022.

[9] Duca, R., Sarojini, D., Bloemer, S., Chakraborty, I., Briceno, S. I., and Mavris, D. N., "Effects of Epistemic Uncertainty on Empennage Loads During Dynamic Maneuvers," *2018 AIAA Aerospace Sciences Meeting*, 2018.

[10] Phillips, B. D., and West, T. K., "Aeroelastic Uncertainty Quantification of a Low-Boom Aircraft Configuration," *2018 AIAA Aerospace Sciences Meeting*, 2018.

[11] Molina-Cristobal, A., Nunez, M., Guenov, M., Laudan, T., and Druot, T., "Black-box model epistemic uncertainty at early design stage. An aircraft power-plant integration case study," *29th Congress of the International Council of the Aeronautical Sciences, ICAS*, 2014.

[12] Zaman, K., and Mahadevan, S., "Robustness-Based Design Optimization of Multidisciplinary System Under Epistemic Uncertainty," *AIAA Journal*, Vol. 51, No. 5, 2013, pp. 1021–1031.

[13] Padulo, M., Campobasso, M. S., and Guenov, M. D., "Novel Uncertainty Propagation Method for Robust Aerodynamic Design," *AIAA Journal*, Vol. 49, No. 3, 2011, pp. 530–543.

[14] Liu, D., Litvinenko, A., Schillings, C., and Schulz, V., "Quantification of Airfoil Geometry-Induced Aerodynamic Uncertainties—Comparison of Approaches," *SIAM/ASA Journal on Uncertainty Quantification*, Vol. 5, No. 1, 2017, pp. 334–352.

[15] Wu, X., Zhang, W., and Song, S., "Uncertainty Quantification and Sensitivity Analysis of Transonic Aerodynamics with Geometric Uncertainty," *International Journal of Aerospace Engineering*, Vol. 2017, 2017, p. 8107190.

[16] Zhu, J., and Qiu, Z., "Interval analysis for uncertain aerodynamic loads with uncertain-but-bounded parameters," *Journal of Fluids and Structures*, Vol. 81, 2018, pp. 418–436.

[17] Sabater, C., Bekemeyer, P., and Görtz, S., "Efficient Bilevel Surrogate Approach for Optimization Under Uncertainty of Shock Control Bumps," *AIAA Journal*, Vol. 58, No. 12, 2020, pp. 5228–5242.

[18] Barklage, A., Römer, U., Bekemeyer, P., Bertram, A., Himisch, J., Radespiel, R., and Badrya, C., "Analysis and uncertainty quantification of a hybrid laminar flow control system," *AIAA SCITECH 2022 Forum*, 2022, p. 2452.

[19] Gumbert, C., Newman, P., and Hou, G., "Effect of Random Geometric Uncertainty on the Computational Design of a 3D Flexible Wing," *20th AIAA Applied Aerodynamics Conference*, 2002, p. 2806.

[20] Marques, S., Badcock, K., Khodaparast, H., and Mottershead, J., "CFD Based Aeroelastic Stability Predictions Under the Influence of Structural Variability," *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2009, pp. 2699–2721.

[21] Garzon, V. E., and Darmofal, D. L., "Impact of Geometric Variability on Axial Compressor Performance ," *Journal of Turbomachinery*, Vol. 125, No. 4, 2003, pp. 692–703.

[22] Kim, H., and Liou, M.-S., "Shape design optimization of embedded engine inlets for N2B hybrid wing-body configuration," *Aerospace Science and Technology*, Vol. 30, No. 1, 2013, pp. 128–149.

[23] Reist, T. A., and Zingg, D. W., "High-Fidelity Aerodynamic Shape Optimization of a Lifting-Fuselage Concept for Regional Aircraft," *Journal of Aircraft*, Vol. 54, No. 3, 2017, pp. 1085–1097.

[24] Sóbester, A., "A Case for Procedural Geometry Generation in Aircraft Design," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017. URL https://arc.aiaa.org/doi/abs/10.2514/6.2017-4439.

[25] Jaynes, E., "Information theory and statistical mechanics," *Physical Review*, Vol. 106, 1957, pp. 620–630.

[26] Bedford, T., and Cooke, R., *Probabilistic Risk Analysis: Foundations and Methods*, Cambridge University Press, 2001.

[27] Moore, R. E., *Methods and applications of interval analysis*, SIAM, 1979.

[28] Caprani, O., Madsen, K., and Nielsen, H., *Introduction to Interval Analysis*, DTU, 2002.

[29] Zadeh, L. A., "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, Vol. 100, 1999, pp. 9–34.

[30] Ferson, S., Kreinovich, V., Ginzburg, L., and Sentz, K., "Constructing Probability Boxes and Dempster-Shafer Structures, SAND2002-4015," Tech. rep., Sandia National Laboratories (SNL), Albuquerque, (NM), United States, 2003.

[31] Faes, M. G. R., and Moens, D., "Recent Trends in the Modeling and Quantification of Non-probabilistic Uncertainty," *Archives of Computational Methods in Engineering*, Vol. 27, 2019, pp. 633–671.

[32] Cicirello, A., and Giunta, F., "Machine learning based optimization for interval uncertainty propagation," *Mechanical Systems and Signal Processing*, Vol. 170, 2022, p. 108619.

[33] Dong, W., and Shah, H. C., "Vertex method for computing functions of fuzzy variables," *Fuzzy Sets and Systems*, Vol. 24, No. 1, 1987, pp. 65–78.

[34] Ferson, S., and Hajagos, J. G., "Arithmetic with uncertain numbers: rigorous and (often) best possible answers," *Reliability Engineering & System Safety*, Vol. 85, No. 1, 2004, pp. 135–152.

[35] De Angelis, M., S., F., E., P., and Kreinovich, V., "Black-box propagation of failure probabilities under epistemic uncertainty," *Proceedings of the 3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2019)*, 2019, pp. 713–723.

[36] Ang, A., and Tang, W., *Probability Concepts in Engineering: Emphasis on Applications to Civil and Environmental Engineering, 2e Instructor Site*, Wiley, 2007.

[37] Bishop, C., *Pattern Recognition and Machine Learning*, Springer New York, 2016.

[38] Gilks, W. R., and Wild, P., "Adaptive Rejection Sampling for Gibbs Sampling," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 41, No. 2, 1992, pp. 337–348.

[39] Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D., *Bayesian Data Analysis, Third Edition*, Taylor & Francis, 2013.

[40] McKay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code." *Technometrics*, Vol. 21, 1979, pp. 239–245.

[41] Morris, M. D., and Mitchell, T. J., "Exploratory designs for computational experiments," *Journal of Statistical Planning and Inference*, Vol. 43, No. 3, 1995, pp. 381–402.

[42] Kreinovich, V., and Trejo, R., "Error estimations for indirect measurements: Randomized vs. deterministic algorithms for "black box" programs, UTEP-CS-00-17," Tech. rep., University of Texas, El Paso, (TX), United States, 2000.

[43] Kreinovich, V., and Ferson, S., "A new Cauchy-based black-box technique for uncertainty in risk analysis," *Reliability Engineering & System Safety*, Vol. 85, No. 1, 2004, pp. 267–279.

[44] Nocedal, J., and Wright, S., *Numerical Optimization*, Springer New York, 2006.

[45] Stork, J., Eiben, A. E., and Bartz-Beielstein, T., "A new taxonomy of global optimization algorithms," *Natural Computing*, Vol. 5, No. 1, 2017, pp. 334–352.

[46] Rutten, D., "Grasshopper," , 2022. URL https://www.grasshopper3d.com/.

[47] Chan, A. S. L., "The Design of Michell Optimum Structures," *Her Majesty's Stationery Office, Ministry Of Aviation Aeronautical Research Council Reports and Memoranda*, Vol. No. 3303, 1960.

[48] Michell, A. G. M., "The limits of economy of material in frame-structures," *Philosophical Magazine*, Vol. 8, No. 47, 1904, pp. 589–597.

[49] Preisinger, C., and Heimrath, M., "Karamba—A Toolkit for Parametric Structural Design," *Structural Engineering International*, Vol. 24, No. 2, 2014, pp. 217–221.

[50] "ABAQUS, Version 2021," , 2021. URL https://www.3ds.com/products-services/simulia/products/abaqus/abaquscae/.

[51] Peherstorfer, B., Willcox, K., and Gunzburger, M., "Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization," Tech. rep., University of Wisonsin-Madison, 2018.

[52] Rutten, D., "Galapagos: On the Logic and Limitations of Generic Solvers," *Architectural Design*, Vol. 83, No. 2, 2013, pp. 132–135.

[53] Zwierzycki, M., "Digital evolutionary tools and their uses in the process of forming the spatial structures (in Polish)," *Archivolta*, Vol. 2, 2013, pp. 54–61.