NPL Report CMSC 40/04
(CMSC 10/01 revised)

Report to the National Measurement System Directorate, Department of Trade and Industry

From the Software Support for Metrology Programme

Software Specifications for Uncertainty Evaluation

By
M G Cox and P M Harris

March 2004

Software Specifications for Uncertainty Evaluation

M G Cox and P M Harris
Centre for Mathematics and Scientific Computing

March 2004

ABSTRACT

This document provides specifications of software units for the evaluation of measurement uncertainty. It is intended to align with and support established guides and extend their functionality in a consistent manner. It is also intended to complement the best-practice guide to uncertainty and statistical modelling that has been produced as part of the UK Department of Industry's National Measurement System *Software Support for Metrology* (SS*f*M) programme. The target audience is those who in their work wish to use software to assist in the evaluation of uncertainty.

This document is a revised edition of a previous report. It contains additional material on automatic differentiation techniques for calculating sensitivity coefficients, and aspects of the Monte Carlo procedure for uncertainty evaluation including sampling from general distributions, shortest coverage intervals, nonlinear sensitivity coefficients, and an adaptive Monte Carlo procedure. The document also accounts for revision of the SS*f*M best-practice guide to uncertainty and statistical modelling, and the preparation of the first Supplemental Guide to the Guide to the Expression of Uncertainty in Measurement.

© Crown copyright 2004
Reproduced by permission of the Controller of HMSO

ISSN 1471–0005

Extracts from this report may be reproduced provided the source is acknowledged
and the extract is not taken out of context

Authorised by Dr Dave Rayner,
Head of the Centre for Mathematics and Scientific Computing

National Physical Laboratory,
Queens Road, Teddington, Middlesex, United Kingdom TW11 0LW

Contents

List of Tables

List of Figures

1 Introduction

This document is intended to align with and help support established guides [4, 15, 40, 41] on uncertainty evaluation and extend their functionality in a consistent manner. It is complementary to the best-practice guide [12] on uncertainty and statistical modelling, produced as part of the UK's Software Support for Metrology (SS*f*M) programme. The first edition of this best-practice guide was published in March 2001, having been developed during the first SS*f*M programme covering the period April 1998 – March 2001. During that period Working Group 1, "Expression of Uncertainty in Measurement", of the Joint Committee for Guides in Metrology (JCGM) started work, following its first meeting in March 2000, on the first *Supplemental Guide* [3] to the Guide to the Expression of Uncertainty in Measurement (GUM) [4] concerned with numerical methods for the propagation of distributions. Material from the evolving best-practice guide was used in various parts of the Supplemental Guide and subsequently refined appropriately for consistency with the remainder of the latter document. This revised report, produced during the second SS*f*M programme, April 2001 – March 2004, takes account of the revision of the best-practice guide and the preparation of the Supplemental Guide. In particular, material from the drafts of the Supplemental Guide prepared during the second programme that had an origin in the first edition of the best-practice guide has been re-used.

The purpose of this document is to provide specifications for software that is relevant to uncertainty evaluation and associated statistical analyses. The specifications relate to software *units* that are useful in this area, rather than packages or systems for uncertainty evaluation. The user or supplier that is implementing software for uncertainty evaluation would need to consider software units of the types covered here.

The software specifications are not intended to be mandatory. They typify constituent parts of uncertainty evaluation. In particular, they indicate the inputs and outputs of the software units and the purpose of each unit, viz., a statement of the computational aim of the unit. The units are presented in the context of the more complete calculations in which they would be used.

The specifications generally have a minimal number of input and output parameters. Computational control parameters, such as those relating to the convergence criteria of iterative techniques, are indicated only broadly. Diagnostic parameters, such as those that indicate failure or degree of success of the computation, are not included.

The scope of the specifications is those that relate to

1. The *law of propagation of uncertainty* (LPU) as summarised in Clause 8 of

the Guide to the Expression of Uncertainty in Measurement (GUM) [4][1]

2. More general calculations, consistent with the GUM, for the *propagation of distributions* based on the use of *Monte Carlo simulation* [3, 12]

3. The validation of (the use of) LPU [3, 12].

It is emphasised that the primary concern is the *specification* of relevant software. Algorithms and software *per se* are not the main consideration: the provision of such material is the responsibility of the user or supplier that is implementing software for uncertainty evaluation. However, two departures from this stance are made:

1. Because of the importance of being able to generate random numbers from a *rectangular* distribution that (a) have sound statistical properties in their own right, (b) can be used as a basis for the many generators that make use of a rectangular random number generator for their function, and (c) in certain circumstances can be re-generated *exactly*, a *specific* rectangular random number generator is recommended (Section 5.3). In addition, generators for Gaussian, t– and multivariate Gaussian distributions are regarded as so fundamental that algorithmic statements for them are included (also in Section 5.3).

2. Some *numerical* considerations accompany the specifications, especially related to the numerical stability of the underlying algorithms. Such considerations are regarded as an important adjunct to the specification: users need to be able to rely on software written in accordance with the specifications to perform reliably in a numerical way as well as in a functional manner.[2]

The number of specifications may be increased in future editions of the document.

The specifications relate to the following combinations of measurement model types, there being $2 \times 2 \times 2 = 8$ model types in all:

- Models with univariate and multivariate output quantities, i.e., a single (scalar) output quantity and more than one (vector) output quantity

- Explicit and implicit models, i.e., models that define the output quantity *directly*, viz., as *formulae*, and models in the form of *equations* that need to be *solved* for the output quantities given the input quantities (e.g., nonlinear least-squares adjustment)

[1] In this document the term "law of propagation of uncertainty" (LPU) will apply to the use of a first-order Taylor series approximation to the measurement model.

[2] Related aspects are treated in the Software Support for Metrology programme as part of the software testing project [11].

- Real- and complex-valued models, i.e., models in which all the quantities are real-valued and those in which one or more of the quantities are complex-valued.

EUROMETROS[3] (the EUROMET Repository of Software) [1] is a system aimed at providing metrologists with access to software appropriate to their needs. It is a state-of-the-art solution to the provision of a metrology software library and is expected to set the model to be followed in the future. Software based on the specifications provided here will subsequently be incorporated in EUROMETROS.

The coverage of the software specifications is divided according to whether they relate to Phase 1 (formulation) or Phase 2 (calculation) of the process of uncertainty evaluation (Section 1.1), or whether the procedure followed is in accordance with LPU [4, Clause 8], Monte Carlo simulation [3, 12] or the validation of LPU [3, 12].

Where a vector of values, e.g., the estimates $\boldsymbol{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$ of the values of N input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$ is needed, it is used in the specifications here as a *column vector* or a *row vector*, as appropriate, for consistency with calculations expressed in linear algebra terms [20]. It is not necessary that the physical means of data storage adopted in software implementations accords with these representations.

The bulk of this document is concerned with a coverage probability of 95% and is couched this way. It is generally extensible to other coverage probabilities.

The report is organised as follows. In the remainder of this section the above two phases of uncertainty evaluation are discussed. In Section 2 the first phase and its outputs are described in terms of the information required for implementing LPU and Monte Carlo simulation. Section 3 covers the numerical evaluation of the measurement model according to the eight categories indicated earlier. In Sections 4, 5 and 6 specifications of software units are provided for the three aspects of uncertainty evaluation considered: LPU, Monte Carlo simulation and the validation of LPU using Monte Carlo simulation. Conclusions are given in Section 7.

1.1 The two phases of uncertainty evaluation

Uncertainty evaluation consists of two phases, *formulation* and *calculation*. In Phase 1 a measurement model is derived and the model inputs are quantified. The value of each model input quantity is represented by a probability density function (PDF). Each such PDF arises from *assigning* a particular distributional form (rectangular, Gaussian, etc.). According to the GUM, PDFs are obtained from an analysis of series of observations [4, Clauses 2.3.2, 3.3.5] or are based on

[3] See www.eurometros.org

scientific judgement using all the relevant information available [4, Clauses 2.3.3, 3.3.5].

Furthermore, a PDF may be "defined" by the output from a previous uncertainty evaluation, as part of a multi-stage uncertainty evaluation process [12]. The distribution may be available analytically in a recognised form (rectangular, Gaussian, etc.) or as an approximation to the distribution function for the value of a quantity obtained from a previous application of Monte Carlo simulation, for example. A means for determining such a distribution function for the value of a quantity is given in Section 5.5.

For an input quantity that is independent of the other input quantities, LPU requires for its operation only three parameters of the PDF for the value of that quantity:

- Its expectation, taken as the estimate of the value of the output quantity
- Its standard deviation, taken as the standard uncertainty associated with the estimate of the value of the output quantity
- The corresponding degrees of freedom.

If there are dependencies, the procedure will also require the covariances associated with estimates of the values of the mutually dependent quantities.[4]

Software can support Phase 1 by providing estimates of location and dispersion and degrees of freedom from which a PDF can be constructed after making appropriate assumptions. In the case of a sufficiently large number of repeated observations of a set of the input quantities, the data can be used to define a covariance matrix or to define a multivariate Gaussian distribution, as appropriate. The specifications below relate to the (arithmetic) mean as a measure of location, and its standard deviation as a measure of dispersion. In addition, the determination of a covariance matrix is covered.

The output of Phase 1 is required to be in a form to act as inputs either to (the Phase 2 computations of) LPU, or to the Monte Carlo calculation procedure. For the LPU calculation phase, the Phase 2 inputs are the Phase 1 outputs identified above. For Monte Carlo simulation, the Phase 2 inputs are (continuous) PDFs, perhaps constructed from discrete values, or, where appropriate, joint PDFs (where there are mutual dependencies) for the values of the input quantities.

The basis of LPU is to "propagate the uncertainties" through the model to provide the uncertainties associated with estimates of the values of the model output quantities. For this purpose it is required to

[4]The degree of mutual dependence associated with the estimates x_i and x_j of the values of the input quantities X_i and X_j is sometimes characterized by the *correlation coefficient* $r(x_i, x_j)$. The covariance and correlation coefficient are related by $u(x_i, x_j) = r(x_i, x_j)u(x_i)u(x_j)$, where $u(x_i)$ and $u(x_j)$ are, respectively, the standard uncertainties associated with the estimates x_i and x_j.

- Determine sensitivity coefficients
- Determine the combined standard uncertainty by propagating the covariance matrix associated with estimates of the values of the input quantities through (a linearization of) the model to yield the covariance matrix associated with estimates of the values of the output quantities
- Apply the Welch-Satterthwaite formula for effective degrees of freedom
- Calculate percentage points of a Gaussian or a t–distribution in order to provide a coverage interval for the value of the output quantity.

These calculations would apply, as appropriate, for each of the model types. In order to operate the Phase 2 (computation of LPU), therefore, software can assist with the above calculations.

For the Monte Carlo procedure it is necessary to "propagate distributions" through the model. For this purpose it is required to

- Sample from the PDFs assigned to the values of the input quantities
- Carry out an appropriate number of Monte Carlo trials
- Obtain an approximation to the PDF for the value of the output quantity
- Obtain a coverage interval for the value of the output quantity from this approximation.

Again, these calculations would apply, as appropriate, for each of the model types, and software can assist with their implementation.

The remaining software specifications are intended to support a recommended procedure for validating LPU [3, 12]. This procedure constitutes operating both LPU and the Monte Carlo procedure for an uncertainty evaluation of concern, and carrying out an appropriate comparison of the results obtained.

2 Formulation phase

In Phase 1, the formulation phase of uncertainty evaluation, it is necessary to carry out various assignments of probability density functions (PDFs) to the values of the input quantities of the measurement model. For the LPU approach only the expectations and standard deviations of these PDFs, and covariances where appropriate, are used. For Monte Carlo simulation the PDFs themselves are used. The way these assignments are undertaken depends on the information that is available about the input quantities. Two cases are considered.

If a set of q independent observations of the value of the input quantity X_i is available, a *statistical analysis* of the observations is undertaken to determine an estimate x_i for the value of X_i together with its associated standard uncertainty $u_i = u(x_i)$ and degrees of freedom ν_i. These are the quantities used in the LPU approach (Section 4). To apply Monte Carlo simulation a PDF with these parameters is assigned. The assignment of PDFs to the values of the input quantities based on the analysis of repeated observations is described in Section 2.1. In the GUM the assignment is referred to as "a Type A evaluation of standard uncertainty".

If repeated observations are not available a PDF for the value of the input quantity X_i is assigned on the basis of, for example, prior knowledge, experience or as the result of a previous uncertainty evaluation. The PDF is used directly in Monte Carlo simulation (Section 5). To apply the LPU approach parameters x_i, u_i and ν_i are derived from the PDF. The derivation for a number of common PDFs is described in Section 2.2, and is referred to in the GUM as "a Type B evaluation of standard uncertainty".

A variant of these analyses applies if some or all of the input quantities are mutually dependent. For the LPU approach a covariance matrix would be assigned to the values of the relevant input quantities. For Monte Carlo simulation a joint (multivariate) PDF would be assigned for the values of these input quantities.

In practice both types of assignment will be required, each applying to a subset of the input quantities. For the LPU approach a covariance matrix would be constructed from those covariance matrices arising from the two types of analysis. For Monte Carlo simulation a joint PDF would be constructed from those PDFs arising from the two types of analysis.

2.1 Formulation based on analysing repeated observations

The statistical analysis of repeated observations can be approached in two parts.

Given a set of observations for the value of an input quantity X_i, assign an estimate x_i for X_i together with the associated standard uncertainty u_i for that estimate and the corresponding degrees of freedom ν_i (Section 2.1.1).

Given a set of observations for the values of a pair of input quantities that are mutually dependent, assign the covariance associated with the estimates of these quantities (Section 2.1.2).

These calculations are the basis for assigning a covariance matrix associated with the estimates of the values of a number of input quantities (Section 2.1.3).

2.1.1 Sample mean and its standard deviation

Suppose $(x_{i,1}, x_{i,2}, \ldots, x_{i,q})$ are q independent observations of the value of the input quantity X_i. The estimate x_i of the value of X_i and its associated standard uncertainty u_i are determined, respectively, as $\bar{x}_i$, the arithmetic mean of the observations, and s_i, the standard deviation of $\bar{x}_i$. The corresponding degrees of freedom ν_i is $q - 1$. Table 1 defines the sample mean of a set of values, the standard deviation of this mean and the degrees of freedom, and specifies the input and output parameters associated with their determination. This is the information required for the LPU procedure (Section 4).

Given the estimate x_i and its associated standard uncertainty s_i so obtained, a PDF for the value of X_i is assigned as follows [12]:

- If the distribution underlying the observations is unknown, assign a Gaussian PDF with expectation x_i and standard deviation s_i
- If the distribution underlying the observations is known to be Gaussian, assign the product of s_i^{-1} and the t–distribution with argument $(\xi - x_i)/s_i$ and $\nu_i = q - 1$ degrees of freedom.

This is the information required for Monte Carlo simulation (Section 5).

2.1.2 Covariance associated with two sample means

Suppose $(x_{i,k}, x_{j,k})^{\mathrm{T}}$, $k = 1, \ldots, q$, are q *pairs* of independent observations of the values of the input quantities X_i and X_j. An estimate x_i of X_i, together with its associated standard uncertainty s_i, may be determined as the mean and its standard deviation for the set of observations $(x_{i,1}, x_{i,2}, \ldots, x_{i,q})$ as in Section 2.1.1, and similarly for X_j. The covariance of X_i and X_j is estimated by the covariance $u_{i,j} = u(x_i, x_j)$ associated with the sample means x_i and x_j. Table 2 defines this covariance in terms of repeated observations of the values of the two input quantities, and specifies the input and output parameters associated with its determination.

2.1.3 Covariance matrix for input quantities

Suppose observations $(x_{i,1}, x_{i,2}, \ldots, x_{i,q})$ are available for the values of *all* input quantities X_i, $i = 1, \ldots, N$, and that these are assembled into the $N \times q$ matrix $\mathbf{\Phi}$.[5] Let $\mathbf{\Phi}'$ be obtained from $\mathbf{\Phi}$ by correcting for the sample means, i.e., the mean

[5] The symbol $\mathbf{\Phi}$ is (reluctantly) used to denote the matrix of observation $x_{i,j}$, since X is used to denote a scalar input quantity and $\boldsymbol{X}$ a vector input quantity.

Input parameters	
q	Number of observations of the value of the ith input quantity
$\boldsymbol{x}_i$	Independent observations $(x_{i,1}, x_{i,2}, \ldots, x_{i,q})$ of the value of X_i
Output parameters	
$\bar{x}_i$	Sample mean, defined by $$\bar{x}_i = \frac{1}{q}\sum_{k=1}^{q} x_{i,k}$$
s_i	Standard deviation of the sample mean, defined by $$s_i^2 = \frac{1}{q(q-1)}\sum_{k=1}^{q}(x_{i,k} - \bar{x}_i)^2$$
ν_i	Degrees of freedom, defined by $$\nu_i = q - 1$$
Numerical analysis	
	It is important to use the above formula for s_i rather than the mathematically equivalent formula $$s_i^2 = \frac{1}{q-1}\left(\frac{1}{q}\sum_{k=1}^{q} x_{i,k}^2 - \bar{x}_i^2\right).$$ For cases in which s_i is very much smaller than $\lvert\bar{x}_i\rvert$ (in which case the $x_{i,k},\ k = 1, \ldots, q$, have a number of leading digits in common) the latter formula suffers from subtractive cancellation (involving a mean square less a squared mean). The cancellation effects can be so severe that the resulting value of s_i may have too few correct significant figures for the the uncertainty evaluation to be valid [9]

Table 1: The sample mean, its standard deviation and degrees of freedom for the value of the ith input quantity.

Input parameters	
q	Number of observations of the values of the ith and of the jth input quantities
$\boldsymbol{x}_i$	Independent observations $(x_{i,1}, x_{i,2}, \ldots, x_{i,q})$ of the value of X_i
$\boldsymbol{x}_j$	Independent observations $(x_{j,1}, x_{j,2}, \ldots, x_{j,q})$ of the value of X_j paired with those for X_i
Output parameters	
$u_{i,j}$	Covariance associated with the sample means for the values of the ith and jth input quantities, defined by $$u(x_i, x_j) = \frac{1}{q(q-1)} \sum_{k=1}^{q} (x_{i,k} - \bar{x}_i)(x_{j,k} - \bar{x}_j),$$ where $\bar{x}_i$ and $\bar{x}_j$ are the sample means of the sets of observations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively
Numerical analysis	
	It is important to use the above formula for $u(x_i, x_j)$ rather than the mathematically equivalent formula $$u(x_i, x_j) = \frac{1}{q-1} \left(\frac{1}{q} \sum_{k=1}^{q} x_{i,k} x_{j,k} - \bar{x}_i \bar{x}_j \right).$$ The latter formula can suffer from subtractive cancellation as in the standard deviation calculation (Table 1) and the resulting value of $u(x_i, x_j)$ may have too few correct figures for the uncertainty evaluation to be valid

Table 2: The covariance associated with the sample means for values of the ith and jth input quantities.

$\bar{x}_i$ of the ith row is subtracted from all elements $x_{i,j},\ j = 1, \ldots, q$, in that row. Then, the covariance matrix $\boldsymbol{V_x}$ of covariances $u_{i,j}$ for the sample means of the values of the input quantities $\boldsymbol{X} = (X_1, X_2, \ldots, X_N)^{\mathrm{T}}$ is estimated by

$$\frac{1}{q(q-1)} \boldsymbol{\Phi}'(\boldsymbol{\Phi}')^{\mathrm{T}}.$$

Table 3 defines the covariance matrix for the sample means of the values of input quantities $\boldsymbol{X}$ in terms of repeated observations for $\boldsymbol{X}$, and specifies the input and output parameters associated with its determination.

	Input parameters
N	Number of input quantities
q	Number of observations of the values of each of the input quantities
$\boldsymbol{\Phi}$	An $N \times q$ matrix containing the observations $x_{i,j}$, where $x_{i,j}$ is the jth measurement of the value of the ith input quantity
	Output parameters
$\boldsymbol{V_x}$	Covariance matrix for the sample means of the values of input quantities $\boldsymbol{X}$, defined by $\frac{1}{q(q-1)} \boldsymbol{\Phi}'(\boldsymbol{\Phi}')^{\mathrm{T}}$, where $\boldsymbol{\Phi}'$ is $\boldsymbol{\Phi}$ corrected for the sample means

Table 3: The covariance matrix for the sample means of the values of input quantities $\boldsymbol{X}$.

Given the estimates $x_i\ (= \bar{x}_i),\ i = 1, \ldots, N$, for the input quantities $\boldsymbol{X}$ and their covariance matrix $\boldsymbol{V_x}$, a multivariate Gaussian distribution is assigned for the purpose of Monte Carlo simulation. This multivariate distribution is discussed in Section 5.3.5.

In practice there may be "simultaneous" observations $(x_{i,1}, x_{i,2}, \ldots, x_{i,q})$ for the values of *some* of the X_i, e.g, for $i = 2$, 4 and 5. In this case the above covariance considerations would apply to this "group". Any other such groups would be handled similarly, and the complete $N \times N$ covariance matrix constructed from these groups, together with the variances associated with estimates of the values of those input quantities that are mutually independent.

2.2 Formulation based on assigning probability distribution functions

If knowledge of an input quantity is based on non-statistical information, a PDF such as a rectangular distribution would be assigned. For the Monte Carlo approach this PDF is used directly. For the LPU approach the mean and the standard uncertainty would be extracted from the PDF and used. Table 4 indicates how the

estimate x_i, the associated standard uncertainty u_i and the degrees of freedom ν_i would be obtained from the PDFs for some common distributions, viz., Gaussian, rectangular, triangular and U-shaped.

Distribution	Parameters	PDF	x_i	u_i	ν_i
Gaussian $N(\mu, \sigma^2)$	Mean μ and standard deviation σ	$\frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$	μ	σ	∞
Rectangular	Limits a and b	$\begin{cases} 0, & x < a, \\ \frac{1}{2w}, & a \le x \le b, \\ 0, & b < x. \end{cases}$	h	$\frac{w}{\sqrt{3}}$	∞
Triangular	Limits a and b	$\begin{cases} 0, & x < a \\ \frac{x-a}{w^2}, & a \le x \le h, \\ \frac{b-x}{w^2}, & h \le x \le b, \\ 0, & b < x. \end{cases}$	h	$\frac{w}{\sqrt{6}}$	∞
U-shaped [23]	Limits a and b	$\begin{cases} 0, & x < a \\ \frac{1}{\pi\sqrt{w^2-(x-h)^2}}, & a \le x \le b, \\ 0, & b < x. \end{cases}$	h	$\frac{w}{\sqrt{2}}$	∞

Table 4: Parameters of common PDFs, where w denotes $(b-a)/2$ and h denotes $(b+a)/2$.

For example, suppose the value of the input quantity X_i is assigned a rectangular distribution between the limits a and b. Information on how to generate pseudo-random numbers from this distribution for the purpose of Monte Carlo simulation is given in Section 5.3.1. For this distribution

$$x_i = \frac{b+a}{2}, \quad u_i = \frac{b-a}{2\sqrt{3}}, \quad \nu_i = \infty$$

(Table 4), and this is the information required for the LPU procedure. For other PDFs assigned to the value of X_i, Table 4 would be used in a similar way.

3 Numerical evaluation of the model

The classification of models in Section 1 covers eight types of measurement model. In principle, any model will naturally fall into one, and only one, of these categories. Some models can be *converted* from one type to another. Whether doing so is desirable depends on circumstances. For instance, it may not be numerically stable to do so. A complex-valued model can always be converted into a real-valued model, by replacing each complex-valued quantity by two real-valued quantities, its real and imaginary parts. Again, doing so is not necessarily desirable for purposes of model evaluation (but see the end of this section regarding uncertainties).

For the categories of *explicit* models, $Y = f(\boldsymbol{X})$ (univariate) and $\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X})$ (multivariate), the calculation of the measurement result y or $\boldsymbol{y}$ requires the evaluation of the *formula* f or $\boldsymbol{f}$. Tables 5 and 6 specify the evaluation of these models, and the input and output parameters required.

The calculation of the measurement result y or $\boldsymbol{y}$ for the corresponding categories of *implicit* models, $h(Y, \boldsymbol{X}) = 0$ and $\boldsymbol{h}(\boldsymbol{Y}, \boldsymbol{X}) = \boldsymbol{0}$, requires the *solution* of an equation or a system of equations. For a univariate model, it is necessary to solve a single equation. For a multivariate model, a system of equations is to be solved. Tables 7 and 8 specify the evaluation of these models, and the input and output parameters required.

The numerical evaluation of *complex-valued* models need be no more complicated than for the categories of real-valued models considered above. Many software packages and languages provide a *complex type* for complex-valued quantities together with functions for performing complex arithmetic. An alternative to using such facilities is to store explicitly each complex-valued quantity in terms of its real and imaginary parts, and to undertake all numerical operations in terms of these two (real-valued) parts.

An issue that requires consideration as part of any implementation for these categories of complex-valued models is the way the "(standard) uncertainty" associated with an estimate of the value of a complex-valued quantity is stored. If X_i is complex-valued with real and imaginary parts X_i^R and X_i^I, the "(standard) uncertainty" associated with an estimate x_i of the value of X_i is described by the 2×2 covariance matrix

$$V_i = \begin{bmatrix} u^2(x_i^R) & u(x_i^R, x_i^I) \\ u(x_i^R, x_i^I) & u^2(x_i^I) \end{bmatrix},$$

where $u^2(x_i^R)$ and $u^2(x_i^I)$ are, respectively, the variances associated with the estimates x_i^R and x_i^I of the real and imaginary parts, and $u(x_i^R, x_i^I)$ is the covariance associated with these estimates. Furthermore, the covariance matrix V_x for the complete set of input quantities $X_i,\ i = 1, \ldots, N$, is a $2N \times 2N$ matrix. Consequently, although the quantities themselves may be regarded as complex-valued, it is necessary to store the corresponding uncertainty information using real-valued matrices, and operate on them using real arithmetic.

4 Law of propagation of uncertainty (LPU): calculation phase

4.1 Procedure

For LPU, the outputs of the formulation phase, Phase 1, are (Section 2):

Input parameters	
N	Number of input quantities
f	Function specifying the model $Y = f(\boldsymbol{X})$ in terms of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$
$\boldsymbol{x}$	Column vector $(x_1, \ldots, x_N)^{\mathrm{T}}$ of estimates of the values of the input quantities $\boldsymbol{X}$
Output parameters	
y	Value obtained by evaluating the formula $$y = f(\boldsymbol{x})$$

Table 5: Numerical evaluation of the univariate, explicit, real-valued measurement model $Y = f(\boldsymbol{X})$.

Input parameters	
N	Number of input quantities
m	Number of output quantities
$\boldsymbol{f}$	Function specifying the model $\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X})$ in terms of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$
$\boldsymbol{x}$	Column vector $(x_1, \ldots, x_N)^{\mathrm{T}}$ of estimates of the values of the input quantities $\boldsymbol{X}$
Output parameters	
$\boldsymbol{y}$	Column vector of values $(y_1, \ldots, y_m)^{\mathrm{T}}$ obtained by evaluating the formula $$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$$

Table 6: Numerical evaluation of the multivariate, explicit, real-valued measurement model $\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X})$.

Input parameters	
N	Number of input quantities
h	Function specifying the model $h(Y, \boldsymbol{X}) = 0$ in terms of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$ and the output quantity Y
$\boldsymbol{x}$	Column vector $(x_1, \ldots, x_N)^{\mathrm{T}}$ of estimates of the values of the input quantities $\boldsymbol{X}$
$\boldsymbol{t}$	Computational control parameters such as an initial estimate of y and technical parameters relating to the equation-solving software
Output parameters	
y	Value obtained by solving the equation $$h(y, \boldsymbol{x}) = 0$$
Numerical analysis	
	A zero-finding algorithm [13, 19], such as the bisection algorithm in cases where suitable lower and upper bounds are known for the measurement result y, can be used to solve the equation

Table 7: Numerical evaluation of the univariate, implicit, real-valued measurement model $h(Y, \boldsymbol{X}) = 0$.

- Estimates $\boldsymbol{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$ of the values of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$
- Standard uncertainties $\boldsymbol{u} = (u_1, \ldots, u_N)^{\mathrm{T}}$ associated with these estimates
- Corresponding degrees of freedom $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_N)^{\mathrm{T}}$
- Where appropriate, the covariances associated with estimates of the values of input quantities that are mutually dependent.

This information is conveniently represented using the three quantities $\boldsymbol{x}$, $\boldsymbol{\nu}$ and $\boldsymbol{V_x}$, where $\boldsymbol{V_x}$ is a *covariance* matrix that holds the standard uncertainties associated with the estimates $\boldsymbol{x}$ and the covariances associated with these estimates. $\boldsymbol{V_x}$ is an $N \times N$ matrix, whose (i, j)th element contains the covariance $u(x_i, x_j)$ associated with x_i and x_j, with $u(x_i, x_i) = u_i^2$.

The three quantities $\boldsymbol{x}$, $\boldsymbol{\nu}$ and $\boldsymbol{V_x}$, together with the measurement model and the required coverage probability p (e.g., 0.95), constitute the inputs to the calculation phase, Phase 2, for the LPU procedure.

For *univariate* measurement models, the following steps constitute the calculation phase:

Input parameters	
N	Number of input quantities
m	Number of output quantities
$\boldsymbol{h}$	Function specifying the model $\boldsymbol{h}(\boldsymbol{Y}, \boldsymbol{X}) = \boldsymbol{0}$ in terms of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$ and the output quantities $\boldsymbol{Y} = (Y_1, \ldots, Y_m)^{\mathrm{T}}$
$\boldsymbol{x}$	Column vector $(x_1, \ldots, x_N)^{\mathrm{T}}$ of estimates of the values of the input quantities $\boldsymbol{X}$
$\boldsymbol{t}$	Computational control parameters such as an initial estimate of $\boldsymbol{y}$ and technical parameters relating to the equation-solving software
Output parameters	
$\boldsymbol{y}$	Column vector of values $(y_1, \ldots, y_m)^{\mathrm{T}}$ obtained by solving the equations $\boldsymbol{h}(\boldsymbol{y}, \boldsymbol{x}) = \boldsymbol{0}$
Numerical analysis	
	An iterative algorithm such as Newton's method [19], starting from a suitable estimate of the measurement result $\boldsymbol{y}$, can be used to solve the system of equations

Table 8: Numerical evaluation of the multivariate, implicit, real-valued measurement model $\boldsymbol{h}(\boldsymbol{Y}, \boldsymbol{X}) = \boldsymbol{0}$.

1. Form numerically the measurement result (an estimate of the value of the output quantity) by evaluating the model at the estimates of the values of the input quantities.

   See Section 3.

2. Form the partial derivatives of first order of the measurement model with respect to the input quantities, and determine numerical values for the sensitivity coefficients by evaluating these partial derivatives at the estimates of the values of the input quantities.[6]

   See Section 4.2.

3. Determine the standard uncertainty associated with the estimate of the value of the model output quantity by combining the standard uncertainties associated with the estimates of the values of the input quantities, the covariances associated with these estimates and the sensitivity coefficients.

   See Section 4.3.

4. Use the Welch-Satterthwaite formula to calculate ν_{eff}, the effective degrees of freedom associated with the estimate of the value of the output quantity, from the standard uncertainties and degrees of freedom associated with the estimates of the values of the input quantities, the sensitivity coefficients and the standard uncertainty associated with the estimate of the value of the output quantity.[7]

   See Section 4.4.

5. Compute the coverage factor corresponding to ν_{eff} and the required coverage probability p as a percentage point of the (standardised) Gaussian distribution ($\nu_{\text{eff}} = \infty$) or a t–distribution ($\nu_{\text{eff}} < \infty$). Hence, compute the expanded uncertainty, and thus an interval having a stipulated coverage probability containing the value of the output quantity, by forming the product of this coverage factor and the standard uncertainty associated with the estimate of the value of the output quantity.

   See Section 4.5.

The computational flow of the LPU calculation phase, indicating the inputs to the phase and the primary output, a coverage interval, is given in Figure 1. This figure applies in the case of a univariate, explicit, real-valued model with no covariance effects. Other model types would give diagrams that constitute a variant of Figure 1.

[6] For an *implicit* model, the partial derivative of first order of the measurement model with respect to the *output* quantity is also required to determine the sensitivity coefficients: see Section 4.2.

[7] LPU does not state how ν_{eff} is to be calculated when the input quantities are correlated.

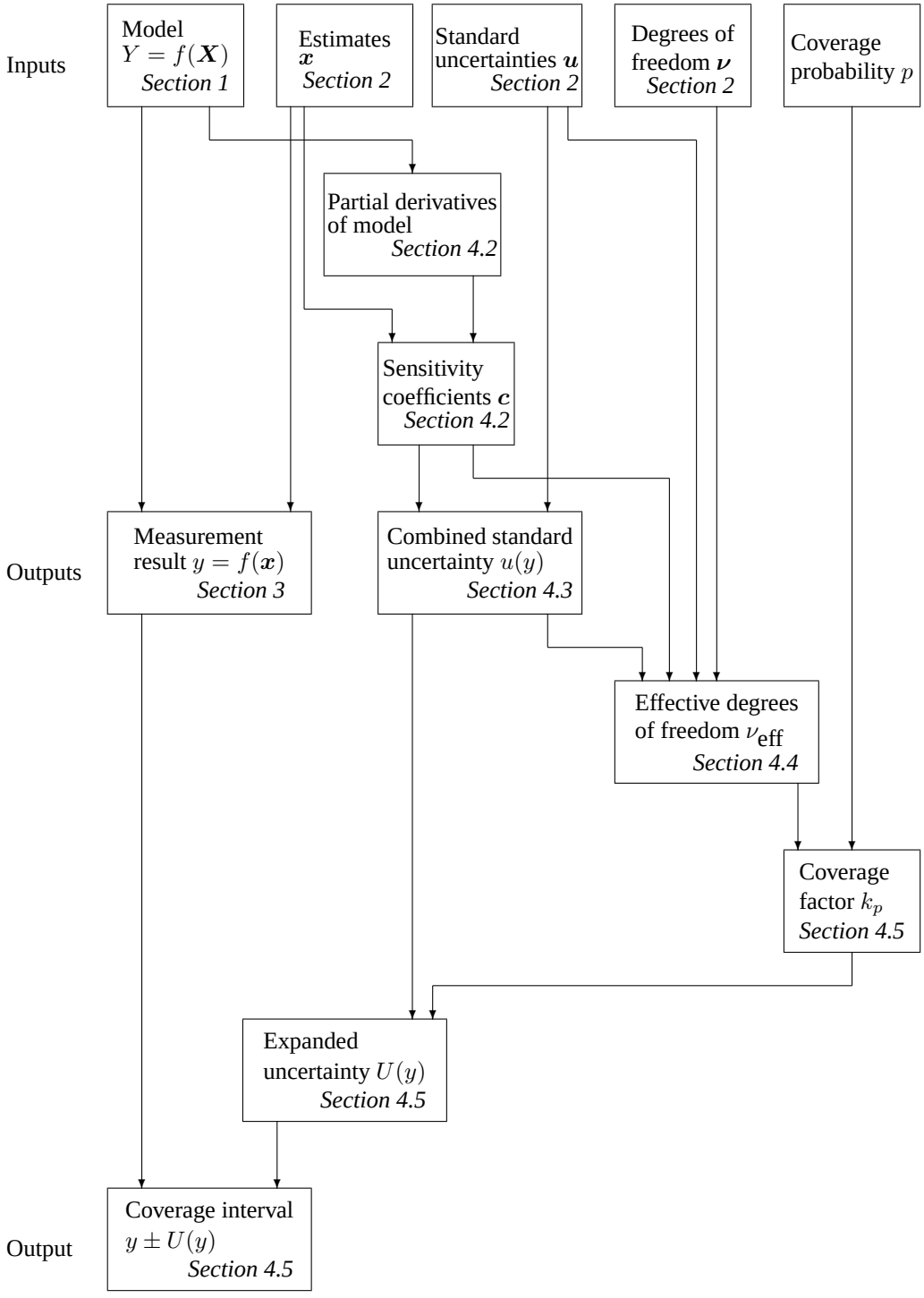

Figure 1: Phase 2, calculation, of uncertainty evaluation using LPU. The stages shown relate to the case of a univariate, explicit, real-valued model with mutually independent input quantities.

For *multivariate* measurement models, Steps 1, 2 and 3 would be performed as above. However, no information is given here regarding steps 4 and 5 for such models. The extension of steps 4 and 5 to the evaluation of coverage *regions* for the values of multivariate output quantities is not straightforward. Consideration of this matter, which is a topic for research [12], will be given in a future version of this report.

Implementation of the above procedure would be achieved in terms of software "units" as described in Sections 3, 4.2, 4.3, 4.4, and 4.5. For example, for a univariate, explicit, real-valued measurement model, these units are specified in Tables 5, 9, 13, 14 and 15.

4.2 Sensitivity coefficients

The LPU procedure (Section 4.1) covers the univariate, explicit, real-valued model of measurement $Y = f(\boldsymbol{X})$. The *sensitivity coefficients* used by that procedure are denoted here by the (row) vector[8] $\boldsymbol{C} = (c_1, \ldots, c_N)$, whose jth element c_j is the partial derivative $\partial f/\partial X_j$ of $f(\boldsymbol{X})$ evaluated at $\boldsymbol{X} = \boldsymbol{x}$. Table 9 specifies the evaluation of sensitivity coefficients for this category of measurement model, and indicates the input and output parameters necessary for their determination.

Input parameters	
N	Number of input quantities
f	Function specifying the model $Y = f(\boldsymbol{X})$ in terms of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$
$\boldsymbol{x}$	Column vector $(x_1, \ldots, x_N)^{\mathrm{T}}$ of estimates of the values of the input quantities $\boldsymbol{X}$
Output parameters	
$\boldsymbol{C}$	A $1 \times N$ vector of sensitivity coefficients, whose jth element is the partial derivative $\partial f/\partial X_j$ of $f(\boldsymbol{X})$ evaluated at $\boldsymbol{X} = \boldsymbol{x}$

Table 9: Sensitivity coefficients for a univariate, explicit, real-valued measurement model $Y = f(\boldsymbol{X})$.

Table 10 specifies the counterpart for a *multivariate*, explicit, real-valued model $\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X})$. In this case $\boldsymbol{C}$ takes the form of a *matrix* of sensitivity coefficients, whose (i, j)th element is the partial derivative of the ith output quantity with respect to the jth input quantity evaluated at $\boldsymbol{X} = \boldsymbol{x}$.

The univariate, *implicit*, real-valued model of measurement is $h(Y, \boldsymbol{X}) = 0$. The sensitivity coefficients $\boldsymbol{C} = (c_1, \ldots, c_N)$ are determined from the partial deriva-

[8] The symbol $\boldsymbol{C}$ rather than the "more natural" $\boldsymbol{c}$ for this vector is used to denote this set of coefficients. The reason for this choice is that for multivariate models $\boldsymbol{C}$ is used to hold an array (matrix) of sensitivity coefficients, and that it is appropriate to use the same symbol throughout.

Input parameters	
N	Number of input quantities
m	Number of output quantities
$\boldsymbol{f}$	Function specifying the model $\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X})$ in terms of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$
$\boldsymbol{x}$	Column vector $(x_1, \ldots, x_N)^{\mathrm{T}}$ of estimates of the values of the input quantities $\boldsymbol{X}$
Output parameters	
$\boldsymbol{C}$	An $m \times N$ matrix of sensitivity coefficients, whose (i, j)th element is the partial derivative $\partial f_i / \partial X_j$ of $\boldsymbol{f}(\boldsymbol{X})$ evaluated at $\boldsymbol{X} = \boldsymbol{x}$

Table 10: Sensitivity coefficients for a multivariate, explicit, real-valued measurement model $\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X})$.

tives of h with respect to both the input quantities $\boldsymbol{X}$ and the output quantity Y. The jth sensitivity coefficient c_j is given by

$$-\left(\frac{\partial h}{\partial Y}\right)^{-1}\left(\frac{\partial h}{\partial X_j}\right),$$

where the partial derivatives are evaluated at $\boldsymbol{X} = \boldsymbol{x}$ and $Y = y$, with y satisfying $h(y, \boldsymbol{x}) = 0$. Table 11 specifies the evaluation of sensitivity coefficients for this category of measurement model, and indicates the input and output parameters necessary for their determination.

The *multivariate*, implicit, real-valued model of measurement is $\boldsymbol{h}(\boldsymbol{Y}, \boldsymbol{X}) = \boldsymbol{0}$. The matrix $\boldsymbol{C}$ of sensitivity coefficients is determined as the solution to the linear system of equations

$$\boldsymbol{H_y C} = \boldsymbol{H_x},$$

where the matrices $\boldsymbol{H_x}$ and $\boldsymbol{H_y}$ contain, respectively, the partial derivatives of the components h_i of $\boldsymbol{h}$ with respect to the input quantities X_j and output quantities Y_j, evaluated at $\boldsymbol{X} = \boldsymbol{x}$ and $\boldsymbol{Y} = \boldsymbol{y}$, with $\boldsymbol{y}$ satisfying $\boldsymbol{h}(\boldsymbol{y}, \boldsymbol{x}) = \boldsymbol{0}$. Table 12 specifies the evaluation of sensitivity coefficients for this category of measurement model, and indicates the input and output parameters necessary for their determination.

Counterparts would apply for *complex-valued* models. It is necessary to form partial derivatives of the real and imaginary parts of the (components of the) measurement model with respect to the real and imaginary parts of the input quantities (and the output quantities for implicit models).

The sensitivity coefficients can be formed [7]

1. Manually, by the algebraic differentiation of, e.g., $f(\boldsymbol{X})$ with respect to each component X_i, $i = 1, \ldots, N$, of $\boldsymbol{X}$, followed by the substitution of the values of $\boldsymbol{x}$ for $\boldsymbol{X}$

Input parameters	
N	Number of input quantities
h	Function specifying the model $h(Y, \boldsymbol{X}) = 0$ in terms of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$ and the output quantity Y
$\boldsymbol{x}$	Column vector $(x_1, \ldots, x_N)^{\mathrm{T}}$ of estimates of the values of the input quantities $\boldsymbol{X}$
y	Estimate of the value of the output quantity Y that satisfies $h(y, \boldsymbol{x}) = 0$
Output parameters	
$\boldsymbol{C}$	A $1 \times N$ vector of sensitivity coefficients, whose jth element is $$-\left(\frac{\partial h}{\partial Y}\right)^{-1}\left(\frac{\partial h}{\partial X_j}\right),$$ where the partial derivatives are evaluated at $\boldsymbol{X} = \boldsymbol{x}$ and $Y = y$

Table 11: Sensitivity coefficients for a univariate, implicit, real-valued measurement model $h(Y, \boldsymbol{X}) = 0$.

Input parameters	
N	Number of input quantities
m	Number of output quantities
$\boldsymbol{h}$	Function specifying the model $\boldsymbol{h}(\boldsymbol{Y}, \boldsymbol{X}) = \boldsymbol{0}$ in terms of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$ and the output quantities $\boldsymbol{Y} = (Y_1, \ldots, Y_m)^{\mathrm{T}}$
$\boldsymbol{x}$	Column vector $(x_1, \ldots, x_N)^{\mathrm{T}}$ of estimates of the values of the input quantities $\boldsymbol{X}$
$\boldsymbol{y}$	Column vector $(y_1, \ldots, y_m)^{\mathrm{T}}$ of estimates of the values of the output quantities $\boldsymbol{Y}$ that satisfy $\boldsymbol{h}(\boldsymbol{y}, \boldsymbol{x}) = \boldsymbol{0}$
Output parameters	
$\boldsymbol{C}$	An $m \times N$ matrix of sensitivity coefficients that solves $$\boldsymbol{H_y C} = \boldsymbol{H_x},$$ where the matrices $\boldsymbol{H_x}$ and $\boldsymbol{H_y}$ contain, respectively, the partial derivatives of the components h_i of $\boldsymbol{h}$ with respect to the input quantities X_j and output quantities Y_j, evaluated at $\boldsymbol{X} = \boldsymbol{x}$ and $\boldsymbol{Y} = \boldsymbol{y}$.
Numerical analysis	
	It is important to solve the above system of equations using a numerically stable procedure, such as Gaussian elimination with a pivoting strategy [20]

Table 12: Sensitivity coefficients for a multivariate, implicit, real-valued measurement model $\boldsymbol{h}(\boldsymbol{Y}, \boldsymbol{X}) = \boldsymbol{0}$.

2. As 1, except by the use of a computer package for algebraic differentiation or a symbolic-algebra package that provides this capability
3. By the use of finite-difference formulae
4. By the use of program differentiation techniques, including forward automatic differentiation and reverse automatic differentiation (which are examples of *operator overloading*), and *source to source transformation*.

The manner in which the partial derivatives required in forming the sensitivity coefficients C in Tables 9–12 are obtained requires careful consideration. These derivatives and hence the sensitivity coefficients can be difficult to determine by hand for models that are complicated. The above options of using a symbolic-algebra package, finite-difference formulae or program differentiation techniques can be attractive in such circumstances. There are learning overheads associated with the use of a symbolic-algebra package and program differentiation techniques: their use can be justified if the user needs to address a sufficient number of complicated models. Finite-difference formulae may provide inadequate accuracy if used inappropriately. These alternatives to the manual determination of sensitivity coefficients are addressed in Appendices A, B and C.

4.3 Combined standard uncertainty

For a univariate, explicit, real-valued measurement model, the combined standard uncertainty $u(y)$ associated with the measurement result y is obtained from the formula [4]

$$u^2(y) = \sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j u(x_i, x_j),$$

where c_i is the sensitivity coefficient for the ith input quantity, and $u(x_i, x_j)$ the covariance associated with the estimates x_i and x_j, with $u(x_i, x_i) = u_i^2$, the variance (squared standard uncertainty) associated with the ith estimate. A compact way of representing the above expression is

$$\boldsymbol{V_y} = \boldsymbol{C}\boldsymbol{V_x}\boldsymbol{C}^{\mathrm{T}}, \tag{1}$$

where $\boldsymbol{V_y} = u^2(y)$, $\boldsymbol{C} = (c_1, \ldots, c_N)$, and $\boldsymbol{V_x}$ is the $N \times N$ covariance matrix associated with the estimates $\boldsymbol{x}$ of the values of the input quantities $\boldsymbol{X}$.

Using this notation, Formula (1) for $\boldsymbol{V_y}$ applies for *all* categories of measurement model. For univariate, real-valued models, $\boldsymbol{V_y}$ is the variance associated with the estimate y of the value of the output quantity Y; for other measurement models, it is the covariance matrix associated with the estimates of the values of the output quantities. Table 13 specifies the evaluation of combined standard uncertainty, and indicates the input and output parameters necessary for its determination.

Input parameters	
N	Number of input quantities
m	Number of output quantities
$\boldsymbol{V_x}$	An $N \times N$ covariance matrix associated with the estimates $\boldsymbol{x}$ of the values of the input quantities $\boldsymbol{X}$
$\boldsymbol{C}$	An $m \times N$ matrix of sensitivity coefficients
Output parameters	
$\boldsymbol{V_y}$	An $m \times m$ covariance matrix associated with the estimate y (or $\boldsymbol{y}$) of the value(s) of the output quantity Y (or $\boldsymbol{Y}$) obtained by evaluating $$\boldsymbol{V_y} = \boldsymbol{C}\boldsymbol{V_x}\boldsymbol{C}^{\mathrm{T}}$$

Table 13: Combined standard uncertainty for a univariate ($m = 1$) or a multivariate (general m) model.

4.4 Effective degrees of freedom

Table 14 specifies the Welch-Satterthwaite formula for evaluating the effective degrees of freedom ν_{eff} for a univariate, real-valued output quantity Y in the case that the input quantities $\boldsymbol{X}$ are mutually independent.

4.5 Expanded uncertainty

Table 15 specifies the calculation of expanded uncertainty $U(y)$ associated with the estimate y of the value of a univariate, real-valued output quantity Y in the case that the input quantities $\boldsymbol{X}$ are mutually independent. The expanded uncertainty is evaluated as the product of the combined standard uncertainty $u(y)$ (Section 4.3) and a coverage factor k_p that depends on the required coverage probability p and the effective degrees of freedom ν_{eff} (Section 4.4).

The calculation of expanded uncertainty depends on knowledge of the distribution of the value of the output quantity. In the LPU approach, a t–distribution with ν_{eff} degrees of freedom is assigned to the random variable

$$t = \frac{y - Y}{u(y)}.$$

It follows that k_p is the percentage point $t_p(\nu_{\text{eff}})$ of the t–distribution such that the probability that $|t|$ is no greater than $t_p(\nu_{\text{eff}})$ is equal to p. Values of k_p for various choices of coverage probability p and degrees of freedom ν_{eff} may be obtained from statistical tables and implemented as a "look-up" table. Many mathematical and statistical software libraries provide an implementation and can be used. For

Input parameters	
N	Number of input quantities
$\boldsymbol{u}$	Column vector $(u_1, \ldots, u_N)^{\mathrm{T}} \equiv (u(x_1), \ldots, u(x_N))^{\mathrm{T}}$ of standard uncertainties associated with the estimates $\boldsymbol{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$ of the values of the input quantities $\boldsymbol{X}$
$\boldsymbol{C}$	A $1 \times N$ (row) vector of sensitivity coefficients, whose jth element is the partial derivative $\partial f / \partial X_j$ of $f(\boldsymbol{X})$ evaluated at $\boldsymbol{X} = \boldsymbol{x}$
$\boldsymbol{\nu}$	Column vector $(\nu_1, \ldots, \nu_N)^{\mathrm{T}}$ of degrees of freedom. If x_i is estimated as the mean of a set of q repeated observations, ν_i is taken as $q - 1$. If x_i is estimated as the mean (mid-point) of a rectangular distribution with accurately-known endpoints, ν_i is taken as infinite (∞). (Since ∞ cannot be represented as such as an input quantity, a convention may be adopted. For instance such a value can be "coded" as 0 (zero), and the procedure would be designed to interpret this value, which cannot occur otherwise, as infinite.) These are two important cases. There are other possibilities [4] associated with assigning the degrees of freedom to estimates of the values of the input quantities. Each case has to be treated on its own merits
$u(y)$	Combined standard uncertainty associated with the measurement result y
Output parameters	
ν_{eff}	Effective degrees of freedom determined from the Welch-Satterthwaite formula $$\frac{u^4(y)}{\nu_{\mathrm{eff}}} = \sum_{i=1}^{N} \frac{c_i^4 u^4(x_i)}{\nu_i}$$

Table 14: Effective degrees of freedom according to the Welch-Satterthwaite formula.

$\nu_{\text{eff}} \geq 473$, $k_p = 1.96$, correct to two decimal places, the corresponding value for the standard Gaussian distribution $N(0, 1)$.

Input parameters	
$u(y)$	Combined standard uncertainty associated with the measurement result y
ν_{eff}	Effective degrees of freedom determined from the Welch-Satterthwaite formula
p	Coverage probability (typically 0.95)
Output parameters	
$U(y)$	Expanded uncertainty, defined by $$U(y) = k_p u_(y),$$ where k_p is a coverage factor, depending on the stipulated coverage probability p, that is obtained from tables of percentage points of the Gaussian distribution ($\nu_{\text{eff}} = \infty$) or a t–distribution ($\nu_{\text{eff}} < \infty$)

Table 15: Expanded uncertainty.

5 Monte Carlo simulation: calculation phase

5.1 Procedure

For Monte Carlo simulation, the outputs of the formulation phase, Phase 1, are (Section 2) the PDFs[9] $\boldsymbol{g}(\boldsymbol{\xi}) = (g_1(\xi_1), \ldots, g_n(\xi_N))^{\mathrm{T}}$ for the values of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$.[10]

The PDFs, together with the measurement model and the required coverage probability p (e.g., 0.95), constitute the inputs to the calculation phase, Phase 2, for the Monte Carlo simulation procedure.

For *univariate* measurement models, the following steps constitute the calculation phase:

1. Select the number M of Monte Carlo trials to be made.

   See Section 5.2.

2. Generate M samples of the (set of N) input quantities.

   See Section 5.3.

[9] A joint (multivariate) PDF for (a subset of) the input quantities is also possible (Section 2.1.3).

[10] The following notation is used: X_i to denote the ith input quantity, x_i an estimate of the value of X_i, and ξ_i a (general) value of X_i. Hence, the PDF for the value of X_i is written as a function of ξ_i. Similarly, Y, y and η are used for a (univariate) output quantity.

3. For each sample, evaluate the model to give the corresponding value of the output quantity.

   See Section 5.4.

4. Sort these values of the output quantity into non-decreasing order, using the sorted values to approximate the distribution function for the value of the output quantity.

   See Section 5.5

5. Form the measurement result (an estimate of the value of the output quantity) and the associated standard uncertainty from this distribution function (or directly from the set of values of the output quantity).

   See Section 5.6.

6. Form the shortest 95% coverage interval for the value of the output quantity from the distribution function.

   See Section 5.7

Figure 2 shows the procedure diagrammatically.

Section 5.8 describes how Monte Carlo simulation may be used to undertake a sensitivity analysis for the measurement model with respect to each input quantity, yielding "nonlinear" sensitivity coefficients that are the counterpart of (linear) sensitivity coefficients necessary for the implementation of LPU.

Section 5.9 indicates a basic implementation of an adaptive Monte Carlo procedure that removes the need to make an *a priori* choice of the number of Monte Carlo trials.

The modifications to the procedure illustrated in Figure 2 for univariate measurement models required for other types of univariate model are straightforward. Multivariate models are considered in Section 5.10.

5.2 The number of Monte Carlo trials

A value of M, the number of Monte Carlo trials to be made, needs to be selected. It can be chosen *a priori*, in which case there will be no direct control over the degree of approximation delivered by the Monte Carlo procedure. The reason is that the number needed to provide a prescribed degree of approximation will depend on the "shape" of the PDF for the value of the output quantity and the coverage probability required. Also, the calculations are *stochastic* in nature, being based on random sampling. However, a value of $M = 10^6$ can often be expected to deliver a 95% coverage interval, having a length with a degree of approximation of one or two significant digits, for the value of the output quantity.

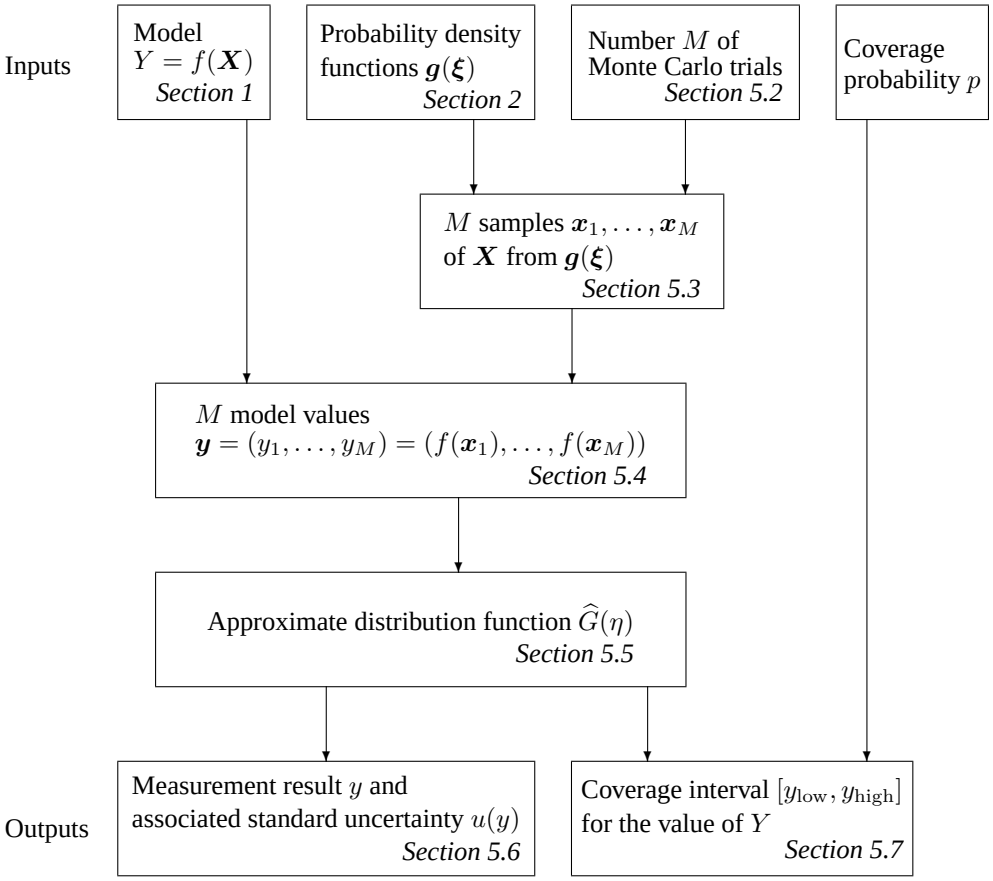

Figure 2: Phase 2, calculation, of uncertainty evaluation using Monte Carlo simulation. The stages shown relate to the case of a univariate, explicit, real-valued model.

Because there is no guarantee that this or any specific number will suffice, it is recommended to use a process that selects M adaptively, i.e., as the trials progress. A property of such a process is that it takes a number of trials that is economically consistent with the achievement of the required degree of approximation [2, 3, 12]. A basic implementation of an adaptive Monte Carlo procedure is described in Section 5.9.

5.3 Sampling from the probability density functions

In an implementation of the Monte Carlo procedure M samples $\boldsymbol{x}_i$, $i = 1, \ldots, M$, are obtained from the PDFs for the values of the input quantities $\boldsymbol{X}$. Samples are obtained from a joint (multivariate) Gaussian PDF when appropriate (Section 2.1.3).

Recommendations concerning the manner in which this sampling should be carried out are given here for the commonest distributions, viz., the rectangular, the Gaussian, the t–distribution and the multivariate Gaussian. It is possible to prepare software for sampling from almost any distribution, and indeed to develop a general framework for doing so (Section 5.3.4).

Tests of randomness of the numbers produced by a generator are indicated.

5.3.1 Rectangular distribution

The ability to generate pseudo-random numbers from a (continuous) rectangular distribution is fundamental in its own right, and also as the basis for generating numbers from any distribution (Section 5.3.4) using an appropriate algorithm or formula. In the latter regard, the quality of the numbers generated from a non-rectangular distribution depends on that of the rectangular generator and on the properties of the algorithm employed. The quality of the non-rectangular generator can therefore be expected to be correlated with that of the rectangular generator. A good rectangular generator and a good algorithm can be expected to provide a good non-rectangular generator. A poor rectangular generator and a good or bad algorithm can be expected to provide a poor non-rectangular generator. It is thus especially important that the underlying rectangular generator is sound (cf. [29]). Unless the user is sure of the pedigree of a rectangular generator it should not be used until adequate testing has been carried out. Invalid results can otherwise be obtained. Some of the "tests for randomness" that should be undertaken are indicated below. A recommended rectangular pseudo-random number generator, that has been shown to perform well in these tests and that is straightforward to implement, is given in this section.

Table 16 defines relevant aspects of the functioning of a procedure for generating

rectangular pseudo-random numbers in the interval $[0,\ 1]$, specifying the input, input-output and output parameters associated with their determination.

Input parameters	
q	Number of rectangular pseudo-random numbers to be generated
Input-output parameters	
$\boldsymbol{t}$	Column vector of parameters required as input quantities and that may be changed as part of the computation. The subsequent values of these quantities are not usually of immediate concern to the user. The parameters are needed to help control the process by which the pseudo-random numbers are produced. The parameters may be realised as global variables and thus not explitly appear as parameters of the procedure. One or more of these parameters may be a seed, used to initiate the sequence of random numbers produced by successive calls of the procedure. By setting the seed(s) to values previously used, the same sequence of random numbers can be produced. Doing so is important as part of software regression testing, used to verify the consistency of results produced using the software with those from previous versions
Output parameters	
$\boldsymbol{z}$	q "draws" from a variable recatngularly distributed between zero and one

Table 16: Rectangular pseudo-random number generation.

A rectangular pseudo-random number x in the interval $[a,\ b]$ can be formed by determining $a + (b - a)z$, where z is a rectangular pseudo-random number in the interval $[0,\ 1]$.

Randomness tests A review [32] has been carried out on the use of random numbers in solving problems using Monte Carlo simulation. It draws conclusions concerning, in particular, the best methods to employ for generating rectangular pseudo-random numbers. The so-called "combination generators" are recommended and are reported as being favoured by experts as fulfilling the requirements of possessing the longest periods[11] and passing a set of statistical tests for randomness.[12]

A combination generator simultaneously uses more than one generator. Each such generator is typically a member of the class of *congruential generators* or the class

[11] A pseudo-random number generator of necessity provides a *sequence* of numbers. The period of the sequence is the number of consecutive values in the sequence before they are repeated.

[12] The tests include the so-called *standard tests* [28], viz., the χ^2 test, the Kolmogorov-Smirnov test, the frequency test, the serial test, the gap test, the poker test, the coupon collector's test and the more stringent *Die Hard tests* [30], that include the overlapping M-tuple test, the overlapping permutation test, the parking lot and lattice test and the birthday-spacing test.

of *shift register generators*, both of which are widely discussed in the literature (see, e.g., [18, 28, 33, 35]).

The KISS[13] generator [31] is a combination of a congruential generator and two shift register generators. A version in the C programming language is available [34, p42] and in Fortran [31].

The Hill-Wichmann generator [25] is a combination of three congruential generators. Both this and the KISS generator satisfy all the above standard and Die Hard tests [32] [34, p41] and are recommended.

A recommended rectangular random number generator Table 17 defines the Hill-Wichmann generator for generating rectangular pseudo-random numbers in the interval $[0,\ 1]$.

Input parameters	
None	
Input-output parameters	
i_A, i_B, i_C	Integer parameters required as input quantities and that are changed by the procedure. Set to integer values between 1 and 30000 before the first call. Do not disturb between calls. Subsequent values of these quantities are not usually of immediate concern to the user. The parameters provide the basis by which the pseudo-random numbers are produced. They may be realised as global variables and thus not explitly appear as parameters of the procedure
Output parameters	
z	Rectangular random number between zero and one
Computation	
1. Form $i_A = 171 i_A \bmod 30269$ 2. Form $i_B = 172 i_B \bmod 30307$ 3. Form $i_C = 170 i_C \bmod 30323$ 4. Form $z = (i_A/30269 + i_B/30307 + i_C/30323) \bmod 1$	

Table 17: The Hill-Wichmann rectangular pseudo-random number generator.

[13] Keep It Simple, Stupid!

5.3.2 Gaussian distribution

The procedure in Table 18 provides a straightforwardly implementable approach [8] to generate values from the standard Gaussian distribution $N(0,\ 1)$ using the Box-Muller transform.

Input parameters	
None	
Output parameters	
z_1, z_2	Two draws from a Gaussian variable with zero expectation and unit standard deviation
Computation	

1. Generate rectangular random variates v_1 and v_2 between zero and one
2. Form $z_1 = \sqrt{-2\log v_1}\cos 2\pi v_2$ and $z_2 = \sqrt{-2\log v_1}\sin 2\pi v_2$
3. Take z_1 and z_2 as *two* standard Gaussian variates

Table 18: The Box-Muller Gaussian pseudo-random number generator.

To sample from the Gaussian distribution $N(\mu, \sigma^2)$, take $X = \mu + \sigma Z$, where Z is a standard Gaussian variable.

5.3.3 t–distribution

The procedure in Table 19 provides an approach [27], [34, p63] to generate values from the t–distribution with ν degrees of freedom, that is also straightforward to implement.

For expectation μ and standard deviation σ and ν degrees of freedom, take $X = \mu + \sigma Z$, where Z is a variable having a t–distribution with ν degrees of freedom.

Sampling from the t–distribution can be used in Monte Carlo simulation when synthesising PDFs assigned to input quantities represented by repeated observations that can be regarded as drawn from a Gaussian distribution (Section 2.1.1).[14]

[14]This use has been criticized in some quarters in that coverage intervals determined for the value of the output quantity can be pessimistically long. Conversely, were a Gaussian distribution to be assigned, i.e., with no account taken of the degrees of freedom that the t–distribution is intended to accommodate, the resulting coverage intervals could be optimistically short. This matter remains a discussion topic by researchers. It is recommended that a decision is agreed and recorded with the details of the uncertainty evaluation in any one instance. In particular, despite the statement in

Input parameters	
ν	Degrees of freedom
Output parameters	
z	Draw from a t–distribution with ν degrees of freedom
Computation	

1. Generate rectangular random variates v_1 and v_2 between zero and one
2. If $v_1 < 1/2$, set $z = 1/(4v_1 - 1)$ and $v = v_2/z^2$; otherwise set $z = 4v_1 - 3$ and $v = v_2$
3. If $v < 1 - |z|/2$ or $v < (1 + z^2/\nu)^{-(\nu+1)/2}$, accept z as a sample from a t–distribution; otherwise repeat from Step 1

Table 19: A t–distribution pseudo-random number generator.

5.3.4 General distributions

Sections 5.3.1–5.3.3 presented procedures for sampling from the PDFs relating to common distributions, viz., rectangular, Gaussian and the t–distribution, respectively. Consideration is given here to the task of sampling from a general distribution defined by its distribution function $G(\xi)$.

A sample z from this distribution is obtained as follows:

1. Sample a random number ψ from a rectangular distribution on the interval $[0,\ 1]$ (Section 5.3.1)
2. Find the value z satisfying $G(z) = \psi$.

The "inversion" step (in 2 above) of forming $z = G^{-1}(\psi)$ may be possible analytically or, otherwise, performed numerically. In the latter case, z is evaluated by solving the equation

$$G(z) - \psi = 0.$$

Upper and lower bounds for z are generally easily found, in which case a "bracketing" algorithm such as bisection can be used to determine z [13, 19].

the GUM that uncertainty evaluations should be realistic, if in doubt it might be wiser and certainly cautious to use the t–distribution, accompanying the results of uncertainty evaluation by a suitable qualifying statement.

5.3.5 Multivariate Gaussian distribution

The most important multivariate distribution is the multivariate Gaussian distribution. An $n \times 1$ vector of expectations $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{V}$ of order n define the parameters of the n-dimensional Gaussian distribution.

Values can be sampled from this multivariate (or joint) Gaussian distribution [36, 39] using the procedure in Table 20.

Input parameters	
n	Dimension of the multivariate Gaussian distribution
$\boldsymbol{\mu}$	$n \times 1$ vector of expectations
$\boldsymbol{V}$	Covariance matrix of order $n \times n$
q	Number of multivariate Gaussian pseudo-random numbers to be generated
Output parameters	
$\boldsymbol{\Phi}$	An $n \times q$ matrix, the jth column of which is a draw from the multivariate Gaussian distribution
Computation	

1. Form the Cholesky factor $\boldsymbol{R}$ of $\boldsymbol{V}$, i.e., the upper triangular matrix satisfying $\boldsymbol{V} = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}$. (To generate q pseudo-random numbers it is necessary to perform this matrix factorization only once.)

2. Form q samples from the n-dimensional standard Gaussian distribution $N(0,1) \times \ldots \times N(0,1)$. Here, $N(\mu, \sigma^2)$ denotes the Gaussian PDF with expectation μ and standard deviation σ. Doing so simply means generating an $n \times q$ array $\boldsymbol{Z}$ of standard Gaussian variates

3. Provide the required sample (the Cholesky factor acts as a transformation from the uncorrelated standardized space to that required):
$$\boldsymbol{\Phi} = \boldsymbol{\mu}\mathbf{1}^{\mathrm{T}} + \boldsymbol{R}^{\mathrm{T}}\boldsymbol{Z},$$
where $\mathbf{1}$ denotes a $q \times 1$ vector of ones

Table 20: A multivariate Gaussian random number generator.

Figure 3 shows 200 points generated from a bivariate Gaussian distribution, using the MULTNORM generator [36], with expectation $\boldsymbol{\mu} = (2,3)^{\mathrm{T}}$ and covariance matrix
$$\boldsymbol{V} = \begin{bmatrix} 2.0 & 1.9 \\ 1.9 & 2.0 \end{bmatrix}.$$

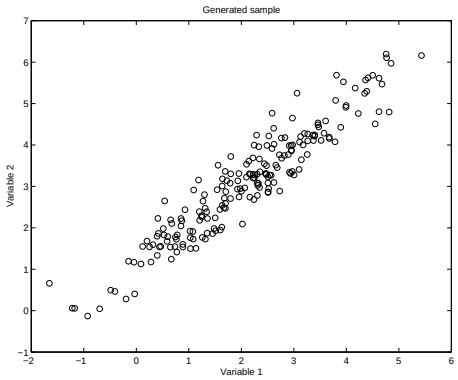

Figure 3: Points sampled from a bivariate Gaussian distribution with positive correlation.

Similar generators are available elsewhere [14].

In this figure the points "span" an elongated angled ellipse. Were the off-diagonal elements of $\boldsymbol{V}$ to be replaced by zero, the points would span a circle. Were the diagonal elements made unequal, and the off-diagonal elements kept at zero, the points would span an ellipse whose axes were parallel to the axes of the graph.

5.4 Evaluation of the model

The model is evaluated for each of the M samples from each of the PDFs for the values of the N input quantities. Specifically, denote the M samples by $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$, where the rth sample $\boldsymbol{x}_r$ contains values $x_{1,r}, \ldots, x_{N,r}$, with $x_{i,r}$ a "draw" from the PDF for the value of X_i. Then, for a univariate, explicit, real-valued model, the model values are

$$y_r = f(\boldsymbol{x}_r),\ r = 1, \ldots, M.$$

For a univariate, *implicit*, real-valued model, the rth model value y_r is determined as the solution to the equation

$$h(y_r, \boldsymbol{x}_r) = 0.$$

The case of a univariate, explicit, real-valued measurement model is covered below, although the presentation would apply similarly for the corresponding implicit model. Multivariate models are considered in Section 5.10.

The information in Section 3 provides advice on model evaluation.

Note that in Monte Carlo simulation the model is evaluated for each sample of the input quantities and hence for a range of values (that may be distanced by "several standard deviations" from the estimates of the values of the input quantities). This

is in contrast to the LPU procedure in which the measurement model is evaluated only at the estimates of the values of the input quantities and, if finite difference approximations are used [4, Clause 5.1.3], also at points perturbed from these estimates by $\pm$ one standard deviation for each variable in turn. For this reason some issues may arise regarding the numerical procedure used to evaluate the model, e.g., ensuring its convergence (where iterative schemes are used) and numerical stability. The user should ensure that, where appropriate, the numerical methods used to evaluate the measurement model are valid for a sufficiently large region centred on the estimates of the values of the input quantities.

5.5 Distribution function for the value of the output quantity

An approximation $\widehat{G}(\eta)$ to the distribution function $G(\eta)$ for the value of the output quantity Y can be obtained as follows. Sort the values y_r, $r = 1, \ldots, M$, provided in Section 5.4 into non-decreasing order. Denote the sorted values by $y_{(r)}$, $r = 1, \ldots, M$. Assign uniformly spaced cumulative probabilities $p_r = (r - 1/2)/M$, $r = 1, \ldots, M$, to the ordered values.[15]

Form $\widehat{G}(\eta)$ as the piecewise-linear function joining the M points $(y_{(r)}, p_r)$, $r = 1, \ldots, M$:

$$\widehat{G}(\eta) = \frac{r - 1/2}{M} + \frac{\eta - y_{(r)}}{M(y_{(r+1)} - y_{(r)})}, \qquad y_{(r)} \leq \eta \leq y_{(r+1)}, \tag{2}$$

for $r = 1, \ldots, M - 1$. Table 21 specifies the evaluation of an approximation to the distribution function for the value of the output quantity, and indicates the input and output parameters necessary for its determination.

The values of $y_{(r)}$ (or y_r), when assembled into a histogram (with suitable cell widths) form a frequency distribution that, when normalised to have unit area, can be taken as an approximation $\widehat{g}(\eta)$ to the PDF $g(\eta)$ for the value of Y. Calculations are not generally carried out in terms of this histogram, the resolution of which depends on the choice of cell size, but in terms of the approximation $\widehat{G}(\eta)$ to the distribution function $G(\eta)$. The histogram can, however, be useful as an aid to understanding the nature of the PDF, e.g., the extent of its asymmetry.

5.6 Measurement result and the associated standard uncertainty

The expectation $\widehat{y}$ of the function $\widehat{G}(\eta)$ (see (2)) approximates the expectation of the PDF $g(\eta)$ for the value of Y and is taken as the estimate y of the value of the output quantity. The standard deviation $u(\widehat{y})$ of $\widehat{G}(\eta)$ approximates the standard

[15] The values p_r, $r = 1, \ldots, M$, are the midpoints of M contiguous probability intervals of width $1/M$ between zero and one.

Input parameters	
M	Number of samples (equal to the number of Monte Carlo trials)
$\boldsymbol{y}$	Model values $(y_1, \ldots, y_M)$ obtained by evaluating the measurement model for each of the M samples $\boldsymbol{x}_r$ from each of the PDFs for the values of the N input quantities, i.e., $y_r = f(\boldsymbol{x}_r)$
Output parameters	
$\tilde{\boldsymbol{y}}$	Model values $(y_{(1)}, \ldots, y_{(M)})$ sorted into non-decreasing order
$\boldsymbol{p}$	Probabilities $(p_1, \ldots, p_M)$, defined by $$p_r = (r - 1/2)/M.$$ The function $\widehat{G}(\eta)$ determined as the piecewise-linear function joining the points $(y_{(r)}, p_r),\ r = 1, \ldots, M$, provides an approaximation to the distribution function for the value of Y. ($G(\eta)$ is defined only for values of η corresponding to values of probability p in the interval $M/2 \leq p \leq 1 - M/2$. Indeed, it should not be used near the endpoints of this interval, because it is less reliable there.)
Numerical analysis	
	It is recommended that a sorting algorithm that takes a number of operations proportional to $M \log M$ be used (e.g., [37]). A naive algorithm would take a time proportional to M^2 and may make the computation time unacceptable

Table 21: Distribution function for the value of the output quantity.

deviation of $g(\eta)$ and is taken as the standard uncertainty $u(y)$ associated with y. $\widehat{y}$ can be taken as the arithmetic mean

$$\widehat{y} = \frac{1}{M}\sum_{r=1}^{M} y_r \tag{3}$$

formed from the M values $y_1, \ldots, y_M$, and the standard deviation $u(\widehat{y})$ determined from

$$u^2(\widehat{y}) = \frac{1}{M-1}\sum_{r=1}^{M}(y_r - y)^2. \tag{4}$$

Table 22 specifies the evaluation of the measurement result y and the associated standard uncertainty $u(y)$ using Formulae (3) and (4), respectively, and indicates the input and output parameters necessary for their determination.

Formulae (3) and (4) do not in general provide values that are identical to the expectation and standard deviation, respectively, of $\widehat{G}(\eta)$. The latter values are given by

$$\widehat{y} = \frac{1}{M}{\sum_{r=1}^{M}}'' y_{(r)} \tag{5}$$

and

$$u^2(\widehat{y}) = \frac{1}{M}\left({\sum_{r=1}^{M}}''(y_{(r)} - \widehat{y})^2 - \frac{1}{6}\sum_{r=1}^{M-1}(y_{(r+1)} - y_{(r)})^2\right), \tag{6}$$

where the double prime on the summation in Expression (5) and on the first summation in Expression (6) indicates that the first and the last terms are to be taken with weight one half. For a sufficiently large value of M, the values obtained using (3) and (4) are generally indistinguishable for practical purposes from those given by (5) and (6).

The value of y so obtained yields the smallest mean squared deviation over all possible choices of the measurement result. However, the value will not in general agree with the model evaluated at the estimates of the values of the input quantities [4, Clause 4.1.4]. Agreement (in a practical sense) will be achieved for a large value of M when the model is linear in the input quantities. Whether this general lack of agreement is important depends on the application. The value of y, even in the limit as $M \to \infty$, is not in general equal to the model evaluated at the expectation values of the PDFs for the values of the input quantities, unless the model is linear [4, Clause 4.1.4].

The evaluations of y and $u(y)$ (Table 22) require the summation of M numbers with M large (typically of the order of 10^5 or 10^6: Section 5.2). The natural way to evaluate, for example, the summation

$$S = \sum_{r=1}^{M} y_r$$

Input parameters	
M	Number of samples (equal to the number of Monte Carlo trials)
$\boldsymbol{y}$	Model values $(y_1, \ldots, y_M)$ obtained by evaluating the measurement model for each of the M samples $\boldsymbol{x}_r$ from each of the PDFs for the values of the N input quantities, i.e., $y_r = f(\boldsymbol{x}_r)$
Output parameters	
y	Measurement result: the arithmetic mean of the model values, defined by $$y = \frac{1}{M}\sum_{r=1}^{M} y_r$$
$u(y)$	Combined standard uncertainty: the standard deviation of the model values, defined by $$u^2(y) = \frac{1}{M-1}\sum_{r=1}^{M}(y_r - y)^2$$
Numerical analysis	
	It is important to use the above formula for $u(y)$ rather than the mathematically equivalent formula $$u^2(y) = \frac{M}{M-1}\left(\frac{1}{M}\sum_{r=1}^{M} y_r^2 - y^2\right).$$ For cases in which $u(y)$ is very much smaller than $\lvert y\rvert$ (in which case the $y_r,\ r = 1, \ldots, M$, have a number of leading digits in common) the latter formula suffers from subtractive cancellation (involving a mean square less a squared mean). The cancellation effects can be so severe that the resulting value of $u(y)$ may have too few correct significant figures for the uncertainty evaluation to be valid [9]

Table 22: The measurement result and the associated standard uncertainty.

is *recursively*, i.e., by the following procedure:

$S = 0$
for $r = 1 : M$
    $S = S + y_r$
end

When implemented using floating-point arithmetic the *computed* result $\widehat{S}$ will differ from the *mathematical* result S by an amount E that is the result of rounding errors associated with each floating-point operation (of which there will be many if M is large). Alternative procedures for undertaking the summation are available and have been designed with the aim of reducing the magnitude of E [24]. One such procedure is *Kahan summation* [26]. This procedure for evaluating S is described in Appendix D.

5.7 Coverage interval for the value of the output quantity

Let α denote any value between zero and $1 - p$, where p is the required coverage probability (e.g., 0.95). The endpoints of a $100p$ % coverage interval for the value of the output quantity are $G^{-1}(\alpha)$ and $G^{-1}(p + \alpha)$, i.e., the α– and $(p + \alpha)$–quantiles of $G(\eta)$. Here, the β–quantile is the value of η for which $G(\eta) = \beta$.

The endpoints of the coverage interval can be obtained from the approximation $\widehat{G}(\eta)$ to $G(\eta)$ obtained in Section 5.5 by inverse linear interpolation. To find the lower endpoint $y_{\rm low}$ such that $\alpha = \widehat{G}(y_{\rm low})$, identify the index r for which the points $(y_{(r)}, p_r)$ and $(y_{(r+1)}, p_{r+1})$ satisfy

$$p_r \le \alpha < p_{r+1}.$$

Then, by inverse linear interpolation,

$$y_{\rm low} = y_{(r)} + \left(y_{(r+1)} - y_{(r)}\right) \frac{\alpha - p_r}{p_{r+1} - p_r}.$$

Similarly, the upper endpoint $y_{\rm high}$ is calculated from

$$y_{\rm high} = y_{(s)} + \left(y_{(s+1)} - y_{(s)}\right) \frac{p + \alpha - p_s}{p_{s+1} - p_s},$$

where the index s is identified to satisfy

$$p_s \le p + \alpha < p_{s+1}.$$

The choice $\alpha = 0.025$ gives the coverage interval defined by the 0.025– and 0.975–quantiles. This choice provides a 95 % coverage interval that is probabilistically

symmetric. The probability is 2.5 % that the value of Y is smaller than the left-hand endpoint of the interval and 2.5 % that it is larger than the right-hand endpoint. If $g(\eta)$ is symmetric about its expectation, this coverage interval is symmetric about the estimate y of the value of the output quantity, and the left-hand and right-hand endpoints of the coverage interval are equidistant from y.

A value of α different from 0.025 would generally be appropriate were the PDF asymmetric. Usually the shortest coverage interval is required, because it corresponds to the best possible location of the value of the output quantity Y for a specified probability. It is given by the value of α satisfying $g(G^{-1}(\alpha)) = g(G^{-1}(p+\alpha))$, if $g(\eta)$ is single-peaked, and in general by the value of α such that $G^{-1}(p+\alpha) - G^{-1}(\alpha)$ is a minimum. If $g(\eta)$ is symmetric, the shortest coverage interval is given by taking $\alpha = (1-p)/2$.

The shortest coverage interval can generally be obtained computationally from $\widehat{G}(\eta)$ by determining α such that $\widehat{G}^{-1}(p+\alpha) - \widehat{G}^{-1}(\alpha)$ is a minimum. A straightforward approach to determining the minimum is to evaluate $\widehat{G}^{-1}(p+\alpha) - \widehat{G}^{-1}(\alpha)$ for a number of choices $\{\alpha_k\}$ of α between zero and $1-p$, and to choose that value α_ℓ from the set $\{\alpha_k\}$ yielding the minimum value from the set $\{\widehat{G}^{-1}(p+\alpha_k) - \widehat{G}^{-1}(\alpha_k)\}$.

5.8 Sensitivity analysis

With Monte Carlo simulation there is no immediate counterpart of a sensitivity coefficient (Section 4.2) since Monte Carlo simulation operates in terms of the actual nonlinear model rather than a linearized counterpart. Recall that with a linear model the sensitivity coefficients "reproduce" linear effects, and for a nonlinear model the sensitivity coefficients provide first-order information. Therefore, those practitioners accustomed to the LPU approach may find the absence of sensitivity coefficients disconcerting.

It is possible and very straightforward, however, to adapt the Monte Carlo simulation procedure such that it provides information that in a sense constitutes a nonlinear counterpart of a sensitivity coefficient. Consider holding the values of all input quantities but one, say X_k, at their nominal values. In this setting the model effectively becomes one having a single input quantity, viz., X_k. Sample values from the PDF for the value of this input quantity and determine an approximation for the distribution of the value of the output quantity with respect to X_k. The standard deviation $\widehat{u}_k(y)$ of this distribution is taken as an approximation to the component of the (combined) standard uncertainty corresponding to X_k.

The use of "nonlinear" sensitivity coefficients in place of "linear" sensitivity coefficients permits individual nonlinear effects to be taken into account. A "nonlinear"

sensitivity coefficient $\widehat{c}_k$ is defined by

$$\widehat{c}_k = \frac{\widehat{u}(y_k)}{u(x_k)}.$$

It will be equal to the magnitude $|c_k|$ of the "linear" sensitivity coefficient c_k in a case where the model is linear in X_k, and be close to its value when the nonlinearity with respect to X_k is negligible. When $\widehat{c}_k$ is appreciably different from c_k the nonlinearity effect may noticeably influence the standard uncertainty $u(y)$. Thus, the deviation of $\widehat{u}(y_k)$ from $u(y_k) = c_k u(x_k)$ can be used as an approximate measure of the influence of model nonlinearity with regards to X_k alone.

The sensitivity coefficients so obtained are not generally to be taken in conjunction with the standard uncertainties associated with the estimates of the input quantities as the only contributions to the standard uncertainty associated with the estimate of the value of the output quantity. There will be further contributions arising from any interaction (i.e., non-additive) terms in the model.

Table 23 specifies the calculation of a nonlinear sensitivity coeficient relating to the input quantity X_k using Monte Carlo simulation.

5.9 Adaptive Monte Carlo procedure

A basic implementation of an adaptive Monte Carlo procedure is described as follows. It is based on carrying out an increasing number of Monte Carlo trials until the various quantities of interest have stabilised in a statistical sense. A quantity is deemed to have stabilised if twice the standard deviation associated with the estimate of its value is less than the degree of approximation required in the standard uncertainty $u(y)$.

A practical approach consists of carrying out a sequence of Monte Carlo calculations, each containing a relatively small number, say $M = 10^4$, trials.[16] For each Monte Carlo calculation in the sequence, y, $u(y)$ and the endpoints of a 95% coverage interval are formed from the results obtained, as in Sections 5.6 and 5.7. Denote by $y^{(h)}$, $u(y^{(h)})$, $y^{(h)}_{\rm low}$ and $y^{(h)}_{\rm high}$ the values of y, $u(y)$ and the left- and right-hand endpoints of the 95% coverage interval for the hth member of the sequence.

After the hth Monte Carlo calculation (apart from the first) in the sequence, the arithmetic mean y of the values $y^{(1)}, \ldots, y^{(h)}$ and the standard deviation s_y associated with this arithmeic mean are formed. The counterparts of these statistics

[16] It is recommended that each sequence of calculations be performed using the same random number generator (albeit using a different part of the generated sequence) or two or more independent generators. This is because the statistical properties of a random number generator are defined "within" sequences and not "between" sequences (as, for example, two sequences obtained from the same number generator may overlap even if initialised with different seeds).

Input parameters	
M	Number of samples (equal to the number of Monte Carlo trials)
$\boldsymbol{y}^k$	Model values $(y_1^k, \ldots, y_M^k)$ obtained by evaluating the measurement model for each of the M samples $\boldsymbol{x}_r^k$ obtained by holding all inputs but one, X_k, at their nominal values and sampling from the input PDF $g_k(\xi_k)$ for the value of X_k
$u(x_k)$	standard uncertainty associated with the estimate x_k of the value of X_k
Output parameters	
$\widehat{u}_k(y)$	Component of the (combined) standard uncertainty: the standard deviation of the model values, defined by $$\widehat{u}_k^2(y) = \frac{1}{M-1}\sum_{r=1}^{M}(y_r^k - y^k)^2$$ where $$y^k = \frac{1}{M}\sum_{r=1}^{M} y_r^k.$$
$\widehat{c}_k$	Nonlinear sensitivity coefficient, defined by $$\widehat{c}_k = \frac{\widehat{u}(y_k)}{u(x_k)}.$$

Table 23: Sensitivity analysis by Monte Carlo simulation.

for y are determined for $u(y)$, $y_{\rm low}$ and $y_{\rm high}$. If the largest of $2s_y$, $2s_{u(y)}$, $2s_{y_{\rm low}}$ and $2s_{y_{\rm high}}$ does not exceed the degree of approximation required in $u(y)$, the overall computation is regarded as having stabilised. The results from the total number of Monte Carlo trials taken are then used to provide the estimate of the value of the output quantity, the associated standard uncertainty and the coverage interval for the value of the output quantity.

5.10 Multivariate Monte Carlo procedure

The above treatment applies to *univariate* real-valued models. It can straightforwardly be adapted to apply Monte Carlo simulation to multivariate or complex-valued models. There is not, however, a ready counterpart of a coverage interval.

The multivariate model is

$$\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X}).$$

M samples $\boldsymbol{x}_r$ of the values of the input quantities $\boldsymbol{X}$ are taken as before. For each $\boldsymbol{x}_r$, evaluate the model as previously, except now the output values $\boldsymbol{y}_r = \boldsymbol{f}(\boldsymbol{x}_r)$ are $m \times 1$ vectors.

Assemble these output vectors into an $m \times M$ matrix:[17]

$$\boldsymbol{\Psi} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M).$$

From this matrix the covariance matrix $\boldsymbol{V}_{\boldsymbol{y}}$ associated with estimates $\boldsymbol{y}$ of the values of the output quantities $\boldsymbol{Y}$ is calculated from

$$\boldsymbol{V}_{\boldsymbol{y}} = \frac{1}{M-1}\boldsymbol{\Psi}'(\boldsymbol{\Psi}')^{\rm T},$$

where $\boldsymbol{\Psi}'$ is $\boldsymbol{\Psi}$ corrected for the sample means over all M trials, i.e., with the arithmetic mean of the jth row subtracted from all elements in that row, for $j = 1, \ldots, M$. Table 24 specifies the evaluation of the covariance matrix $\boldsymbol{V}_{\boldsymbol{y}}$ in terms of the vectors $\boldsymbol{y}_r$, and indicates the input and output parameters necessary for its determination.

This covariance matrix contains (generally a more reliable estimate of) the information that would be delivered by a linear analysis such as LPU. (In fact, it provides more than LPU, since that procedure does not in general cover multivariate models.) The matrix $\boldsymbol{\Psi}$ provides much richer information, however, in the following sense. Any column of $\boldsymbol{\Psi}$ corresponds to the values of the output quantities for one choice (sample) of the input quantities. *Any* (scalar) derived quantity can be determined from this single set of output values. This quantity can be calculated for

[17] The symbol $\boldsymbol{\Psi}$ is (reluctantly) used to denote the matrix of y-vectors, since Y is used to denote a scalar output quantity and $\boldsymbol{Y}$ a vector output quantity.

Input parameters	
m	Number of output quantities
M	Number of samples (equal to the number of Monte Carlo trials)
$\boldsymbol{\Psi}$	An $m \times M$ matrix containing the model output values $\boldsymbol{y}_r$, i.e., $$\boldsymbol{\Psi} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M).$$
Output parameters	
$\boldsymbol{V}_{\boldsymbol{y}}$	Covariance matrix associated with estimates $\boldsymbol{y}$ of values of the output quantities $\boldsymbol{Y}$, defined by $$\frac{1}{M-1}\boldsymbol{\Psi}'(\boldsymbol{\Psi}')^{\mathrm{T}},$$ where $\boldsymbol{\Psi}'$ is $\boldsymbol{\Psi}$ corrected for the sample means

Table 24: The covariance matrix associated with estimates $\boldsymbol{y}$ of the values of the output quantities $\boldsymbol{Y}$.

all columns, the resulting $1 \times M$ row vector being used to provide an approximation to the distribution function for the value of that quantity (as in Section 5.5). In particular, the distribution function can be used to provide a coverage interval for the value of the derived quantity (as in Section 5.7). Another quantity could be so introduced and the two row vectors used to compute any statistics required (mean, median, etc.) and the pair of vectors used to estimate correlation effects. Thus, the matrix $\boldsymbol{\Psi}$ is a very valuable array, being the basis of almost unlimited statistical information.[18]

6 Validation of the law of propation of uncertainty

LPU has some limitations [4, 12]. Although the procedure can be expected to work well in many circumstances, it is generally difficult to quantify the effects of the approximations involved, viz., linearization, the Welch-Satterthwaite formula for the effective degrees of freedom and the assumption that the value of the output quantity is Gaussian (i.e., that the Central Limit Theorem is applicable). Indeed, the degree of difficulty of doing so would typically be considerably greater than that required to apply Monte Carlo simulation. Therefore, since these circumstances cannot readily be tested, any cases of doubt should be validated. To this end,

[18] Note, however, that consideration is not given to using the matrix $\boldsymbol{\Psi}$ to obtain an approximation to the (joint) distribution function for the value of the (multivariate) output quantity $\boldsymbol{Y}$. This is in contrast to the case of a univariate output quantity Y (Section 5.5).

since the propagation of distributions is more general, it is recommended that both LPU and the Monte Carlo approach are applied and the results compared. If the comparison is favourable, LPU can be used on this occasion and for sufficiently similar problems in the future. Otherwise, consideration can be given to using Monte Carlo simulation instead.

Specifically, it is recommended that the two steps below and the following comparison process are carried out.

1. Apply LPU to yield a 95% coverage interval $y \pm U(y)$ for the value of the output quantity.
2. Apply Monte Carlo simulation to yield the standard uncertainty $u(y)$ associated with an estimate of the value of the output quantity and the endpoints y_{low} and y_{high} of a 95% coverage interval for the value of the output quantity.

A comparison procedure is based on the following objective: determine whether the coverage intervals obtained by LPU and Monte Carlo simulation agree to a stipulated degree of approximation. This degree of approximation is assessed in terms of the endpoints of the coverage intervals and corresponds to that given by expressing the standard uncertainty $u(y)$ to what is regarded as a meaningful number of *significant decimal digits*.

The procedure is as follows:

1. Let n_{ndig} denote the number of significant digits regarded as meaningful in the numerical value of $u(y)$. Usually, $n_{\text{ndig}} = 1$ or $n_{\text{ndig}} = 2$. Express the value of $u(y)$ in the form $a \times 10^r$, where a is an n_{ndig}–digit integer and r an integer. The comparison accuracy is
$$\delta = \frac{1}{2}10^{-r}.$$
2. Compare the coverage intervals obtained by LPU and Monte Carlo simulation to determine whether the required number of correct digits in the coverage interval provided by LPU has been obtained. Specifically, determine the quantities
$$|y - U(y) - y_{\text{low}}|$$
and
$$|y + U(y) - y_{\text{high}}|,$$
viz., the absolute values of the differences of the respective endpoints of the two coverage intervals. Then, if both these quantities are no larger than δ the comparison is successful and LPU has been validated in this instance.

Example. The estimate of the value of the output quantity for a nominally 100 g standard of mass [4, Clause 7.2.2] is $y = 100.021\ 47$ g. The standard uncertainty $u(y) = 0.000\ 35$ g, Thus, $n_{\text{ndig}} = 2$ and $u(y)$ is expressed as 35×10^{-5} g, and so $a = 35$ and $r = -5$. Take $\delta = \frac{1}{2} \times 10^{-5}$ g $= 0.000\ 005$ g.

7 Conclusions

Software specifications have been proposed for three aspects of uncertainty evaluation, viz.,

1. The law of propagation of uncertainty (LPU) procedure of the Guide to the Expression of Uncertainty in Measurement (GUM) [4]
2. A general approach [3, 12] based on Monte Carlo simulation as an implementation of the propagation of distributions
3. Validation of LPU, the use of Monte Carlo simulation to validate LPU in circumstances where there is doubt concerning its applicability [3, 12].

The specifications are not intended to be mandatory but indicative of the software units that are required for implementation of the above aspects.

The second edition of this document has been produced in the Software Support for Metrology (SS*f*M) programme 2001–2004. It is anticipated that a third edition, extending and revising the document further and keeping it in line with the evolving SS*f*M best-practice guide [12], will be produced in the SS*f*M 2004–2007 programme.

Acknowledgements

The work described here was supported by the National Measurement System Directorate of the UK Department of Trade and Industry as part of its NMS Software Support for Metrology programme.

The authors thank Martyn Byng (Numerical Algorithms Group Ltd.) for comments on a draft of this report.

References

[1] R. M. Barker. Good Practice Guide No. 5. Guide to EUROMETROS: a manual for users, contributors and testers. Technical report, National Physical Laboratory, Teddington, UK, 2004. To be published.

[2] P. Mac Berthouex and L. C. Brown. *Statistics for Environmental Engineers*. CRC Press, USA, 1994.

[3] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. Guide to the Expression of Uncertainty in Measurement, Supplement 1, Numerical Methods for the Propagation of Distributions. To be published.

[4] BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, and OIML. Guide to the Expression of Uncertainty in Measurement, 1995. ISBN 92-67-10188-9, Second Edition.

[5] C. Bischof, A. Carle, P. Hovland, P. Khademi, and A. Mauer. *ADIFOR 2.0 Users' Guide*, 1998.

[6] C. Bischof, L. Roh, and A. Mauer. *ADIC – An extensible automatic differentiation tool for ANSI-C*. ANL/MCS-P626-1196.

[7] R. Boudjemaa, M. G. Cox, A. B. Forbes, and P. M. Harris. Automatic differentiation techniques and their application in metrology. Technical Report CMSC 26/03, National Physical Laboratory, Teddington, UK, 2003.

[8] G. E. P. Box and M. Muller. A note on the generation of random normal variates. *Ann. Math. Statist.*, 29:610–611, 1958.

[9] M. G. Cox, M. P. Dainton, and P. M. Harris. Testing spreadsheets and other packages used in metrology. Testing functions for the calculation of standard deviation. Technical Report CMSC 07/00, National Physical Laboratory, Teddington, UK, 2000.

[10] M. G. Cox, A. B. Forbes, and P. M. Harris. Best Practice Guide No. 11. Numerical analysis for algorithm design in metrology. Technical report, National Physical Laboratory, Teddington, UK, 2004. To be published.

[11] M. G. Cox and P. M. Harris. Guidelines to help users select and use software for their metrology applications. Technical Report CMSC 04/00, National Physical Laboratory, Teddington, UK, 2000.

[12] M. G. Cox and P. M. Harris. Best Practice Guide No. 6. Uncertainty evaluation. Technical report, National Physical Laboratory, Teddington, UK, 2004.

[13] T. J. Dekker. Finding a zero by means of successive linear interpolation. In B. Dejon and P. Henrici, editors, *Constructive Aspects of the Fundamental Theorem of Algebra*, London, 1969. Wiley Interscience.

[14] L. Devroye. *Non-Uniform Random Number Generation*. Springer, New York, 1986.

[15] EA. Expression of the uncertainty of measurement in calibration. Technical Report EA-4/02, European Co-operation for Accreditation, 1999.

[16] C. Faure. Automatic differentiation for adjoint code generation: An introduction to automatic adjoint code generation. Technical Report 3555, INRIA, 1998.

[17] C. Faure and Y. Papegay. Odyssée version 1.6: The user's reference manual. Technical report, INRIA, 1997.

[18] G. S. Fisher. *Monte Carlo*. Springer-Verlag, New York, 1996.

[19] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, 1981.

[20] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, MD, USA, 1996. Third edition.

[21] A. Griewank and G. F. Corliss, editors. *Automatic Differentiation of Algorithms: Theory, Implementation and Applications*, Philadelphia, 1991. Society for Industrial and Applied Mathematics.

[22] A. Griewank, D. Juedes, H. Mitev, J. Utke, O. Vogel, and A. Walther. ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM TOMS*, 22(2):131–167, 1996.

[23] I. A. Harris and F. L. Warner. Re-examination of mismatch uncertainty when measuring microwave power and attenuation. *Proc. IEEE*, 128:35–41, 1981.

[24] N. J. Higham. The accuracy of floating point summation. *SIAM J. Sci. Comput.*, 14(4):783–799, 1993.

[25] I. D. Hill and B. A. Wichmann. Algorithm AS183. An efficient and portable pseudo-random number generator. *Appl. Statist.*, 31:188–190, 1982.

[26] W. Kahan. Further remarks on removing truncation errors. *Comm. ACM*, 8, 1995.

[27] A. Kinderman, J. Monahan, and J. Ramage. Computer methods for sampling from Student's t-distribution. *Math. Comput*, 31:1009–1018, 1977.

[28] D. E. Knuth. *The Art of Computer Programming, Vol. 2: Semi-numerical algorithms*. Addison-Wesley, Reading, Ma., USA, 1981.

[29] J. Leydold. Automatic sampling with the ratio-of-uniforms method. *ACM Trans. Math. Software*, 26:78–98, 2000.

[30] G. Marsaglia. A current view of random number generators. In *Computer Science and Statistics: The Interface*, Amsterdam, 1985. Elsevier.

[31] G. Marsaglia and A. Zaman. The KISS generator. Technical report, Department of Statistics, University of Florida, USA, 1993.

[32] M. McCoy. A review on random number generation and methods for generating normal distributions. Technical Report 113/2000, National Engineering Laboratory, UK, 2000.

[33] B. D. Ripley. *Stochastic Simulation*. J. Wiley, New York, 1987.

[34] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 1999.

[35] R. Y. Rubenstein. *Simulation and the Monte Carlo Method*. J. Wiley, New York, 1981.

[36] M. J. Salter, N. M. Ridler, and M. G. Cox. Distribution of correlation coefficient for samples taken from a bivariate normal distribution. Technical Report CETM 22, National Physical Laboratory, Teddington, UK, 2000.

[37] R. S. Scowen. QUICKERSORT, Algorithm 271. *Comm. ACM*, 8:669, 1965.

[38] D. Shiriaev and A. Griewank. *Computational Differentiation: Techniques, Applications, and Tools*, chapter ADOL-F: Automatic differentiation of Fortran codes. SIAM, 1996.

[39] G. Strang and K. Borre. *Linear Algebra, Geodesy and GPS*. Wiley, Wellesley-Cambridge Press, 1997.

[40] B. N. Taylor and C. E. Kuyatt. Guidelines for evaluating and expressing the uncertainty of NIST measurement results. Technical Report TN1297, National Institute of Standards and Technology, 1994.

[41] UKAS. The expression of uncertainty and confidence in measurement. Technical Report M 3003, United Kingdom Accreditation Service, 1997.

A Use of symbolic-algebra packages

This appendix is concerned with the use of a symbolic-algebra package to provide partial derivatives of a model, from which sensitivity coefficients can be determined (Section 4.2).

Table 25 specifies for a univariate, explicit, real-valued model, the use of a symbolic-algebra package to obtain sensitivity coefficients and indicates the input and output parameters associated with their determination. The input parameter f is used to specify the measurement model.

Counterparts would apply for the other categories of measurement model.

	Input parameters
N	Number of input quantities
f	Function specifying the model $Y = f(\boldsymbol{X})$ in terms of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$
$\boldsymbol{x}$	Column vector $(x_1, \ldots, x_N)^{\mathrm{T}}$ of estimates of the values of the input quantities $\boldsymbol{X}$. Some packages may not be able to make use of $\boldsymbol{x}$ in that they return algebraic expressions for $(\partial f/\partial X_1, \ldots, \partial f/\partial X_N)$ rather than these expressions evaluated at $\boldsymbol{X} = \boldsymbol{x}$
	Output parameters
$\boldsymbol{f'}$	Functions $(f'_1, \ldots, f'_N)$ representing algebraic expressions for $(\partial f/\partial X_1, \ldots, \partial f/\partial X_N)$, the partial derivatives of first order of the function f
$\boldsymbol{C}$	A $1 \times N$ vector of sensitivity coefficients, whose jth element is the partial derivative $\partial f/\partial X_j$ of $f(\boldsymbol{X})$ evaluated at $\boldsymbol{X} = \boldsymbol{x}$. Some symbolic-algebra packages may produce just $\boldsymbol{f'}$ and it will be the user's responsibility to evaluate $\boldsymbol{f'}$ at $\boldsymbol{X} = \boldsymbol{x}$

Table 25: Sensitivity coefficients obtained using a symbolic-algebra package.

When a symbolic-algebra package system is used, care needs to be taken that the mathematical expressions generated are stable with respect to their evaluation at the estimates of the values of the input quantities. For instance, suppose that (part of) a model is

$$Y = (X_1 - K_0)^4,$$

where K_0 is a specified constant. An automatic system might involve expansions such as Taylor series to generate the partial derivative of Y in the form

$$\frac{\partial Y}{\partial X_1} = 4X_1^3 - 12X_1^2K_0 + 12X_1K_0^2 - 4K_0^3, \tag{7}$$

and perhaps not contain a facility to generate directly or simplify this expression to

the *mathematically* equivalent form

$$\frac{\partial Y}{\partial X_1} = 4(X_1 - K_0)^3, \tag{8}$$

that would typically be obtained *manually*.

Suppose the estimate of the value of X_1 is $x_1 = 10.1$ and $K_0 = 9.9$. The value c_1 of the resulting sensitivity coefficient is $4(x_1 - K_0)^3 = 0.032$, correct to two significant figures. Both Formulae (7) and (8) deliver this value to this number of figures. The second, more compact, form is, however, much to be preferred. The reason is that Formula (8) suffers negligible loss of numerical precision when it is used to evaluate c_1, whereas, in contrast, Formula (7) loses figures in forming this value. To see why this the case, consider the contributions to the expression, evaluated and displayed here to a constant number of decimal places (corresponding to seven significant figures in the contribution of greatest magnitude):

$$\begin{aligned} 4x_1^3 &= 4121.204, \\ -12x_1^2K_0 &= -12118.79, \\ 12x_1K_0^2 &= 11878.81, \\ -4K_0^3 &= -3881.196. \end{aligned}$$

The sum of these values constitutes the value of c_1. To the numerical accuracy held, this value is 0.028, compared with the correct value of 0.032. The important point is that a value of order 10^{-2} has been obtained by the sum of positive and negative values of magnitude up to order 10^4. Almost inevitably, a loss of some six figures of numerical precision has resulted, as a consequence of *subtractive cancellation*.

For different values of x_1 and K_0 or in other situations the loss of figures could be greater or less. The concerning matter is that this loss has resulted from such a simple model. The effects in the case of a sophisticated model or a multi-stage model [12] could well be compounded, with the consequence that there are dangers that the sensitivity coefficients formed in this way will be insufficiently accurate. Therefore, care must be taken in using sensitivity coefficients that are evaluated from the expressions provided by some software for algebraic differentiation. Such a system, if used, should evidently be chosen with care. One criterion in making a choice is whether the system offers comprehensive facilities for carrying out algebraic simplification, thus ameliorating the danger of loss of figures. Even then, some form of validation should be applied to the numerical values so obtained.[19]

[19] Numerical analysis issues such as that discussed are addressed as part of the numerical analysis project of the SS*f*M programme [10].

B Use of finite-difference formulae

This appendix is concerned with the use of a finite-difference formula to provide partial derivatives of a model, from which sensitivity coefficients can be determined (Section 4.2).

Numerical approximations to the values of derivatives can be obtained using finite-differences techniques. Given a value i ($1 \leq i \leq N$), set all $X_k = x_k$, apart from X_i, i.e., hold the values of all input quantities, apart from the ith, at their nominal values. Denote the resulting function of X_i by $f_i(X_i)$.

A typical finite difference approximation to $\partial Y/\partial X_i$ evaluated at $\boldsymbol{x}$ is

$$\left.\frac{\partial Y}{\partial X_i}\right|_{\boldsymbol{X}=\boldsymbol{x}} \approx \frac{f_i(x_i+\delta_i) - f_i(x_i)}{\delta_i},$$

where δ_i is a "suitably small" increment in x_i (see below). Note that $f_i(x_i) \equiv f(\boldsymbol{x})$ will already have been formed in evaluating the model at the estimates $\boldsymbol{x}$ of the input quantities.

The approximation can be perceived as follows. Consider the graph of $f_i(X_i)$. The formula gives the gradient of the chord joining the points $(x_i,\ f_i(x_i))$ and $(x_i+\delta_i,\ f_i(x_i+\delta_i))$. This gradient approximates the gradient of the tangent at $(x_i,\ f_i(x_i))$ to the graph of the function, which is of course the required derivative.

The choice of δ_i is important. If it is too great, the formula gives a large approximation error, i.e., the tangent and the chord point in appreciably different directions. If it is too small, the formula gives a large subtractive cancellation error, since the values of $f_i(x_i)$ and $f_i(x_i+\delta_i)$ will have many common leading figures.

A generally more accurate form, requiring an additional function evaluation, is

$$\left.\frac{\partial Y}{\partial X_i}\right|_{\boldsymbol{X}=\boldsymbol{x}} \approx \frac{f_i(x_i+\delta_i) - f_i(x_i-\delta_i)}{2\delta_i}.$$

For a given value of δ_i, the magnitude of the approximation error is generally reduced using this form. Thus the value of δ_i can be larger, affording a better balance between approximation and cancellation errors.

The GUM, in Clause 5.1.3, suggests the use of the second formula with $\delta_i = u(x_i)$. This choice can generally be expected to be acceptable, although there may be circumstances when it is not.[20]

Table 26 specifies, for a univariate, explicit, real-valued model, the use of a finite-difference formula to obtain sensitivity coefficients, and indicates the input and

[20] For example, in cases where $u(x_i)$ is large and the non-linearity of f as a function of X_i is appreciable.

output parameters associated with their determination. The input parameter f is used to provide information about the measurement model, and may take the form of a function for evaluating the model as in Section 3, Table 5.

Counterparts would apply for the other categories of measurement model.

Input parameters	
N	Number of input quantities
f	Function specifying the model $Y = f(\boldsymbol{X})$ in terms of the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$
$\boldsymbol{x}$	Column vector $(x_1, \ldots, x_N)^{\mathrm{T}}$ of estimates of the values of the input quantities $\boldsymbol{X}$
$\boldsymbol{u}$	Column vector $(u_1, \ldots, u_N)^{\mathrm{T}} \equiv (u(x_1), \ldots, u(x_N)^{\mathrm{T}}$ of standard uncertainties associated with the estimates of the values of the input quantities. This parameter would be used if the GUM recommendation for estimating the sensitivity coefficients were adopted. It would not be used if a finite-difference formula were used that attempted to provide a sensible compromise between the loss of accuracy due to truncation (approximation) error and that due to subtractive cancellation
Output parameters	
$\boldsymbol{C}$	A $1 \times N$ vector of sensitivity coefficients, whose jth element is an estimate obtained using finite differences of the partial derivative $\partial f / \partial X_j$ of $f(\boldsymbol{X})$ evaluated at $\boldsymbol{X} = \boldsymbol{x}$

Table 26: Sensitivity coefficients obtained using a finite-difference formula.

C Use of program differentiation techniques

Automatic differentiation (AD) is a set of techniques aimed at "differentiating the program" that computes a function value [7, 16]. AD is an accurate method in the sense that AD applies the rules of calculus in a repetitive way to an algorithmic specification of a function and, in exact arithmetic, will produce the exact answer. Like symbolic algebra, AD relies on the fact that any program, no matter how complex it is, can be broken down into a finite combination of elementary operators such as arithmetic operations (e.g., $+$, $-$) and elementary functions (e.g., $\sin x$, $\cos x$, e^x) [6, 21].

There are two different approaches to differentiating program code: *operator overloading* and *source-to-source transformation*. In the operator overloading approach, the basic arithmetic operations and intrinsic functions are assigned routines that calculate the derivatives of the operator output in addition to the calculation of the function value. The source code of the function is progressively differentiated by calling these routines at the same time as each operation is performed in the eval-

uation of the program. Operator overloading is only allowed by a limited number of programing languages such as Fortran 90, ADA, C++ and Matlab. Examples of software packages that use this approach are ADOL-C [22] and ADOL-F [38].

The source-to-source approach defines a new source code for calculating the derivatives explicitly obtained from the program function evaluation source code. However, the implementation of this method requires considerable programming effort. Examples of software packages that use this approach are ADIFOR [5] and ODYSSEE [17].

D Use of Kahan summation

Suppose a and b are floating-point numbers with $|a| \geq |b|$, and let $\widehat{s}$ denote the floating-point sum of a and b:

$$\widehat{s} = fl(a + b).$$

Then, the floating-point value $\widehat{e}$:

$$\widehat{e} = fl(-(((a + b) - a) - b)) = fl((a - \widehat{s}) + b)$$

is an estimate of the error $(a+b)-\widehat{s}$. Kahan's summation procedure [26] uses this result to apply a correction $\widehat{e}$ at every step of a recursive summation procedure for evaluating

$$S = \sum_{r=1}^{M} y_r.$$

The procedure may be written as follows:

$S = 0$
$e = 0$
for $r = 1 : M$
    $a = S$
    $b = y_r + e$
    $S = a + b$
    $e = (a - S) + b$
end

The method has two weaknesses: $\widehat{e}$ is not necessarily the exact correction and the addition $b = y_r + e$ is not performed exactly. Nevertheless, the use of the procedure brings a benefit in the form of an improved error bound compared with that for the (basic) recursive scheme for evaluating S [24].