NPL
National Physical Laboratory

Report

NPL Report
DEM-ES-010 (CMSC 40/04 revised)

Software specifications for uncertainty evaluation

M G Cox and P M Harris

September 2006

National Physical Laboratory | Hampton Road | Teddington | Middlesex | United Kingdom | TW11 0LW
Switchboard 020 8977 3222 | **NPL Helpline** 020 8943 6880 | **Fax** 020 8614 0446 | **www.npl.co.uk**

Department for
Innovation,
Universities &
Skills

NPL Report DEM-ES-010 (CMSC 40/04 revised)

Software specifications for uncertainty evaluation

M G Cox and P M Harris
Mathematics and Scientific Computing Group

September 2006

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 evaluation 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, accounting for revision of the SS*f*M best-practice guide to uncertainty evaluation and the preparation of the first Supplement to the "Guide to the expression of uncertainty in measurement" (GUM).

Revised/Corrected November 2008

NPL Report DEM-ES-010 (CMSC 40/04 revised)

ISSN 1754–2960

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

We gratefully acknowledge the financial support of the UK Department for Innovation, Universities and Skills (National Measurement System Directorate)

Approved on behalf of the Managing Director, NPL by Jonathan Williams,
Knowledge Leader for the Electrical and Software team

Contents

1 Introduction 1

1.1 The stages of uncertainty evaluation . . . 3

1.2 Document history . . . 5

2 Formulation 6

2.1 Formulation based on analysing repeated indications . . . 6

2.1.1 Sample mean and its standard deviation . . . 7

2.1.2 Covariance associated with two sample means . . . 7

2.1.3 Uncertainty matrix for input quantities . . . 10

2.2 Formulation based on assigning probability distributions . . . 11

3 Numerical evaluation of the measurement model 12

4 GUM uncertainty framework 15

4.1 Procedure . . . 15

4.2 Sensitivity coefficients . . . 16

4.3 Uncertainty associated with the estimate of the output quantity . . . 21

4.4 Effective degrees of freedom . . . 21

4.5 Expanded uncertainty . . . 22

5 Monte Carlo method 24

5.1 Procedure . . . 24

5.2 The number of Monte Carlo trials . . . 26

5.3 Sampling from the probability density functions . . . 26

5.3.1 Rectangular distribution . . . 27

5.3.2 Gaussian distribution . . . 28

5.3.3 t–distribution . . . 30

5.3.4 General distributions . . . 31

5.3.5 Multivariate Gaussian distribution . . . 31

5.4 Numerical evaluation of the model . . . 33

5.5 Estimate of the output quantity and the associated standard uncertainty . . . 35

5.5.1 Approximation to the probability density function . . . 37

5.6 Discrete representation of the distribution function . . . 39

5.6.1 Approximation to the distribution function . . . 39

5.7 Coverage interval . . . 41

5.8 Sensitivity analysis . . . 42

5.9 Adaptive Monte Carlo procedure . . . 43

5.10 Multivariate Monte Carlo procedure . . . 45

6 Validation of the GUM uncertainty framework 46

7 Conclusions 48

References **49**

A Use of symbolic-algebra packages **52**

B Use of finite-difference formulae **54**

C Use of program differentiation techniques **56**

D Use of Kahan summation **57**

E Updating procedure in an implementation of a Monte Carlo method **58**

List of Tables

1 Sample mean, standard deviation and degrees of freedom 8
2 Covariance associated with two sample means 9
3 Uncertainty matrix from repeated indications 10
4 Expectation, standard uncertainty and degrees of freedom for some common probability distributions 11
5 Evaluation of a univariate, explicit, real model 13
6 Evaluation of a multivariate, explicit, real model 13
7 Evaluation of a univariate, implicit, real model 14
8 Evaluation of a multivariate, implicit, real model 14
9 Sensitivity coefficients for a univariate, explicit, real model 18
10 Sensitivity coefficients for a multivariate, explicit, real model 18
11 Sensitivity coefficients for a univariate, implicit, real model 19
12 Sensitivity coefficients for a multivariate, implicit, real model 20
13 Uncertainty associated with the estimate of the output quantity 22
14 Effective degrees of freedom 23
15 Expanded uncertainty 23
16 Pseudo-random number generation 27
17 Pseudo-random number generator for the rectangular distribution 29
18 Pseudo-random number generator for the Gaussian distribution 30
19 Pseudo-random number generator for the t–distribution 30
20 Pseudo-random number generator for the multivariate Gaussian distribution 32
21 Estimate of the output quantity and the associated standard uncertainty . . . 36
22 Approximation to the probability density function 38
23 Discrete representation of the distribution function 39
24 Approximation to the distribution function 40
25 Shortest coverage interval 42
26 Sensitivity analysis using a Monte Carlo procedure 44
27 Uncertainty matrix for a multivariate model 46
28 Sensitivity coefficients obtained using a symbolic-algebra package 52
29 Sensitivity coefficients obtained using a finite-difference formula 55

List of Figures

1 Uncertainty evaluation using the GUM uncertainty framework 17
2 Uncertainty evaluation using a Monte Carlo method 25
3 Points drawn from a bivariate Gaussian distribution 34

1 Introduction

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 document 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 [12] to uncertainty evaluation 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 [11], accounting for revision of the SS*f*M best-practice guide to uncertainty evaluation and the preparation of the first Supplement [2] to the "Guide to the expression of uncertainty in measurement" (GUM) [3].

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 *GUM uncertainty framework* as summarised in Clause 8 of the "Guide to the expression of uncertainty in measurement" (GUM) [3][1]
2. More general calculations, consistent with the GUM, for the *propagation of distributions* based on the use of a *Monte Carlo method* [2, 12]
3. The validation of (the use of) the GUM uncertainty framework using a Monte Carlo method [2, 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] In this document the term "GUM uncertainty framework" will apply to the use of a first-order Taylor series approximation to the measurement model.

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 regenerated *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 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., non-linear least-squares adjustment)

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

The coverage of the software specifications is divided according to whether they relate to the *formulation* or *calculation* stages of the process of uncertainty evaluation (Section 1.1), or whether the procedure followed is in accordance with the GUM uncertainty framework [3, Clause 8], a Monte Carlo method [2, 12] or the validation of the GUM uncertainty framework using a Monte Carlo method [2, 12].

Where a vector of values, e.g., the estimates $\boldsymbol{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$ of the 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 [21]. 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 coverage intervals corresponding to a coverage probability of 95 % and is couched this way. It is generally extensible to other coverage probabilities.

[2] Related aspects are treated in the Software Support for Metrology programme as part of the software testing project [10].

The report is organised as follows. In the remainder of this section the above stages of uncertainty evaluation are discussed. In Section 2 the formulation stage and its outputs are described in terms of the information required for implementing the GUM uncertainty framework and a Monte Carlo method. 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: the GUM uncertainty framework, a Monte Carlo method and the validation of the GUM uncertainty framework using a Monte Carlo method. Conclusions are given in Section 7.

1.1 The stages of uncertainty evaluation

The software specifications are divided according to whether they relate to the *formulation* or *calculation* stages of the process of uncertainty evaluation.

In the formulation stage a measurement model is derived and the model inputs are quantified. Each model input quantity is characterized 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 indications [3, Clauses 2.3.2, 3.3.5] or are based on scientific judgement using all the relevant information available [3, Clauses 2.3.3, 3.3.5]. A PDF may also 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 obtained from a previous application of a Monte Carlo method, for example. A means for determining such an approximation is given in Section 5.6.

In the calculation stage the PDFs are propagated through the measurement model to obtain the PDF for the model output quantity. This PDF is used to obtain the expectation of the output quantity, taken as an estimate of the output quantity, the standard deviation of the quantity, taken as the standard uncertainty associated with the estimate, and a coverage interval corresponding to a specified coverage probability. The GUM uncertainty framework and a Monte Carlo method both provide approaches to undertaking the calculation stage of the process of uncertainty evaluation.

For an input quantity that is independent of the other input quantities, the GUM uncertainty framework requires for its operation only three parameters that summarize the information about the quantity represented by its PDF:

- The expectation, taken as an estimate of the input quantity
- The standard deviation, taken as the standard uncertainty associated with the estimate
- The corresponding degrees of freedom.

If there are dependencies, the procedure will also require the covariances associated with estimates of the mutually dependent quantities.[3]

Software can support the formulation stage 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 indications of a set of the input quantities, the data can be used to define an uncertainty matrix[4] or a multivariate Gaussian distribution, as appropriate. The specifications below relate to the (arithmetic) mean as a measure of location, and its associated standard deviation as a measure of dispersion. In addition, the determination of an uncertainty matrix is covered.

The output of the formulation stage is required to be in a form to act as inputs either to the GUM uncertainty framework or to a Monte Carlo procedure. For the GUM uncertainty framework, the inputs are the outputs of the formulation stage identified above. For a Monte Carlo method, the inputs are PDFs or, where appropriate, joint PDFs (where there are mutual dependencies) for the input quantities.

The basis of the GUM uncertainty framework is to "propagate the uncertainties" associated with estimates of the input quantities through the model to provide the uncertainties associated with estimates of the model output quantities. For this purpose it is required to

- Evaluate the model at the estimates of the input quantities to obtain estimates of the output quantities
- Determine sensitivity coefficients
- Propogate the uncertainty matrix associated with estimates of the input quantities through (a linearization of) the model to yield the uncertainty matrix associated with estimates of the output quantities
- Apply the Welch-Satterthwaite formula for effective degrees of freedom
- Determine percentage points of a Gaussian or a t–distribution in order to provide a coverage interval for the output quantity.

These calculations would apply, as appropriate, for each of the model types, and software can assist with the calculations.

For the Monte Carlo procedure it is necessary to "propagate (discrete representations of) the distributions" for the input quantities through the model. For this purpose it is required to

- Sample from the PDFs assigned to the input quantities

[3] The degree of mutual dependence associated with the estimates x_i and x_j 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.

[4] An uncertainty matrix is also known as a covariance or variance-covariance matrix [2].

- Evaluate the model for each sample of values for the input quantities to obtain a set of values for the output quantities

- Use the set of values for the output quantities to form a discrete representation of the distribution for the output quantities

- Determine from the discrete representation estimates of the output quantities and the associated uncertainty matrix

- Determine from the discrete representation a coverage interval for the output quantity.

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 the GUM uncertainty framework using a Monte Carlo method [2, 12]. This procedure constitutes operating both the GUM uncertainty framework and a Monte Carlo method for an uncertainty evaluation of concern, and carrying out an appropriate comparison of the results obtained.

1.2 Document history

This document is intended to align with and help support established guides [3, 15, 41, 42] on uncertainty evaluation and extend their functionality in a consistent manner. It is complementary to the best-practice guide [12] on uncertainty evaluation, 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 *Supplement* [2] to the "Guide to the expression of uncertainty in measurement" (GUM) [3] concerned with numerical methods for the propagation of distributions. Material from the evolving best-practice guide was used in various parts of the Supplement and subsequently refined appropriately for consistency with the remainder of the latter document. This revised report, produced during the third SS*f*M programme, April 2004 – March 2007, takes account of the revision of the best-practice guide (now in its third edition) and the preparation of the Supplement. In particular, material from the drafts of the Supplement prepared during the second and third programmes that had an origin in the first edition of the best-practice guide has been re-used.

2 Formulation

In the formulation stage of uncertainty evaluation, it is necessary to carry out various assignments of probability density functions (PDFs) to the input quantities of the measurement model. In the GUM uncertainty framework only the expectations and standard deviations of the quantities characterized by these PDFs, and covariances where appropriate, are used. For a Monte Carlo method 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 indications, obtained independently, of the input quantity X_i is available, a *statistical analysis* of the indications is undertaken to determine an estimate x_i of X_i together with the associated standard uncertainty $u_i = u(x_i)$ and corresponding degrees of freedom ν_i. These are the quantities used in the GUM uncertainty framework (Section 4). To apply a Monte Carlo method these parameters are used to assign a PDF to X_i. The assignment of PDFs to the input quantities based on the analysis of repeated indications is described in Section 2.1. In the GUM the assignment is referred to as "a Type A evaluation of standard uncertainty".

If repeated indications are not available a PDF for 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 a Monte Carlo method (Section 5). To apply the GUM uncertainty framework parameters x_i, u_i and ν_i are determined from the PDF. Results of the evaluation for a number of common PDFs are given in Section 2.2. In the GUM the assignment is referred to 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 GUM uncertainty framework an uncertainty matrix associated with the estimates of the relevant input quantities is evaluated. For a Monte Carlo method a joint (multivariate) PDF would be assigned to these input quantities.

In practice both types of assignment will be required, each applying to a subset of the input quantities. For the GUM uncertainty framework an uncertainty matrix would be constructed from those uncertainty matrices arising from the two types of analysis. For a Monte Carlo method a joint PDF would be constructed from those PDFs arising from the two types of analysis.

2.1 Formulation based on analysing repeated indications

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

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

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

These calculations are the basis for determining an uncertainty matrix associated with the estimates 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 indications, obtained independently, of the input quantity X_i. The estimate x_i of X_i and the associated standard uncertainty u_i are determined, respectively, as $\bar{x}_i$, the arithmetic mean of the indications, and s_i, the standard deviation associated with $\bar{x}_i$. The corresponding degrees of freedom ν_i is $q-1$. Table 1 defines the sample mean of a set of indications, the standard deviation associated with 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 GUM uncertainty framework (Section 4).

Given the estimate $\bar{x}_i$ and the associated standard uncertainty s_i so obtained, a PDF for X_i is assigned as follows [12]:

- If the distribution underlying the indications is unknown, assign a Gaussian PDF $\mathrm{N}(\bar{x}_i, s_i^2)$
- If the distribution underlying the indications is known to be Gaussian, assign a scaled and shifted t–distribution $\mathrm{t}_{\nu_i}(\bar{x}_i, s_i^2)$ with $\nu_i = q-1$ degrees of freedom.

This is the information required for a Monte Carlo method (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 indications, each pair obtained independently of the remaining pairs, of the input quantities X_i and X_j. An estimate x_i of X_i, together with the associated standard uncertainty u_i, may be determined as the mean and the standard deviation associated with the mean for the set of indications $(x_{i,1}, x_{i,2}, \ldots, x_{i,q})$ as in Section 2.1.1, and similarly for X_j. The covariance of X_i and X_j is taken as 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 indications of the two input quantities, and specifies the input and output parameters associated with its determination.

	Input parameters
q	Number of indications of the ith input quantity
$\boldsymbol{x}_i$	Indications $(x_{i,1}, x_{i,2}, \ldots, x_{i,q})$, obtained independently, of X_i
	Output parameters
$\bar{x}_i$	Sample mean, defined by $\displaystyle \bar{x}_i = \frac{1}{q}\sum_{k=1}^{q} x_{i,k}$
s_i	Standard deviation associated with the sample mean, defined by $\displaystyle s_i^2 = \frac{1}{q(q-1)}\sum_{k=1}^{q}(x_{i,k} - \bar{x}_i)^2$
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 $\displaystyle s_i^2 = \frac{1}{q-1}\left(\frac{1}{q}\sum_{k=1}^{q} x_{i,k}^2 - \bar{x}_i^2\right).$ 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 uncertainty evaluation to be valid [8]

Table 1: Sample mean, associated standard deviation and corresponding degrees of freedom from repeated indications of the ith input quantity.

	Input parameters
q	Number of indications of the ith and of the jth input quantities
$\boldsymbol{x}_i$	Indications $(x_{i,1}, x_{i,2}, \ldots, x_{i,q})$ of X_i
$\boldsymbol{x}_j$	Indications $(x_{j,1}, x_{j,2}, \ldots, x_{j,q})$ of X_j paired with those of X_i. Each pair $(x_{i,k}, x_{j,k})^{\mathrm{T}}$ is obtained independently of the remaining pairs
	Output parameter
$u_{i,j}$	Covariance associated with the sample means for the ith and jth input quantities, defined by $\displaystyle u_{i,j} = \frac{1}{q(q-1)} \sum_{k=1}^{q} (x_{i,k} - \bar{x}_i)(x_{j,k} - \bar{x}_j),$ where $\bar{x}_i$ and $\bar{x}_j$ are the sample means of the sets of indications $\boldsymbol{x}_i$ 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 $\displaystyle u_{i,j} = \frac{1}{q-1} \left(\frac{1}{q} \sum_{k=1}^{q} x_{i,k} x_{j,k} - \bar{x}_i \bar{x}_j \right).$ 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: Covariance associated with the sample means from repeated indications of the ith and jth input quantities.

2.1.3 Uncertainty matrix for input quantities

Suppose $(x_{1,k}, x_{2,k}, \ldots, x_{N,k})^{\mathrm{T}}$, $k = 1, \ldots, q$, are q N-tuples of indications, each N-tuple obtained independently of the remaining N-tuples, of the input quantities X_i, $i = 1, \ldots, N$, and that these are assembled into the $N \times q$ matrix $\boldsymbol{\Phi}$.[5] Let $\boldsymbol{\Phi}'$ be obtained from $\boldsymbol{\Phi}$ by correcting for the sample means, i.e., the mean $\bar{x}_i$ of the elements in the ith row is subtracted from all elements $x_{i,j}$, $j = 1, \ldots, q$, in that row. Then, the uncertainty matrix $\boldsymbol{V_x}$ of covariances $u_{i,j}$ associated with the sample means $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N$, of the input quantities $\boldsymbol{X} = (X_1, X_2, \ldots, X_N)^{\mathrm{T}}$ is given by

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

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

Input parameters	
N	Number of input quantities
q	Number of indications of each of the input quantities
$\boldsymbol{\Phi}$	$N \times q$ matrix containing the indications $x_{i,j}$, where $x_{i,j}$ is the jth measured value of the ith input quantity. Each N-tuple $(x_{1,k}, x_{2,k}, \ldots, x_{N,k})^{\mathrm{T}}$ is obtained independently of the remaining N-tuples
Output parameter	
$\boldsymbol{V_x}$	Uncertainty matrix of order N associated with the sample means for the input quantities $\boldsymbol{X}$, defined by $\dfrac{1}{q(q-1)}\boldsymbol{\Phi}'(\boldsymbol{\Phi}')^{\mathrm{T}},$ where $\boldsymbol{\Phi}'$ is $\boldsymbol{\Phi}$ corrected for the sample means

Table 3: Uncertainty matrix associated with the sample means from repeated indications of input quantities.

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

In practice there may be "simultaneous" indications $(x_{i,1}, x_{i,2}, \ldots, x_{i,q})$ for *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 uncertainty

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

matrix of order N constructed from these groups, together with the variances associated with estimates of those input quantities that are mutually independent.

2.2 Formulation based on assigning probability distributions

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 method this PDF is used directly. For the GUM uncertainty framework the expectation and the standard uncertainty of the input quantity would be evaluated in terms of the PDF and used. Table 4 gives the estimate x_i, the associated standard uncertainty u_i and the corresponding degrees of freedom ν_i for an input quantity characterized by some common distributions, viz., Gaussian, rectangular, triangular and U-shaped [16, 24].

Distribution	PDF	x_i	u_i	ν_i
Gaussian N(μ, σ^2)	$\frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$	μ	σ	∞
Rectangular R(a, 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 T(a, 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 U(a, b)	$\begin{cases} 0, & x \le a \\ \frac{1}{\pi\sqrt{w^2-(x-h)^2}}, & a < x < b, \\ 0, & b \le x. \end{cases}$	h	$\frac{w}{\sqrt{2}}$	∞

Table 4: Expectation, standard uncertainty and degrees of freedom for an input quantity characterized by some common probability distributions, where w denotes $(b-a)/2$ and h denotes $(b+a)/2$.

For example, suppose the input quantity X_i is assigned a rectangular distribution with limits a and b. Information on how to generate pseudo-random numbers from this distribution for the purpose of a Monte Carlo method 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 GUM uncertainty framework. For other PDFs assigned to X_i, Table 4 would be used in a similar way.

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

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

$$\boldsymbol{V}_i = \begin{bmatrix} u^2(x_i^{\mathrm{R}}) & u(x_i^{\mathrm{R}}, x_i^{\mathrm{I}}) \\ u(x_i^{\mathrm{R}}, x_i^{\mathrm{I}}) & u^2(x_i^{\mathrm{I}}) \end{bmatrix}$$

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

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 input quantities $\boldsymbol{X}$
Output parameter	
y	Value obtained by evaluating the formula $$y = f(\boldsymbol{x})$$

Table 5: Evaluation of a univariate, explicit, real model.

Input parameters	
N	Number of input quantities
m	Number of output quantities
$\boldsymbol{f}$	Function with m components 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 input quantities $\boldsymbol{X}$
Output parameter	
$\boldsymbol{y}$	Column vector of values $(y_1, \ldots, y_m)^{\mathrm{T}}$ obtained by evaluating the formula $$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$$

Table 6: Evaluation of a multivariate, explicit, real model.

	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 input quantities $\boldsymbol{X}$
$\boldsymbol{t}$	Computational control parameters such as an initial approximation to y and technical parameters relating to the equation-solving software
	Output parameter
y	Value obtained by solving the equation $h(y, \boldsymbol{x}) = 0$
	Numerical analysis
	A zero-finding algorithm [13, 20], such as the bisection algorithm in cases where suitable lower and upper bounds are known for y, can be used to solve the equation

Table 7: Evaluation of a univariate, implicit, real model.

	Input parameters
N	Number of input quantities
m	Number of output quantities
$\boldsymbol{h}$	Function with m components 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 input quantities $\boldsymbol{X}$
$\boldsymbol{t}$	Computational control parameters such as an initial approximation to $\boldsymbol{y}$ and technical parameters relating to the equation-solving software
	Output parameter
$\boldsymbol{y}$	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 [20], starting from a suitable approximation to $\boldsymbol{y}$, can be used to solve the system of equations

Table 8: Evaluation of a multivariate, implicit, real model.

4 GUM uncertainty framework

4.1 Procedure

For the application of the GUM uncertainty framework, the outputs of the formulation stage are (Section 2):

- Estimates $\boldsymbol{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$ 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, covariances associated with estimates of the 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 an *uncertainty* matrix that holds the standard uncertainties associated with the estimates $\boldsymbol{x}$ and the covariances associated with these estimates. $\boldsymbol{V_x}$ is a matrix of order N, whose (i, j)th element contains the covariance $u(x_i, x_j)$ associated with x_i and x_j, with $u(x_i, x_i) = u_i^2$.

$\boldsymbol{x}$, $\boldsymbol{\nu}$ 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 stage of the GUM uncertainty framework.

For *univariate* measurement models, the procedure is as follows:

1. Calculate the estimate of the output quantity by evaluating the model at the estimates 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 calculate the sensitivity coefficients by evaluating these partial derivatives at the estimates of the input quantities.[6]

   See Section 4.2.

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

   See Section 4.3.

[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.

4. For the case that the input quantities $\boldsymbol{X}$ are mutually independent, use the Welch-Satterthwaite formula to calculate ν_{eff}, the effective degrees of freedom associated with the standard uncertainty, from the standard uncertainties associated with the estimates of the input quantities, the corresponding degrees of freedom, the sensitivity coefficients and the standard uncertainty associated with the estimate of the output quantity. The GUM uncertainty framework does not state how ν_{eff} is to be calculated for the case that the input quantities are correlated.

   See Section 4.4.

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

   See Section 4.5.

The computational flow of the calculation stage for the GUM uncertainty framework, indicating the inputs and the outputs, an estimate of the output quantity, the standard uncertainty associated with this estimate and a coverage interval, is given in Figure 1. This figure applies in the case of a univariate, explicit, real model with no covariance effects. Other model types would give diagrams that constitute a variant of Figure 1.

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 measurement model, these units are specified in Tables 5, 9, 13, 14 and 15.

4.2 Sensitivity coefficients

The procedure for the GUM uncertainty framework (Section 4.1) covers the univariate, explicit, real model of measurement $Y = f(\boldsymbol{X})$. The *sensitivity coefficients* used by that procedure are denoted here by the (row) vector[7] $\boldsymbol{C} = (c_1, \dots, c_N)$, 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

[7] 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.

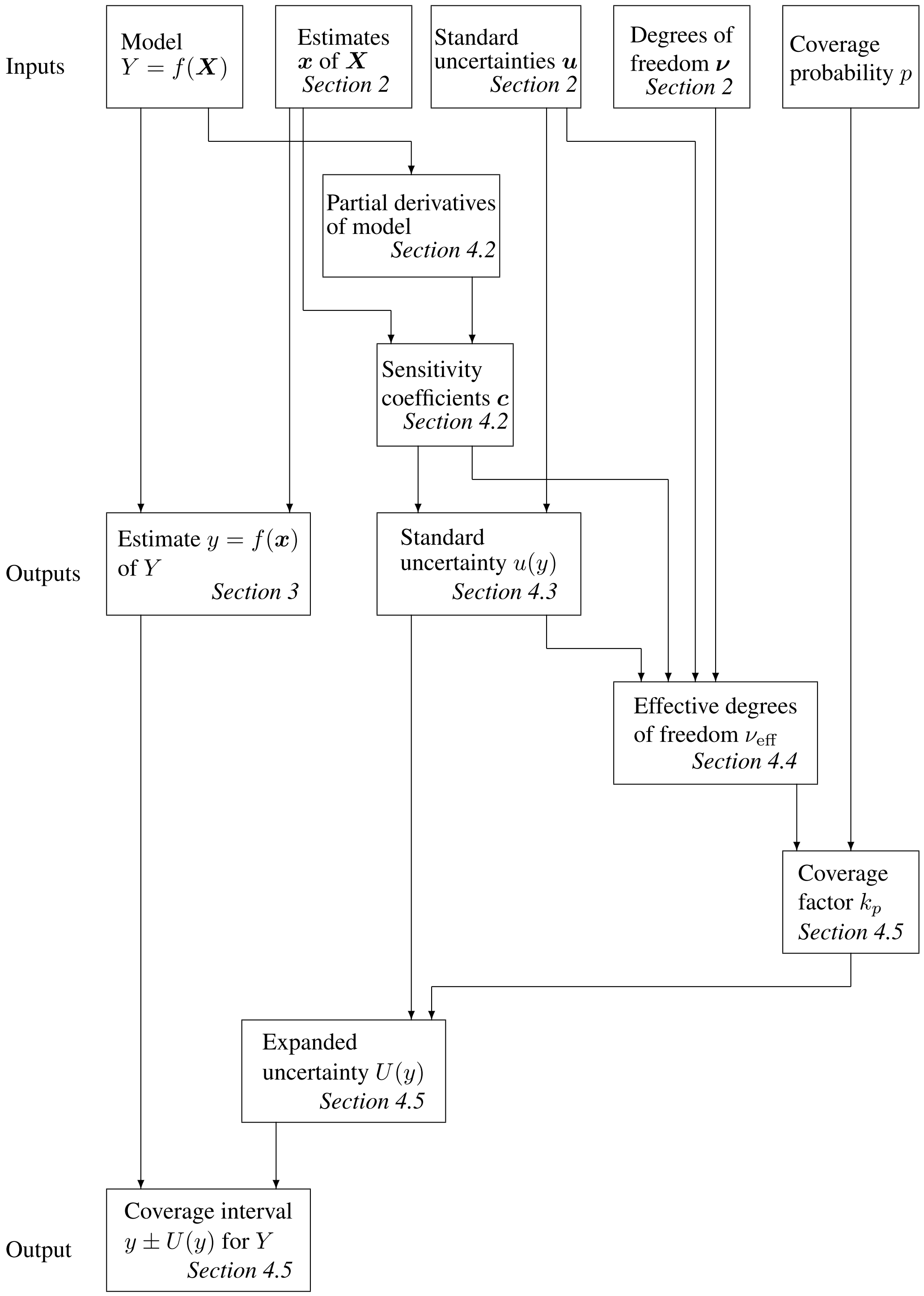
Inputs
Model $Y = f(\boldsymbol{X})$
Estimates $\boldsymbol{x}$ of $\boldsymbol{X}$ *Section 2*
Standard uncertainties $\boldsymbol{u}$ *Section 2*
Degrees of freedom $\boldsymbol{\nu}$ *Section 2*
Coverage probability p
Partial derivatives of model *Section 4.2*
Sensitivity coefficients $\boldsymbol{c}$ *Section 4.2*
Outputs
Estimate $y = f(\boldsymbol{x})$ of Y *Section 3*
Standard uncertainty $u(y)$ *Section 4.3*
Effective degrees of freedom ν_{eff} *Section 4.4*
Coverage factor k_p *Section 4.5*
Expanded uncertainty $U(y)$ *Section 4.5*
Output
Coverage interval $y \pm U(y)$ for Y *Section 4.5*

Figure 1: Uncertainty evaluation using the GUM uncertainty framework for a univariate, explicit, real model with mutually independent input quantities.

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 input quantities $\boldsymbol{X}$
Output parameter	
$\boldsymbol{C}$	$1 \times N$ 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 model.

Table 10 specifies the counterpart for a *multivariate*, explicit, real 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 $\boldsymbol{X} = \boldsymbol{x}$.

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 input quantities $\boldsymbol{X}$
Output parameter	
$\boldsymbol{C}$	$m \times N$ 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 model.

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

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

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* 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).

	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 input quantities $\boldsymbol{X}$
y	Estimate of the output quantity Y that satisfies $h(y, \boldsymbol{x}) = 0$
	Output parameter
$\boldsymbol{C}$	$1 \times N$ vector of sensitivity coefficients, whose jth element is $-\left(\dfrac{\partial h}{\partial Y}\right)^{-1}\left(\dfrac{\partial h}{\partial X_j}\right),$ where the partial derivatives are evaluated at $\boldsymbol{X} = \boldsymbol{x}$ and $Y = y$

Table 11: Sensitivity coefficients for a univariate, implicit, real model.

The sensitivity coefficients can be formed [6]

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 setting $\boldsymbol{X} = \boldsymbol{x}$
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 $\boldsymbol{C}$ in Tables 9–12 are obtained requires careful consideration. These derivatives and hence the

	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 input quantities $\boldsymbol{X}$
$\boldsymbol{y}$	Column vector $(y_1, \ldots, y_m)^{\mathrm{T}}$ of estimates of the output quantities $\boldsymbol{Y}$ that satisfy $\boldsymbol{h}(\boldsymbol{y}, \boldsymbol{x}) = \boldsymbol{0}$
	Output parameter
$\boldsymbol{C}$	$m \times N$ matrix of sensitivity coefficients that solves $\boldsymbol{H}_{\boldsymbol{y}}\boldsymbol{C} = -\boldsymbol{H}_{\boldsymbol{x}},$ where the matrices $\boldsymbol{H}_{\boldsymbol{x}}$ and $\boldsymbol{H}_{\boldsymbol{y}}$ 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 [21]

Table 12: Sensitivity coefficients for a multivariate, implicit, real model.

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 Uncertainty associated with the estimate of the output quantity

For a univariate, explicit, real measurement model, the standard uncertainty $u(y)$ associated with the estimate y of Y is obtained from the formula [3]

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

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 uncertainty matrix of order N associated with the estimates $\boldsymbol{x}$ 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 models, $\boldsymbol{V_y}$ is the variance associated with the estimate y of the output quantity Y; for other measurement models, it is the uncertainty matrix associated with the estimates of the output quantities. Table 13 specifies the evaluation of the uncertainty associated with the estimate of the output quantity, and indicates the input and output parameters necessary for its determination.

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 output quantity Y when the input quantities $\boldsymbol{X}$ are mutually independent. The GUM uncertainty framework does not state how ν_{eff} is to be calculated when the input quantities are correlated.

Input parameters	
N	Number of input quantities
m	Number of output quantities
$\boldsymbol{V_x}$	Uncertainty matrix of order N associated with the estimates $\boldsymbol{x}$ of the input quantities $\boldsymbol{X}$
$\boldsymbol{C}$	$m \times N$ matrix of sensitivity coefficients
Output parameter	
$\boldsymbol{V_y}$	Uncertainty matrix of order m associated with the estimate y (or $\boldsymbol{y}$) 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: Standard uncertainty associated with the estimate y of Y for a univariate ($m = 1$) model or the uncertainty matrix associated with the estimate $\boldsymbol{y}$ of $\boldsymbol{Y}$ for a multivariate (general m) model.

4.5 Expanded uncertainty

Table 15 specifies the calculation of expanded uncertainty $U(y)$ associated with the estimate y of a univariate, real output quantity Y when the input quantities $\boldsymbol{X}$ are mutually independent. The expanded uncertainty is evaluated as the product of the 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 that characterizes the output quantity Y. In the GUM uncertainty framework, 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 $\nu_{\text{eff}} \geq 473$, $k_p = 1.96$, correct to two decimal places, the corresponding value for the standard Gaussian distribution $\mathrm{N}(0, 1)$.

	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 input quantities $\boldsymbol{X}$
$\boldsymbol{C}$	$1 \times N$ (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 the estimate x_i of X_i is taken as the mean of a set of q repeated indications, ν_i is taken as $q-1$. If a rectangular distribution with accurately known endpoints is assigned to X_i, ν_i is taken as infinite (∞). (Since ∞ cannot be represented as such as an input parameter, 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 [3] for assigning the degrees of freedom to the standard uncertainties associated with estimates of the input quantities. Each case has to be treated on its own merits
$u(y)$	Standard uncertainty associated with the estimate y of the output quantity
	Output parameter
ν_{eff}	Effective degrees of freedom determined from the Welch-Satterthwaite formula $$\frac{u^4(y)}{\nu_{\text{eff}}} = \sum_{i=1}^{N} \frac{c_i^4 u^4(x_i)}{\nu_i}$$

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

	Input parameters
$u(y)$	Standard uncertainty associated with the estimate y of the output quantity
ν_{eff}	Effective degrees of freedom determined from the Welch-Satterthwaite formula
p	Coverage probability (typically 0.95)
	Output parameter
$U(y)$	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 for a univariate, real model.

5 Monte Carlo method

5.1 Procedure

For the application of a Monte Carlo method, the outputs of the formulation stage are (Section 2) the PDFs[8] $\boldsymbol{g}(\boldsymbol{\xi}) = (g_1(\xi_1), \ldots, g_n(\xi_N))^{\mathrm{T}}$ for the input quantities $\boldsymbol{X} = (X_1, \ldots, X_N)^{\mathrm{T}}$.[9] The PDFs, together with the measurement model and the required coverage probability p (e.g., 0.95), constitute the inputs to the calculation stage of the Monte Carlo procedure.

For *univariate* measurement models, the procedure is as follows:

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

   See Section 5.2.

2. Generate M vectors by sampling from the PDFs for the (set of N) input quantities.

   See Section 5.3.

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

   See Section 5.4.

4. Calculate the estimate of the output quantity and the associated standard uncertainty as the (arithmetic) mean and standard deviation of the model values. Optionally, use the model values to form an approximation to the PDF for the output quantity.

   See Section 5.5.

5. Sort the model values into non-decreasing order, and use the sorted values to provide a discrete representation of the distribution function for the output quantity. Optionally, use the discrete representation to form a (continuous) approximation to the distribution function for the output quantity.

   See Section 5.6.

6. Use the discrete representation of the distribution function to calculate a 95 % coverage interval for the output quantity.

   See Section 5.7

Figure 2 shows the procedure diagrammatically.

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

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

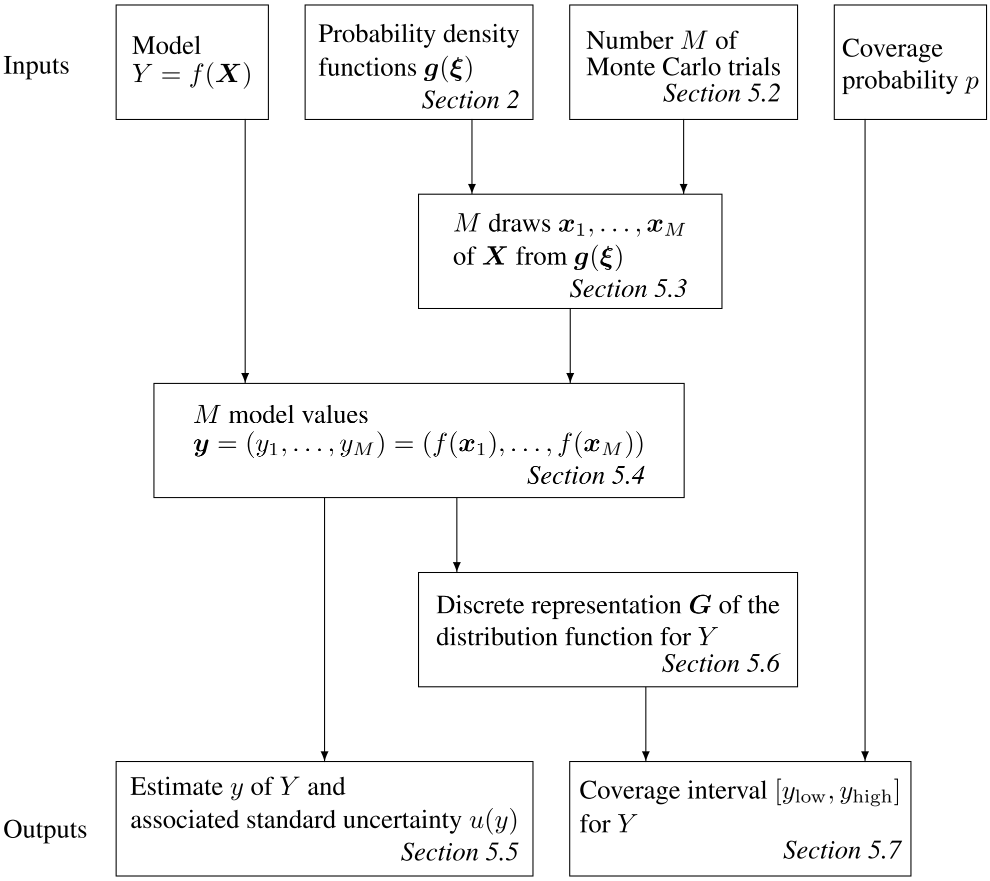
Inputs
Model $Y = f(\boldsymbol{X})$
Probability density functions $\boldsymbol{g}(\boldsymbol{\xi})$ *Section 2*
Number M of Monte Carlo trials *Section 5.2*
Coverage probability p
M draws $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$ of $\boldsymbol{X}$ from $\boldsymbol{g}(\boldsymbol{\xi})$ *Section 5.3*
M model values $\boldsymbol{y} = (y_1, \ldots, y_M) = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_M))$ *Section 5.4*
Discrete representation $\boldsymbol{G}$ of the distribution function for Y *Section 5.6*
Outputs
Estimate y of Y and associated standard uncertainty $u(y)$ *Section 5.5*
Coverage interval $[y_{\rm low}, y_{\rm high}]$ for Y *Section 5.7*

Figure 2: Uncertainty evaluation using a Monte Carlo method for a univariate, explicit, real model.

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

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 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 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 decimal digits, for the output quantity.

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 [1, 2, 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 vectors $\boldsymbol{x}_i$, $i = 1, \ldots, M$, are drawn from the PDFs for the input quantities $\boldsymbol{X}$. Draws are made 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 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 numbers generated from the rectangular distribution 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. [30]). 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 parameter	
q	Number of rectangular pseudo-random numbers to be generated
Input-output parameter	
$\boldsymbol{t}$	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 realized 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 parameter	
$\boldsymbol{r}$	q draws from a rectangular distribution between zero and one

Table 16: Pseudo-random number generation.

A draw from the rectangular distribution on the interval $(a,\ b)$ can be formed from $a + (b - a)r$, where r is a draw from the rectangular distribution on the interval $(0,\ 1)$.

Randomness tests A review [33] has been carried out on the use of random numbers in solving problems using a Monte Carlo procedure. 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[10] and passing a set of statistical tests for randomness.[11]

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 of *shift register generators*, both of which are widely discussed in the literature [19, 28, 34, 36].

The KISS[12] generator [32] is a combination of a congruential generator and two shift register generators. A version in the C programming language is available [35, p42] and in Fortran [32].

The test suite TestU01 [29] may be used to carry out an extensive test of the statistical properties of any generator submitted to it. The suite is very detailed, with many individual tests, including the so-called "Big Crush". Several generators that pass the suite of tests are listed by Wichmann and Hill [43]. An enhanced Wichmann-Hill generator also passes the test and is recommended [2].

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

5.3.2 Gaussian distribution

The procedure in Table 18 provides a straightforwardly implementable approach [7] to generate draws from the standard Gaussian distribution $\mathrm{N}(0,\ 1)$ using the Box-Muller transform.

A draw from the Gaussian distribution $\mathrm{N}(\mu, \sigma^2)$ can be formed from $\mu + \sigma z$, where z is a draw from the standard Gaussian distribution $\mathrm{N}(0, 1)$.

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

[11] 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* [31], that include the overlapping M-tuple test, the overlapping permutation test, the parking lot and lattice test and the birthday-spacing test.

[12] Keep It Simple, Stupid!

Input parameter	
	None
Input-output parameters	
ix, iy, iz, it	Integer parameters required as input parameters and that are changed by the procedure. Set to integers between 1 and 2 147 483 647 before the first call. Do not disturb between calls. Subsequent values of these parameters are not usually of concern to the user. The parameters provide the basis by which the pseudo-random numbers are generated. They may be realized as global variables and thus not appear explicitly as parameters of the procedure
Output parameter	
r	A draw from a rectangular distribution between zero and one
Computation	

1. Form $ix = 11\,600 \times (ix \bmod 185\,127) - 10\,379 \times \lfloor ix \div 185\,127 \rfloor$
2. Form $iy = 47\,003 \times (iy \bmod 45\,688) - 10\,479 \times \lfloor iy \div 45\,688 \rfloor$
3. Form $iz = 23\,000 \times (iz \bmod 93\,368) - 19\,423 \times \lfloor iz \div 93\,368 \rfloor$
4. Form $it = 33\,000 \times (it \bmod 65\,075) - 8\,123 \times \lfloor it \div 65\,075 \rfloor$
5. If $ix < 0$, then form $ix = ix + 2\,147\,483\,579$
6. If $iy < 0$, then form $iy = iy + 2\,147\,483\,543$
7. If $iz < 0$, then form $iz = iz + 2\,147\,483\,423$
8. If $it < 0$, then form $it = it + 2\,147\,483\,123$
9. Form $w = wx + wy + wz + wt$, where

$$\begin{aligned} wx &= ix/2\,147\,483\,579.0, \\ wy &= iy/2\,147\,483\,543.0, \\ wz &= iz/2\,147\,483\,423.0, \\ wt &= it/2\,147\,483\,123.0. \end{aligned}$$

10. Form $r = w - \lfloor w \rfloor$

Table 17: Enhanced Wichmann-Hill pseudo-random number generator for the rectangular distribution. $ix \bmod n$ denotes the modulus (or remainder) after division by n, and $\lfloor w \rfloor$ denotes the integer part of w.

Input parameter	
	None
Output parameters	
z_1, z_2	Two draws from a Gaussian distribution with zero expectation and unit standard deviation
Computation	
	1. Generate draws r_1 and r_2 independently from the rectangular distribution between zero and one 2. Form $z_1 = \sqrt{-2 \log r_1} \cos 2\pi r_2$ and $z_2 = \sqrt{-2 \log r_1} \sin 2\pi r_2$

Table 18: Box-Muller pseudo-random number generator for the Gaussian distribution.

5.3.3 t–distribution

The procedure in Table 19 provides an approach [27], [35, p63] to generate draws from the t–distribution t_ν with ν degrees of freedom, that is also straightforward to implement.

Input parameter	
ν	Degrees of freedom
Output parameter	
t	A draw from a t–distribution with ν degrees of freedom
Computation	
	1. Generate draws r_1 and r_2 independently from the rectangular distribution between zero and one 2. If $r_1 < 1/2$, form $t = 1/(4r_1 - 1)$ and $v = r_2/t^2$; otherwise set $t = 4r_1 - 3$ and $v = r_2$ 3. If $v < 1 - \lvert t \rvert/2$ or $v < (1 + t^2/\nu)^{-(\nu+1)/2}$, accept t as a draw from the t–distribution; otherwise repeat from Step 1

Table 19: Pseudo-random number generator for the t–distribution.

A draw from the t–distribution $\mathrm{t}_\nu(\mu, \sigma^2)$ with shift parameter μ and scale parameter σ can be formed from $\mu + \sigma t$, where t is a draw from the t–distribution t_ν.

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 draw z from this distribution is obtained as follows:

1. Generate a draw ψ from a rectangular distribution between zero and one (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, 20].

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. Draws can be made from this multivariate (or joint) Gaussian distribution [37, 40] using the procedure in Table 20.

Figure 3 shows three examples of 1000 points generated from bivariate Gaussian distributions using the MULTNORM generator [37]. In all three cases the distributions characterize a quantity with expectation $\boldsymbol{\mu} = (2, 3)^{\mathrm{T}}$. In the top graph, the quantity has covariance matrix

$$\boldsymbol{V} = \left[\begin{array}{cc} 2.0 & 0.0 \\ 0.0 & 2.0 \end{array} \right],$$

i.e., the components of the quantity are independent and have the same standard deviation, and the cloud of points resembles a disk with centre at the expectation $\boldsymbol{\mu}$. In the middle graph, the quantity has covariance matrix

$$\boldsymbol{V} = \left[\begin{array}{cc} 1.0 & 0.0 \\ 0.0 & 4.0 \end{array} \right],$$

i.e., the components are independent and have different standard deviations, and the cloud of points resembles an ellipse with major and minor axes parallel to the co-ordinate axes.

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 draws from the multivariate Gaussian distribution

Output parameter	
$\boldsymbol{\Phi}$	$n \times q$ 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. Generate q draws from the n-dimensional standard Gaussian distribution $\mathrm{N}(0,1) \times \cdots \times \mathrm{N}(0,1)$. Doing so simply means generating an $n \times q$ array $\boldsymbol{Z}$ of draws from the standard Gaussian distribution

3. Provide the required draw (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: Pseudo-random number generator for the multivariate Gaussian distribution.

In the bottom graph, the quantity has covariance matrix

$$\boldsymbol{V} = \begin{bmatrix} 2.0 & 1.9 \\ 1.9 & 2.0 \end{bmatrix},$$

i.e., the components are correlated, and the cloud of points resembles an ellipse with axes oriented to the co-ordinate axes by an angle that is determined by the covariance of the components.

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 Numerical evaluation of the model

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

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

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

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

The model values $y_1, \ldots, y_M$ are used in the evaluation of the estimate y of Y and the associated standard uncertainty $u(y)$ (Section 5.5), and as the basis for calculating an approximation to the PDF for Y (Section 5.5.1). The model values are also used to provide a discrete representation of the distribution function for Y (Section 5.6) in terms of which are obtained a (continuous) approximation to the distribution function (Section 5.6.1) and a coverage interval for Y (Section 5.7). The case of a univariate, explicit, real measurement model is covered below, although the presentation would apply similarly for the corresponding implicit model. Multivariate models are considered in Section 5.10.

An "updating" procedure for forming y, $u(y)$ and approximations to the PDF and distribution function for the output quantity, which avoids the need to store and sort the complete set of model values, is described in Appendix E.

The information in Section 3 provides advice on model evaluation. Note that in the Monte Carlo procedure the model is evaluated for each draw of the input quantities and hence for values (that may be distanced by "several standard deviations" from the estimates of the input quantities). This is in contrast to the GUM uncertainty framework in which the

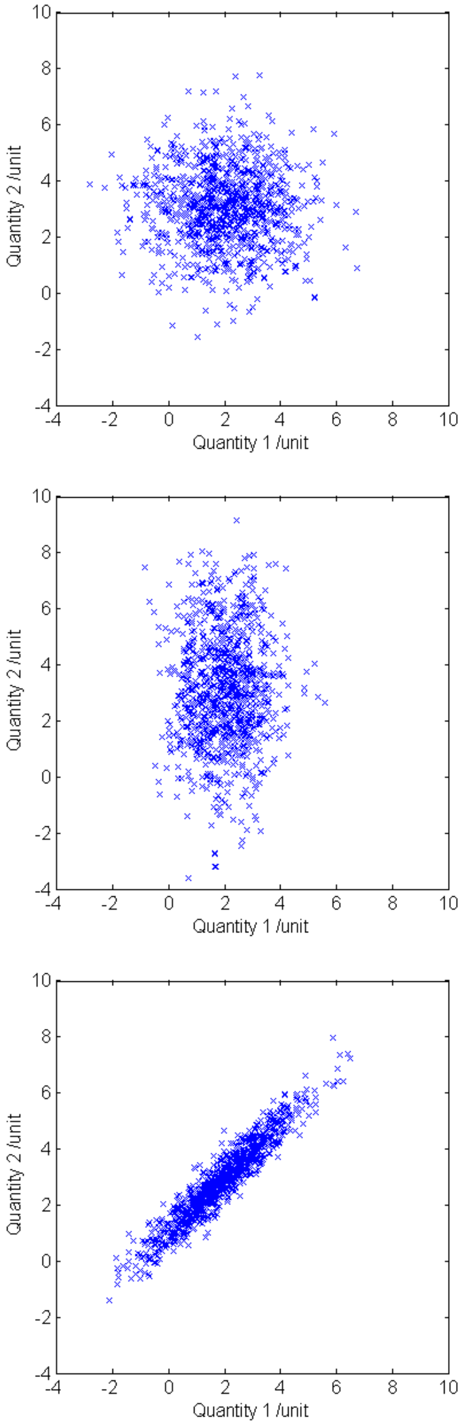

Figure 3: Points drawn from bivariate Gaussian distributions used to characterize quantities that are independent with the same standard deviation (top), independent having different standard deviations (middle), and correlated (bottom).

measurement model is evaluated only at the estimates of the input quantities and, if finite difference approximations are used [3, Clause 5.1.3], also at points perturbed from these estimates by ± one standard deviation for each quantity 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 input quantities.

5.5 Estimate of the output quantity and the associated standard uncertainty

The average $\widehat{y}$ of the values $y_r,\ r = 1, \ldots, M$, of the output quantity is taken as the estimate y of the output quantity, and the standard deviation $u(\widehat{y})$ of the values is taken as the standard uncertainty $u(y)$ associated with y. $\widehat{y}$ is evaluated from

$$\widehat{y} = \frac{1}{M}\sum_{r=1}^{M} y_r, \tag{2}$$

and the standard deviation $u(\widehat{y})$ from

$$u^2(\widehat{y}) = \frac{1}{M-1}\sum_{r=1}^{M}(y_r - \widehat{y})^2. \tag{3}$$

Table 21 specifies the evaluation of the estimate y of Y and the associated standard uncertainty $u(y)$ using Formulae (2) and (3), respectively, and indicates the input and output parameters necessary for their determination.

The evaluations of y and $u(y)$ (Table 21) require the summation of M numbers with M large (typically of the order of 10^5 or 10^6: Section 5.2). A procedure for undertaking these summations, designed to reduce the effect of rounding errors associated with the floating-point operations (of which there are many), is described in Appendix D. An “updating” procedure for evaluating the estimate of the output quantity and the associated standard uncertainty that avoids the need to store the complete set of model values is described in Appendix E.

The value of y so obtained yields the smallest mean squared deviation over all possible choices of the estimate of the output quantity. However, the value will not in general agree with the model evaluated at the estimates of the input quantities [3, 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 input quantities, unless the model is linear [3, Clause 4.1.4].

	Input parameters
M	Number of draws (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 draws $\boldsymbol{x}_r$ from each of the PDFs for the N input quantities, i.e., $y_r = f(\boldsymbol{x}_r)$
	Output parameters
y	Estimate of the output quantity: the arithmetic mean of the model values, defined by $y = \frac{1}{M}\sum_{r=1}^{M} y_r$
$u(y)$	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 [8]

Table 21: Estimate of the output quantity and the associated standard uncertainty.

5.5.1 Approximation to the probability density function

The values $y_r,\ r = 1, \ldots, M$, when assembled into a histogram (with suitable bin 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 output quantity. Calculations are not generally carried out in terms of this histogram, the resolution of which depends on the choice of bin size. The histogram can, however, be useful as an aid to understanding the nature of the PDF, e.g., the extent of its asymmetry.

An approximation $\widehat{g}(\eta)$ to the PDF $g(\eta)$ for the output quantity is obtained as follows. Let $[\eta_0, \eta_b]$ be an interval of values of the output quantity partitioned into b subintervals $\{B_k : k = 1, \ldots, b\}$, where

$$B_k = \begin{cases} [\eta_{k-1}, \eta_k), & k = 1, \ldots, b-1, \\ [\eta_{k-1}, \eta_k], & k = b, \end{cases}$$

and

$$\eta_0 \le \min\{y_r : r = 1, \ldots, M\}, \qquad \max\{y_r : r = 1, \ldots, M\} \le \eta_b,$$

i.e., each of the values y_r of the output quantity lies in exactly one of the intervals B_k.[13] Define[14]

$$h_k = \frac{\mathrm{card}(\{y_r \in B_k : r = 1, \ldots, M\})}{\eta_k - \eta_{k-1}}, \qquad k = 1, \ldots, b,$$

and

$$g_k = \frac{h_k}{M}, \qquad k = 1, \ldots, b.$$

h_k is the height of the kth bar corresponding to the bin B_k in a histogram of the values y_r. The height is chosen so that the area of the bar is proportional to the number of values of the output quantity contained in the bin. g_k is a probability density obtained by scaling the heights of the bars so that the total area of the bars is unity. The scaled histogram defined by the bins B_k and probability densities $g_k,\ k = 1, \ldots, b$, defines an approximation to the PDF for the output quantity. Table 22 specifies the evaluation of an approximation to the PDF for the output quantity, and indicates the input and output parameters necessary for its determination.

The approximation $\widehat{g}(\eta)$ may alternatively be defined as the piecewise-linear function joining the b points $(\widehat{\eta}_k, g_k),\ k = 1, \ldots, b$, where $\widehat{\eta}_k$ is the midpoint of the interval defining the kth bin, i.e.,

$$\widehat{\eta}_k = \frac{\eta_{k-1} + \eta_k}{2}, \qquad k = 1, \ldots, b,$$

and g_k is the probability density defined above.

[13] Often the bins will be chosen to have equal width, $\delta\eta$ say, where $\delta\eta = \eta_k - \eta_{k-1},\ k = 1, \ldots, b$.

[14] card(A) is used to denote the *cardinality* of the set A, i.e., the number of elements in the set.

Input parameters	
M	Number of draws (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 draws $\boldsymbol{x}_r$ from each of the PDFs for the N input quantities, i.e., $y_r = f(\boldsymbol{x}_r)$
b	Number of bins in approximation to PDF for the output quantity
$\boldsymbol{\eta}$	Values $(\eta_0, \ldots, \eta_b)$, with $\eta_0 \leq \min\{y_r : r = 1, \ldots, M\}$ and $\max\{y_r : r = 1, \ldots, M\} \leq \eta_b$, that define bins $B_k,\ k = 1, \ldots, b$, where $B_k = \begin{cases} [\eta_{k-1}, \eta_k), & k = 1, \ldots, b-1, \\ [\eta_{k-1}, \eta_k], & k = b. \end{cases}$
Output parameter	
$\boldsymbol{g}$	Probability densities $(g_1, \ldots, g_b)$, defined by $g_k = \dfrac{h_k}{M}, \qquad k = 1, \ldots, b,$ where $h_k = \dfrac{\mathrm{card}(\{y_r \in B_k : r = 1, \ldots, M\})}{\eta_k - \eta_{k-1}}.$ The scaled histogram defined by bins B_k and probability densities g_k defines an approximation to the PDF for the output quantity

Table 22: Approximation to the probability density function for the output quantity.

5.6 Discrete representation of the distribution function

A discrete representation $\boldsymbol{G}$ of the distribution function for the output quantity is obtained by sorting the values y_r, $r = 1, \ldots, M$, provided in Section 5.4 into non-decreasing order. Denoting the sorted values by $y_{(r)}$, $r = 1, \ldots, M$, the discrete representation is given by $\boldsymbol{G} = (y_{(1)}, \ldots, y_{(M)})$. Table 23 specifies the evaluation of a discrete representation of the distribution function for the output quantity, and indicates the input and output parameters necessary for its determination.

The discrete representation is used as the basis for calculating a coverage interval for the output quantity (Section 5.7). It is also used as the basis for obtaining a (continuous) approximation to the distribution function for the output quantity (Section 5.6.1) that may used, for example, to obtain random draws from the distribution for the output quantity (in the manner described in Section 5.3.4).

	Input parameters
M	Number of draws (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 draws $\boldsymbol{x}_r$ from each of the PDFs for the N input quantities, i.e., $y_r = f(\boldsymbol{x}_r)$
	Output parameter
$\boldsymbol{G}$	Discrete representation of the distribution function for the output quantity, where $\boldsymbol{G} = (y_{(1)}, \ldots, y_{(M)})$, the model values sorted into non-decreasing order
	Numerical analysis
	It is recommended that a sorting algorithm that takes a number of operations proportional to $M \log M$ be used (e.g., [38]). A naive algorithm would take a time proportional to M^2 and may make the computation time unacceptable

Table 23: Discrete representation of the distribution function for the output quantity.

5.6.1 Approximation to the distribution function

An approximation $\widehat{G}(\eta)$ to the distribution function $G(\eta)$ for the output quantity is obtained as follows. Assign uniformly spaced cumulative probabilities $p_r = (r - 1/2)/M$, $r = 1, \ldots, M$, to the ordered values $y_{(r)}$ in the discrete representation $\boldsymbol{G}$ of the distribution function for the output quantity.[15] Form $\widehat{G}(\eta)$ as the piecewise-linear function joining

[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.

the M points $(y_{(r)}, p_r),\ r = 1, \ldots, M$:

$$\widehat{G}(\eta) = \frac{r - 1/2}{M} + \frac{\eta - y_{(r)}}{M(y_{(r+1)} - y_{(r)})}, \qquad y_{(r)} \le \eta \le y_{(r+1)}, \tag{4}$$

for $r = 1, \ldots, M - 1$. Table 24 specifies the evaluation of an approximation to the distribution function for the output quantity, and indicates the input and output parameters necessary for its determination.

Input parameters	
M	Number of draws (equal to the number of Monte Carlo trials)
$\boldsymbol{G}$	Discrete representation of the distribution function for the output quantity, where $\boldsymbol{G} = (y_{(1)}, \ldots, y_{(M)})$, the model values sorted into non-decreasing order
Output parameter	
$\boldsymbol{p}$	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 approximation to the distribution function for the output quantity. ($\widehat{G}(\eta)$ is defined only for values of η corresponding to values of probability p in the interval $M/2 \le p \le 1 - M/2$. Indeed, it should not be used near the endpoints of this interval, because it is less reliable there.)

Table 24: Approximation to the distribution function for the output quantity.

Formulae (2) and (3) for the estimate of the output quantity and the associated standard uncertainty do not in general provide values that are identical to the expectation and standard deviation of the quantity characterized by the distribution function $\widehat{G}(\eta)$. 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. However, for a sufficiently large value of M, the values obtained using Expressions (2) and (3) are generally indistinguishable for practical purposes from those given by Expressions (5) and (6).

5.7 Coverage interval

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 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 the output quantity 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 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 output quantity 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 $\alpha = (1-p)/2$, corresponding to the probabilistically symmetric interval.

The endpoints of a coverage interval can be obtained from the discrete representation of the distribution function for the output quantity (Section 5.6) as follows. Let $q = pM$, if pM is an integer, or the integer part of $pM + 1/2$, otherwise. Then, $[y_{\text{low}}, y_{\text{high}}] = [y_{(r)}, y_{(r+q)}]$ for any $r = 1, \ldots, M-q$, is a $100p$ % coverage interval. The probabilistically symmetric $100p$ % coverage interval is given by $r = (M-q)/2$ if $(M-q)/2$ is an integer, or the integer part of $(M-q+1)/2$, otherwise. The shortest $100p$ % coverage interval is given by determining $r = r^*$ such that, for $r = 1, \ldots, M-q$, $y_{(r^*+q)} - y_{(r^*)} \le y_{(r+q)} - y_{(r)}$. Table 25 specifies the determination of the shortest coverage interval from a discrete representation of the distribution function for the output quantity, and indicates the input and output parameters necessary for their determination.

The endpoints of a coverage interval can also be obtained from the approximation $\widehat{G}(\eta)$ to $G(\eta)$ obtained in Section 5.6.1 or Appendix E. For a sufficiently large value of M, the coverage interval obtained using the discrete representation $\boldsymbol{G}$ of $G(\eta)$ can be expected to be indistinguishable for practical purposes from those obtained using the approximation $\widehat{G}(\eta)$. To find the left-hand endpoint y_{low} such that $\alpha = \widehat{G}(y_{\text{low}})$, 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,

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

Input parameters	
M	Number of draws (equal to the number of Monte Carlo trials)
$\boldsymbol{G}$	Discrete representation of the distribution function for the output quantity, where $\boldsymbol{G} = (y_{(1)}, \ldots, y_{(M)})$, the model values sorted into non-decreasing order
p	Coverage probability (e.g., 0.95)
Output parameters	
$y_{\text{low}}, y_{\text{high}}$	Endpoints of the shortest $100p$ % coverage interval for the output quantity, given by $[y_{(r^*)}, y_{(r^*+q)}]$, where $q = pM$, if pM is an integer, or the integer part of $pM + 1/2$, otherwise, and r^*, an integer between 1 and $M - q$, is chosen so that for $r = 1, \ldots, M - q$, $y_{(r^*+q)} - y_{(r^*)} \leq y_{(r+q)} - y_{(r)}$

Table 25: Shortest coverage interval obtained from a discrete representation of the distribution function for the output quantity.

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

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

where the index s is identified to satisfy

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

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 sufficient 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

In the application of a Monte Carlo method there is no immediate counterpart of a sensitivity coefficient (Section 4.2) since the Monte Carlo procedure operates in terms of the actual non-linear model rather than a linearized counterpart. Recall that with a linear model the sensitivity coefficients “reproduce” linear effects, and for a non-linear model the sensitivity coefficients provide first-order information. Therefore, those practitioners accustomed to the GUM uncertainty framework may find the absence of sensitivity coefficients disconcerting.

It is possible and very straightforward, however, to adapt the Monte Carlo procedure such that it provides information that in a sense constitutes a non-linear counterpart of a sensitivity coefficient. Consider holding 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 randomly from the PDF for this input quantity and determine a discrete representation for the distribution function of the output quantity with respect to X_k. The standard deviation $\widehat{u}_k(y)$ evaluated in terms of this discrete representation is taken as an approximation to the component of the standard uncertainty $u(y)$ corresponding to X_k.

The use of "non-linear" sensitivity coefficients in place of "linear" sensitivity coefficients permits individual non-linear effects to be taken into account. A "non-linear" 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 non-linearity with respect to X_k is negligible. When $\widehat{c}_k$ is appreciably different from c_k the non-linearity 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 non-linearity 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 output quantity. There will be further contributions arising from any interaction (i.e., non-additive) terms in the model.

Table 26 specifies the calculation of a non-linear sensitivity coeficient relating to the input quantity X_k using the Monte Carlo procedure.

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 the quantity 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 $M_{\text{adap}} = 10^4$, trials.[16] For each Monte Carlo calculation in the sequence, y, $u(y)$ and the endpoints of a 95 % coverage interval are

[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 draws (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 draws $\boldsymbol{x}_r^k$ obtained by holding all input quantities but one, X_k, at their nominal values and sampling from the PDF $g_k(\xi_k)$ for X_k
$u(x_k)$	Standard uncertainty associated with the estimate x_k of X_k
Output parameters	
$\widehat{u}_k(y)$	Component of the standard uncertainty $u(y)$: 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 $y^k = \frac{1}{M}\sum_{r=1}^{M} y_r^k.$
$\widehat{c}_k$	Non-linear sensitivity coefficient, defined by $\widehat{c}_k = \frac{\widehat{u}(y_k)}{u(x_k)}.$

Table 26: Sensitivity analysis using a Monte Carlo procedure.

formed from the results obtained, as in Sections 5.5 and 5.7. Denote by $y^{(h)}$, $u(y^{(h)})$, $y_{\text{low}}^{(h)}$ and $y_{\text{high}}^{(h)}$ 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 of the values $y^{(1)}, \ldots, y^{(h)}$ and the standard deviation s_y associated with this arithmetic mean are formed. The counterparts of these statistics for y are determined for $u(y)$, y_{low} and y_{high}. If the largest of $2s_y$, $2s_{u(y)}$, $2s_{y_{\text{low}}}$ and $2s_{y_{\text{high}}}$ 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 output quantity, the associated standard uncertainty and the coverage interval for the output quantity.

5.10 Multivariate Monte Carlo procedure

The treatment described in Sections 5.4–5.9 applies to *univariate* real models. It can straightforwardly be adapted to apply to multivariate or complex models. There is not, however, a ready counterpart of a coverage interval.

The multivariate model is

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

M draws $\boldsymbol{x}_r$ 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 uncertainty matrix $\boldsymbol{V_y}$ associated with estimates $\boldsymbol{y}$ of the output quantities $\boldsymbol{Y}$ is calculated from

$$\boldsymbol{V_y} = \frac{1}{M-1}\,\boldsymbol{\Psi}'(\,\boldsymbol{\Psi}')^{\mathrm{T}},$$

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 27 specifies the evaluation of the uncertainty matrix $\boldsymbol{V_y}$ in terms of the vectors $\boldsymbol{y}_r$, and indicates the input and output parameters necessary for its determination.

This uncertainty matrix contains (generally a more reliable estimate of) the information that would be delivered by a linear analysis such as the GUM uncertainty framework. (In fact, it provides more than the GUM uncertainty framework, 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

[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 draws (equal to the number of Monte Carlo trials)
$\boldsymbol{\Psi}$	$m \times M$ matrix containing the model output values $\boldsymbol{y}_r$, i.e., $\boldsymbol{\Psi} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M).$
Output parameter	
$\boldsymbol{V_y}$	Uncertainty matrix associated with estimates $\boldsymbol{y}$ 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 27: Uncertainty matrix associated with estimates of a multivariate output quantity.

quantities for one choice (draw) of the input quantities. *Any* (scalar) derived quantity can be determined from this single set of output values. This quantity can be calculated for all columns, the resulting $1 \times M$ row vector being used to provide a discrete representation of the distribution function for that quantity (as in Section 5.6). In particular, the discrete representation can be used to provide a coverage interval for 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 GUM uncertainty framework

The GUM unceertainty framework has some limitations [3, 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 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 a Monte Carlo method. Therefore, since these circumstances cannot readily be tested, any cases of doubt should be validated. To this end, since the propagation of distributions is more general, it is recommended that both the GUM uncertainty framework and the Monte Carlo approach are applied and the results compared. If the comparison is favourable, the GUM uncertainty

[18] Note, however, that consideration is not given to using the matrix $\boldsymbol{\Psi}$ to obtain a discrete representation of the (joint) distribution function for the (multivariate) output quantity $\boldsymbol{Y}$. This is in contrast to the case of a univariate output quantity Y (Section 5.6).

framework can be used on this occasion and for sufficiently similar problems in the future. Otherwise, consideration can be given to using a Monte Carlo method instead.

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

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

A comparison procedure is based on the following objective: determine whether the coverage intervals obtained by the GUM uncertainty framework and a Monte Carlo method 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 decimal 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 the GUM uncertainty framework and a Monte Carlo method to determine whether the required number of correct digits in the coverage interval provided by the GUM uncertainty framework 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 favourable and the GUM uncertainty framework has been validated in this instance.

Example. The estimate of the output quantity for a nominally 100 g standard of mass [3, 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 GUM uncertainty framework [3]
2. A Monte Carlo method [2, 12] as an implementation of the propagation of distributions
3. Validation of the GUM uncertainty framework using a Monte Carlo method [2, 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 third edition of this document has been produced in the Software Support for Metrology (SS*f*M) programme 2004–2007. It is anticipated that future editions, 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 subsequent SS*f*M programmes.

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 previous editions of this report.

References

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

[2] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. Guide to the Expression of Uncertainty in Measurement. Supplement 1—Propagation of Distributions using a Monte Carlo method. Draft.

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

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

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

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] M. G. Cox and P. M. Harris. Software specifications for uncertainty evaluation. Technical Report CMSC 40/04, National Physical Laboratory, Teddington, UK, 2004.

[12] M. G. Cox and P. M. Harris. SS*f*M Best Practice Guide No. 6, Uncertainty evaluation. Technical Report DEM-ES-011, National Physical Laboratory, Teddington, UK, 2006.

[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] M. Evans, N. Hastings, and B. Peacock. *Statistical distributions*. Wiley, 2000.

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

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

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

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

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

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

[23] 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.

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

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

[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] P. L'Ecuyer and R. Simard. TestU01: A software library in ANSI C for empirical testing of random number generators, 2005. Version 6.0, January 14, 2005, http://www.iro.umontreal.ca/
lecuyer/.

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

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

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

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

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

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

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

[37] 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.

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

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

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

[41] 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, USA, 1994.

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

[43] B. A. Wichmann and I. D. Hill. Generating good pseudo-random numbers. *Computational Statistics and Data Analysis*, 2005. To appear.

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 28 specifies for a univariate, explicit, real 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 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
$\boldsymbol{C}$	$1 \times N$ 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 28: 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 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 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 is 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 [9].

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 ($1 \le i \le N$), set all $X_k = x_k$, apart from X_i, i.e., hold 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 29 specifies, for a univariate, explicit, real model, the use of a finite-difference formula to obtain sensitivity coefficients, and indicates the input and 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.

[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.

	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 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 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 parameter
$\boldsymbol{C}$	$1 \times N$ 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 29: 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 [6, 17]. 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, 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) [5, 22].

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 evaluation of the function. 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 [23] and ADOL-F [39].

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 [4] and ODYSSEE [18].

D Use of Kahan summation

The natural way to evaluate, for example, the summation

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

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 [25]. One such procedure is *Kahan summation* [26].

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 uses this result to apply a correction $\widehat{e}$ at every step of a recursive summation procedure for evaluating S. The procedure takes the following form:

```
$S = 0$
$e = 0$
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 [25].

E Updating procedure in an implementation of a Monte Carlo method

A consideration for an implementation of a Monte Carlo method, as described in Section 5, is the need to store large quantities of data and to perform numerical operations on that data. For a univariate, real model, an implementation of the method requires

- evaluating M model values and storing those values (Section 5.4)
- evaluating the arithmetic mean and standard deviation of the M values to provide the estimate of the output quantity and the associated standard uncertainty (Section 5.5)
- sorting the M values, and storing the sorted values, to provide a discrete representation of the distribution function for the output quantity (Section 5.6)
- evaluating a coverage interval in terms of the discrete representation (Section 5.7).

An approach is described here that does not require all M model values to be stored at the same time. The approach is based on expressing the results (including the estimate of the output quantity, associated standard uncertainty, etc.) for $M_0 + M_{\text{seq}}$ Monte Carlo trials in terms of the results for M_0 trials and M_{seq} additional model values. The procedure is applied iteratively starting with $M_0 = 0$. The approach only requires that the M_{seq} model values are stored, where M_{seq} will generally be small compared to the total number M required, and avoids the need to sort the complete set of M model values. It is also straightforward to integrate the approach with the adaptive Monte Carlo procedure described in Section 5.9 by taking $M_{\text{seq}} = M_{\text{adap}}$.

Suppose M_0 Monte Carlo trials have been undertaken, and the corresponding model values $y_{0,r},\ r = 1, \ldots, M_0$, are summarised by the arithmetic mean

$$y_0 = \frac{1}{M_0}\sum_{r=1}^{M_0} y_{0,r},$$

the variance

$$u^2(y_0) = \frac{1}{M_0 - 1}\sum_{r=1}^{M_0}(y_{0,r} - y_0)^2,$$

and the histogram $\{B_k, h_{0,k} : k = 1, \ldots, b\}$ (see Tables 21 and 22).

Suppose a further M_{seq} Monte Carlo trials are undertaken giving model values $y_r,\ r = 1, \ldots, M_{\text{seq}}$. Then, the arithmetic mean of the combined set of $M_0 + M_{\text{seq}}$ model values is

$$y = y_0 + \frac{1}{M_0 + M_{\text{seq}}}\sum_{r=0}^{M_{\text{seq}}}(y_r - y_0), \qquad (9)$$

and the variance is

$$u^2(y) = \frac{1}{M_0 + M_{\mathrm{seq}} - 1}\left((M_0 - 1)u^2(y_0) + M_0(y - y_0)^2 + \sum_{r=0}^{M_{\mathrm{seq}}}(y_r - y)^2\right). \quad (10)$$

For the evaluation of y and $u^2(y)$ using Formulae (9) and (10) it is not necessary to store the model values from the first M_0 trials, but only the arithmetic mean y_0 and variance $u^2(y_0)$ evaluated in terms of those values.

The histogram for the combined set of model values can similarly be updated. The new "heights" of the bars of the histogram corresponding to the bins $B_k,\ k = 1, \ldots, b$, are given by

$$h_k = h_{0,k} + \frac{\mathrm{card}(\{y_r \in B_k : r = 1, \ldots, M_{\mathrm{seq}}\})}{\eta_k - \eta_{k-1}}, \qquad k = 1, \ldots, b. \quad (11)$$

In cases where there are model values y_r satisfying $y_r < \eta_0$ or $y_r > \eta_b$, those values do not lie in any of the bins B_k and are, consequently, not counted in the application of Formula (11). It is necessary to store separately those values in order to determine a histogram for the complete set of model values.

Let $y_r^{\mathrm{L}},\ r = 1, \ldots, M_{\mathrm{L}}$, denote the values to the left of η_0 and $y_r^{\mathrm{R}},\ r = 1, \ldots, M_{\mathrm{R}}$, those to the right of η_b. Then, additional bars in the histogram for the complete set of model values are defined by the bins

$$B_0 = [\min\{y_r^{\mathrm{L}},\ r = 1, \ldots, M_{\mathrm{L}}\}, \eta_0)$$

with height

$$h_0 = \frac{M_{\mathrm{L}}}{\eta_0 - \min\{y_r^{\mathrm{L}},\ r = 1, \ldots, M_{\mathrm{L}}\}},$$

and

$$B_{b+1} = (\eta_b, \max\{y_r^{\mathrm{R}},\ r = 1, \ldots, M_{\mathrm{R}}\}]$$

with height

$$h_{b+1} = \frac{M_{\mathrm{R}}}{\max\{y_r^{\mathrm{R}},\ r = 1, \ldots, M_{\mathrm{R}}\} - \eta_b}.$$

Provided a "sensible" (initial) choice of bins $B_k,\ k = 1, \ldots, b$, is made, it can generally be expected that the numbers M_{L} and M_{R} of model values not contained in (one of) the bins are small. An approximation to the PDF for the output quantity is then given by the scaled histogram defined by $\{B_k, g_k : k = 0, \ldots, b + 1\}$, where

$$g_k = \frac{h_k}{M}, \qquad k = 0, \ldots, b + 1.$$

Let $y_{(r)}^{\mathrm{L}},\ r = 1, \ldots, M_{\mathrm{L}}$, and $y_{(r)}^{\mathrm{R}},\ r = 1, \ldots, M_{\mathrm{R}}$, denote the values y_r^{L} and y_r^{R} sorted into non-decreasing order. For the values

$$\boldsymbol{G} = \left(y_{(1)}^{\mathrm{L}}, \ldots, y_{M_{\mathrm{L}}}^{\mathrm{L}}, \eta_0, \ldots, \eta_b, y_{(1)}^{\mathrm{R}}, \ldots, y_{M_{\mathrm{R}}}^{\mathrm{R}}\right)$$

of the output quantity, assign cumulative probabilities

$$\boldsymbol{p} = \left(\frac{1/2}{M}, \ldots, \frac{M_{\mathrm{L}} - 1/2}{M}, g_0, \ldots, \sum_{k=1}^{b} g_k, \frac{M - M_{\mathrm{R}} + 1/2}{M}, \ldots, \frac{M - 1/2}{M} \right).$$

An approximation to the distribution function for the output quantity is then given by the piecewise-linear function joining the points $(G_k, p_k),\ k = 1, \ldots, M_{\mathrm{L}} + (b+1) + M_{\mathrm{R}}$. A coverage interval for the output quantity can be formed from this approximation to the distribution function in the manner described in Section 5.7.