Journal of Physics: Conference Series

Mathematica tools for uncertainty analysis

Mathematica tools for uncertainty analysis

M D Mikhailov, V Y Aibe
Inmetro, Fluid Dynamics Metrology, Rio de Janeiro, Brazil

E-mail: dinam@inmetro.gov.br

Abstract. Some functions of the author´s *Mathematica* 9 package are presented. Links are given to the author´s interactive demonstrations. *Mathematica* supports 143 types of statistical distribution covering not only metrology but almost all statistical areas. The Supplement 1 to the ISO GUM mentions that analytical methods are ideal ones, but assume they are applicable only in simple cases, and recommends Monte Carlo method. In contrast, this paper presents *Mathematica* functions that can explore a multitude of methods in order to find analytical, numerical, or statistical results using (usually) one million random variates.

1. Introduction

The GUM Guide [1] provides a well-known framework for assessment and evaluation of uncertainties based on the law of propagation of uncertainty. This guide has generated the appearance of a great number of computer systems and calculators dedicated in helping estimate the uncertainty of measurements. Typing the key "uncertainty calculator" in Google gives back about 2,460,000 results. Wikipedia alone describes 18 uncertainty propagation software systems [2].

The authors of the present work have previously devised an uncertainty calculus based on the GUM in Mathematica 6 [3]. An object $x \pm \Delta x$ called uncertainty number was defined. After assuming $\Delta x/x \ll 1$ the terms Δx were neglected in relational operations (<, =, >). Our Mathematica rules transform one expression of independent uncertainty numbers into single uncertainty number. Later on we were able to extend these rules to cover also fully dependent uncertainty numbers. Special cases of the rules we used in the interactive demonstrations [4, 5, 6, and 7].

The GUM approach [1] is only exact for linear models, since it is based on the first order Taylor expansion, neglecting higher order terms. Thus direct application of the GUM rules in nonlinear models can potentially give quite misleading results. In order to account for non-linear scenarios, GUM allows for techniques other than the law of propagation of uncertainty (ref. [1], section G.1.5). The Supplement 1 to the GUM [8] recommends propagation of distributions, which can be applied in non-linear problems. In this Supplement, it is mentioned that analytical methods are ideal, but only viable in simple cases. Because of this, the Supplement recommends the Monte-Carlo method.

In this paper we continue our previous work [3] by presenting a new *Mathematica* functions able to compute symbolically or numerically the propagation of statistical distributions, thereby extending our uncertainty calculus into the non-linear domain, at the same time avoiding the limitations present in the GUM approach. Interactive examples using this newly developed package are available on-line in Computable Document Format (CDF) files [9]. There users can interactively compare the exact uncertainties versus GUM uncertainties for `Sin` and `Cos` function [10]. The free *Mathematica* Player [11] is needed to evaluate the interactive tools [3, 4, 5, 6, 7, 9, 10].

Mathematica support 143 statistical distributions covered different research areas such as Actuarial

Journal of Physics: Conference Series **575** (2015) 012042 doi:10.1088/1742-6596/575/1/012042

Science, Finance, Metrology, Risk, Reliability etc. From these distributions, there are many ways to define new, user-derived distributions that behave just like any other built-in distribution. Recently, the authors have published a quick, interactive reference guide to the logical relationships obeyed by the statistical parameters of the built-in *Mathematica* distributions [12], and the percentile plots of 78 continuous statistical distributions [13].
All functions, input and output lines bellow are written in the standard for *Mathematica* Currier font.

2. Objectives

This paper aims is the comparison of results obtained by original *Mathematica* functions based on propagation of uncertainty and propagation of distributions.
Mathematica use **`NormalDistribution[μ,σ]`** for the Gaussian distribution with mean μ and standard deviation σ. Since almost all *Mathematica* built-in objects are full English names beginning with capital letters, we follow this convention for our package functions as well. The functions demonstrated are: **`UncertainCalculus`**, **`GUM`**, **`CoverageInterval`**, and **`Measurand`**.

3. UncertainCalculus

The first version of uncertain calculus is demonstrated at ENCIT 2008 [3]. The function **UncertainCalculus[*case*]** evaluate input line with expression including uncertain numbers like $\mu\pm\sigma$ (**`PlusMinus[μ,σ]`**). Similarly to **`GUM`** function ***case*** = 1 activate the rule for independent variables and ***case*** = 3 activate a rule for fully correlated variables. The ***case = 0*** deactivate **UncertainCalculus**.
The **`UncertainCalculus[1]`** activate the rule for independent variables.
Then the following input line gives the corresponding output lines.

`c1 (μ1 ± σ1) + c2 (μ2 ± σ2)`

$$(c_1\mu_1 + c_2\mu_2) \pm \sqrt{c_1^2\sigma_1^2 + c_2^2\sigma_2^2} \qquad \text{Eqs 1}$$

In case of detectable error, like **`UncertainCalculus[2]`** the warning message appears.
The **`UncertainCalculus[3]`** activate the rule for fully correlated variables.
Then the following input line gives the corresponding output lines.

`c1 (μ1 ± σ1) + c2 (μ2 ± σ2)`

$$(c_1\ \mu_1+c_2\ \mu_2)\pm(c_1\ \sigma_1+c_2\ \sigma_2) \qquad \text{Eqs 2}$$

For one variable independent and fully correlated variables rules give the same results.
Next examples evaluate two nonlinear functions

`(μ ± σ)^2`

$$\mu^2\pm 2\ \mu\ \sigma \qquad \text{Eqs 3}$$

`Sin[μ±σ]`

`Sin[μ]±σ Cos[μ]` Eqs 4

Finally, the **`UncertainCalculus[0]`** deactivate the uncertain calculus.

4. GUM

The function **`GUM`** implements the formulas given in [1].
`GUM[expr,x==μ±σ,case]` or **`GUM[expr,{x1==μ1±σ1,...,xn==μn±σn},case]`**
gives first order series approximation for expectation ±standard deviation of **`expr`**.
Mathematica use **`lhs==rhs`** to denote equation, while **`lhs=rhs`** to evaluate **`rhs`** and assign the result to **`lhs`**.
`case` = 1 or IdentityMatrix[n] (independent variables).
`case` = 2 nonlinear expr (more series terms included in case 1).
`case` = 3 or n x n constant matrix with elements 1 (fully correlated variables).
`case` = n x n symmetric matrix with main diagonal 1 and elements in the closed interval -1 to 1 .

```
GUM[c1 x1 + c2 x2, {x1 == μ1 ± σ1, x2 == μ2 ± σ2}, 1]
```

$(c_1\mu_1 + c_2\mu_2) \pm \sqrt{c_1^2\sigma_1^2 + c_2^2\sigma_2^2}$ Eqs 5

```
GUM[c1 x1 + c2 x2, {x1 == μ1 ± σ1, x2 == μ2 ± σ2}, 3]
```

$(c_1\ \mu_1+c_2\ \mu_2) \pm (c_1\ \sigma_1+c_2\ \sigma_2)$ Eqs 6

```
GUM[x^2, x == μ ± σ, 1]
```

$\mu^2 \pm 2\ \mu\ \sigma$ Eqs 7

```
GUM[Sin[x], x == μ ± σ, 1]
```

$\mathrm{Sin}[\mu]\ \pm\ \sigma\ \mathrm{Cos}[\mu]$ Eqs 8

The uncertain calculus (Eqs 1 to Eqs 4), and GUM (Eqs 5 to Eqs 8) gives the same results.
The uncertain calculus could not use the correlation matrix **R**, where **0≤ρ≤1**.

$$R = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

Eqs 9

```
GUM[c1 x1 + c2 x2, {x1 == μ1 ± σ1, x2 == μ2 ± σ2}, R]
```

$(c_1\ \mu_1+c_2\ \mu_2) \pm \sqrt{(c_1^2\ \sigma_1^2+2\ c_1\ c_2\ \rho\ \sigma_1\ \sigma_2+\sigma_1^2\ \sigma_2^2)}$ Eqs 10

The solution given by Eqs 5 and Eqs 6 are special cases of Eqs 10 for **ρ=0** and **ρ=1**, respectivelly.
For the non-linear measurand **x^2** , **Sin[x]**, and **Cos[x]** the results are

```
GUM[x^2, x == μ ± σ, 2]
```

$\mu^2 \pm \sqrt{4\mu^2\sigma^2 + 2\sigma^2}$ Eqs 11

```
FullSimplify[GUM[Sin[x], x == μ ± σ, 2]]
```

$Sin[\mu] \pm \sqrt{\sigma^2 \cdot Cos[\mu]^2 - \frac{1}{4}\cdot\sigma^4\cdot(1+3\cdot Cos[2\mu])}$ Eqs 12

```
FullSimplify[GUM[Cos[x], x == μ ± σ, 2]]
```

$Cos[\mu] \pm \sqrt{\frac{1}{4}\cdot\sigma^4\cdot(-1+3\cdot Cos[2\mu])+\sigma^2\cdot Sin[\mu]^2}$ Eqs 13

5. Uncertainty

The **Uncertainty** function propagate distributions to find the exact results analytically, numerically, or statistically.

Uncertainty[expr,case] in domain of symbols, rationals, or reals attempt to calculate the uncertainty (expectation ±sandard deviation) of expr.

Uncertainty[expr,case,n] in domain of reals calculate uncertainty of expr by using n pseudoradom variates.

The argument case specify statistical distributions of the variables as in the *Mathematica* function Expectation.

```
Uncertainty[c1 x1 + c2 x2,
{x1 ≈ NormalDistribution[μ1, σ1],
x2 ≈ NormalDistribution[μ2, σ2]}]
```

$(c_1\mu_1 + c_2\mu_2) \pm \sqrt{c_1^2\sigma_1^2 + c_2^2\sigma_2^2}$ Eqs 14

```
MatrixForm[Ξ = MultinormalCovarianceMatrix[{σ1, σ2}]]
```

$$\begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2 \\ \sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

Eqs 15

```
PowerExpand[Simplify[Uncertainty[c1 x1 + c2 x2,
```

```
   {x1, x2} ≈ MultinormalDistribution[{μ1, μ2}, Ξ]]]]
```

$(c_1\ \mu_1+c_2\ \mu_2)\pm(c_1\ \sigma_1+c_2\ \sigma_2)$ Eqs 16

```
PowerExpand[Simplify[Uncertainty[c1 x1 + c2 x2,
   {x1, x2} ≈ MultinormalDistribution[{μ1, μ2}, Ξ R]]]]
```

$(c_1\ \mu 1+c_2\ \mu_2)\pm\sqrt{(c_1^2\ \sigma_1^2+2\ c_1\ c_2\ \rho\ \sigma_1\ \sigma_2+c_2^2\ \sigma_2^2)}$ Eqs 17

Since the measurand is linear the GUM solutions Eqs 5, Eqs 6, and Eqs10 coincide with Eqs14, Eqs16, and Eqs17.

For the non-linear measurand **`x^2`** , **`Sin[x]`**, and **`Cos[x]`** the results are

```
Uncertainty[x^2, x ≈ NormalDistribution[μ, σ]] /.
Sqrt[a_] Sqrt[b_] :> Sqrt[Expand[a b]]
```

$(\mu^2+\sigma^2)\pm\sqrt{4\mu^2\sigma^2+2\sigma^4}$ Eqs 18

The exact analitical solutions for **`Sin[x]`** and **`Cos[x]`** needs more time measured in second by the function **`Timing`**.

```
Timing[Uncertainty[Sin[x], x ≈ NormalDistribution[μ, σ]] /.
   Sqrt[x_] :> Sqrt[FullSimplify[ExpToTrig[x]]] /.
  (x_ + Exp[y_]) :> Exp[y] (1 + Exp[-y] x)]
```

$$\left\{213.986572, e^{-\frac{\sigma^2}{2}}\cdot Sin[\mu]\pm\frac{1}{\sqrt{2}}\cdot\sqrt{(1-e^{-\sigma^2})(1+e^{-\sigma^2}\cdot Cos[2\mu])}\right\}$$ Eqs 19

```
Timing[Uncertainty[Cos[x], x ≈ NormalDistribution[μ, σ]] /.
 Sqrt[x_] :> Sqrt[FullSimplify[ExpToTrig[x]]] /.
  (x_ + Exp[y_]) :> Exp[y] (1 + Exp[-y] x)]
```

$$\left\{142.647314, e^{-\frac{\sigma^2}{2}}\cdot Cos[\mu]\pm\frac{1}{\sqrt{2}}\cdot\sqrt{(1-e^{-\sigma^2})(1-e^{-\sigma^2}\cdot Cos[2\mu])}\right\}$$ Eqs 20

The exact results for **`x^2`** , **`Sin[x]`**, and **`Cos[x]`** obtained by **`Uncertainty`** function (Eqs 18, Eqs 19, and Eqs 20) correspond to (Eqs 11, Eqs 12 and Eqs 13) obtained by **`GUM`** function.

Finally for a nonlinear measurand we generate 5 times 1 million random variates and extract the mean and standard deviation from the obtained data. The speed of computation is because *Mathematica* use a package array that permitted to do manipulations at once.

```
Timing[TableForm[Table[Uncertainty[x1^2 + x2^3,
    {x1 ≈ NormalDistribution[10, 0.1],
     x2 ≈ NormalDistribution[5,0.2]}, 10^6], {5}]]]
```

```
            225.627±15.2012
            225.632±15.1982
{2.059213,  225.579±15.1832},                          Eqs 21
            225.598±15.1781
            225.6±15.1789
```

The exact numerical result is:

```
Uncertainty[x1^2 + x2^3,
{x1 ≈ NormalDistribution[10, 0.1],
```

```
x2 ≈ NormalDistribution[5, 0.2]}]
225.61 ± 15.1803
```
Eqs 22

6. CoverageInterval

CoverageInterval[expr,case,q] evaluate **CoverageInterval[expr,case,(1-q)/2,(1+q)/2]**

CoverageInterval[expr,case,q,n] evaluate **CoverageInterval[expr,case,(1-q)/2,(1+q)/2,n]**

CoverageInterval[expr,case,{q1,q2}] attempt to calculate coverage interval from **q1** to **q2**,
where **0<q1<q2<1**and **q1+q2==1**

CoverageInterval[expr, case, {q1,q2}, n] calculate coverage interval from **q1** to **q2** by using n pseudoradom variates.

The argument case specifies statistical distributions of the variables as in the *Mathematica* function Expectation.

The function **ToInterval** transform the uncertainty given by Eqs 17 as:

```
ToInterval[Eqs17]
```
$\text{Interval}[\{c_1\ \mu_1+c_2\ \mu_2-\sqrt{(c_1^2\ \sigma_1^2+2\ c_1\ c_2\ \rho\ \sigma_1\ \sigma_2+c_2^2\ \sigma_2^2)},$
$c_1\ \mu_1+c_2\ \mu_2+\sqrt{(c_1^2\ \sigma_1^2+2\ c_1\ c_2\ \rho\ \sigma_1\ \sigma_2+c_2^2\ \sigma_2^2)}\}],$ Eqs 23

For the same problem the function **CoverageInterval** gives

```
PowerExpand[Simplify[
  CoverageInterval[c1 x1 + c2 x2,
   {x1, x2} ≈ MultinormalDistribution[{μ1, μ2}, Ξ R], q], 0 < q < 1]]
```
$\text{Interval}[\{c_1\ \mu1+c_2\ \mu_2-\sqrt{2}\ c_1^2\ \sigma_1^2+2\ c_1\ c_2\ \rho\ \sigma_1\ \sigma_2+c_2^2\ \sigma_2^2)\ \text{InverseErfc}[1-q],$
$c_1\ \mu1+c_2\ \mu_2-\sqrt{2}\ \sqrt{(c_1^2\ \sigma_1^2+2\ c_1\ c_2\ \rho\ \sigma_1\ \sigma_2+c_2^2\ \sigma_2^2)}\ \text{InverseErfc}[1+q]\}],$ Eqs 24

We like to find such a parameter q that transform Eqs 24 in Eqs 23

```
FullSimplify[Solve[Eqs23[[1, 2]] == Eqs24[[1, 2]], q]][[1, 1]]
  q→Erf[1/√2]
```

The obtained probability q is exact and could be computed numerically with any number of digits

```
N[%, 12]
q → 0.682689492137
```
Eqs 25

The following example shows that exact numerical solution spends more time that random variate method with a million variate.

```
Timing[CoverageInterval[
x^2, x ≈ NormalDistribution[10, 0.1], 0.95]]
{7.503648, Interval[{96.1185, 103.958}]}
```
Eqs 26

```
 Timing[CoverageInterval[
x^2, x ≈ NormalDistribution[10, 0.1], 0.95, 10^6]]
{1.357209, Interval[{96.1248, 103.972}]},
```
Eqs 27

7. Feasible Region

In the domain of symbols, the function **Uncertainty** obtain exact solution for uncertainty of the measurand **Exp[x^2]**, when **x** is normally distributed with mean μ, and standard deviation σ.

The exact solution is [15]:

$$\frac{e^{\frac{2\mu^2}{1-2\sigma^2}}}{\sqrt{1-2\sigma^2}} \pm \sqrt{\frac{e^{\frac{2\mu^2}{1-4\sigma^2}}}{\sqrt{1-4\sigma^2}} - \frac{e^{\frac{2\mu^2}{1-2\sigma^2}}}{1-2\sigma^2}}$$

For the value used in [14] μ**=0**, and σ**=1** the obtained solution becomes $-i\pm\sqrt{1\cdot \; i \; \sqrt{3}}$, where $i=\sqrt{\cdot\ 1}$.
For this reason "Monte Carlo techniques is computationally problematic" [14]
The full analysis of the exact solution is not object of the present notebook. In addition to the well-known restriction σ>**0**, we take two more restrictions. Than we **Reduce** function finds the following feasible region:

```
 Reduce[{σ>0,1-2σ2>0,1-4 σ2>0},σ],
0<σ<1/2,
```

Eqs 27

In the domain of **Rationals**, the function **Uncertainty** needs about 2 second to obtain uncertainty of **Exp[x^2]**, when **x** is normally distributed with mean **1**, and standard deviation **1/10**.

```
Timing[Uncertainty[Exp[x^2], x ≈ NormalDistribution[1, 1/10]]]
```

$$\left\{1.562500, \frac{5}{7}\sqrt{2}\cdot e^{\frac{50}{49}} \pm \sqrt{-\frac{50\cdot e^{\frac{100}{49}}}{49} + \frac{5\cdot e^{\frac{25}{12}}}{2\sqrt{6}}}\right\}$$

Eqs 28

Since the result is also in the domain of **Rationals** it could be transformed in **Reals** with arbitrary number of digits.
Next line show 20 digits:

```
2.8024933887978839525 ± 0.58552102828175616088
```

```
Timing[TableForm[
  Table[{Uncertainty[Exp[x^2],

     x ≈ NormalDistribution[1., .1], 10^6]}, {5}]]]
```

```
           2.80168±0.58508
           2.80251±0.58524
{0.468750, 2.80227±0.585468}
           2.80199±0.586149
           2.80262±0.585053
```

Eqs 29

For the values μ**=0**, and σ**=1** used by Bruce Christianson and Maurice Cox the solution is complex number and **Uncertainty** function return the input line after 327 seconds.
Finally we use different number of random variates to obtain completely non convergent results

"random variates"	"uncertainty"
1000	110.54±1880.48
10000	417.604±12989.1
100000	3182.44±431251
1000000	$183460.\pm1.24656\times10^{8}$

Eqs 30

For **UnitSetep** function the feasible regions of the results are built-in *Mathematica.*

```
Uncertainty[UnitStep[x - a], x ≈ UniformDistribution[{2, m}]]
```

$$\left(\begin{cases} 1 & a \; 2 \;\&\&\; m \; 2 \\ \frac{\cdot\, a \; m}{\cdot\, 2 \; m} & a \; 2 \;\&\&\; a \cdot m \; 0 \\ 0 & \text{True} \end{cases}\right) \; \sqrt{\begin{cases} \frac{2\, a \cdot a^2 \cdot 2\, m \; a\, m}{|\cdot 2 \; m|^2} & a \; 2 \;\&\&\; a \cdot m \; 0 \\ 0 & \text{True} \end{cases}}$$

Eqs 31

8. Acknowledgments

The authors are thankful to the Inmetro - National Institute of Metrology, Quality and Technology for the financial support to this research through a CNPq Research Fellowship - Edital MCT/ CNPq / Inmetro nº 059/2010 - PROMETRO Processo Nº 563061/2010-3. This work is a contribution of the INMETRO and is not subject to copyright in the Brazil.

9. References

[1] JCGM 100:2008 - GUM 1995 with minor corrections
http://www.bipm.org/en/publications/guides/gum.html
[2]. Uncertainty propagation software
http://en.wikipedia.org/wiki/List_of_uncertainty_propagation_software
[3]. V Y Aibe and M D Mikhailov, Uncertainty Calculus in Metrology, Proceedings of ENCIT 2008, 12th Brazilian Congress of Thermal Engineering and Sciences, Belo Horizonte, MG, November 10-14, 2008
http://repositorios.inmetro.gov.br/handle/10926/448
[4]. V Y Aibe, M D Mikhailov,
http://demonstrations.wolfram.com/SolvingMatrixSystemsWithRealIntervalOrUncertainElements/
[5]. V Y Aibe, M D Mikhailov,
http://demonstrations.wolfram.com/SolvingALinearSystemWithUncertainCoefficients/
[6]. V Y Aibe, M D Mikhailov,
http://demonstrations.wolfram.com/AreaPerimeterAndDiagonalOfARectangleWithUncertainSides/
[7]. V Y Aibe, M D Mikhailov,
http://demonstrations.wolfram.com/FitInterpolationOrPolynomialInterpolationInUncertainCalculus/
[8]. JCGM 101:2008 Supplement 1 to the GUM
http://www.sci.utah.edu/~kpotter/Library/Papers/jcgm:2008:PDMC/
[9]. V Y Aibe and M D Mikhailov, Uncertainty and UncertaintyGUM *Mathematica* functions,
http://repositorios.inmetro.gov.br/handle/10926/1898
[10]. M D Mikhailov and V Y Aibe.
http://demonstrations.wolfram.com/GUMVersus
ExactUncertaintyOfSinXAndCosXWhenXIsNormallyDistrib/
[11]. M D Mikhailov and V Y Aibe.
. http://www.wolfram.com/cdf-player/
[12]. M D Mikhailov and V Y Aibe.
http://demonstrations.wolfram.com/LogicalExpressionForAssumptionsOnParametersInMathematicasSym/
[13]. M D Mikhailov, V Y Aibe, and I Yasuda
http://demonstrations.wolfram.com/PercentilesOfContinuousProbabilityDistributions
[14]. Bruce Christianson and Maurice Cox, Automatic propagation of uncertainties,
In H. M. Bücker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, editors,
Automatic Differentiation: Applications, Theory, and Implementations,
Lecture Notes in Computational Science and Engineering, pages 47-58, Springer, 2005.
[15] I Yasuda, M D Mikhailov and V Y Aibe, New *Mathematica* Uncertainty Analysis Tools, Proceedings of Flomeko 2013, 17th International Flow Measurement Conference, Paris - France, November 24-26, 2013.