Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Institute of Metrology METAS

METAS Uncertainty Library

Michael Wollensack

Outline

Introduction

Right Triangle Example

METAS UncLib

Python Wrapper

Pendulum Example

Conclusion

Introduction

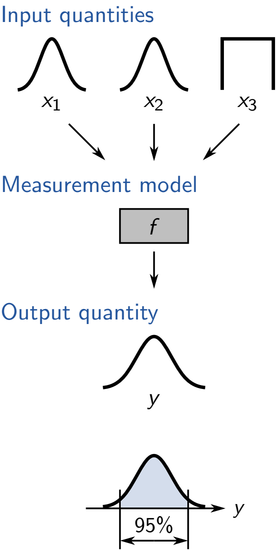
Problem

Computation of measurement uncertainties according to ISO-GUM.

Solution

1. Identify all uncertainty influences.
2. Set up a measurement model.
3. Propagate all uncertainties through this model.

Uncertainty according to ISO-GUM (scalar)



Input quantities

Measurement model

Output quantity

Linear uncertainty propagation

Input standard uncertainties $\mathrm{u}(x_1)$, $\mathrm{u}(x_2)$, $\mathrm{u}(x_3)$

Sensitivities $\frac{\partial f}{\partial x_1}$, $\frac{\partial f}{\partial x_2}$, $\frac{\partial f}{\partial x_3}$

Output standard uncertainty $\mathrm{u}^2(y) =$

$$\left(\frac{\partial f}{\partial x_1}\right)^2 \mathrm{u}^2(x_1) + \left(\frac{\partial f}{\partial x_2}\right)^2 \mathrm{u}^2(x_2) + \left(\frac{\partial f}{\partial x_3}\right)^2 \mathrm{u}^2(x_3)$$

Expanded uncertainty $\mathrm{U}^2(y) = (1.96)^2\, \mathrm{u}^2(y)$

Derivative $f'(x) = \frac{d}{dx} f(x)$

The derivative $f'(x)$ of a function $y = f(x)$ of a real variable measures the sensitivity to change of the function value (output value y) with respect to a change in its argument (input value x).

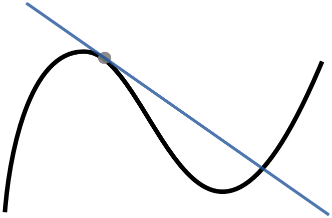


Figure: The graph of a function, drawn in black, and a tangent line to that function, drawn in blue. The slope of the tangent line is equal to the derivative of the function at the marked point.

The basic rules for derivatives

Derivatives $f'(x) = \frac{d}{dx} f(x)$

- $\frac{d}{dx} ax = a$
- $\frac{d}{dx} x^b = b x^{b-1}$
- $\frac{d}{dx} e^x = e^x$
- $\frac{d}{dx} \ln(x) = \frac{1}{x}$
- $\frac{d}{dx} \sin(x) = \cos(x)$
- $\frac{d}{dx} \cos(x) = -\sin(x)$

Partial derivatives $\frac{\partial}{\partial x_i} f(x_1, \ldots, x_N)$

- $\frac{\partial}{\partial a}(a+b) = 1$, $\frac{\partial}{\partial b}(a+b) = 1$
- $\frac{\partial}{\partial a} ab = b$ and $\frac{\partial}{\partial b} ab = a$

Sum rule

$(\alpha f + \beta g)' = \alpha f' + \beta g'$

Chain rule

If $f(x) = h(g(x))$ then

$$f'(x) = h'(g(x)) \cdot g'(x)$$

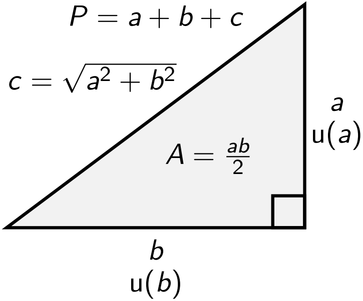
If $z = h(y)$ and $y = g(x)$ then

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}$$

Right triangle example

- Cathetus $a = 3$, $\mathrm{u}(a) = 0.3$
- Cathetus $b = 4$, $\mathrm{u}(b) = 0.4$

- What's the value and unc of the hypotenuse c?
- What's the value and unc of the perimeter P?
- What's the value and unc of the area A?



Solving the right triangle example by hand

Step	Function	Partial Derivatives	Chain Rule
1, 2	$a = 3,\ b = 4$		
3	$x_1 = a^2 = 9$	$\frac{\partial x_1}{\partial a} = 2a = 6$	
4	$x_2 = b^2 = 16$	$\frac{\partial x_2}{\partial b} = 2b = 8$	
5	$x_3 = x_1 + x_2 = 25$	$\frac{\partial x_3}{\partial x_1} = 1,\ \frac{\partial x_3}{\partial x_2} = 1$	$\frac{\partial x_3}{\cancel{\partial x_1}} \frac{\cancel{\partial x_1}}{\partial a} = 6,\ \frac{\partial x_3}{\partial b} = 8$
6	$c = \sqrt{x_3} = 5$	$\frac{\partial c}{\partial x_3} = \frac{1}{2\sqrt{x_3}} = 0.1$	$\frac{\partial c}{\partial a} = 0.6,\ \frac{\partial c}{\partial b} = 0.8$
7	$x_4 = a + b = 7$	$\frac{\partial x_4}{\partial a} = 1,\ \frac{\partial x_4}{\partial b} = 1$	
8	$P = x_4 + c = 12$	$\frac{\partial P}{\partial x_4} = 1,\ \frac{\partial P}{\partial c} = 1$	$\frac{\partial P}{\partial a} = 1.6,\ \frac{\partial P}{\partial b} = 1.8$
9	$x_5 = ab = 12$	$\frac{\partial x_5}{\partial a} = b,\ \frac{\partial x_5}{\partial b} = a$	
10	$A = \frac{x_5}{2} = 6$	$\frac{\partial A}{\partial x_5} = 0.5$	$\frac{\partial A}{\partial a} = 2,\ \frac{\partial A}{\partial b} = 1.5$

Solving the right triangle example by hand

11. What's the value and unc of the hypotenuse c?
    - $c = 5$
    - $\mathrm{u}^2(c) = \left(\frac{\partial c}{\partial a}\right)^2 \mathrm{u}^2(a) + \left(\frac{\partial c}{\partial b}\right)^2 \mathrm{u}^2(b) = (0.6 \cdot 0.3)^2 + (0.8 \cdot 0.4)^2$
    - $\mathrm{u}(c) = \sqrt{0.1348} = 0.3672$
12. What's the value and unc of the perimeter P?
    - $P = 12$
    - $\mathrm{u}^2(P) = \left(\frac{\partial P}{\partial a}\right)^2 \mathrm{u}^2(a) + \left(\frac{\partial P}{\partial b}\right)^2 \mathrm{u}^2(b) = (1.6 \cdot 0.3)^2 + (1.8 \cdot 0.4)^2$
    - $\mathrm{u}(P) = \sqrt{0.7488} = 0.8653$
13. What's the value and unc of the area A?
    - $A = 6$
    - $\mathrm{u}^2(A) = \left(\frac{\partial A}{\partial a}\right)^2 \mathrm{u}^2(a) + \left(\frac{\partial A}{\partial b}\right)^2 \mathrm{u}^2(b) = (2 \cdot 0.3)^2 + (0.8 \cdot 1.5)^2$
    - $\mathrm{u}(A) = \sqrt{0.7200} = 0.8485$

Solving the right triangle example using a PC

Which steps could be automated?

1. Decomposition of the measurement model into elementary functions. Implemented in any compiler or interpreter.
2. Partial derivatives of elementary functions $\frac{\partial}{\partial x_i} f(x_1, \ldots, x_N)$.
3. Sum $(\alpha f + \beta g)' = \alpha f' + \beta g'$ and Chain rule $\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial x}$.
4. Uncertainty propagation
$$\mathrm{u}^2(y) = \left(\frac{\partial f}{\partial x_1}\right)^2 \mathrm{u}^2(x_1) + \left(\frac{\partial f}{\partial x_2}\right)^2 \mathrm{u}^2(x_2) + \ldots + \left(\frac{\partial f}{\partial x_N}\right)^2 \mathrm{u}^2(x_N).$$

Combination of steps 2 and 3 is called automatic differentiation.
METAS UncLib implements steps 2 to 4 in the LinProp module.

Solving the right triangle example using Python

```
#  Definition of the input uncertainty objects
from metas_unclib import *        # import METAS UncLib
use_linprop()                     # linear unc propagation
a = ufloat(3.0, 0.3, desc=’a’)  # value 3, stdunc 0.3
b = ufloat(4.0, 0.4, desc=’b’)  # value 4, stdunc 0.4

#  Compute output quantities
c = umath.sqrt(a**2 + b**2)       # hypotenuse
P = a + b + c                     # perimeter
A = a * b / 2.0                   # area
```

```
--> c =  (5.0 +/- 0.367151)
--> P = (12.0 +/- 0.865332)
--> A =  (6.0 +/- 0.848528)
```

METAS UncLib

- METAS UncLib is a C# software library.
- It supports creation of uncertainty objects and subsequent calculation with them as well as storage of the results.
- It's able to handle complex-valued and multivariate quantities.
- Object-oriented implementation and overloaded operators hide the complexity from the user.

- It has been developed with Visual Studio 2008, 2013 and 2019.
- METAS UncLib V2.8 requires the .NET Framework V4.6.2.
- An Installer (msi-file) for METAS UncLib is available for free at www.metas.ch/unclib.

Python Wrapper

- METAS UncLib Python is an extension to Python.
- It supports creation of uncertainty objects and subsequent calculation with them as well as storage of the results.
- It's able to handle complex-valued and multivariate quantities.

- It has been developed with Python V3.6 (numpy, pythonnet).
- It requires the C# library METAS UncLib in the background.
- The classes `ufloat`, `ucomplex`, `ufloatarray` and `ucomplexarray` wrap METAS UncLib to Python over the .NET interface (pythonnet).
- The Python wrapper, several examples and a tutorial are part of the METAS UncLib Installer (msi-file) or the metas-unclib (pypi package).

Pendulum example

Gravitational acceleration in $\mathrm{m/s^2}$

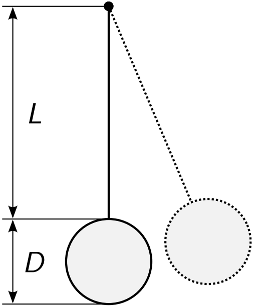
$$g = \frac{4\pi^2}{\tau^2}\left(L + \frac{D}{2} + \frac{D^2}{5(2L+D)}\right)$$

where

τ is the period in s

L is the length in m

D is the diameter in m



Solving the pendulum example using Python

```
from metas_unclib import *  # import METAS UncLib
use_linprop()               # linear unc propagation

# Input quantaties time measurement
n_periods = 20

n_tau_meas = ufloatfromdistribution(
  StudentTDistribution(24.9060, 0.0712 / np.sqrt(10), 9),
  desc='Time␣N␣Periods␣Measurement␣/␣s')

dtau_react = ufloatfromdistribution(
  UniformDistribution(-0.1, 0.1),
  desc='Time␣Reaction␣/␣s')

dtau_res = ufloatfromdistribution(
  UniformDistribution(-0.005, 0.005),
  desc='Time␣Resolution␣/␣s')
```

Solving the pendulum example using Python

```python
# Input quantaties length measurement
L_meas = ufloatfromdistribution(
  StudentTDistribution(0.33316, 0.00026 / np.sqrt(6), 5),
  desc='Length␣Measurement␣/␣m')

dL_res = ufloatfromdistribution(
  UniformDistribution(-0.0005, 0.0005),
  desc='Length␣Resolution␣/␣m')

dL_read = ufloatfromdistribution(
  TriangularDistribution(-0.0005, 0.0005),
  desc='Length␣Reading␣/␣m')
```

Solving the pendulum example using Python

```
# Input quantaties diameter measurement
D_meas = ufloatfromdistribution(
  StudentTDistribution(0.1005, 0.000058 / np.sqrt(4), 3),
  desc='Diameter␣Measurement␣/␣m')

dD_res = ufloatfromdistribution(
  UniformDistribution(-0.001, 0.001),
  desc='Diameter␣Resolution␣/␣m')

dD_read = ufloatfromdistribution(
  TriangularDistribution(-0.0005, 0.0005),
  desc='Diameter␣Reading␣/␣m')
```

Solving the pendulum example using Python

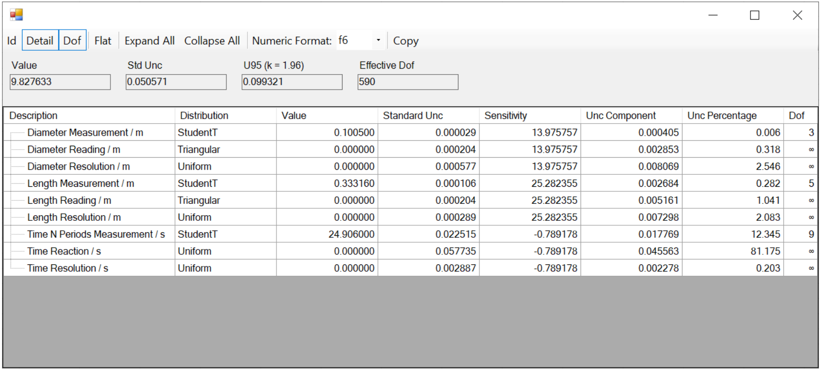
```
# Measurement model
n_tau = n_tau_meas + dtau_res + dtau_react
tau = n_tau / n_periods
L = L_meas + dL_res + dL_read
D = D_meas + dD_res + dD_read

g = 4 * np.pi**2 / tau**2 * (L + D / 2 +
                            (D**2 / (5 * (2 * L + D))))

# Output quantaty
print(g)
unc_budget(g)
```

```
--> g = (9.8276 +/- 0.0507)
```

Solving the pendulum example using Python



Numeric Format: f6

Value: 9.827633

Std Unc: 0.050571

U95 (k = 1.96): 0.099321

Effective Dof: 590

Description	Distribution	Value	Standard Unc	Sensitivity	Unc Component	Unc Percentage	Dof
Diameter Measurement / m	StudentT	0.100500	0.000029	13.975757	0.000405	0.006	3
Diameter Reading / m	Triangular	0.000000	0.000204	13.975757	0.002853	0.318	∞
Diameter Resolution / m	Uniform	0.000000	0.000577	13.975757	0.008069	2.546	∞
Length Measurement / m	StudentT	0.333160	0.000106	25.282355	0.002684	0.282	5
Length Reading / m	Triangular	0.000000	0.000204	25.282355	0.005161	1.041	∞
Length Resolution / m	Uniform	0.000000	0.000289	25.282355	0.007298	2.083	∞
Time N Periods Measurement / s	StudentT	24.906000	0.022515	-0.789178	0.017769	12.345	9
Time Reaction / s	Uniform	0.000000	0.057735	-0.789178	0.045563	81.175	∞
Time Resolution / s	Uniform	0.000000	0.002887	-0.789178	0.002278	0.203	∞

Figure: Uncertainty budget of gravitational acceleration g in $\mathrm{m/s^2}$

Conclusion

METAS UncLib

- Generic software library for propagation of uncertainties.
- Is able to handle complex-valued and multivariate quantities.
- Can be used from C# (Visual Studio), MATLAB or Python.
- Used in many labs at METAS and around the world.
- Used in large applications like METAS VNA Tools.
- Available for free since 2009 at www.metas.ch/unclib.

Validation

- Only derivatives of elementary functions are programmed.
  Elementary functions are easy to check.
- Overloaded operators hide the complexity from the user.
- User has the possibility to check if the unc budget is plausible.