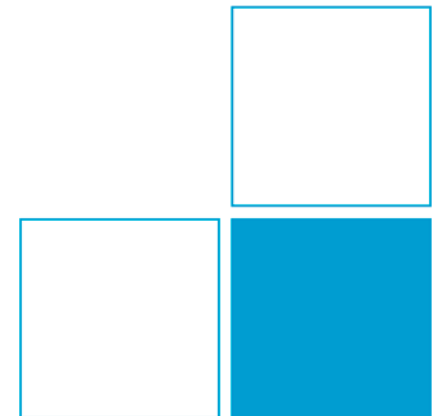


# Freie Software zur Messunsicherheitsberechnung

Dirk Röske

Physikalisch-Technische Bundesanstalt, Braunschweig



## Darstellung der Einheit

Das Drehmoment ist nach der Definition

$$\vec{M} = \vec{r} \times \vec{F}$$

$$\text{skalar: } M = r \cdot F \cdot \sin(\vec{r}, \vec{F})$$

eine **abgeleitete** Größe und wird **primär** rückgeführt auf

- die nationalen Normale für Masse, Länge und Zeit.

Die Einheit  $\text{N} \cdot \text{m}$  löst sich auf zu  $\text{kg} \cdot \text{m}^2 \cdot \text{s}^{-2}$  und folgt aus der Definition der Einheit der Kraft, des Newton:

$$\vec{F} = m \cdot \vec{a} = m \cdot \frac{d\vec{v}}{dt} = m \cdot \frac{d^2\vec{r}}{dt^2}$$

$$\text{skalar: } F = m \cdot a$$

$$\text{mit der Einheit: } [F] = [m] \cdot [a] = \text{kg} \cdot \text{m} \cdot \text{s}^{-2}$$

$$\Rightarrow \text{Einheit des Momentes: } [M] = [r] \cdot [F] = \text{kg} \cdot \text{m}^2 \cdot \text{s}^{-2} = \text{kg m}^2 \text{ s}^{-2}$$

## Rückführung

In der Praxis wird für den Abstand  $r$  die **Hebellänge**  $l$  eines Hebels und für die Beschleunigung  $a$  der Masse  $m$  die **lokale Fallbeschleunigung**  $g$  im Schwerfeld der Erde verwendet, so dass sich unter Berücksichtigung des Auftriebs in der Luft Folgendes ergibt:

$$M = l \cdot m \cdot g \cdot \left( 1 - \frac{\rho_{\text{Luft}}}{\rho_m} \right) \cdot \sin(\vec{l}, \vec{g})$$

Die Größen  $\rho_{\text{Luft}}$  und  $\rho_m$  sind – entsprechend – die Dichten der Luft und des Materials der Belastungsmassen.

Für eine Direktbelastungseinrichtung müssen diese Größen genau bestimmt („kalibriert“) werden. Zusätzlich sind zusätzliche Einflussgrößen wie

- Lagerreibung sowie weitere
- Störkräfte und –momente

zu berücksichtigen.

# Modellfunktion und Eingangsgrößen

Bestimmung der Messunsicherheit einer Drehmoment-Normalmesseinrichtung

Modell:

$$M = m \cdot g \cdot l \cdot \left(1 - \frac{\rho_{\text{air}}}{\rho_{\text{m}}}\right) \cdot \cos(\alpha) + \sum_{i=1}^n \Delta M_i$$

$M$  – das dargestellte Drehmoment

$m$  – Gesamtmasse der aktiven Gewichte

$g$  – lokale Fallbeschleunigung

$l$  – Hebellänge

$\rho_{\text{air}}$  – Dichte der umgebenden Luft

$\rho_{\text{m}}$  – Dichte des Materials der Belastungsmassen

$\alpha$  – Neigungswinkel des Hebels

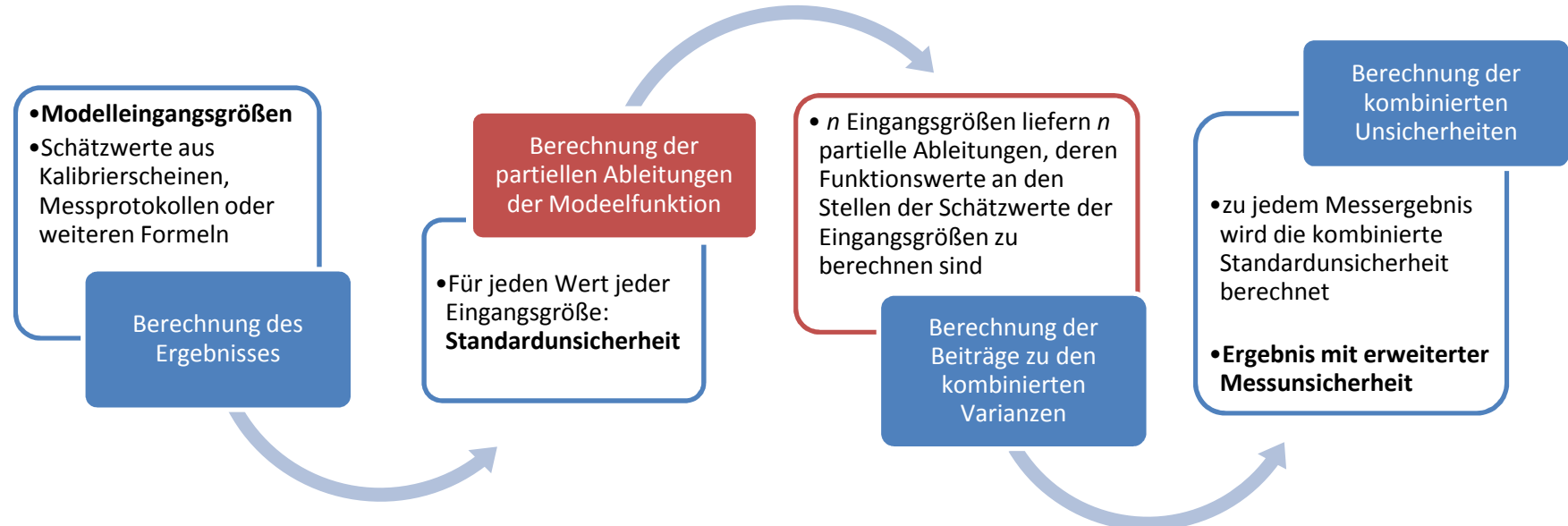
$\Delta M_i$  – zusätzliche Einflussgrößen

## Berechnung in Excel

MS Excel:

$$\begin{aligned}
 &= B2*B3*(1-B4/B5)*COS(B6) \\
 &= B1*B3*(1-B4/B5)*COS(B6) \\
 &= B1*B2*(1-B4/B5)*COS(B6) \\
 &= -B1*B2*B3/B5*COS(B6) \\
 &= B1*B2*B3*B4/B5^2*COS(B6) \\
 &= \dots
 \end{aligned}$$

$$=2*WURZEL(K4+L4+\dots R4+S4)$$



$$\begin{aligned}
 &= B1*B2*B3*(1-B4/B5)*COS(B6) \\
 &= m*g*I*(1-\rho_A/\rho_m)*COS(\alpha)
 \end{aligned}$$

$$\begin{aligned}
 K4 &= (B2*B3*(1-B4/B5)*COS(B6)*C1)^2 \\
 L4 &= (B1*B3*(1-B4/B5)*COS(B6)*C2)^2 \\
 &\dots
 \end{aligned}$$

# Allgemeiner Fall – Anwendung des GUM

GUM:

$$M = m \cdot g \cdot l \cdot \left(1 - \frac{\rho_{\text{air}}}{\rho_m}\right) \cdot \cos(\alpha) + \sum_{i=1}^n \Delta M_i$$

$$M = f(m, g, l, \rho_{\text{air}}, \rho_m, \alpha, \Delta M_i)$$

$$u^2(M) = \sum_{j=1}^N \left( \frac{\partial f}{\partial x_j} \right)^2 \cdot u^2(x_j)$$



Empfindlichkeitskoeffizient

Ein Computer-Algebrasystem (CAS) wie Maxima kann hier helfen.

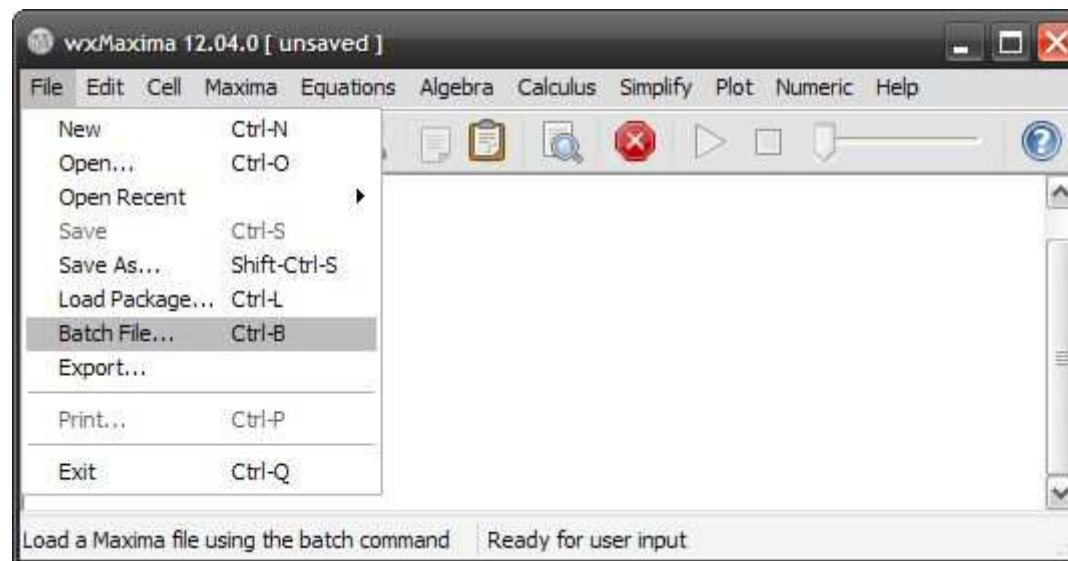
Ergebnis:

$$M \pm k \cdot u(M), \text{ mit } k = 2$$

# Maxima

Maxima und wxMaxima (als grafische Bedienoberfläche) können aus dem Internet geladen werden:

<http://maxima.sourceforge.net/>



# Arbeit mit Maxima

## Grundlegende Kommandos in Maxima

### Operation | Notation

### Beispiel

Zuweisung eines Wertes zu einer Variablen (g0) | :  
g0: 9.812524

Zuweisung einer Liste von Werten zu einer Variablen (m0List) | : [..., ...]  
m0List: [0.00040773594, 0.004077139, 0.040770601]

Definition einer Function | define(f(x1, ..., xn), expression)  
define(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), m\*g\*l\*(1-rhoA/rhoM)\*cos(alpha) + mR + mA + mF)

Definition einer Liste (MD0List) | : makelist(expression, n, n0, nmax)  
MD0List : makelist(), generates an empty list

Hinzufügen eines Wertes zu einer Liste (MD0List) | append(listname, expression)  
MD0List : append(MD0List, [MD0])

Zeichnen eines 2d Diagramms | plot2d (...)  
plot2d ([discrete, MD0List, yList],[style, points])



# Maxima

## Eingabedatei

```

1 kill(all) $ load("basic") $
2
3 m0List : [0.00040773594,0.004077139,0.040770601,0.4077028]$
4 g0      : 9.812524$
5 rhoA0   : 1.2$
6 rhoM0   : 7975$
7 l0      : 0.25$
8 alpha0  : 0$
9 mR0     : 0$
10 mA0    : 0$
11 mF0    : 0$
12
13 umList  : [0.00000001,0.00000002,0.00000003,0.00000027];
14 ug      : 0.000005$
15 urhoA   : 0.0462$
16 urhoM   : 6.09$
17 ul      : 0.000005$
18 ualpha  : 0.000714$
19 umR     : 0.00000029$
20 umA     : 0.00000029$
21 umF     : 0.00000029$
22
23 define(MD      (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), m*g*l*(1-rhoA/rhoM)*cos(alpha) + mR + mA + mF)$
24 define(dMD_m   (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), m))$
25 define(dMD_g   (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), g))$
26 define(dMD_l   (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), l))$
27 define(dMD_rhoA(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), rhoA))$
28 define(dMD_rhoM(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), rhoM))$
29 define(dMD_alpha(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), alpha))$
30 define(dMD_mR  (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), mR))$
31 define(dMD_mA  (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), mA))$
32 define(dMD_mF  (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), mF))$
33
34 MD0List : makelist() $ yList : makelist()$
35
36 for i: 0 thru 3 do (
37   m0 : pop(m0List),
38   um : pop(umList),
39
40   MD0      : MD      (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0),
41
42   cMD_m    : dMD_m   (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *um,
43   cMD_g    : dMD_g   (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *ug,
44   cMD_l    : dMD_l   (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *ul,
45   cMD_rhoA : dMD_rhoA(m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *urhoA,
46   cMD_rhoM : dMD_rhoM(m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *urhoM,
47   cMD_alpha : dMD_alpha(m0, g0, l0, rhoA0, rhoM0, ualpha, mR0, mA0, mF0) *ualpha,
48   cMD_mR   : dMD_mR  (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *umR,
49   cMD_mA   : dMD_mA  (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *umA,
50   cMD_mF   : dMD_mF  (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *umF,
51
52   W      : 2*sqrt(cMD_m^2+cMD_g^2+cMD_l^2+cMD_rhoA^2+cMD_rhoM^2+cMD_alpha^2+cMD_mR^2+cMD_mA^2+cMD_mF^2)/MD0,
53   MD0List : append(MD0List, [MD0]),
54   yList   : append(yList, [W])
55 );
56
57 MD0List; yList;
58 load(draw)$
59 plot2d ([discrete, MD0List, yList],[style, points],[xlabel, "Torque in N*m"],[ylabel, "Rel. exp. unc."],[logx],[logy])$

```

Maxima  
Eingabedatei  
mit  
Textblöcken

```

1 kill(all) $ load("basic") $
2
3 m0List : [0.00040773594,0.004077139,0.040770601,0.4077028]$
4 g0 : 9.812524$
5 rhoA0 : 1.2$
6 rhoM0 : 7975$
7 l0 : 0.25$
8 alpha0 : 0$
9 mR0 : 0$
10 mA0 : 0$
11 mF0 : 0$
12
13 umList : [0.00000001,0.00000002,0.00000003,0.00000027];
14 ug : 0.000005$
15 urhoA : 0.0462$
16 urhoM : 6.09$
17 ul : 0.000005$
18 ualpha : 0.000714$
19 umR : 0.00000029$
20 umA : 0.00000029$
21 umF : 0.00000029$
22
23 define MD (m, g, l, rhoA, rhoM, alpha, mR, mA, mF),
24 define dMD_m (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), m)$
25 define dMD_g (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), g)$
26 define dMD_l (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), l)$
27 define dMD_rhoA (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), rhoA)$
28 define dMD_rhoM (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), rhoM)$
29 define dMD_alpha (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), alpha)$
30 define dMD_mR (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), mR)$
31 define dMD_mA (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), mA)$
32 define dMD_mF (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), mF)$
33
34 MD0List : makelist() $ yList : makelist()$
35
36 for i: 0 thru 3 do (
37 m0 : pop(m0List),
38 um : pop(umList),
39
40 MD0 : MD m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0 ,
41
42 cMD_m : dMD_m m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0 +um,
43 cMD_g : dMD_g m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0 +ug,
44 cMD_l : dMD_l m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0 +ul,
45 cMD_rhoA : dMD_rhoA m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0 +urhoA,
46 cMD_rhoM : dMD_rhoM m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0 +urhoM,
47 cMD_alpha : dMD_alpha m0, g0, l0, rhoA0, rhoM0, ualpha, mR0, mA0, mF0 +ualpha,
48 cMD_mR : dMD_mR m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0 +umR,
49 cMD_mA : dMD_mA m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0 +umA,
50 cMD_mF : dMD_mF m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0 +umF,
51
52 W : 2*sqrt(cMD_m^2+cMD_g^2+cMD_l^2+cMD_rhoA^2+cMD_rhoM^2+cMD_alpha^2+cMD_mR^2+cMD_mA^2+cMD_mF^2)/MD0,
53 MD0List : append(MD0List, [MD0]),
54 yList : append(yList, [W])
55 );
56
57 MD0List; yList;
58 load(draw)$
59 plot2d ([discrete, MD0List, yList],[style, points],[xlabel, "Torque in N*m"],[ylabel, "Rel. exp. unc."],[logx],[logy])$

```

## Veränderliche Eingabdaten

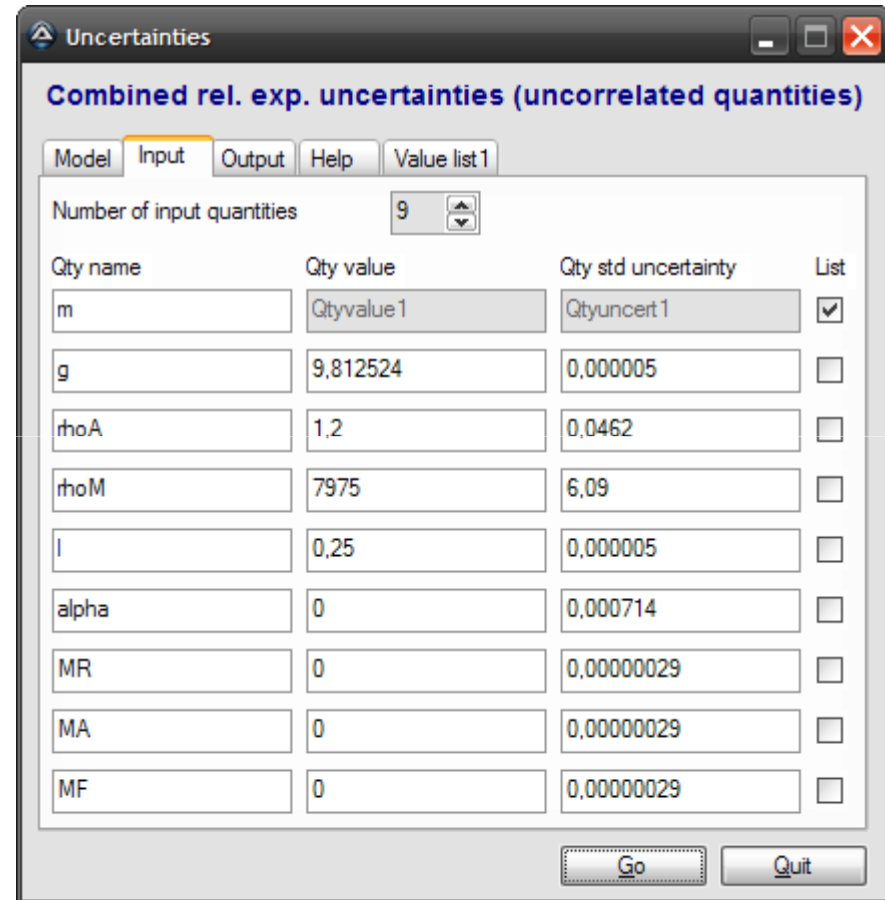
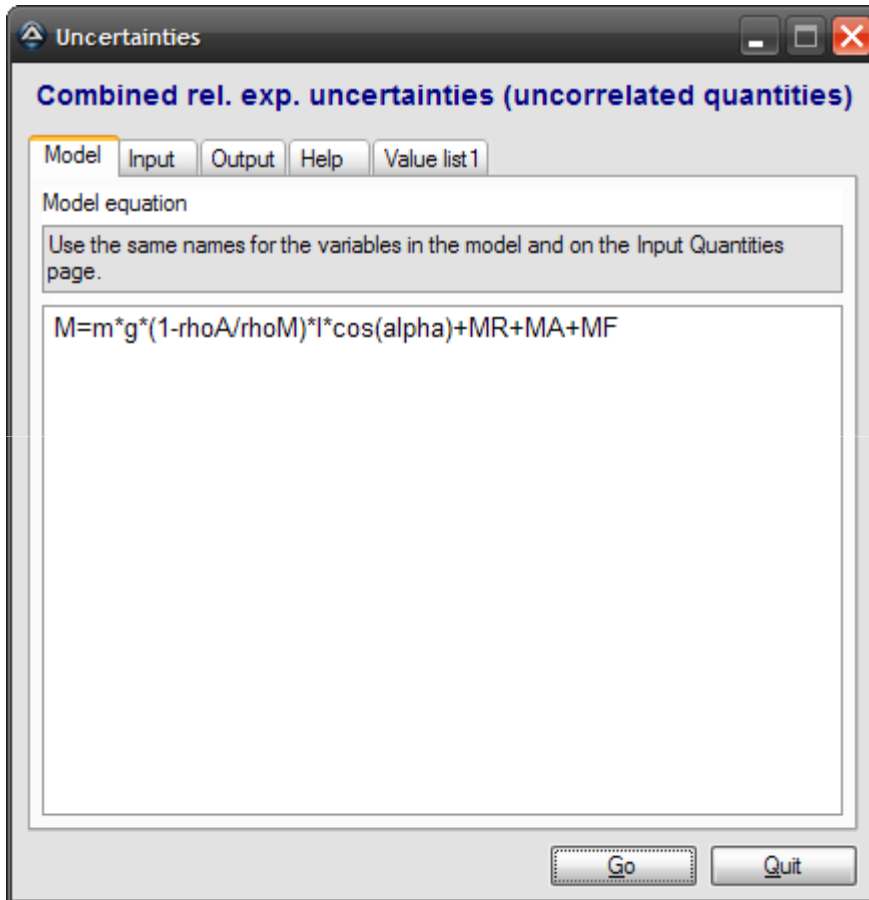
```

1 kill(all) $ load("basic") $
2
3 m0List : [0.00040773594,0.004077139,0.040770601,0.4077028]$
4 g0     : 9.812524$
5 rhoA0  : 1.2$
6 rhoM0  : 7975$
7 l0     : 0.25$
8 alpha0 : 0$
9 mR0    : 0$
10 mA0   : 0$
11 mF0   : 0$
12
13 umList : [0.00000001,0.00000002,0.00000003,0.00000027];
14 ug     : 0.000005$
15 urhoA  : 0.0462$
16 urhoM  : 6.09$
17 ul     : 0.000005$
18 ualpha : 0.000714$
19 umR    : 0.00000029$
20 umA    : 0.00000029$
21 umF    : 0.00000029$
22
23 define(MD (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), m*g*l*(1-rhoA/rhoM)*cos(alpha) + mR + mA + mF)$
24 define(dMD_m (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), m))$
25 define(dMD_g (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), g))$
26 define(dMD_l (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), l))$
27 define(dMD_rhoA (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), rhoA))$
28 define(dMD_rhoM (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), rhoM))$
29 define(dMD_alpha (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), alpha))$
30 define(dMD_mR (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), mR))$
31 define(dMD_mA (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), mA))$
32 define(dMD_mF (m, g, l, rhoA, rhoM, alpha, mR, mA, mF), diff(MD(m, g, l, rhoA, rhoM, alpha, mR, mA, mF), mF))$
33
34 MD0List : makelist() $ yList : makelist()$
35
36 for i: 0 thru 3 do (
37   m0 : pop(m0List),
38   um : pop(umList),
39
40   MD0 : MD (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0),
41
42   cMD_m : dMD_m (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *um,
43   cMD_g : dMD_g (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *ug,
44   cMD_l : dMD_l (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *ul,
45   cMD_rhoA : dMD_rhoA (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *urhoA,
46   cMD_rhoM : dMD_rhoM (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *urhoM,
47   cMD_alpha : dMD_alpha (m0, g0, l0, rhoA0, rhoM0, ualpha, mR0, mA0, mF0) *ualpha,
48   cMD_mR : dMD_mR (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *umR,
49   cMD_mA : dMD_mA (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *umA,
50   cMD_mF : dMD_mF (m0, g0, l0, rhoA0, rhoM0, alpha0, mR0, mA0, mF0) *umF,
51
52   W : 2*sqrt(cMD_m^2+cMD_g^2+cMD_l^2+cMD_rhoA^2+cMD_rhoM^2+cMD_alpha^2+cMD_mR^2+cMD_mA^2+cMD_mF^2)/MD0,
53   MD0List : append(MD0List, [MD0]),
54   yList : append(yList, [W])
55 );
56
57 MD0List; yList;
58 load(draw)$
59 plot2d ([discrete, MD0List, yList],[style, points],[xlabel, "Torque in N*m"],[ylabel, "Rel. exp. unc."],[logx],[logy])$

```

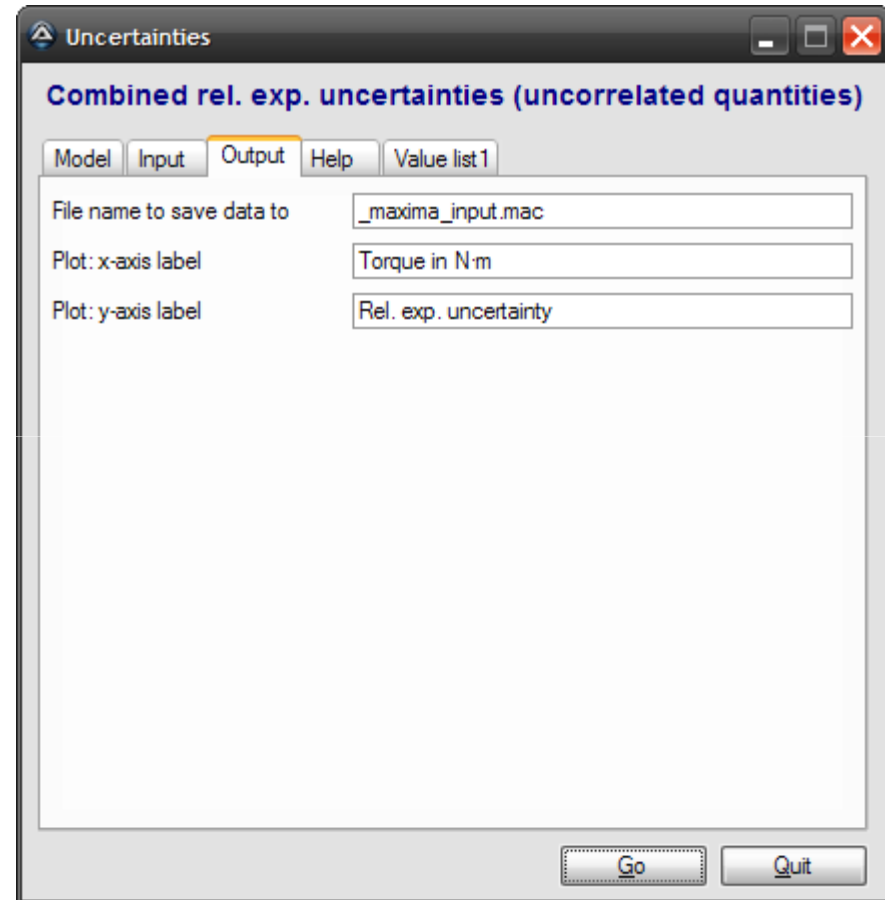
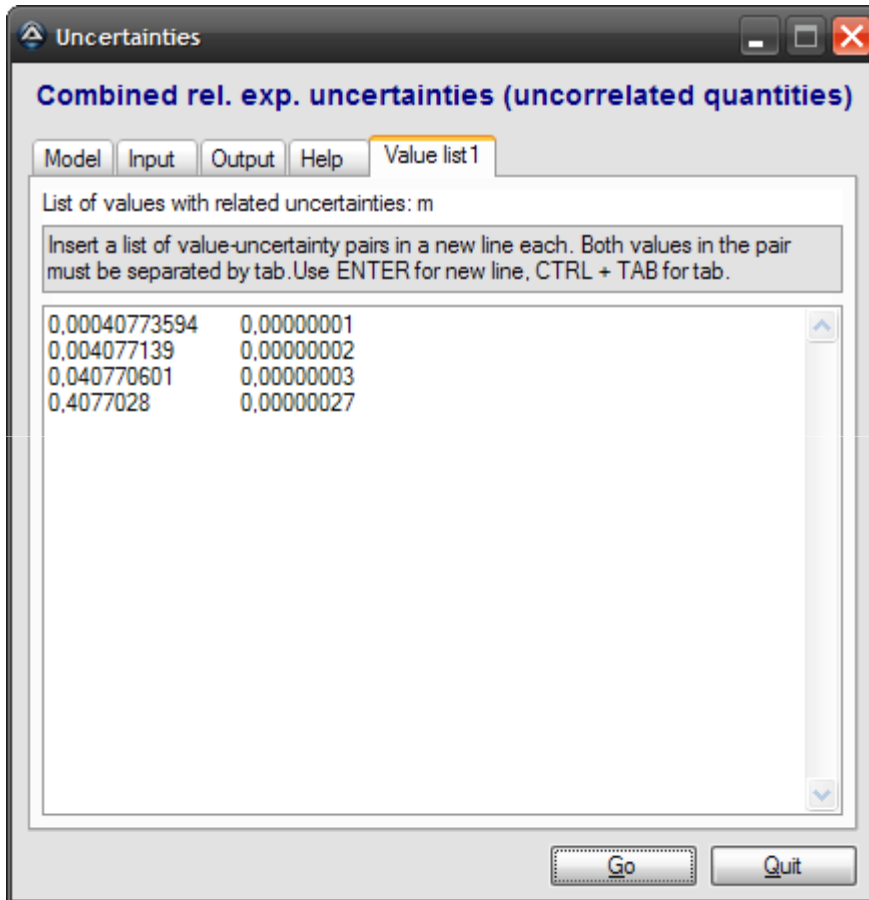
# Hilfsprogramm für Maxima

Mit AutoIt (Freeware) wurde ein Hilfsprogramm zur programmgesteuerten Erzeugung der Maxima-Eingabedatei erstellt.



# Hilfsprogramm für Maxima

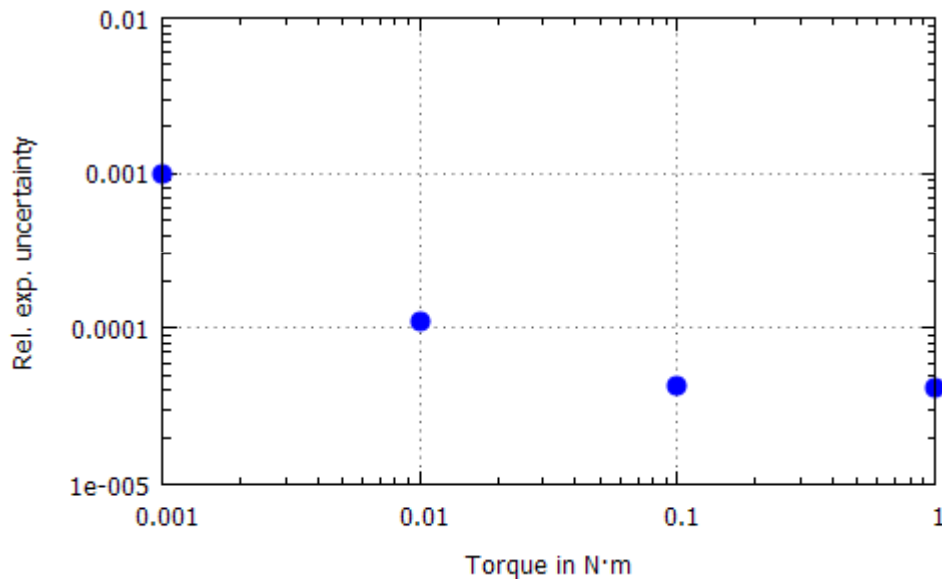
Eingabe der Modellfunktion, der Eingangsgrößen, der Werte der Eingangsgrößen samt beigeordneter Messunsicherheiten (auch als Liste).



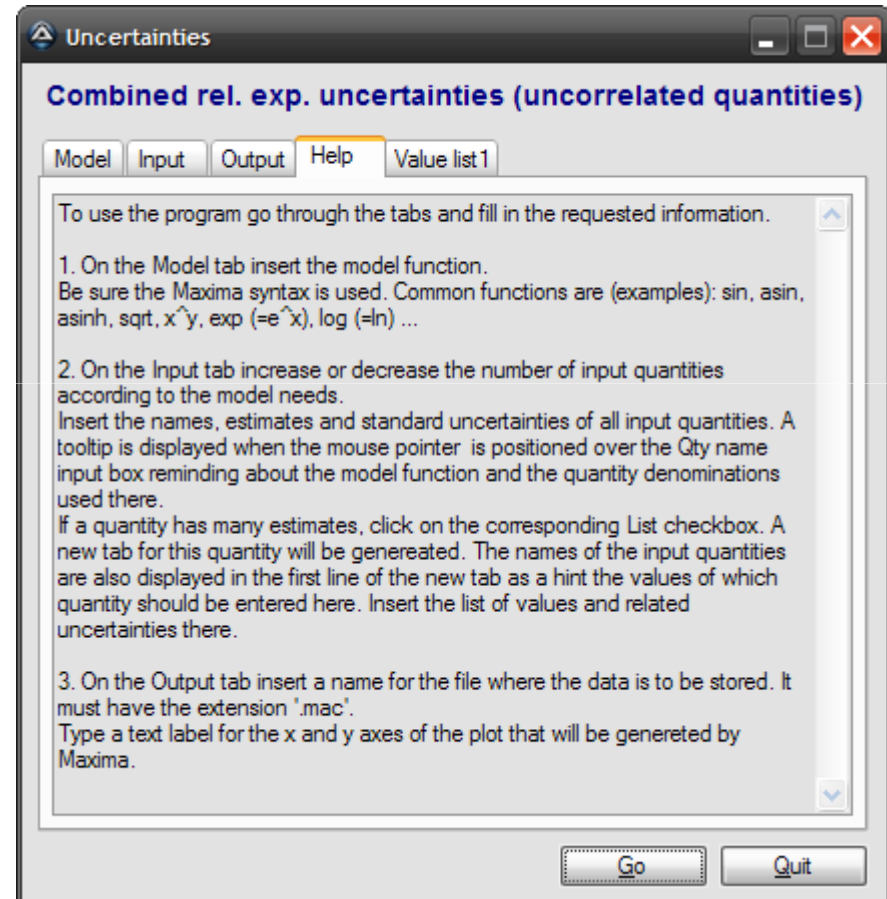
Weitere Angaben: Dateiname für Ausgabe, Achsenbeschriftungen ...

# Ergebnis von Maxima

Ausgabediagramm



Hilfefenster

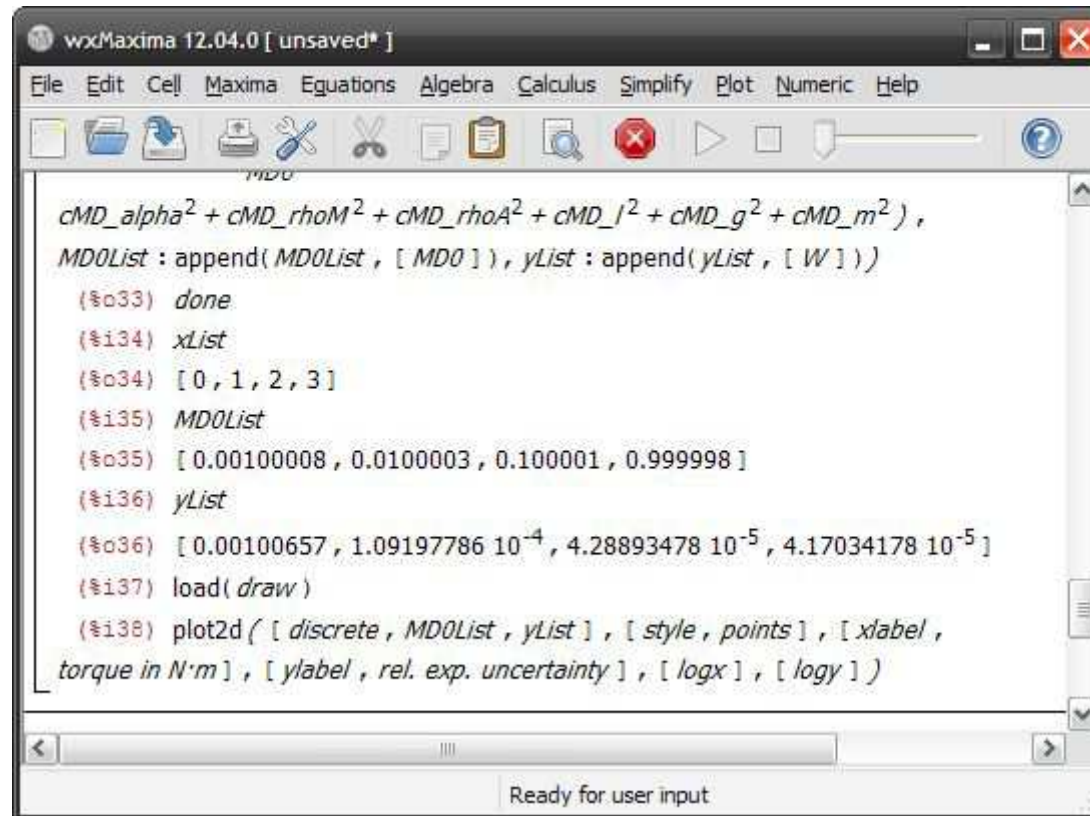


## Ergebnis von Maxima

Liste der Ergebnisse im Ausgabefenster von wxMaxima

MD0List: Drehmomente,

yList: zugeordnete erweiterte ( $k = 2$ ) relative Messunsicherheiten



```
wxMaxima 12.04.0 [unsaved* ]
File Edit Cell Maxima Equations Algebra Calculus Simplify Plot Numeric Help
cMD_alpha^2 + cMD_rhoM^2 + cMD_rhoA^2 + cMD_l^2 + cMD_g^2 + cMD_m^2),
MD0List : append( MD0List , [ MD0 ]), yList : append( yList , [ W ]))
(%o33) done
(%i34) xList
(%o34) [ 0, 1, 2, 3 ]
(%i35) MD0List
(%o35) [ 0.00100008, 0.0100003, 0.100001, 0.999998 ]
(%i36) yList
(%o36) [ 0.00100657, 1.09197786 10^-4, 4.28893478 10^-5, 4.17034178 10^-5 ]
(%i37) load( draw )
(%i38) plot2d( [ discrete, MD0List, yList ], [ style, points ], [ xlabel,
torque in N·m ], [ ylabel, rel. exp. uncertainty ], [ logx ], [ logy ] )
Ready for user input
```

# Zusammenfassung

## Maxima

- kann zur Berechnung von Messunsicherheiten eingesetzt werden,
- ist frei im Internet verfügbar,
- hilft insbesondere bei komplizierteren Modellfunktionen,
- kommt ohne Lizenzgebühren und ist damit bestens geeignet für den Austausch mit Kollegen (ohne Lizenzprobleme),
- kann eine ideale Ergänzung zur Überprüfung der auf anderen Wegen gewonnenen Ergebnisse sein und
- ist mit ein wenig Programmierkenntnis:
  - an eigene Anforderungen anpassbar und dadurch
  - flexibel im Einsatz.

Die hier vorgestellte Erweiterung (programmiert mit der ebenfalls freien Software AutoIt für Windows) macht die Arbeit mit Maxima intuitiv und einfach. Sie deckt bei Weitem nicht alle Möglichkeiten ab, sondern sie soll eher eine eng begrenzte Aufgabe – an die sie flexibel angepasst werden kann – schnell und effektiv erledigen.



# Nachtrag

Hinweis von Frau Kathrin Günther,

Senatsverwaltung für Stadtentwicklung und Umwelt  
Abteilung integrativer Umweltschutz  
Strahlenmessstelle des Landes Berlin

Rubensstraße 111 / 12157 Berlin  
Tel. 030 90166 424 / Fax 030 90166 444

Eine weitere freie Software gibt es für die Berechnung der Unsicherheit und der Nachweisgrenze. Das Programm soll als Hilfe zur Bestimmung der charakteristischen Grenzen bei Messungen mit ionisierender Strahlung dienen (DIN ISO 11929). Die Auswertung der Messungen und die Behandlung der Messunsicherheiten werden nach dem Verfahren gemäß GUM durchgeführt.

Das Programm kann auch in anderen Bereichen hilfreich sein und verfügt über eine Beispielsammlung und eine Hilfe.

Link:

<http://www.ti.bund.de/de/startseite/institute/fi/forschungsbereiche/leitstelle-umweltradioaktivitaet/uncertradio.html>

**Vielen Dank  
für Ihr Interesse  
an dieser Präsentation.**

	<b>Physikalisch-Technische Bundesanstalt Braunschweig und Berlin</b> Bundesallee 100 38116 Braunschweig	
	Dr. Dirk Röske Arbeitsgruppe 1.22 „Darstellung Drehmoment“	
		Telefon: 0531 592-1131 E-Mail: <a href="mailto:dirk.roeske@ptb.de">dirk.roeske@ptb.de</a> Web: <a href="http://www.ptb.de">www.ptb.de</a>