In [40]:
```python
import math
```

In [41]:
```python
import numpy as np
```

In [42]:
```python
from metas_unclib import *
```

In [44]:
```python
from Metas.UncLib.LinProp import UncBudget
```

In [ ]:
```python
import pandas as pd

import itables

itables.init_notebook_mode(all_interactive=True)
```

In [47]:
```python
def unc_budget(unc_item, show_table=True):
    tree = UncBudget.ComputeTreeUncBudget(unc_item.net_object)

    table = pd.DataFrame(
        columns=("description", "uncertainty component", "uncertain
        index=range(len(tree)+1)
    )

    for i, elem in enumerate(tree):
        #print(element)
        table.loc[i] = (
            elem.get_Description(),
            elem.get_UncComponent(),
            elem.get_UncPercentage(),
        )
    table.loc[len(tree)] = (
        "SUMMARY",
        unc_item.stdunc,
        100.,
    )

    return table.sort_values("uncertainty percentage", ascending=Fa
```

In [4]:
```python
def tolerance(value, a):
    """
    producer tolerance of value +/- a

    returns UniformDistribution(value - a, value + a)
    """
    return UniformDistribution(value - a, value + a)
```

$$\Delta_{sol} H_B = -UIt \frac{M_B}{m_B} \frac{\Delta T_{sol}}{\Delta T_C} = -UI \frac{M_B}{m_B} \frac{\Delta T_{sol}}{b}$$

B: Amonium Nitrate NH_4NO_3

In [5]:
```
U = ufloatfromdistribution(
  tolerance(10.0, 0.05),  # V
  desc='voltage / V'
)
I = ufloatfromdistribution(
  tolerance(1.61, 0.005),  # V
  desc='current / A'
)
P = U * I
P
```

Out[5]:
```
16.1 ± 0.0547121254080546
```

In [10]:
```
amm_M = ufloat(
    80.04,
    desc="molar mass ammonium nitrate NH4NO3 / g/mol",
)
```

In [20]:
```
amm_m = ufloatfromsamples(
    (0.600, 0.595, 0.601),
    desc="absolute mass of ammonium nitrate NH4NO3 / g",
)
amm_m
```

Out[20]:
```
0.5986666666666666 ± 0.0040742511466699575
```

In [32]:
```
amm_T_delta = ufloatfromsamples(
    (-0.3815051, -0.3662502, -0.3717871),
    desc="delta T of solution / K",
)
amm_T_delta
```

Out[32]:
```
-0.37318080000000003 ± 0.009787627577074718
```

In [33]:
```
# Standardfehler der Regression wird ignoriert, da um über 1 Grösse
amm_b = ufloatfromsamples(
    (0.02780959, 0.02764169, 0.02755107),
    desc="heating rate / K/s",
)
amm_b
```

Out[33]:
```
0.02766745 ± 0.00016625143430306562
```

In [37]:
```
amm_H_sol = - U * I * amm_M / amm_m * amm_T_delta / amm_b  # J / mo
amm_H_sol / 1000  # kJ / mol
```

Out[37]:
```
29.03340925691424 ± 0.8118233607977087
```

In [48]:
```
unc_budget(amm_H_sol)
```

Out[48]:

	description	uncertainty component
5	SUMMARY	811.823361
2	delta T of solution / K	761.475931
0	absolute mass of ammonium nitrate NH4NO3 / g	197.588086
3	heating rate / K/s	174.459371
4	voltage / V	83.812233
1	current / A	52.057288

```
with old values: amm_T_delta = -0.3840371, -0.3723012, -0.3814346

amm_H_sol = 29506.18596234154 ± 671.6938464947295
```

In [ ]:
```
# reference value @25°C: 25.69 kJ/mol
# https://en.wikipedia.org/wiki/Enthalpy_change_of_solution#Depende

# Medvedev et al.
# @ 25°C
# https://pdf.sciencedirectassets.com/271405/1-s2.0-S0040603100X028
# 25.53 ± 0.24 J/mol (k=2)
```