In [1]:
```
import math

from metas_unclib import *

from Metas.UncLib.LinProp import UncBudget

import pandas as pd

import pandas as pd

import itables

itables.init_notebook_mode(all_interactive=True)
```

In [2]:
```
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):
        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 [3]:
```
def tolerance(value, a):
    """
    producer tolerance of value +/- a

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

In [ ]:
```

```

$$\Delta_m H_E^{sp} = c_W^{sp} \left( \theta_{1f} \frac{M_{EW}}{M_E} \frac{\Delta\theta_2}{\Delta\theta_1} - \theta_{2f} \right)$$

In [4]:
```
#water_c_sp = ufloat(4182, desc="water specific heat capacity / J/K
water_c_sp = 4182
```

In [5]:
```
water_m1 = np.array((955.78, 988.3, 1002.60))  # g
water_m2 = np.array((937.38, 970.8,  984.50))  # g
water_m = water_m1 - water_m2  # g

ice_m1 = np.array((20.65, 24.97, 26.04))  # g
ice_m2 = np.array(( 6.31,  7.26,  8.49))  # g
ice_m = ice_m1 - ice_m2

(water_m, ice_m)
```

Out[5]:
```
(array([18.4, 17.5, 18.1]), array([14.34, 17.71, 17.55]))
```

In [ ]:
```

```

In [6]:
```
ice_temp_final = np.array((8.898531, 6.263302, 6.412726))  # °C
ice_temp_delta = np.array((-10.48647, -12.13266, -11.59748))  # K

water_temp_final = np.array((17.54753, 17.22439, 16.99892))  # °C
water_temp_delta = np.array((-2.524441, -2.409830, -2.492528))  # K
```

In [ ]:
```

```

In [7]:
```
ice_H_sp = (water_temp_final * water_m / ice_m * ice_temp_delta / wa
ice_H_sp
```

Out[7]:
```
array([353926.84412912, 332164.77441224, 314320.21564004])
```

In [11]:
```
ufloatfromsamples(ice_H_sp) / 1000
```

Out[11]:
```
333.4706113937995 ± 25.140414947719453
```

In [9]:
```
"330 ± 50 kJ/kg"
```

Out[9]:
```
'330 ± 50 kJ/kg'
```

```
Quelle Wikipedia: 333,5 kJ/kg
```

In [10]:
```
ice_H_sp.std() / np.sqrt(2) * 4.3
```

Out[10]:
```
49243.92859011987
```