In [1]:
```
from IPython.display import display, Math
```

In [2]:
```
import math
```

In [3]:
```
import pandas as pd
```

```
/Users/Janjo/opt/anaconda3/lib/python3.8/site-packages/pandas/core/computation/expres
sions.py:20: UserWarning: Pandas requires version '2.7.3' or newer of 'numexpr' (vers
ion '2.7.1' currently installed).
  from pandas.core.computation.check import NUMEXPR_INSTALLED
```

In [4]:
```
# METAS UncLib
from metas_unclib import *  # import METAS UncLib
use_linprop()               # linear unc propagation
```

In [5]:
```
from Metas.UncLib.LinProp import UncBudget as _LinPropUncBudget
```

In [6]:
```
# unc_budget bug fix

def unc_budget(unc_item):
    tree = _LinPropUncBudget.ComputeTreeUncBudget(unc_item.net_object)

    data = np.zeros((len(tree) + 1, 2))
    desc = [""]*(len(tree) + 1)

    desc[0] = "SUMMARY"
    data[0] = (
        unc_item.stdunc,
        100.,
    )

    for i, elem in enumerate(tree):
        data[i + 1] = (
            elem.get_UncComponent(),
            elem.get_UncPercentage(),
        )
        desc[i + 1] = elem.get_Description()

    try:
        # render pandas.DataFrame
        import pandas as pd
        return pd.DataFrame(columns=("description", "unc component", "unc / %"), data
            "description": desc,
            "unc component": data[:,0],
            "unc / %": data[:,1],
        }).sort_values("unc / %", ascending=False)

    # fallback if pandas not installed
    except ModuleNotFoundError:
        print("unc component | unc / %    | label")
        print("--------------|------------|--------------")
        for label, (component, percent) in zip(desc, data):
            print("{:.4E}    | {:.4E} | {}".format(component, percent, label))
    return data, desc
```

Formelübersicht

$$\Delta_V H_{\text{subst}} = \Delta_V H_{\text{ref}} \frac{\rho_{\text{ref}}}{\rho_{\text{subst}}} \frac{M_{\text{subst}}}{M_{\text{ref}}} \frac{A_{\text{subst}}}{A_{\text{ref}}}$$

$$\int \Delta T dt = A \propto Q = n \Delta_V H$$

Q: Verdunstungswärme

n: Stoffmenge

M: Molmasse

ρ: spezifische Dichte

$\Delta_V H$: molare Verdampfungsenthalpie

A: Peak-Fläche

Funktionen

In [7]:
```
def calc_h(H_ref, rho_ref, rho_subst, M_ref, M_subst, A_ref, A_subst):
    return (
        H_ref * (rho_ref / rho_subst)
        * (M_subst / M_ref)
        * (A_subst / A_ref)
    )
```

In [8]:
```
def get_areas(path):
    return list(pd.read_csv(path, sep=" ").iloc[0])
```

Measured values and constants

In [9]:
```
# rho values literature [kg/dm3]

rho_lit_m = ufloat(0.7914, desc="rho methanol (literature)")
rho_lit_a = ufloat(0.7899, desc="rho acetone (literature)")
rho_lit_n = ufloat(0.6603, desc="rho n-hexane (n-hexane)")
```

In [10]:
```
# uncertainties of devices
u_m = 0.002  # [g]
u_V = 0.08  # [mL]
```

In [11]:
```
# density acetone measured in exercise 3
m_a = ufloat(19.76, u_m, desc="mass acetone")  # [g]
V_a = ufloat(25., u_V, desc="volume acetone")  # [mL]
rho_a = m_a / V_a  # [kg/dm3]
rho_a
```

Out[11]: 0.7904000000000001 ± 0.0025305448659132686

In [12]:
```
# acetone calculated with method of the Meister book
1/V_a.value * u_m + abs(-m_a.value / V_a.value**2) * u_V  # [kg/dm3]
```

Out[12]: 0.0026092800000000007

In [13]:
```
# density n-hexane measured in exercise 3
m_n = ufloat(16.45, u_m, desc="mass n-hexane")  # [g]
V_n = ufloat(25., u_V, desc="volume n-hexane")  # [mL]
rho_n = m_n / V_n  # [kg/dm3]
rho_n
```

Out[13]: 0.6579999999999999 ± 0.002107119208777709

In [14]:
```
# n-hexane calculated with method of the Meister book
1/V_n.value * u_m + abs(-m_n.value / V_n.value**2) * u_V  # [kg/dm3]
```

Out[14]: 0.0021856

In [15]:
```
# Molar masses [kg/mol]
M_m = ufloat(32.04, desc="M methanol")
M_a = ufloat(58.08, desc="M acetone")
M_n = ufloat(86.18, desc="M acetone")
```

In [16]:
```
# evaporation enthalpy
H_m = ufloat(37.43, 0.17, desc="H methanol")
H_m
```

Out[16]:
```
37.43 ± 0.17
```

Reference: Methanol

In [17]:
```
# read in measured areas
met_samples = get_areas("../ddr2_exports/methanol_areas.csv")
met_samples
```

Out[17]:
```
[-28.655065, -29.9144020000002, -29.660473]
```

In [18]:
```
# transform to uncertainty object
A_met = ufloatfromsamples(met_samples, desc="are methanol")
A_met
```

Out[18]:
```
-29.409980000000065 ± 0.8441024984395747
```

Acetone

In [19]:
```
# read in measured areas
ac_samples = get_areas("../ddr2_exports/acetone_areas.csv")
ac_samples
```

Out[19]:
```
[-12.963891, -12.7988540000001, -13.5183319999999]
```

In [20]:
```
# transform to uncertainty object
A_ac = ufloatfromsamples(ac_samples, desc="area acetone")
A_ac
```

Out[20]:
```
-13.093692333333335 ± 0.4776896236957493
```

In [21]:
```
# calculate evaporation enthalpy of acetone
H_ac = calc_h(
    H_ref=H_m,
    rho_ref=rho_lit_m,
    rho_subst=rho_a,
    #rho_subst=rho_lit_a,
    M_subst=M_a,
    M_ref=M_m,
    A_ref=A_met,
    A_subst=A_ac,
)  # [kJ/mol]
H_ac
```

Out[21]:
```
30.24617388090784 ± 1.4140226667388984
```

In [22]:
```
H_ac.stdunc
```

Out[22]:
```
1.4140226667388984
```

In [23]:
```
H_ac_result = r"\Delta_VH_{{\text{{acetone}}}} = {:.1f} \pm{:.1f} \text{{ kJ/mol }}(9
display(Math(H_ac_result))
```

$$\Delta_V H_{\text{acetone}} = 30.2 \pm 2.8 \text{ kJ/mol } (95\ \%)$$

In [24]:
```
print(H_ac_result)
```

```
\Delta_VH_{\text{acetone}} = 30.2 \pm2.8 \text{ kJ/mol }(95 \space \%)
```

n-Hexane

In [25]:
```
# read in measured areas
nhex_samples = get_areas("../ddr2_exports/n-hexane_areas.csv")
nhex_samples
```

Out[25]:
```
[-7.7363979999999, -7.77194099999995, -7.79426499999992]
```

In [26]:
```
# transform to uncertainty object
A_nhex = ufloatfromsamples(nhex_samples, desc="area n-hexane")
A_nhex
```

Out[26]:
```
-7.767534666666591 ± 0.036989056992450586
```

In [27]:
```
# calculate evaporation enthalpy of n-hexane
H_nhex = calc_h(
    H_ref=H_m,
    rho_ref=rho_lit_m,
    rho_subst=rho_n,
    #rho_subst=rho_lit_n,
    M_subst=M_n,
    M_ref=M_m,
    A_ref=A_met,
    A_subst=A_nhex,
)  # [kJ/mol]
H_nhex
```

Out[27]:
```
31.981028970326822 ± 0.9472648790759943
```

In [28]:
```
H_nhex_result = r"\Delta_VH_{{\text{{n-hexane}}}} = {:.1f} \pm{:.1f} \text{{ kJ/mol }
display(Math(H_nhex_result))
```

$$\Delta_V H_{\text{n-hexane}} = 32.0 \pm 1.9 \,\text{kJ/mol}\ (95\ \%)$$

In [29]:
```
print(H_nhex_result)
```

```
\Delta_VH_{\text{n-hexane}} = 32.0 \pm1.9 \text{ kJ/mol }(95 \space \%)
```

In [30]:
```
unc_budget(H_nhex)
```

Out[30]:

	description	unc component	unc / %
0	SUMMARY	0.947265	100.000000
1	are methanol	0.917895	93.895095
2	area n-hexane	0.152294	2.584771
3	H methanol	0.145252	2.351257
5	volume n-hexane	0.102339	1.167191
4	mass n-hexane	0.003888	0.001685