Setup folder structure for simulation

In [ ]:
```
#import MILK env
import MILK
config = MILK.load_json('milk.json')
```

In [ ]:
```
#create folders based on number of MAUD instances and milk.json preferences
from pathlib import Path
import shutil

NCPU = config["compute"]["n_maud"]
folder_names=[]
for i in range(0,NCPU):
    folder_names.append(config["folders"]["run_dirs"].replace("(wild)",f"{i}
    if not Path(folder_names[-1]).is_dir():
        shutil.copytree(src="base_files",dst=folder_names[-1])
```

Simulate different e.g. phase fractions

In [ ]:
```
# Setup MILK editor and text interface instances
editor = MILK.parameterEditor.editor()
editor.parseConfig(config)
maudText = MILK.maud.maudText()
maudText.parseConfig(config)
maudText.wild_range=[[0,NCPU-1]]
```

In [ ]:
```
# Generate random volume fractions to simulate
import numpy as np
import pandas as pd
from math import ceil

#sample parameters
NSAMPLES = NCPU * ceil(100 / NCPU)
NPHASE = 3

#do sampling
phase = np.zeros((NSAMPLES,NPHASE))
for i in range(0,NPHASE):
    phase[:,i]=np.random.uniform(0,1,NSAMPLES)

#normalize and add to dataframe
norm = np.sum(phase,axis=1)
df=pd.DataFrame()
for i in range(0,NPHASE):
    phase[:,i] = phase[:,i]/norm
    df[f"phase_frac_{i}"]=phase[:,i]
```

In [ ]:
```
#build the spectra file list
spectra_files = []
for folder_name in folder_names:
    spectra_files.append(Path(folder_name) / f"{config['ins']['output_summed
```

In [ ]:
```
from MILK.MAUDText.spectraReader import extract_spectra
from tqdm import tqdm

spectra = []
for i in tqdm(range(0,NSAMPLES,NCPU),total = NSAMPLES / NCPU):
    #setup the simulation runs
    for j in range(0,NCPU):
        for k in range(0,NPHASE):
            editor.set_val(key="_pd_phase_atom_",value=f"{df[f'phase_frac_{k

    #use maud to export the pattern
    #setting itr=0 causes maud to only compute the pattern.
    maudText.refinement(itr='0',inc_step=False,simple_call=True,ifile=editor

    #get data and append to list
    spectra = spectra + extract_spectra(files=spectra_files,poolsize=NCPU)
```

```
100%|██████████| 13/13.0 [01:03<00:00,  4.89s/it]
```

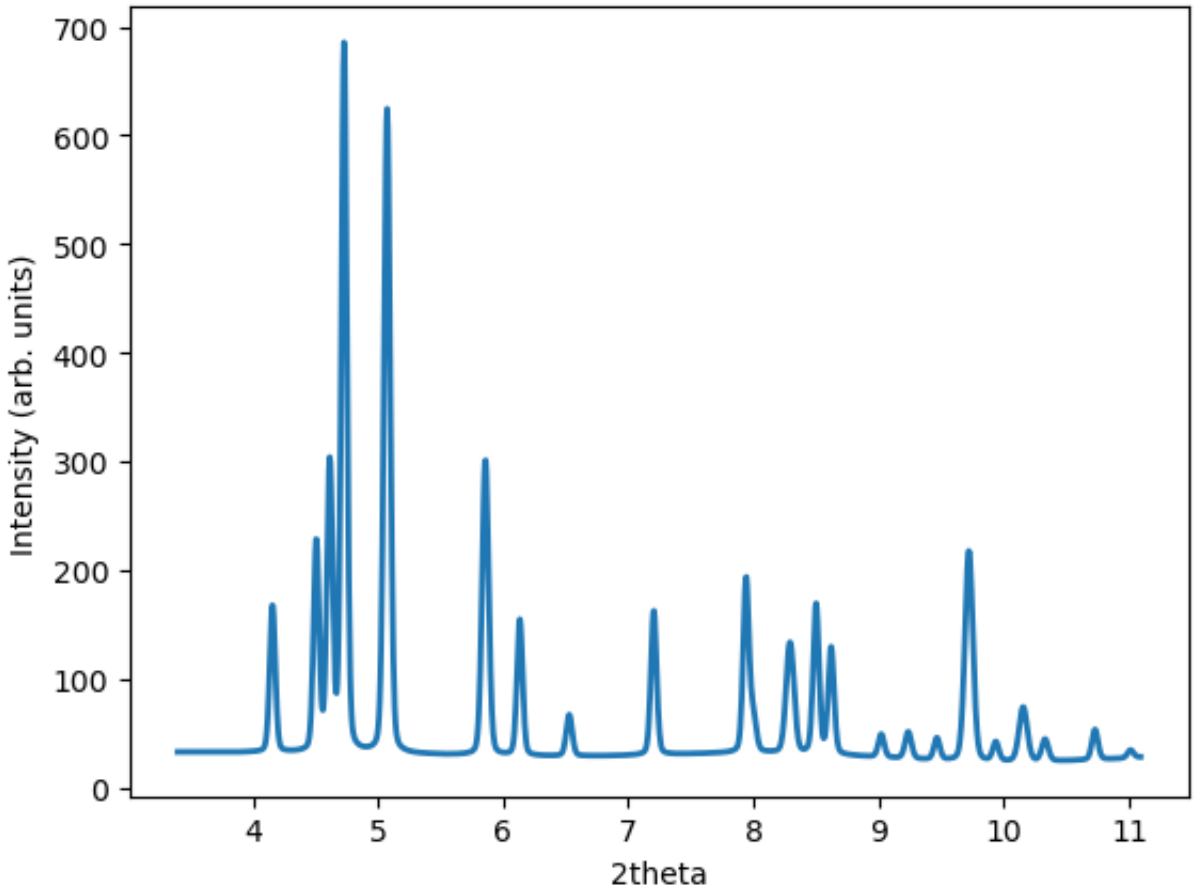
With 8 cores on a M1 mac we are simulating 1.65 patterns per second.

Visualize the simulation

In [ ]:
```
import matplotlib.pyplot as plt
def plot_spectra(spectra):
    """Takes a MAUD exported spectra and """
    y = spectra['calculated_intensity']
    x = spectra['2theta/d_coordinate']

    fig, ax = plt.subplots()
    ax.plot(x,y, linewidth=2.0)
    ax.set_ylabel("Intensity (arb. units)")
    ax.set_xlabel("2theta")
    plt.show()
```

In [ ]:
```
plot_spectra(spectra[0][0])
```



In [ ]:
```
plot_spectra(spectra[1][0])
```