1 Nanonis Metadata and Data Read

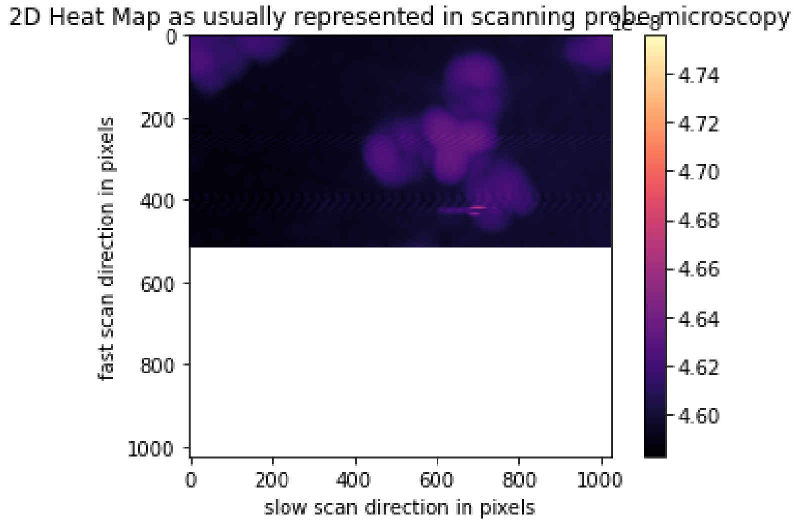
In [1]:

```
# Read Nanonis metadata dictionary #CAP 23.03.11

import os
import re
import itertools

######################

folder_path = "."
file_ext = ".sxm"
metadata_lines = 97 #need better metadata EOF

######################

def Convert(a): # change list to dictionary, keys and values not yet debugged
    it = iter(a)
    res_dct = dict(zip(it, it))
    return res_dct

a=[]
for file_name in os.listdir(folder_path):
    try:
        if file_name.endswith(file_ext):
                #with open(os.path.join(folder_path, file_name), "r", errors="ignore")
            with open(os.path.join(folder_path, file_name), "r", errors="ignore") as f:
                a = re.split('\n:|:\n', f.read())[:metadata_lines]
    except:
        print('An error occured or there is no file in the current directory with the e

#print(file_name)
print("*Keys*")
print((Convert(a)).keys())
print("*Values*")
print((Convert(a)).values())
```

```
*Keys*
dict_keys([':NANONIS_VERSION', 'SCANIT_TYPE', 'REC_DATE', 'REC_TIME', 'RE
C_TEMP', 'ACQ_TIME', 'SCAN_PIXELS', 'SCAN_FILE', 'SCAN_TIME', 'SCAN_RANG
E', 'SCAN_OFFSET', 'SCAN_ANGLE', 'SCAN_DIR', 'BIAS', 'Z-CONTROLLER', 'COM
MENT', 'Current>Current (A)', 'Current>Calibration (A/V)', 'Current>Offse
t (A)', 'Current>Gain', 'Z-Controller>Z (m)', 'Z-Controller>Controller na
me', 'Z-Controller>Controller status', 'Z-Controller>Setpoint', 'Z-Contro
ller>Setpoint unit', 'Z-Controller>P gain', 'Z-Controller>I gain', 'Z-Con
troller>Time const (s)', 'Z-Controller>TipLift (m)', 'Z-Controller>Switch
off delay (s)', 'Piezo Configuration>Active Calib.', 'Piezo Configuration
>Calib. X (m/V)', 'Piezo Configuration>Calib. Y (m/V)', 'Piezo Configurat
ion>Calib. Z (m/V)', 'Piezo Configuration>HV Gain X', 'Piezo Configuratio
n>HV Gain Y', 'Piezo Configuration>HV Gain Z', 'Piezo Configuration>Tilt
X (deg)', 'Piezo Configuration>Tilt Y (deg)', 'Piezo Configuration>Curvat
ure radius X (m)', 'Piezo Configuration>Curvature radius Y (m)', 'Piezo C
onfiguration>2nd order corr X (V/m^2)', 'Piezo Configuration>2nd order co
rr Y (V/m^2)', 'Piezo Configuration>Drift X (m/s)', 'Piezo Configuration>
Drift Y (m/s)', 'Piezo Configuration>Drift Z (m/s)', 'Piezo Configuration
>Drift correction status (on/off)', 'DATA_INFO'])
*Values*
dict_values(['2', '              FLOAT            MSBFIRST', ' 11.11.202
2', '14:58:57', '      290.0000000000', '      1066.2', '       1024
1024', 'C:\\Users\\SPM-PEEM\\Desktop\\DATA_Nanonis\\20220711_CreaTec_Serv
ice_Benchmarks_LHe\\Nanonis-Session-PMD100-HVHU_CreaTec_Service_PalmaLabB
erlin220711\\221110_TiSe2_tryanCN_5K0031.sxm', '             1.024E+0
1.024E+0', '           6.000000E-9           6.000000E-9', '
-2.944796E-7         -2.103954E-7', '            -9.000E+1', 'down', '
1.000E-1', '\tName\ton\tSetpoint\tP-gain\tI-gain\tT-const\n\tlog Current
\t1\t1.000E-12 A\t1.000E-11 m\t4.886E-8 m/s\t2.047E-4 s', 'TSTM=5.2K', '-
1.0308E-12', '100E-12', '35.0304E-15', 'Not switchable', '45.8914E-9', 'l
og Current', 'ON', '1E-12', 'A', '10E-12', '48.861E-9', '204.662E-6', '0E
+0', '0E+0', 'LHe', '3.8E-9', '3.8E-9', '900E-12', '14.5', '14.5', '14.
5', '-2.0396', '5.19673', 'Inf', 'Inf', '0E+0', '0E+0', '0E+0', '0E+0',
'0E+0', 'FALSE', '\tChannel\tName\tUnit\tDirection\tCalibration\tOffset\n
\t14\tZ\tm\tboth\t1.305E-8\t0.000E+0\n\t0\tCurrent\tA\tboth\t1.000E-10\t
3.503E-14\n\t20\tLI_Demod_2_X\tA\tboth\t1.000E-10\t0.000E+0\n\t22\tLI_Dem
od_1_X\tA\tboth\t1.000E-10\t0.000E+0\n'])
```

In [3]:

```
import nanonispy as nap #CAP 23.03.11
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
%matplotlib inline

scan = nap.read.Scan('test.sxm')


#dictionary of signals
scan.signals

#only one channel

data=scan.signals['Z']['forward']
np.shape(data) # number of pixels in x and y

plt.imshow( data , cmap = 'magma' )

# Adding details to the plot
plt.title( "2D Heat Map as usually represented in scanning probe microscopy" )
plt.xlabel('slow scan direction in pixels')
plt.ylabel('fast scan direction in pixels')

# Adding a color bar to the plot
plt.colorbar()

# Displaying the plot
plt.show()
```



# Other methods parsing methods for Nanonis data format (not implemented)

In [ ]:

```
import pySPM
print(pySPM.__version__)

import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
%matplotlib inline

import os
from IPython import display

filename = get_data("test.sxm")
S = pySPM.SXM(filename)
S.list_channels()
fig, ax = plt.subplots(1,2,figsize=(14,7))
S.get_channel('Z').show(ax=ax[0]);
p = S.get_channel('Current').show(ax=ax[1], cmap='viridis');
```

In [ ]:

```

import gwyfile
import scipy.ndimage.filters
# Load a Gwyddion file into memory

obj = gwyfile.load('./test.sxm')
#obj = gwyfile.load(f'{file_name}')

# Return a dictionary with the datafield titles as keys and the
# datafield objects as values.
channels = gwyfile.util.get_datafields(obj)

channels['Z (Forward)'].data # One data channel out of many available

channel = channels['Z (Forward)']
channel.xreal
# Datafield objects have a `data` property to access their
# two-dimensional data as numpy arrays.
data = channel.data

# Plot the data using matplotlib.
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
ax.imshow(data, interpolation='none', origin='upper',
        extent=(0, channel.xreal, 0, channel.yreal), cmap='hot')
plt.show()
```

In [ ]:

```

```