basic_usage

February 13, 2023

[ ]:
```
%load_ext autoreload
%autoreload 2
%reset
```

[ ]:
```
%matplotlib inline
```

0.1 Repeatability Test

Reading and resampling surface data from FEM analysis.

[ ]:
```
from pySurf.readers.instrumentReader import matrixdat_reader
import numpy as np
from pySurf.points import resample_grid, plot_points, level_points,␣
 ↪points_autoresample
import os
from pySurf.data2D_class import Data2D
from dataIO.span import span
from matplotlib import pyplot as plt
```

VC 2022/12/22 Test di ripetibilita' e sensibilita' usando specchio di test da 100 mm di Pecchioli con alleggerimento.

Tre fiducial marcati in corrispondenza di riferimento "ALTO" (N) e corrispettivi a 90 gradi destra e sx (E,W).

01, 02, 03, 04 misura ripetuta a breve distanza senza toccare niente.

05 stessa misura senza toccare niente.

06, 07, 08, 09, 10 rimisuro rimuovendo e rimettendo il campione ogni volta in modo che i fiducial corrispondano. Nota che ogni volta devo riannullare le frange. Dal 7 in poi viene lasciato stabilizzare due minuti dopo allineamento.

11 - 24 a partire da stessa posizione di 10 (11 senza spostamenti), ruoto leggermente in senso antiorario e riacquisisco dopo aver livellato frange, per acquisire reference. Aspetto un minuto prima di ogni acquisizione. N.B.: la base viene toccata per sbaglio prima di 14. 22 non ha richiesto di riallineare.

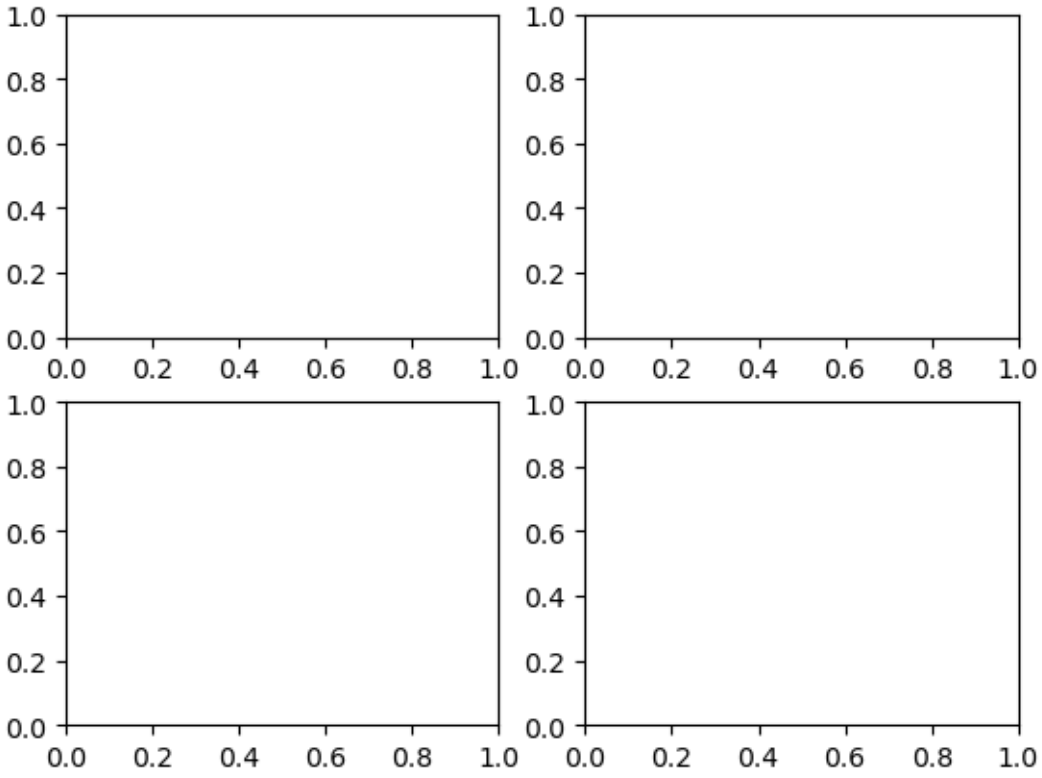
[ ]:
```
from pySurf.scripts.dlist import dcouples_plot, plot_data_repeat
from plotting.backends import maximize
from plotting.multiplots import commonscale
```

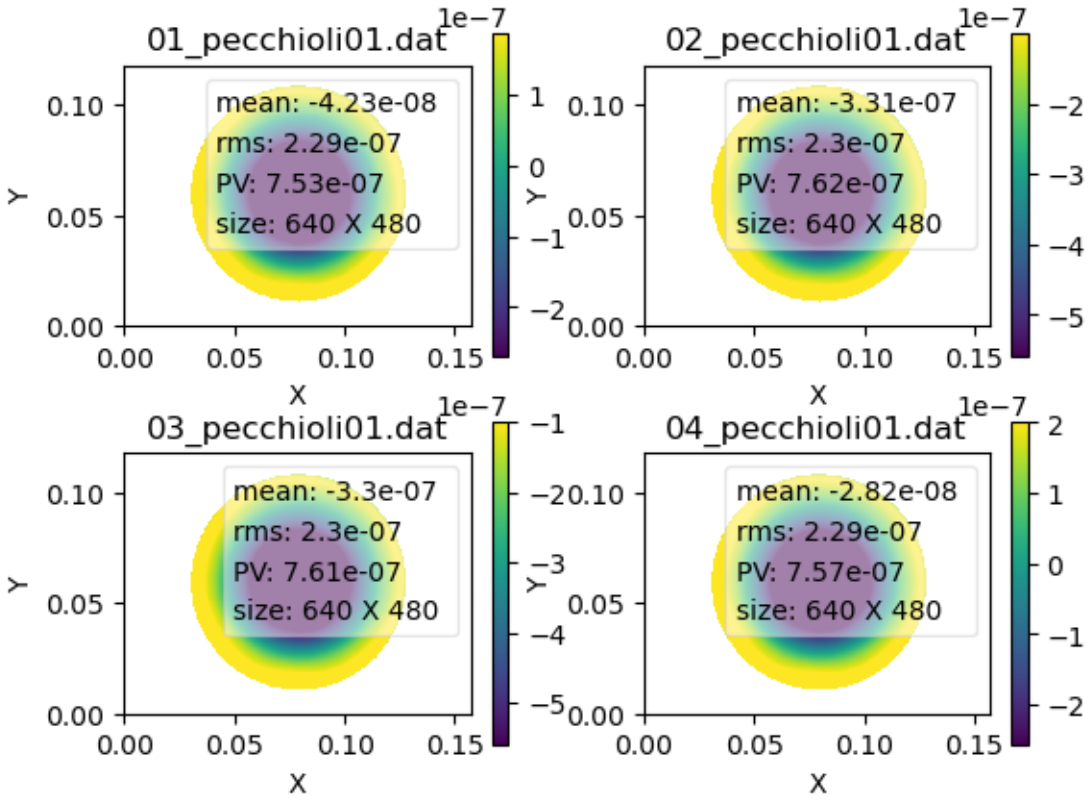
[ ]:
```
infolder = r"G:\Shared␣
 ↪drives\MUSE\dati\801B_100mm_Pecchioli\zygo\20221222_pecchioli01_repatability"
fl = sorted([f for f in os.listdir(infolder) if os.path.splitext(f)[-1] == '.
 ↪dat'])
```

[ ]:
```
plt.ion()
```

[ ]: <matplotlib.pyplot._IonContext at 0x23057c249d0>

[ ]:
```
from pySurf.scripts.dlist import Dlist, load_dlist

dl = load_dlist([os.path.join(infolder,f) for f in fl[:
 ↪4]],reader=matrixdat_reader)
dl.plot(type='grid',center = (0,0),strip=True)
```





[ ]:
```
a = dd.level()
dd.plot()
#a.plot()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_10956\3963769781.py in <cell line: 2>()
      1 a = dd.level()
----> 2 dd.plot()
      3 #a.plot()

c:\users\kovor\documents\python\pyxtel\source\dataIO\superlist.py in␣
 ↪newfunc(*args, **kwargs)
    317         if hasattr(attr, '__call__'):
    318             def newfunc(*args, **kwargs):
--> 319                 result = attr(*args, **kwargs)
    320                 return result
    321                 #return self.__class__(result)  #doesn't work, big cras

c:\users\kovor\documents\python\pyxtel\source\pySurf\scripts\dlist.py in␣
 ↪plot(self, type, *args, **kwargs)
```

```
    507             maximize()
    508         for ax,d in zip(axes,self):
--> 509             plt.sca(ax)
    510             d.plot(*args,**prep_kw(plt.plot,kwargs))
    511         #print(args)

c:\Users\kovor\anaconda3\lib\site-packages\matplotlib\pyplot.py in sca(ax)
   1103     """
   1104     figure(ax.figure)
-> 1105     ax.figure.sca(ax)
   1106
   1107

c:\Users\kovor\anaconda3\lib\site-packages\matplotlib\figure.py in sca(self, a)
   1533     def sca(self, a):
   1534         """Set the current Axes to be *a* and return *a*."""
-> 1535         self._axstack.bubble(a)
   1536         self._axobservers.process("_axes_change_event", self)
   1537         return a

c:\Users\kovor\anaconda3\lib\site-packages\matplotlib\figure.py in bubble(self,
 ↪a)
     80         Move the given axes, which must already exist in the stack, to␣
 ↪the top.
     81         """
---> 82         return super().bubble(self._entry_from_axes(a))
     83
     84     def add(self, a):

c:\Users\kovor\anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in␣
 ↪bubble(self, o)
    636         """
    637         if o not in self._elements:
--> 638             raise ValueError('Given element not contained in the stack'
    639         old_elements = self._elements.copy()
    640         self.clear()

ValueError: Given element not contained in the stack
```

```
<Figure size 640x480 with 0 Axes>
```

```
<Figure size 640x480 with 0 Axes>
```

```
<Figure size 640x480 with 0 Axes>
```

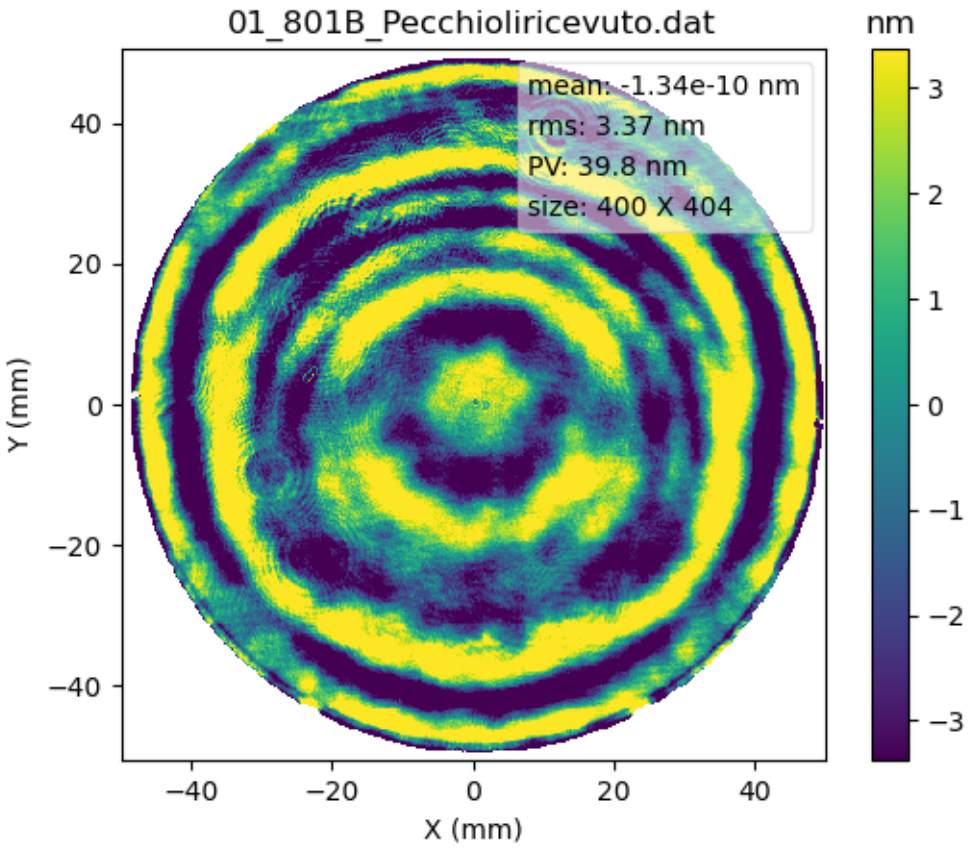
```
<Figure size 640x480 with 0 Axes>
```

[ ]:
```
dd = Dlist(dl[:4])

maximize()
dcouples_plot(dd.level(), level=False)
commonscale()
plt.tight_layout()
plt.show()

plt.figure()
plot_data_repeat(dd.level())
plt.tight_layout()
dcouples_plot(dd)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_10956\1659969996.py in <cell line: 1>()
----> 1 dd = Dlist(dl[:4])
      2
      3 maximize()
      4 dcouples_plot(dd.level(), level=False)
      5 commonscale()

c:\users\kovor\documents\python\pyxtel\source\dataIO\superlist.py in␣
 ↪__getitem__(self, index)
    327         """This implements slicing, according to chatGPT suggestions."""
    328
--> 329         result = super().__getitem__(index)
    330         if isinstance(result, list):
    331             return self.__class__(result) #Superlist(result)

TypeError: super(type, obj): obj must be an instance or subtype of type
```

[ ]:
```
# file = r'G:\Shared drives\MUSE\dati\models\DESIGN1_43percent\00063691-Static␣
 ↪2-Results-Displacement1-43%LW.csv'
file =  r'G:\Shared␣
 ↪drives\MUSE\dati\801B_100mm_Pecchioli\20220908_zygo_initial\01_801B_Pecchioliricevuto.
 ↪dat'

d = Data2D(file = file, reader = matrixdat_reader,
           center = (0,0),
           scale = (1000,1000,1000000000),units=['mm','mm','nm'])
d.level((10,10)).plot()
```

```
<matplotlib.image.AxesImage at 0x1c5979dfac0>
```


01_801B_Pecchioliricevuto.dat
mean: -1.34e-10 nm
rms: 3.37 nm
PV: 39.8 nm
size: 400 X 404
nm
3
2
1
0
−1
−2
−3
40
20
0
−20
−40
Y (mm)
−40
−20
0
20
40
X (mm)