

```

import numpy as np
from tkinter import filedialog
import pycubelib.files_functions as ff
import pycubelib.general_functions as gf
import pycubelib.plotting_functions as pf
import pycubelib.data_functions as df

pi = np.pi
pi2 = pi * 2
pi2limit = (-pi, pi)
sxy = (.35, .35)
roi = (1000, 900, 10, 10)
z0_roi = 0.

# txt_path = filedialog.askopenfilename(title='TXT file path', filetypes=[('txt files',
# '*.txt')])
txt_path = '/media/mkpaulkim/Ultra Touch/{UT White}/Dropbox/[PROJECTS.dbox ]' \
            '/project folders 2021/proj 2021-10 AlphaSigma/temp_data/eee7.txt'
notes, txt_path = ff.read_txt(txt_path)
note = notes[notes.find('%%'):]
nx = gf.find_param(note, 'nx', int)
ny = gf.find_param(note, 'ny', int)
dx = gf.find_param(note, 'dx', float)
nw = gf.find_param(note, 'nw', int)
wln = gf.find_param(note, 'wln', float)
if wln[0] != 0.:
    wln = [0.] + wln

print(f'> txt_path = {txt_path}')
print(f'> notes: \n{notes}')
blank = np.zeros((ny, nx))

''' read hhh '''
hhhp = []
for n in range(nw + 1):
    if n == 0:
        hh_path = txt_path.replace('.txt', f'_aa.png')
        hh, _ = ff.read_png(hh_path, alimit=(0., 1.))
    else:
        hh_path = txt_path.replace('.txt', f'_{n}p.png')
        hh, _ = ff.read_png(hh_path, alimit=pi2limit)
        capA, _ = gf.path_parts(hh_path)
        # pf.plotAAB(hh, capA=capA, roi=roi, sxy=sxy, pause=1)
        hhhp += [hh]
hha = hhhp[0]

''' get lam_lns '''
lam_lns = np.zeros(nw + 1)
lam1 = wln[1]
for n in range(2, nw + 1):
    lam_lns[n] = (lam1 * wln[n]) / (lam1 - wln[n])
# lam_lns[5] = 500
lam12 = lam_lns[2]

print(gf.prn_list('wln', wln, 8))
print(gf.prn_list('lam_lns', lam_lns, 1))

''' make zz_lns '''
zz_lns = [blank, blank]
ep_lns = [blank, blank]
for n in range(2, nw + 1):
    lam1n = lam_lns[n]

```

```

    ep1n = np.mod(hhhp[n] - hhhp[1] + pi, pi2) - pi
    zz1n_ = ep1n * lam1n / pi2
    _, z_roi = gf.roi_cyclic_measure(zz1n_, roi, lam1n)
    zz1n = np.mod(zz1n_ - z_roi + z0_roi + lam1n/2, lam1n) - lam1n/2
    # _, z1_roi = gf.roi_measure(zz1n, roi)
    # print(f'< z_roi = {z_roi:.1f}, z0_roi = {z0_roi:.1f}, z1_roi = {z1_roi:.1f}')

    ep_1ns += [ep1n]
    zz_1ns += [zz1n]
    pf.plotAAB(zz1n, capA=f'ZZ1{n}', roi=roi, sxy=sxy, pause=1)

''' make zz_12ns '''
zz_12ns = [blank, blank, zz_1ns[2]]
for n in range(3, nw + 1):

    lam_1ns[n] = df.calib_lam1n(zz_12ns[n-1], zz_1ns[n], lam12, lam_1ns[n], roi)
    zz_12n = df.stitch(zz_12ns[n-1], zz_1ns[n], lam_1ns[2], lam_1ns[n])
    zz_12ns += [zz_12n]
    pf.plotAAB(zz_12n, capA=f'ZZ12{n}: lam_1{n} = {lam_1ns[n]:.1f}', roi=roi, sxy=sxy,
    pause=1)

''' graph all '''
graphs = []
ix, iy, rx, ry = roi
# print(f'< roi = {roi}')
for n in range(1, nw + 1):
    graphs += [(hhhp[n][iy, :], (n-1, 0), f'HH{n}p', pi2limit)]
for n in range(2, nw + 1):
    lam_1n = lam_1ns[n]
    graphs += [(zz_1ns[n][iy, :], (n-1, 1), f'ZZ_1{n}', (-lam_1n/2, lam_1n/2))]
for n in range(3, nw + 1):
    graphs += [(zz_12ns[n][iy, :], (n-1, 2), f'ZZ_12{n}', (-lam12/2, lam12/2))]

pf.graph_many(graphs, col_row=(3, nw), sxy=(.25, .25), pause=1)

pf.mayaviAA(zz_12ns[-1])

pf.plt.show()

```