forsep_test

May 17, 2019

0.0.1 MaxEnt Fourier-space based component separation

Based on the algorithm described in Hobson et al, Foreground separation methods for satellite observations of the cosmic microwave background, MNRAS, 300, p1 (1998)

```
In [1]: import sys
        import os

        cwd = os.getcwd()
        sys.path.insert(0, cwd)

        import fastmem as fm
        import numpy as np

        import scipy as sp
        import scipy.optimize as opt

        from astropy.io import fits

        import matplotlib.pyplot as plt

        import multiprocessing
        from multiprocessing import Pool
        import time

        from ipywidgets import IntProgress
        from IPython.display import display
```

```
In [2]: inputdir    = './data/'
        outputdir   = './data/'
```

```
In [3]: multithreaded = True
```

```
In [4]: datafile     = "testdata.fits"
        noisefile    = "testnoise.fits"
        compfile     = "testcomp.fits"
        reconfile    = 'test_reconstruction.fits'
        errorfile    = 'test_errors.fits'
```

In [5]:
```
datafile     = inputdir  + datafile
noisefile    = inputdir  + noisefile
compfile     = inputdir  + compfile
reconfile    = outputdir + reconfile
errorfile    = outputdir + errorfile
```

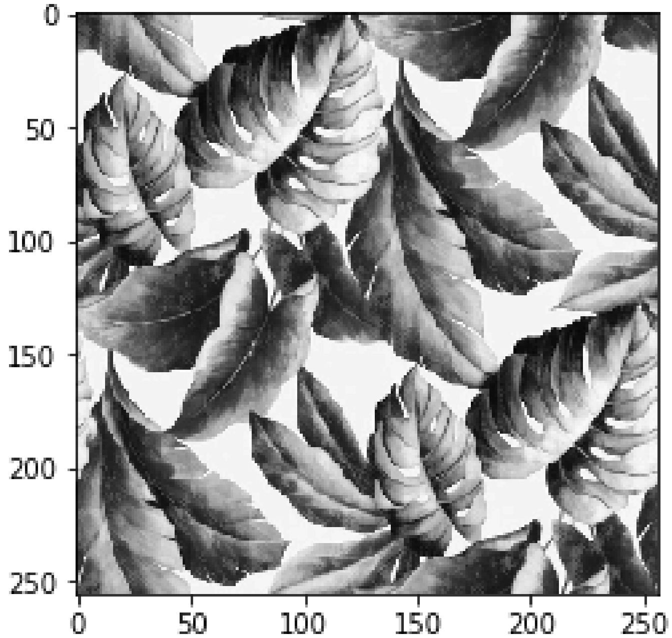
In [6]:
```
# Read component models (to calculate ICFs)
components, components_data = fm.read_fits_map(compfile)
components[0].header
```

```
Filename: ./data/testcomp.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      26   (256, 256, 2)   float64
```

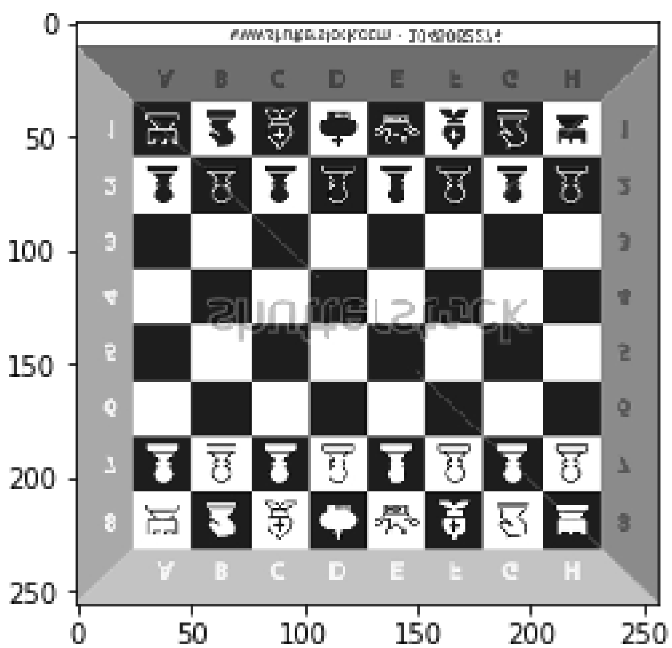
Out[6]:
```
SIMPLE  =                    T / conforms to FITS standard
BITPIX  =                  -64 / array data type
NAXIS   =                    3 / number of array dimensions
NAXIS1  =                  256
NAXIS2  =                  256
NAXIS3  =                    2
WCSAXES =                    3 / Number of coordinate axes
CRPIX1  =                128.0 / Pixel coordinate of reference point
CRPIX2  =                128.0 / Pixel coordinate of reference point
CRPIX3  =                  1.0 / Pixel coordinate of reference point
CDELT1  =            -0.003125 / [deg] Coordinate increment at reference point
CDELT2  =             0.003125 / [deg] Coordinate increment at reference point
CDELT3  =             100000.0 / [Hz] Coordinate increment at reference point
CUNIT1  = 'deg'                / Units of coordinate increment and value
CUNIT2  = 'deg'                / Units of coordinate increment and value
CUNIT3  = 'Hz'                 / Units of coordinate increment and value
CTYPE1  = 'RA---TAN'           / Right ascension, gnomonic projection
CTYPE2  = 'DEC--TAN'           / Declination, gnomonic projection
CTYPE3  = 'FREQ'               / Frequency (linear)
CRVAL1  =                180.0 / [deg] Coordinate value at reference point
CRVAL2  =                  0.0 / [deg] Coordinate value at reference point
CRVAL3  =          100000000.0 / [Hz] Coordinate value at reference point
LONPOLE =                180.0 / [deg] Native longitude of celestial pole
LATPOLE =                  0.0 / [deg] Native latitude of celestial pole
RADESYS = 'ICRS'               / Equatorial coordinate system
BUNIT   = 'K       '
```

In [7]:
```
# Show components
for i in range(components[0].header['NAXIS3']):
    print(i, round(np.mean(components_data[i]),5), round(np.std(components_data[i]),5)
    plt.imshow(components_data[i], cmap='gray')
    plt.show()
```

```
0 0.6439 0.23718
```




```
1 0.58717 0.35104
```



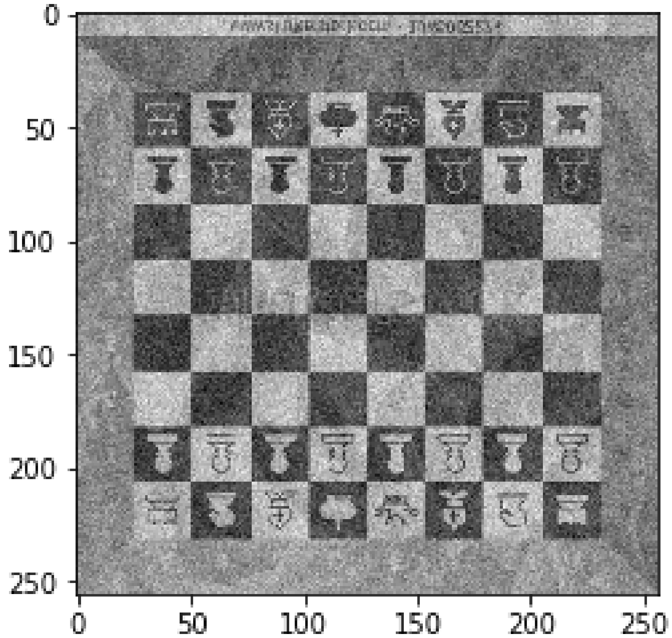
```
In [8]: # Read input data file
        hera_hdl, hera_data = fm.read_fits_map(datafile)
        hera_hdl[0].header
```

```
Filename: ./data/testdata.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      26   (256, 256, 3)   float64
```

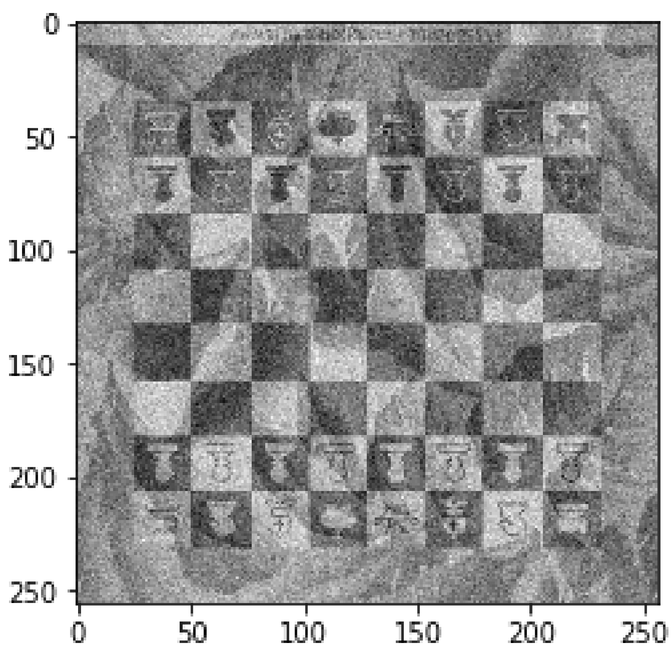
```
Out[8]: SIMPLE  =                    T / conforms to FITS standard
        BITPIX  =                  -64 / array data type
        NAXIS   =                    3 / number of array dimensions
        NAXIS1  =                  256
        NAXIS2  =                  256
        NAXIS3  =                    3
        WCSAXES =                    3 / Number of coordinate axes
        CRPIX1  =                128.0 / Pixel coordinate of reference point
        CRPIX2  =                128.0 / Pixel coordinate of reference point
        CRPIX3  =                  1.0 / Pixel coordinate of reference point
        CDELT1  =            -0.003125 / [deg] Coordinate increment at reference point
        CDELT2  =             0.003125 / [deg] Coordinate increment at reference point
        CDELT3  =          100000000.0 / [Hz] Coordinate increment at reference point
        CUNIT1  = 'deg'                / Units of coordinate increment and value
        CUNIT2  = 'deg'                / Units of coordinate increment and value
        CUNIT3  = 'Hz'                 / Units of coordinate increment and value
        CTYPE1  = 'RA---TAN'           / Right ascension, gnomonic projection
        CTYPE2  = 'DEC--TAN'           / Declination, gnomonic projection
        CTYPE3  = 'FREQ'               / Frequency (linear)
        CRVAL1  =                180.0 / [deg] Coordinate value at reference point
        CRVAL2  =                  0.0 / [deg] Coordinate value at reference point
        CRVAL3  =          100000000.0 / [Hz] Coordinate value at reference point
        LONPOLE =                180.0 / [deg] Native longitude of celestial pole
        LATPOLE =                  0.0 / [deg] Native latitude of celestial pole
        RADESYS = 'ICRS'               / Equatorial coordinate system
        BUNIT   = 'K       '
```

```
In [9]: # Show mixed frequency channels
        for i in range(hera_hdl[0].header['NAXIS3']):
            print(print(i, round(np.mean(hera_data[i]),5), round(np.std(hera_data[i]),5)))
            plt.imshow(hera_data[i], cmap='gray')
            plt.show()
```

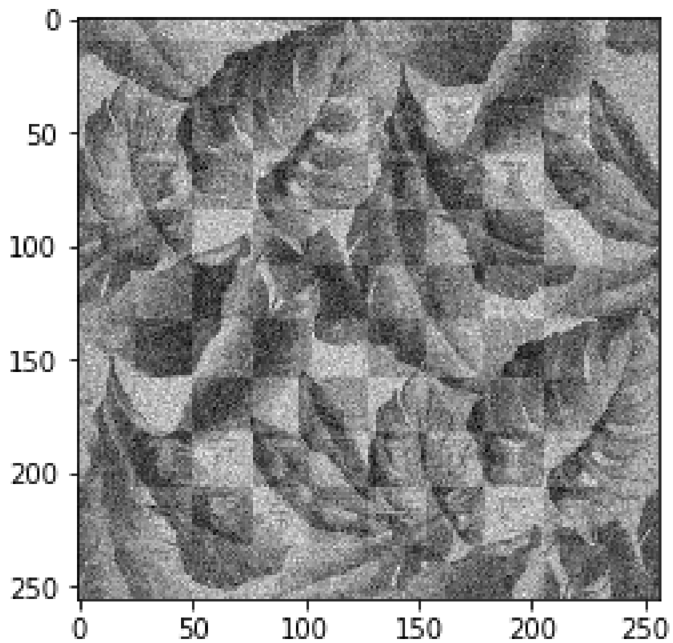
```
0 2.40901 1.14848
None
```



```
1 2.46221 0.93597
None
```



```
2 2.51876 0.88706
None
```



In [10]:
```
# Read noise models (to calculate it's power spectum)
noise, noise_data = fm.read_fits_map(noisefile)
noise[0].header
```

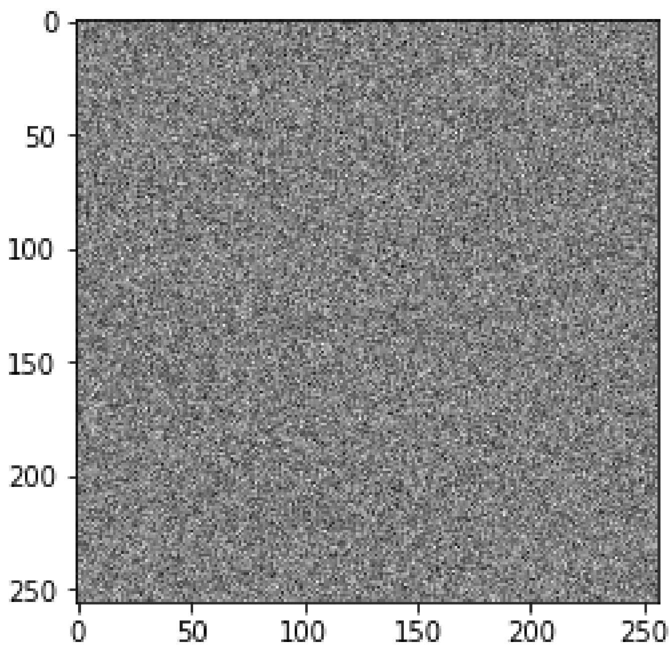
```
Filename: ./data/testnoise.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      26   (256, 256, 3)   float64
```

Out[10]:
```
SIMPLE  =                    T / conforms to FITS standard
BITPIX  =                  -64 / array data type
NAXIS   =                    3 / number of array dimensions
NAXIS1  =                  256
NAXIS2  =                  256
NAXIS3  =                    3
WCSAXES =                    3 / Number of coordinate axes
CRPIX1  =                128.0 / Pixel coordinate of reference point
CRPIX2  =                128.0 / Pixel coordinate of reference point
CRPIX3  =                  1.0 / Pixel coordinate of reference point
CDELT1  =            -0.003125 / [deg] Coordinate increment at reference point
CDELT2  =             0.003125 / [deg] Coordinate increment at reference point
CDELT3  =          100000000.0 / [Hz] Coordinate increment at reference point
```

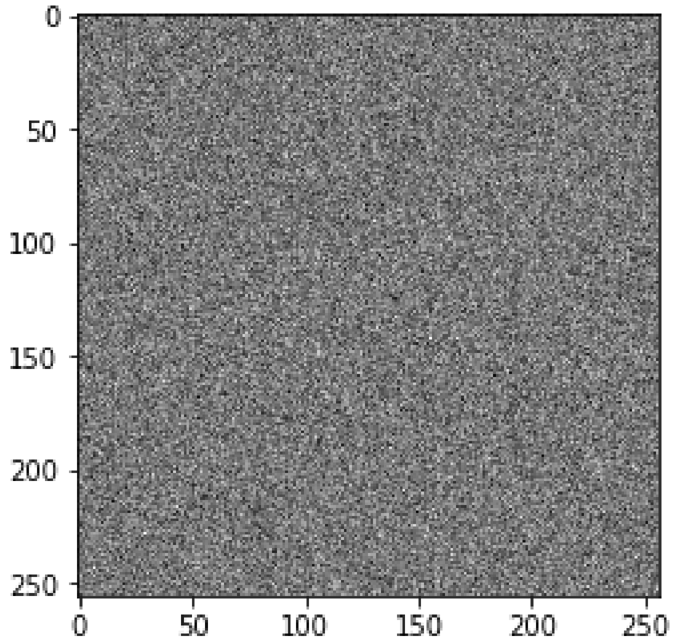
```
CUNIT1  = 'deg'                / Units of coordinate increment and value
CUNIT2  = 'deg'                / Units of coordinate increment and value
CUNIT3  = 'Hz'                 / Units of coordinate increment and value
CTYPE1  = 'RA---TAN'           / Right ascension, gnomonic projection
CTYPE2  = 'DEC--TAN'           / Declination, gnomonic projection
CTYPE3  = 'FREQ'               / Frequency (linear)
CRVAL1  =                180.0 / [deg] Coordinate value at reference point
CRVAL2  =                  0.0 / [deg] Coordinate value at reference point
CRVAL3  =          100000000.0 / [Hz] Coordinate value at reference point
LONPOLE =                180.0 / [deg] Native longitude of celestial pole
LATPOLE =                  0.0 / [deg] Native latitude of celestial pole
RADESYS = 'ICRS'               / Equatorial coordinate system
BUNIT   = 'K       '
```

In [11]:
```
# Show noise maps
nvar = []
for i in range(noise[0].header['NAXIS3']):
    nvar_field = np.var(noise_data[i])
    nvar.append(nvar_field)
    print(i, round(np.mean(noise_data[i]),5), round(np.std(noise_data[i]),5))
    plt.imshow(noise_data[i], cmap='gray')
    plt.show()
```

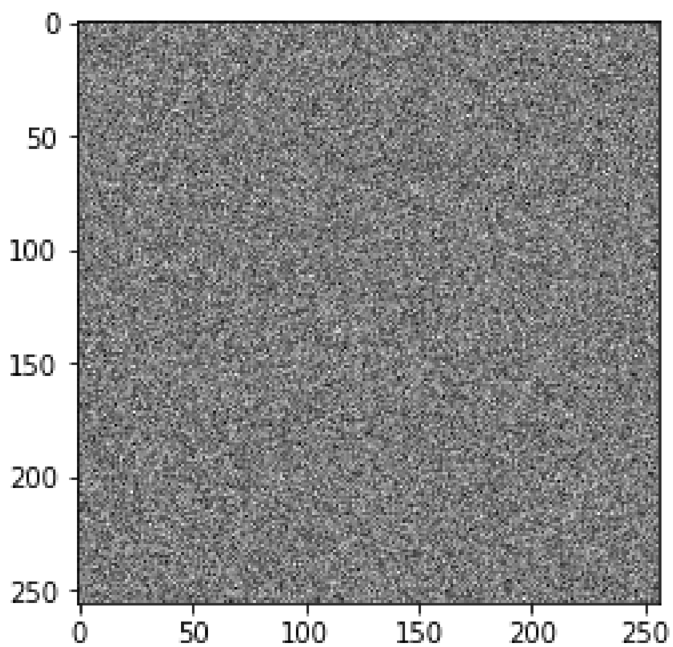
```
0 0.00359 0.40041
```



```
1 6e-05 0.40114
```




```
2 -0.00012 0.40104
```



In [12]:
```
# Check the shape of the maps
assert(noise[0].header['NAXIS1'] == hera_hdl[0].header['NAXIS1'])
assert(noise[0].header['NAXIS2'] == hera_hdl[0].header['NAXIS2'])
assert(noise[0].header['NAXIS3'] == hera_hdl[0].header['NAXIS3'])

assert(components[0].header['NAXIS1'] == hera_hdl[0].header['NAXIS1'])
assert(components[0].header['NAXIS2'] == hera_hdl[0].header['NAXIS2'])
```

In [13]:
```
# Pixel size in degrees
skycell = np.abs(hera_hdl[0].header['CDELT1'])
assert(skycell > 0.0)
```

In [14]:
```
# Get/calculate frequency conversion matrix (FCM, or mixing matrix)
conv = fm.get_testconvert()
print(conv, conv.shape)
```

```
[[1. 3.]
 [2. 2.]
 [3. 1.]] (3, 2)
```

In [15]:
```
conv.shape
```

Out[15]:
```
(3, 2)
```

In [16]:
```
# Get central frequencies
freqs = fm.get_testfrequencies()
print(freqs, np.mean(freqs))
```

```
[100. 200. 300.] 200.0
```

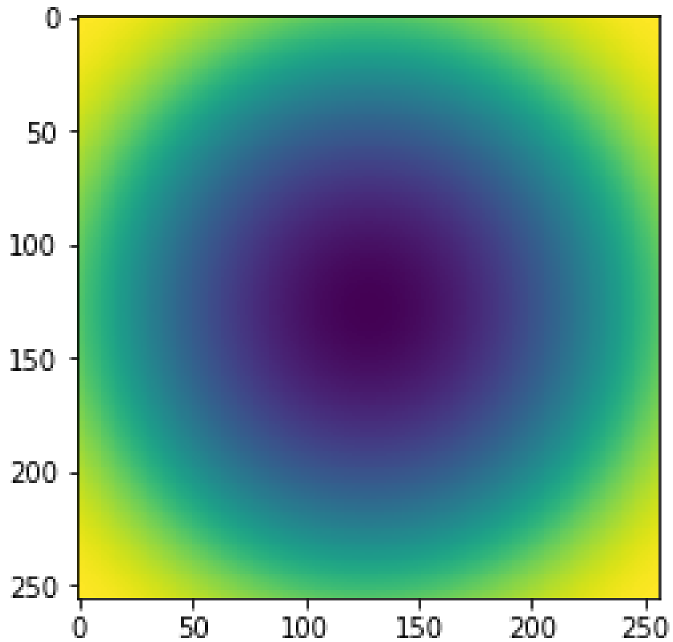
In [17]:
```
# Get FWHMs, arcmin (same for all freqs)
fwhm, sigma = fm.get_fwhm(freqs.shape[0], 0.01)
print(fwhm, np.mean(fwhm))
print(sigma, np.mean(sigma))
```

```
[0.01 0.01 0.01] 0.01
[7.07768167e-05 7.07768167e-05 7.07768167e-05] 7.077681669066825e-05
```

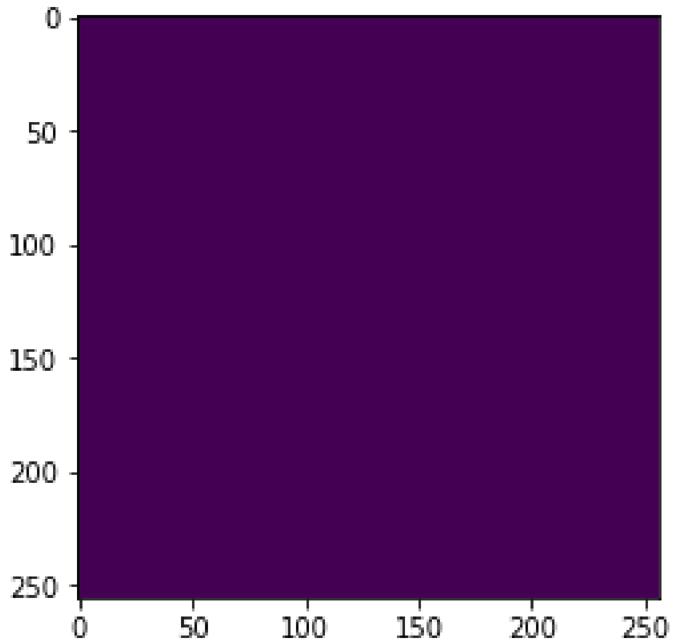
In [18]:
```
# Check shapes
print(conv.shape, freqs.shape, fwhm.shape)
assert(conv.shape[0] == freqs.shape[0])
assert(fwhm.shape[0] == freqs.shape[0])
```

```
(3, 2) (3,) (3,)
```

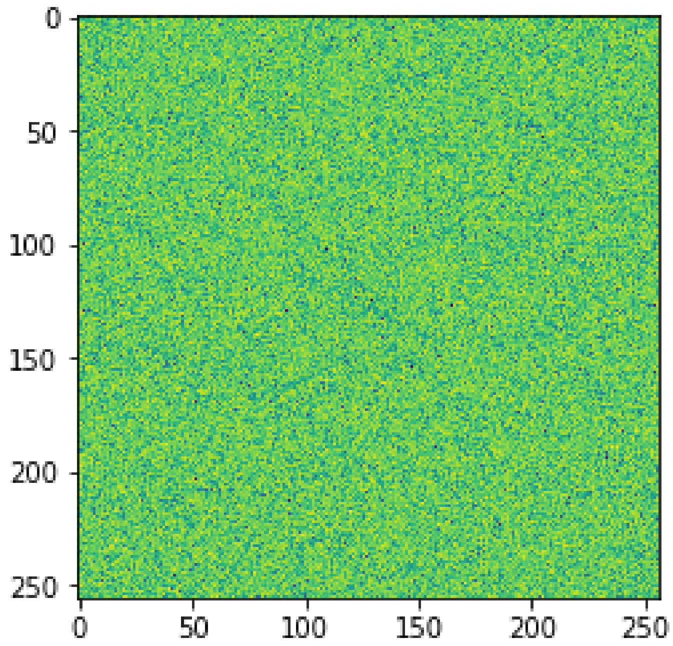
```
In [19]: nx = hera_hdl[0].header['NAXIS1']
         ny = hera_hdl[0].header['NAXIS2']
         nf = hera_hdl[0].header['NAXIS3']
         nc = components[0].header['NAXIS3']

In [20]: # Convert variances from a list to an array
         nvar = np.asarray(nvar)
         # Find the variance in Fourier domain (per Re or Im field)
         nvar = nvar*nx*ny/2.0

In [21]: print("Assumed noise variance per Re or Im part of the pixel in Fourier domain, nvar:
         print(nvar)

Assumed noise variance per Re or Im part of the pixel in Fourier domain, nvar:
[5253.64703276 5272.79470965 5270.10199165]


In [22]: bins, topbin = fm.calcbins(nx,ny)

In [23]: bins

Out[23]: array([[5923, 5922, 5921, ..., 5919, 5921, 5922],
                [5922, 5920, 5918, ..., 5916, 5918, 5920],
                [5921, 5918, 5915, ..., 5912, 5915, 5918],
                ...,
                [5919, 5916, 5912, ..., 5908, 5912, 5916],
                [5921, 5918, 5915, ..., 5912, 5915, 5918],
                [5922, 5920, 5918, ..., 5916, 5918, 5920]])

In [24]: topbin

Out[24]: 5923

In [25]: plt.imshow(bins)
         plt.show()
```




```
In [26]: beam, beamarea = fm.makebeam(nx,ny,sigma,skycell)

In [27]: beamarea

Out[27]: array([2.97477949e-09, 2.97477949e-09, 2.97477949e-09])

In [28]: beamfft = np.fft.fft2(beam)
         beamfft = np.fft.fftshift(beamfft, axes=(1,2))

In [29]: plt.imshow(abs(beamfft[0]))
         plt.show()
```



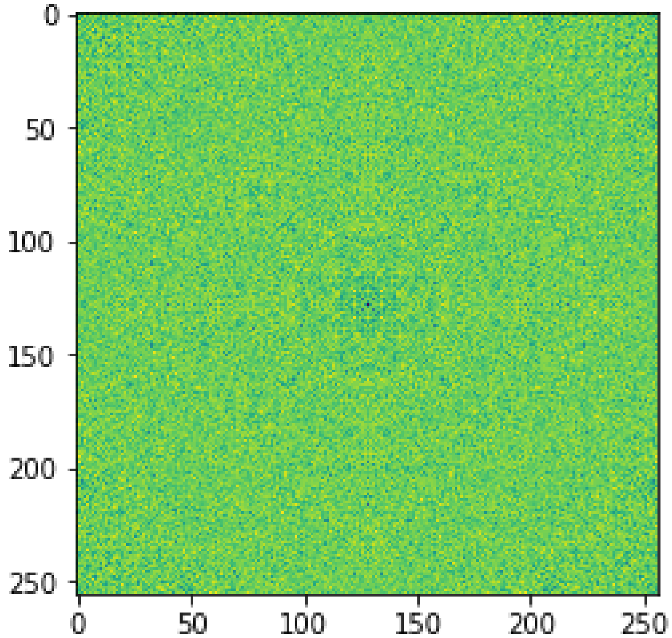

```
In [30]: datafft = np.fft.fft2(hera_data)
         datafft = np.fft.fftshift(datafft, axes=(1,2))

In [31]: noisefft = np.fft.fft2(noise_data)
         noisefft = np.fft.fftshift(noisefft, axes=(1,2))

In [32]: compfft = np.fft.fft2(components_data)
         compfft = np.fft.fftshift(compfft, axes=(1,2))

In [33]: plt.imshow(np.log(abs(noisefft[nf-1])))
         plt.show()
```



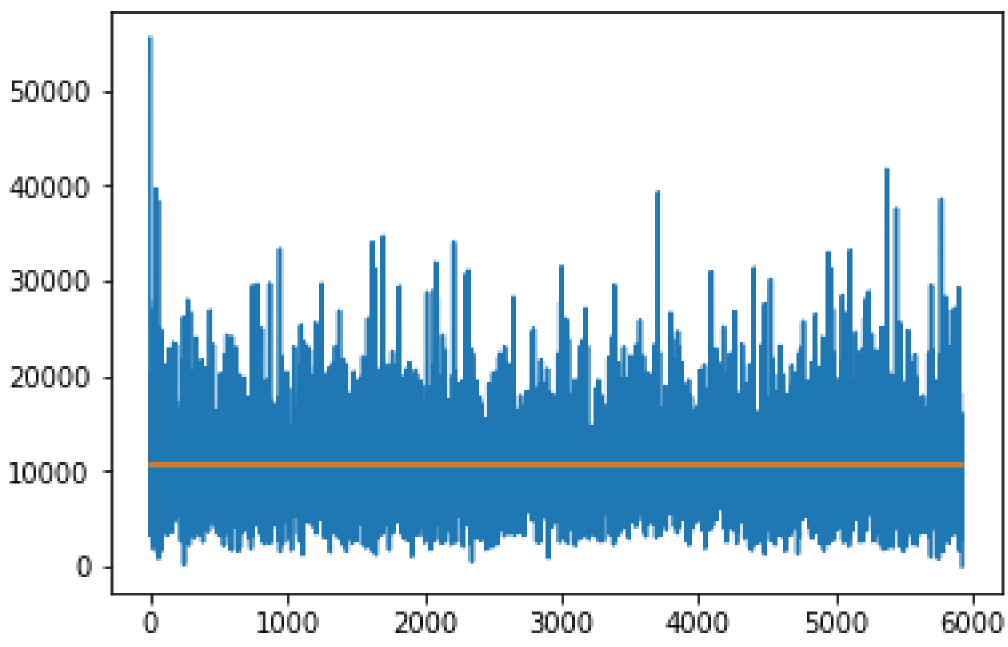
In [34]:
```
# Finding power spectra for the noise realisations
noiseps = np.zeros((nf, topbin+1))
for i in range(nf):
    workspace = np.abs(noisefft[i,:,:]**2)
    print("Noise level calculated ", round(np.mean(workspace)/2,5), " and assumed ", r
    summ, num = fm.azav(workspace, nx, ny, topbin, bins)
    print("Noise level after azav ", round(np.mean(workspace)/2,5))
    noiseps[i,:] = summ
```

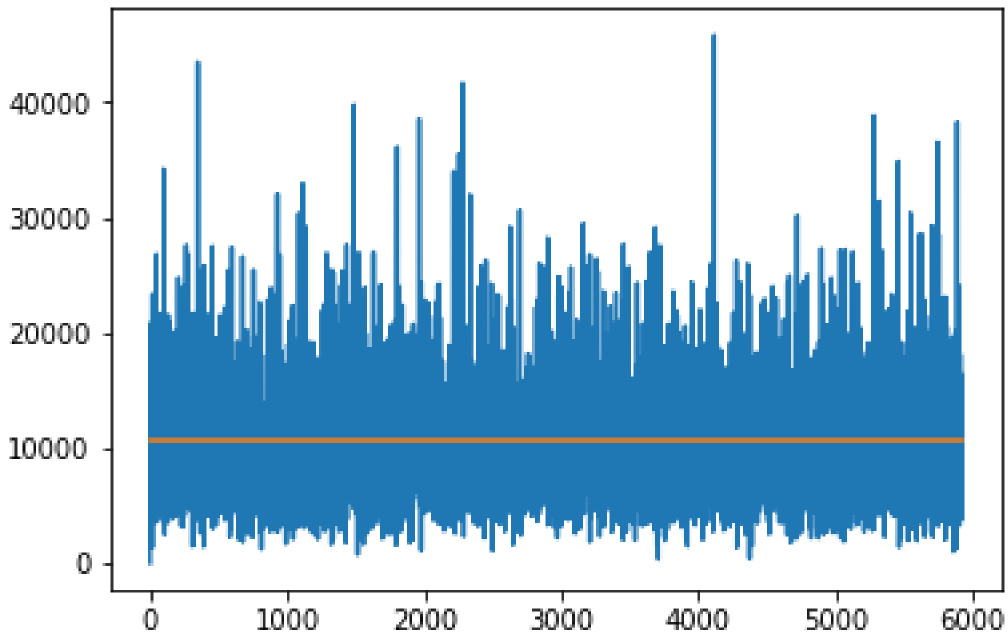
```
Noise level calculated  5254.06981  and assumed  5253.64703
Noise level after azav  5254.06981
Noise level calculated  5272.79484  and assumed  5272.79471
Noise level after azav  5272.79484
Noise level calculated  5270.10249  and assumed  5270.10199
Noise level after azav  5270.10249
```

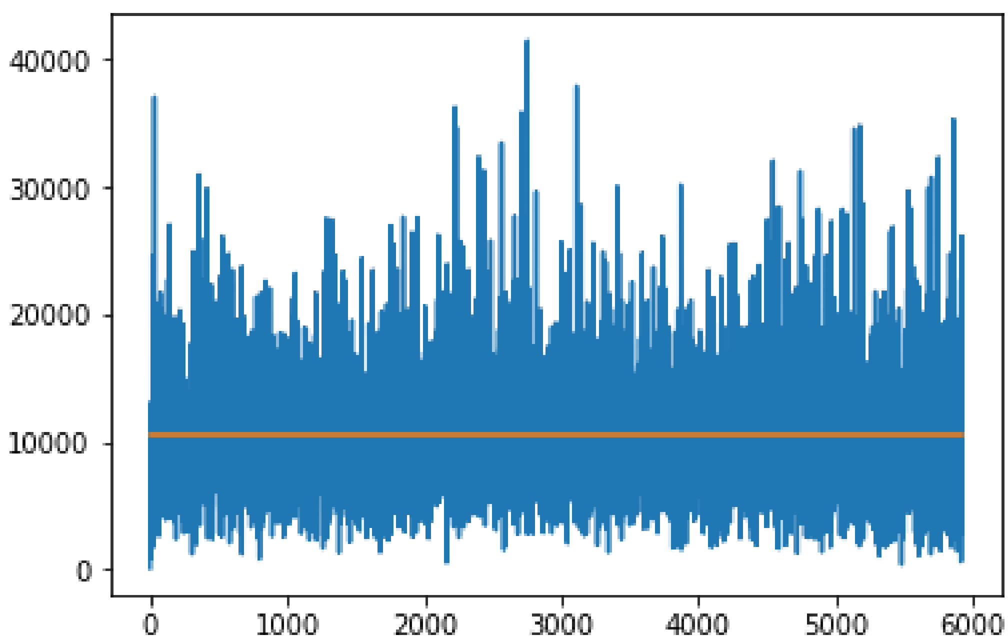
In [35]:
```
plt.imshow(np.log(workspace))
plt.show()
```



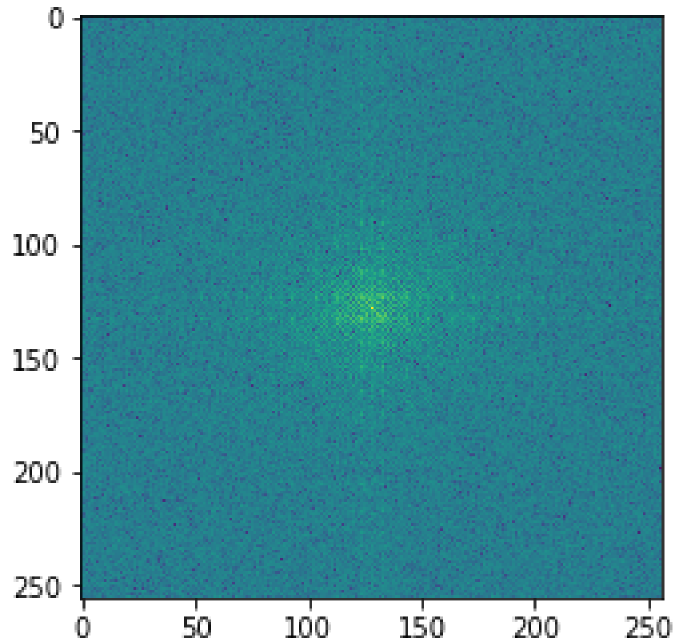
In [36]:
```
# Plot the noise power spectra and the noise floor for each frequency
for i in range(nf):
    plt.plot(noiseps[i,:]), plt.plot([0,topbin-1],[nvar[i]*2, nvar[i]*2])
    plt.show()
```



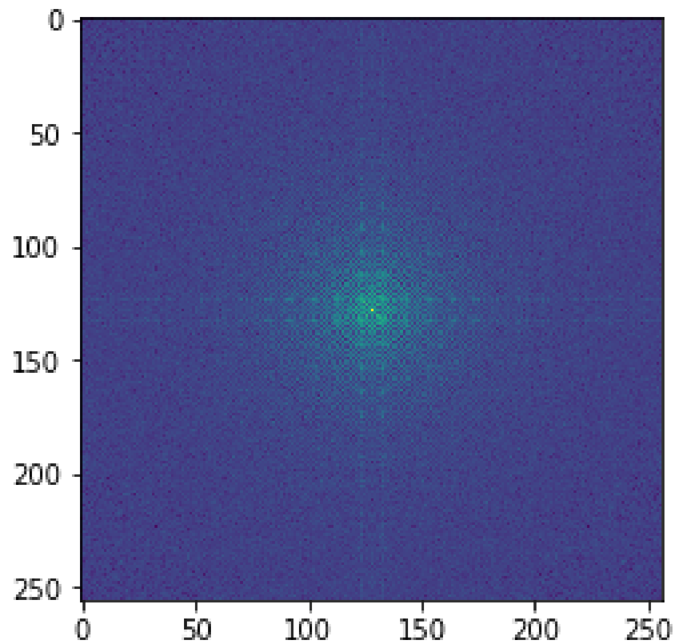




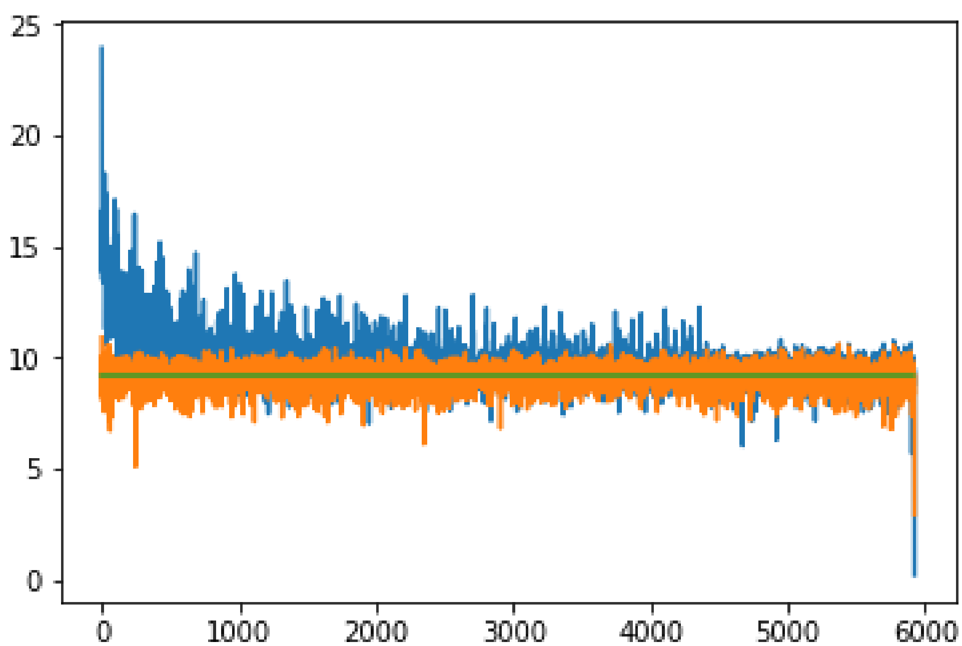

```
In [37]: plt.imshow(np.log(abs(datafft[nf-1])))
         plt.show()
```

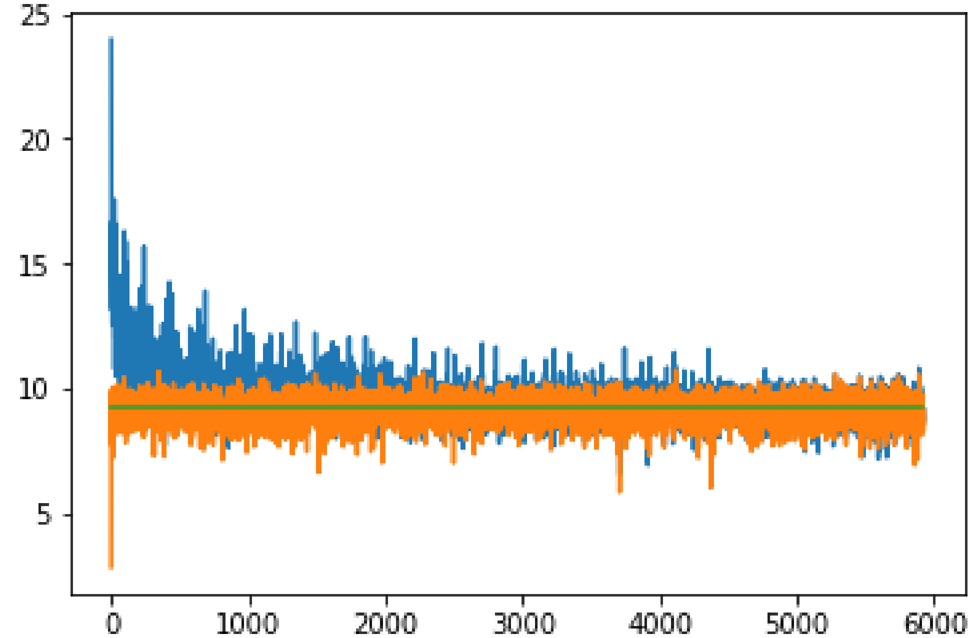


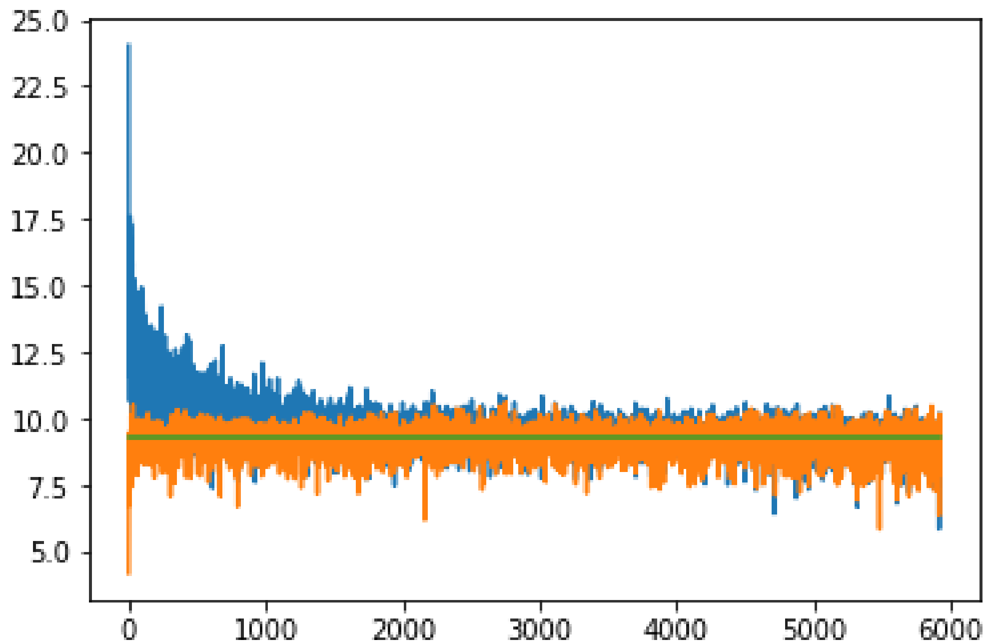

```
In [38]: # Finding power spectra for the data
         dataps = np.zeros((nf, topbin+1))
         for i in range(nf):
             workspace = np.abs(datafft[i,:,:]**2)
             summ, num = fm.azav(workspace, nx, ny, topbin, bins)
             dataps[i,:] = summ

In [39]: plt.imshow(np.log(workspace))
         plt.show()
```




```
In [40]: # Show logpower spectra of noisy data and noise floor
         for i in range(nf):
             plt.plot(np.log(dataps[i,:])), plt.plot(np.log(noiseps[i,:])), plt.plot([0,topbin
             plt.show()
```





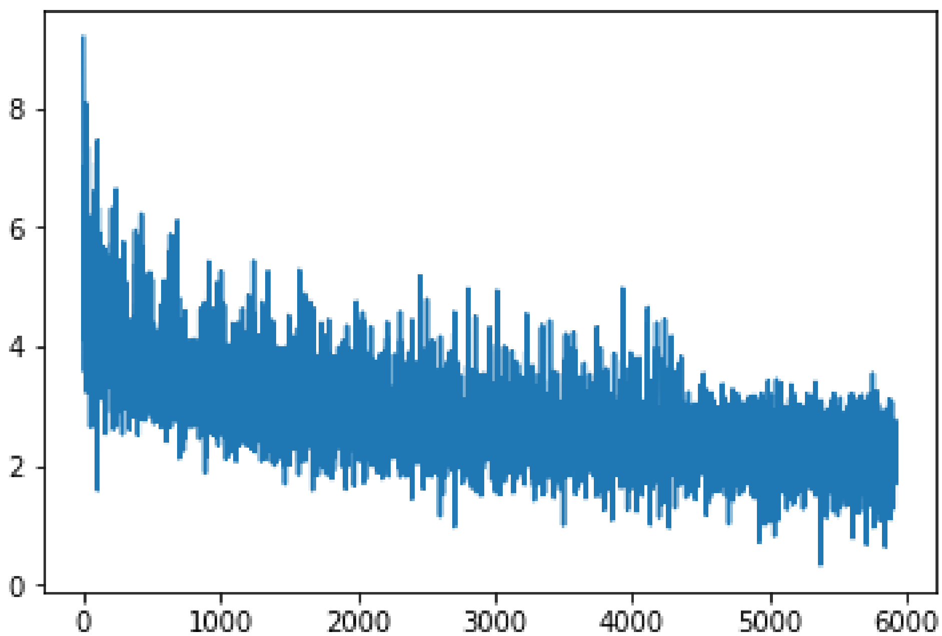


In [41]:
```
# Find Intrinsic Correlation Function (ICF)
icf = fm.make_icf(components_data,nx,ny,nc, topbin, bins, usecross=True)
```

```
0 0
1 0
1 1
WARNING: choldc failed at mode =  0
WARNING: choldc failed at mode =  5923
```

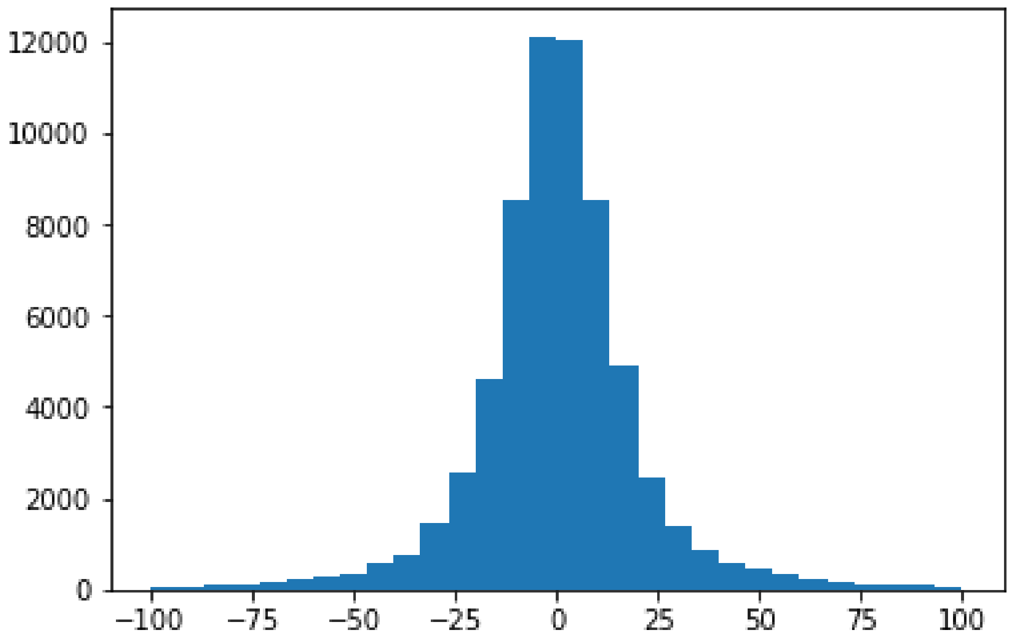
```
In [42]: # Show the off-diagonal ICF values
         plt.plot(np.log(icf[1,0,:]))
         plt.show()
```

```
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:2: RuntimeWarning: divide by zero
```

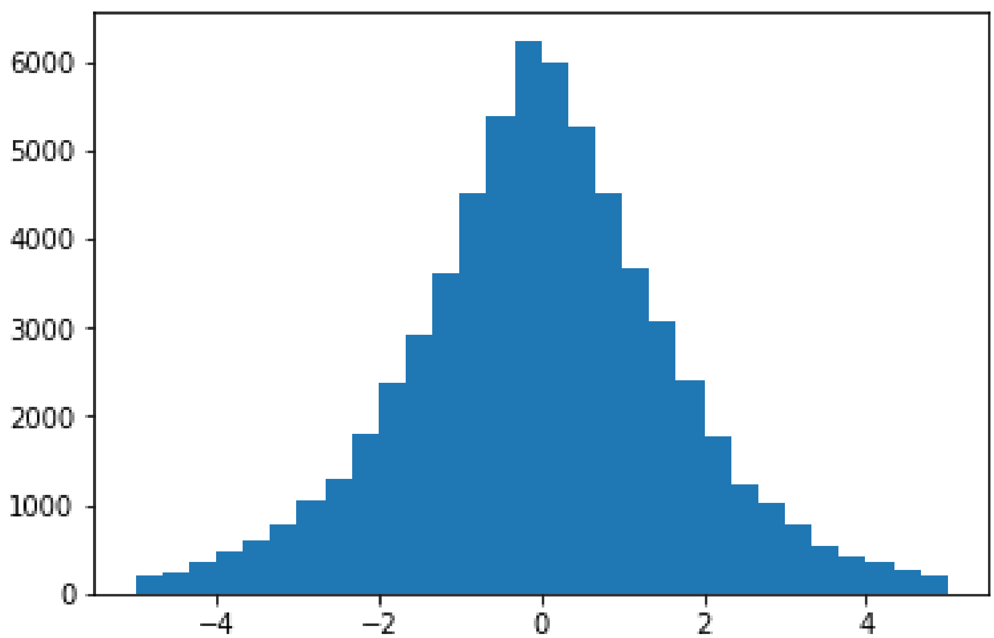



```
In [43]: # Find hidden (uncorrelated) variables
         mapsfft = fm.vistohid(icf,bins,compfft)
```

```
In [44]: # Pixel value distribution of the visible Fourier image
         plt.hist(np.real(compfft[1,:,:]).reshape(nx*nx), range=(-100,100), bins=30)
         plt.show()
```




```
In [45]: # Pixel value distribution of the hidden Fourier image (uncorrelated part)
         plt.hist(np.real(mapsfft[1,:,:]).reshape(nx*nx), range=(-5,5), bins=30)
         plt.show()
```



In [46]:
```python
# Calculate RL = FCM*L, where FCM is the mixing matrix and L is the Cholesky-decompos
rl = fm.updaterl(topbin, conv, icf)
```

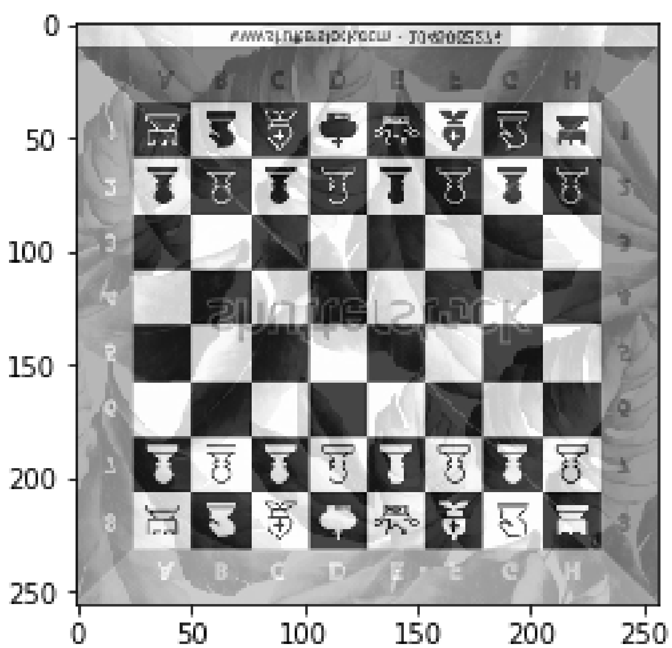
In [47]:
```python
datapfft = np.zeros((nf,nx,ny), dtype=complex)
```

In [48]:
```python
# Calculate predicted frequency maps with the component mixture
for i in range(nx):
    for j in range(ny):
        ibin = bins[i,j]
        datapfft[:,i,j] = fm.predict_mode(mapsfft[:,i,j], beamfft[:,i,j], rl[:,:,ibin]
```

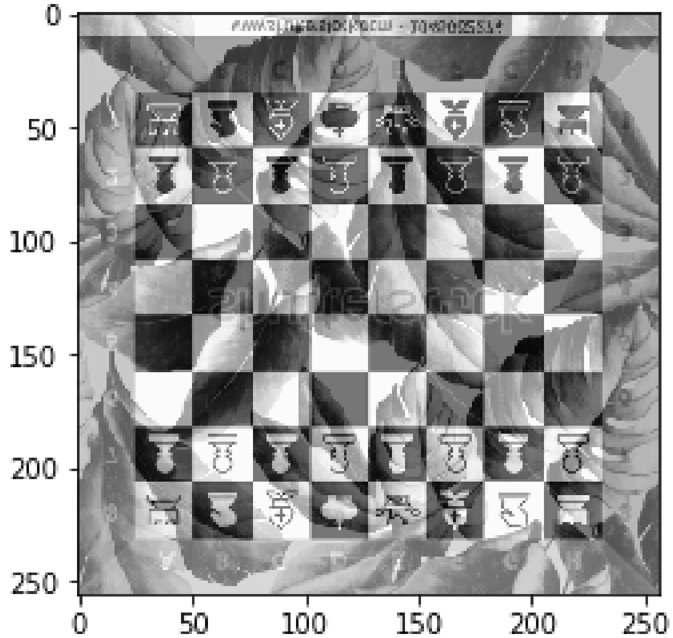
In [49]:
```python
datap = np.fft.ifft2(datapfft)
datap = np.fft.ifftshift(datap, axes=(1,2))
```

In [50]:
```python
#Show predicted frequency maps
for i in range(nf):
    print(i, freqs[i])
    plt.imshow(abs(datap[i]), cmap='gray')
    plt.show()
```

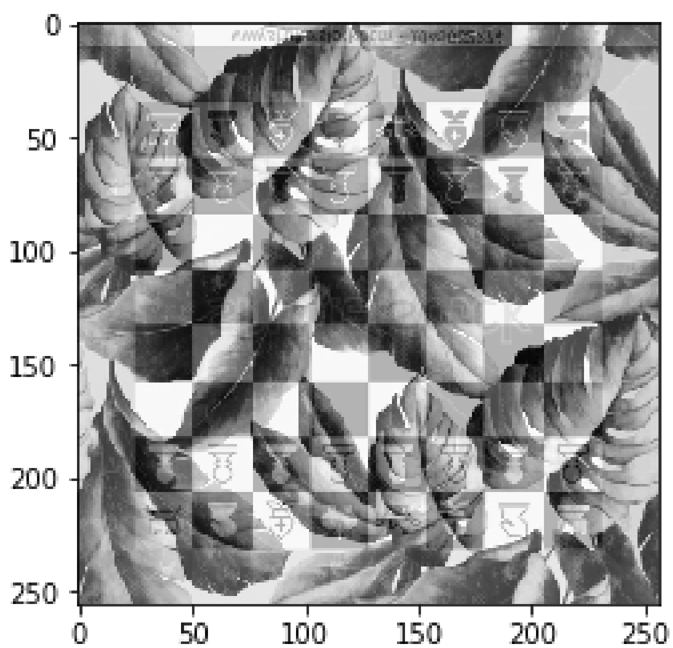
```
0 100.0
```




```
1 200.0
```



2 300.0



```
In [51]: comprec = np.empty_like(mapsfft)
```

```
In [52]: # MEM functional to minimise
         def F_mem(x,i,j):
             n = x.shape[0]
             endRe = int(n/2)
             xRe = x[:endRe]
             xIm = x[endRe:]
             xCmplx = xRe +1j*xIm
         # Predict data
             ibin = bins[i,j]
             datap = fm.predict_mode(xCmplx, beamfft[:,i,j], rl[:,:,ibin])
         # Find Chi2 term
             chi2Re = np.sum((np.real(datap - datafft[:,i,j]))**2/nvar)
             chi2Im = np.sum((np.imag(datap - datafft[:,i,j]))**2/nvar)
             chi2 = chi2Re + chi2Im

         # Entropic part
             sfftr = xRe
             sffti = xIm
             psir = np.sqrt(sfftr**2 + 4.0*modelp*modeln)
             psii = np.sqrt(sffti**2 + 4.0*modelp*modeln)
             tempr = psir + sfftr
             tempi = psii + sffti
             valuer = psir-modelp-modeln-sfftr*np.log(tempr/(2.0*modelp))
             valuei = psii-modelp-modeln-sffti*np.log(tempi/(2.0*modelp))
             value = np.sum(valuer + valuei)
             fmem = 0.5*chi2 - alpha*value
             return fmem

In [53]: # Chi2 functional to minimise
         def F_chi2(x,i,j):
             n = x.shape[0]
             endRe = int(n/2)
             xRe = x[:endRe]
             xIm = x[endRe:]
             xCmplx = xRe +1j*xIm
         # Predict data
             ibin = bins[i,j]
             datap = fm.predict_mode(xCmplx, beamfft[:,i,j], rl[:,:,ibin])
         # Find Chi2 term
             chi2Re = np.sum((np.real(datap - datafft[:,i,j]))**2/nvar)
             chi2Im = np.sum((np.imag(datap - datafft[:,i,j]))**2/nvar)
             chi2 = chi2Re + chi2Im
             return 0.5*chi2

In [54]: # MEM functional to minimise, returns F_MEM and the gradient
         def F_mem_grad(x,i,j):
             n = x.shape[0]
             endRe = int(n/2)
```

```
    xRe = x[:endRe]
    xIm = x[endRe:]
    xCmplx = xRe +1j*xIm
# Predict data
    ibin = bins[i,j]
    datap = fm.predict_mode(xCmplx, beamfft[:,i,j], rl[:,:,ibin])
    dfftr = np.real(datap - datafft[:,i,j])
    dffti = np.imag(datap - datafft[:,i,j])
# Find Chi2 term
    chi2Re = np.sum(dfftr**2/nvar)
    chi2Im = np.sum(dffti**2/nvar)
    chi2 = chi2Re + chi2Im

# Find Chi2 gradient
    cgradr = np.zeros(nc)
    cgradi = np.zeros(nc)

    for l in range(nc):
        for k in range(nf):
            resp = np.sum(np.real(beamfft[k,i,j])*conv[k,:]*icf[:,l,ibin])
            factor = -2.*resp/nvar[k]
            valuer = factor*dfftr[k]
            valuei = factor*dffti[k]
            cgradr[l] += valuer
            cgradi[l] += valuei

# Entropic part
    sfftr = xRe
    sffti = xIm
    psir = np.sqrt(sfftr**2 + 4.0*modelp*modeln)
    psii = np.sqrt(sffti**2 + 4.0*modelp*modeln)
    tempr = psir + sfftr
    tempi = psii + sffti
    valuer = psir-modelp-modeln-sfftr*np.log(tempr/(2.0*modelp))
    valuei = psii-modelp-modeln-sffti*np.log(tempi/(2.0*modelp))
    value = np.sum(valuer + valuei)

# Find entropic gradient
    egradr = -np.log(tempr/2./modelp)
    egradi = -np.log(tempi/2./modelp)

# Total gradient for each Re and Im part
    fmemgradr = 0.5*cgradr - alpha*egradr
    fmemgradi = 0.5*cgradi - alpha*egradi
    fmemgrad = np.array((fmemgradr, fmemgradi))

# Total F_mem value
    fmem = 0.5*chi2 - alpha*value
```

```
        return fmem, -1.0*fmemgrad.flatten()
```

```
In [55]: # Chi2 functional to minimise, returns Chi2 and the gradient
         def F_chi2_grad(x,i,j):
             n = x.shape[0]
             endRe = int(n/2)
             xRe = x[:endRe]
             xIm = x[endRe:]
             xCmplx = xRe +1j*xIm
         # Predict data
             ibin = bins[i,j]
             datap = fm.predict_mode(xCmplx, beamfft[:,i,j], rl[:,:,ibin])
             dfftr = np.real(datap - datafft[:,i,j])
             dffti = np.imag(datap - datafft[:,i,j])
         # Find Chi2 term
             chi2Re = np.sum(dfftr**2/nvar)
             chi2Im = np.sum(dffti**2/nvar)
             chi2 = chi2Re + chi2Im

         # Find Chi2 gradient
             cgradr = np.zeros(nc)
             cgradi = np.zeros(nc)

             for l in range(nc):
                 for k in range(nf):
                     resp = np.sum(np.real(beamfft[k,i,j])*conv[k,:]*icf[:,l,ibin])
                     factor = -2.*resp/nvar[k]
                     valuer = factor*dfftr[k]
                     valuei = factor*dffti[k]
                     cgradr[l] += valuer
                     cgradi[l] += valuei

             fmemgradr = 0.5*cgradr
             fmemgradi = 0.5*cgradi
             fmemgrad = np.array((fmemgradr, fmemgradi))

         # Total F_mem value
             fmem = 0.5*chi2
             return fmem, -1.0*fmemgrad.flatten()
```

```
In [56]: def F_mem_deriv(x,i,j):
             n = x.shape[0]
             endRe = int(n/2)
             xRe = x[:endRe]
             xIm = x[endRe:]
             xCmplx = xRe +1j*xIm
         # Predict data
             ibin = bins[i,j]
```

```
        datap = fm.predict_mode(xCmplx, beamfft[:,i,j], rl[:,:,ibin])
        dfftr = np.real(datap - datafft[:,i,j])
        dffti = np.imag(datap - datafft[:,i,j])

    # Find Chi2 gradient
        cgradr = np.zeros(nc)
        cgradi = np.zeros(nc)

        for l in range(nc):
            for k in range(nf):
                resp = np.sum(np.real(beamfft[k,i,j])*conv[k,:]*icf[:,l,ibin])
                factor = -2.*resp/nvar[k]
                valuer = factor*dfftr[k]
                valuei = factor*dffti[k]
                cgradr[l] += valuer
                cgradi[l] += valuei

    # Find entropic gradient
        sfftr = xRe
        sffti = xIm
        psir = np.sqrt(sfftr**2 + 4.0*modelp*modeln)
        psii = np.sqrt(sffti**2 + 4.0*modelp*modeln)
        tempr = psir + sfftr
        tempi = psii + sffti
        egradr = -np.log(tempr/2./modelp)
        egradi = -np.log(tempi/2./modelp)

    # Total gradient for each Re and Im part
        fmemgradr = 0.5*cgradr - alpha*egradr
        fmemgradi = 0.5*cgradi - alpha*egradi
        fmemgrad = np.array((fmemgradr, fmemgradi))

        return -1.0*fmemgrad.flatten()
```

```
In [57]: def F_mem_minimize(args):
             f,x,i,j = args
             xmin = opt.minimize(f, x, method = 'CG', args = (i,j)).x
             xmin = xmin[:nc] + 1j*xmin[nc:]
             print(i,j, xmin)
             return i,j,xmin
```

```
In [58]: def F_mem_grad_minimize(args):
             f,x,i,j = args
         #    xmin = opt.minimize(f, x, method = 'BFGS', jac = True, args = (i,j)).x
             xmin = opt.minimize(f, x, method = 'CG', jac = True, args = (i,j)).x
             xmin = xmin[:nc] + 1j*xmin[nc:]
         # Print information on some modes
             if(not i%10):
```

```
            if(not j%10):
                print(i, j, xmin, multiprocessing.current_process().name)
        return i,j,xmin
```

In [59]:
```
# Initial starting value - zero assumption
x0 = np.array([[0.,0.],[0.,0.]]).flatten()
# Assumed unit model for entropic term
modelp = np.array([1,1])
modeln = modelp
# Regularizing parameter
alpha = 0.1
```

In [60]:
```
print(opt.minimize(F_mem_grad, x0, method = 'BFGS', jac=True, args=(120,100), options=
```

```
Warning: Desired error not necessarily achieved due to precision loss.
         Current function value: 0.763183
         Iterations: 5
         Function evaluations: 113
         Gradient evaluations: 101
[ 0.74402051  0.93473288  1.06505702 -3.42298227]
```

In [61]:
```
print(opt.minimize(F_mem, x0, method = 'BFGS', args=(120,100), options={'disp': True})
```

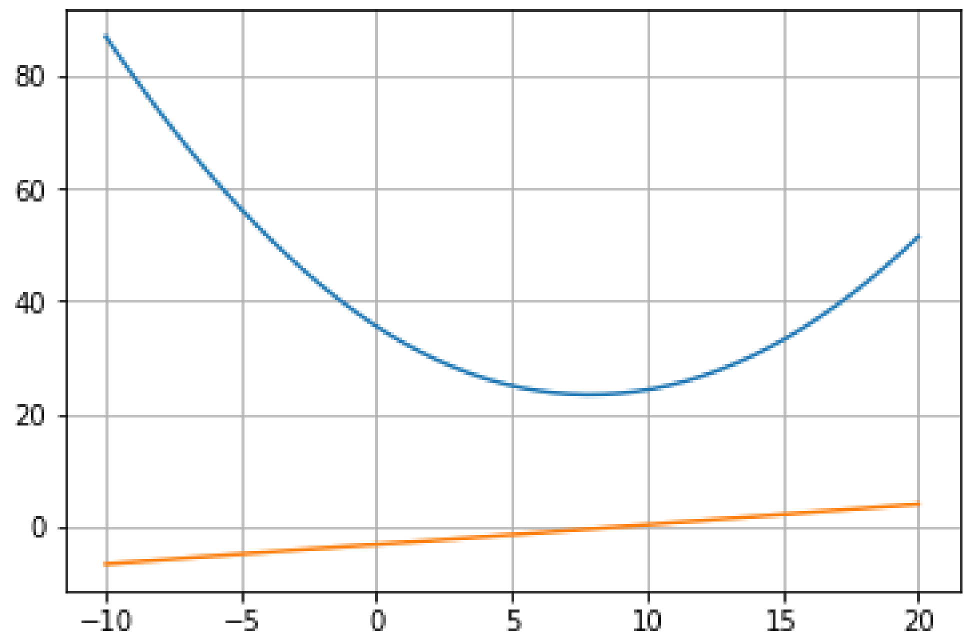
```
Optimization terminated successfully.
         Current function value: 0.578411
         Iterations: 14
         Function evaluations: 102
         Gradient evaluations: 17
[ 0.69191621  0.82063262  1.00013513 -3.21728479]
```

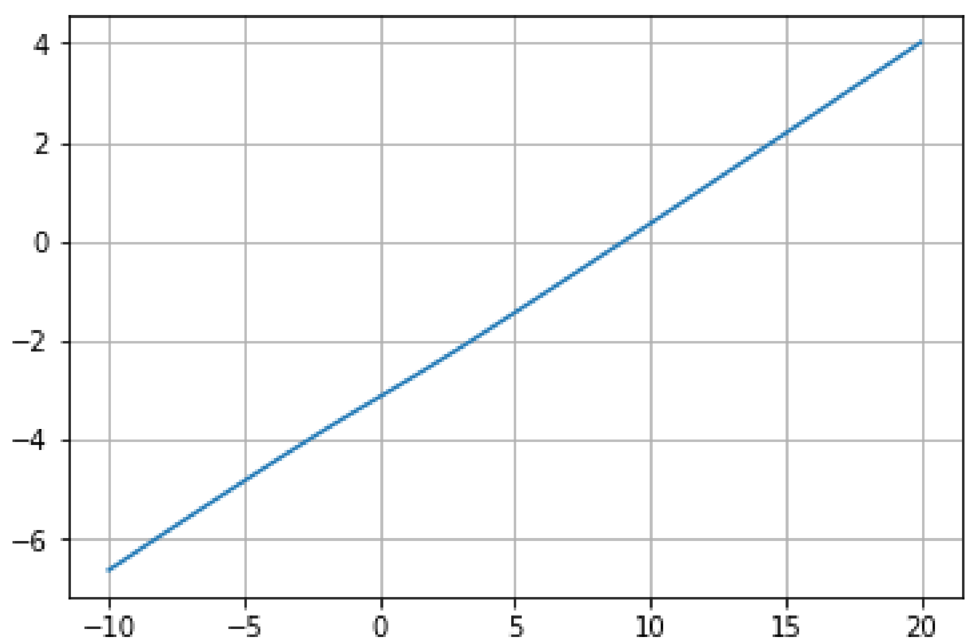
In [62]:
```
args = (F_mem_grad,x0,10,10)
```

In [63]:
```
f = []
fg = []
iaxis = 1
x0 = np.array([[0.,0.],[0.,0.]]).flatten()
ax = np.linspace(-10,20)
lenax = len(ax)
f = np.zeros(lenax)
fg = np.zeros((len(x0), lenax))
for indx, xx in enumerate(ax):
    x0[iaxis] = xx
    f0, fg0 = F_mem_grad(x0,120,100)
    f[indx] = f0
    fg[:,indx] = fg0
```

In [64]:
```
plt.plot(ax,f), plt.plot(ax, fg[iaxis,:])
plt.grid(True)
```

```
plt.show()
plt.plot(ax, fg[iaxis,:])
plt.grid(True)
plt.show()
```





```
In [65]: # Make reconstruction of the components
         if (not multithreaded):

             max_count = nx*ny-1

             fprog = IntProgress(min=0, max=max_count) # instantiate the bar
             display(fprog) # display the bar
             start_time = time.time()
             for i in range(nx):
                 for j in range(ny):
                     ibin = bins[i,j]
                     xmin = opt.minimize(F_mem, x0, method = 'BFGS', jac =F_mem_deriv , args=(
                     xmin = xmin[:nc] + 1j*xmin[nc:]
                     comprec[:,i,j] = xmin
         #            print(i, j, ibin, comprec[0,i,j])
                     fprog.value += 1
             elapsed_time = time.time() - start_time
             print("Single-threaded minimization took ", elapsed_time, " sec")

In [66]: F_mem_grad(x0,10,10)

Out[66]: (13.6529436421852, array([ 1.82736657,  0.18359874, -0.73829923, -0.25858483]))

In [67]: if multithreaded :
             nthreads = 8
             args = []
             for i in range(nx):
                 for j in range(ny):
                     args.append((F_mem_grad,x0,i,j))
         #            args.append((F_chi2_grad,x0,i,j))

             start_time = time.time()
             p = Pool(nthreads)
             datarec = p.map(F_mem_grad_minimize,args)
             elapsed_time = time.time() - start_time
             print("Multi-threaded minimization took ", elapsed_time, " sec")

             for dataelement in datarec:
                 i = dataelement[0]
                 j = dataelement[1]
                 comprec[:,i,j] = dataelement[2]


40 0 [-1.75653496+1.83366314j 20.00853103+0.15827204j] ForkPoolWorker-6
0 0 [ 0.+0.j 20.+0.j] ForkPoolWorker-1
0 10 [ 0.97705745+2.08037433j -0.44516113-7.13718027j] ForkPoolWorker-1
40 10 [-0.08808859+1.96735273j 20.24628989+0.12837728j] ForkPoolWorker-6
0 20 [-0.67380536+0.52376383j 20.23201825-0.00300874j] ForkPoolWorker-1
40 20 [-5.18099382-0.90207123j 18.96893892-0.17269538j] ForkPoolWorker-6
```

```
0 30 [ 0.+0.j 20.+0.j] ForkPoolWorker-1
40 30 [-2.88543909+1.14020918j 19.90556633+0.1397837j ] ForkPoolWorker-6
0 40 [-2.70081115-1.82026898j 20.01577308-0.12417366j] ForkPoolWorker-1
40 40 [-4.55187214+2.17136541j 19.90916241+0.24470554j] ForkPoolWorker-6
40 50 [-4.94284332-0.76818651j 19.32538291-0.15994114j] ForkPoolWorker-6
0 50 [-2.20799031+0.1100273j  20.14889773-0.00629522j] ForkPoolWorker-1
40 60 [-1.36024724+2.07645286j  5.34590175-4.60416074j] ForkPoolWorker-6
0 60 [-3.45334459-0.18832158j  9.46050215+4.84098607j] ForkPoolWorker-1
40 70 [-2.30212636-1.41334086j 20.04580579-0.17718869j] ForkPoolWorker-6
40 80 [-4.92808261-1.06899112j 19.73056098-0.03409353j] ForkPoolWorker-6
0 70 [-2.69852046-2.98957653j 20.11252578-0.29892722j] ForkPoolWorker-1
40 90 [-2.37670588+0.38490448j 15.14644279+1.52833555j] ForkPoolWorker-6
0 80 [-3.68778162-1.05555847j 11.33821388+1.28975312j] ForkPoolWorker-1
40 100 [0.16559078+1.88105281j 0.79438827-5.97461846j] ForkPoolWorker-6
0 90 [-3.30419428+0.15130674j 19.50246029-0.00350855j] ForkPoolWorker-1
0 100 [-3.69625407-1.559485j    6.27190903+4.84821408j] ForkPoolWorker-1
40 110 [ 1.23127476-1.51135085j -3.63854182+1.70114849j] ForkPoolWorker-6
0 110 [-5.35736753+0.36927528j 19.01140136-0.10461088j] ForkPoolWorker-1
40 120 [-4.91648334-1.01577768j 19.4703072 -0.13532472j] ForkPoolWorker-6
0 120 [-0.77521581+0.63948439j 19.98918796+0.10041544j] ForkPoolWorker-1
0 130 [-0.95004022+0.46821806j  1.1517568 -1.14736475j] ForkPoolWorker-1
40 130 [0.29651873+0.51529089j 3.51601986+3.79878277j] ForkPoolWorker-6
0 140 [2.04457015 +4.79559089j 0.6202474 -11.56618788j] ForkPoolWorker-1
40 140 [ 2.26823821+0.51186714j -5.93662697-6.76869191j] ForkPoolWorker-6
0 150 [ 1.36967561+0.57384626j -0.66736021-1.45152972j] ForkPoolWorker-1
40 150 [0.01563078+1.60275498j 1.97751691+0.19064571j] ForkPoolWorker-6
40 160 [-2.27864431+0.88270205j 17.22159149+1.42265502j] ForkPoolWorker-6
0 160 [-10.09217647-1.73542765j  14.44784846-0.94401648j] ForkPoolWorker-1
0 170 [-5.89509341+0.31505117j 17.62652561-0.1555576j ] ForkPoolWorker-1
40 170 [-6.7928255 +0.05081317j 16.60440945-0.80307327j] ForkPoolWorker-6
0 180 [-1.85342844-0.61449285j 20.78443795-0.00820658j] ForkPoolWorker-1
40 180 [-3.90459665-1.55367471j 19.51797635-0.23395907j] ForkPoolWorker-6
0 190 [-3.00051715+0.1089836j  20.04598634+0.01083082j] ForkPoolWorker-1
40 190 [-1.00384193-0.10687282j 20.00671047-0.03057053j] ForkPoolWorker-6
0 200 [-3.72646955+1.54777502j 19.99694434+0.19851414j] ForkPoolWorker-1
40 200 [-0.83725645-1.22270623j  5.44579117+0.01052731j] ForkPoolWorker-6
0 210 [-6.33067867-1.04772423j 17.62990413+0.07959669j] ForkPoolWorker-1
40 210 [-4.88971011+0.83343901j 19.20242095-0.07665833j] ForkPoolWorker-6
0 220 [-7.24651165-1.65752444j 16.92150429-0.74158328j] ForkPoolWorker-1
40 220 [-5.28387096-1.62721607j 18.80743571-0.45660215j] ForkPoolWorker-6
0 230 [-3.23203388+1.84767587j  1.83197939-1.81222725j] ForkPoolWorker-1
40 230 [-6.68351857-1.04263991j 18.31539904+0.07448228j] ForkPoolWorker-6
0 240 [-0.58147987+3.98329351j 20.20821521+0.27110962j] ForkPoolWorker-1
40 240 [-3.07882197+0.24322463j 19.83224435+0.07581782j] ForkPoolWorker-6
0 250 [-3.8911944 -1.23015691j 19.45339374-0.13510243j] ForkPoolWorker-1
40 250 [-5.72903118-1.57216826j 19.53471084-0.45455247j] ForkPoolWorker-6
50 0 [-1.06605893+1.02322385j 20.21946531+0.06641511j] ForkPoolWorker-7
50 10 [-1.14124715+1.61313909j 19.95157916+0.41640444j] ForkPoolWorker-7
```

```
50 20 [-6.12449529-1.49867725j 19.79485423+5.19272463j] ForkPoolWorker-7
10 0 [-3.42900406+0.39746767j 12.60177326-1.21376951j] ForkPoolWorker-2
50 30 [-2.5506819 +2.31110991j 11.47319146-2.66444126j] ForkPoolWorker-7
10 10 [-3.93932719+1.59158117j 19.60420885+0.55744166j] ForkPoolWorker-2
50 40 [-3.7272447 +0.0714168j  19.64216018+0.13218132j] ForkPoolWorker-7
10 20 [-1.45437213+0.3454244j  4.83777877+3.9048313j] ForkPoolWorker-2
50 50 [-2.71729726-0.47560585j 19.68920966-0.1293598j ] ForkPoolWorker-7
10 30 [-7.5393217 -0.41357263j 16.76308105-0.31112385j] ForkPoolWorker-2
50 60 [-0.58518523+0.62409191j  3.26774108-1.1524723j ] ForkPoolWorker-7
10 40 [-2.061079  +0.01322389j 19.94374449-0.12014692j] ForkPoolWorker-2
50 70 [-3.16004504-1.32250211j  6.59236634+3.08639016j] ForkPoolWorker-7
10 50 [-2.70466639-0.81181799j 19.75127972-0.22962185j] ForkPoolWorker-2
50 80 [-0.33598621+1.65930714j  2.98307303-3.23588224j] ForkPoolWorker-7
10 60 [-2.73554541-0.39463648j 19.59217826-0.18202693j] ForkPoolWorker-2
50 90 [1.30549323-0.10494188j 1.53857012+2.27998626j] ForkPoolWorker-7
10 70 [-5.82061539+1.8738929j  18.79519363+0.37215771j] ForkPoolWorker-2
50 100 [0.84510141+0.65965441j 0.73635211-2.65696275j] ForkPoolWorker-7
10 80 [-2.13934839+0.7163905j  20.00578022-0.01342094j] ForkPoolWorker-2
50 110 [ 0.03416732-0.77185797j -0.40456641+6.12275241j] ForkPoolWorker-7
10 90 [-0.15769161+0.29405949j -1.73676288-0.25004407j] ForkPoolWorker-2
50 120 [-0.70889179+2.14101697j  1.31207849-2.45299375j] ForkPoolWorker-7
10 100 [-3.47725251+0.45676034j 18.99282654+0.36183304j] ForkPoolWorker-2
50 130 [ 1.6885301 +1.08325863j -0.16089245-2.30154912j] ForkPoolWorker-7
10 110 [-4.10892051-1.42150388j 18.82442609-0.26674926j] ForkPoolWorker-2
50 140 [ 1.14390551-1.70875312j -1.8903213 -0.85431723j] ForkPoolWorker-7
10 120 [-4.11301344-0.75269198j  6.82428842-2.14688839j] ForkPoolWorker-2
50 150 [-3.04474195+3.13138949j  5.15729405-5.24158288j] ForkPoolWorker-7
50 160 [-0.52437163-0.3333324j  2.13323966-0.2849496j] ForkPoolWorker-7
10 130 [-4.82489278+0.46166605j 17.67248331+0.71237472j] ForkPoolWorker-2
50 170 [ 0.49401341+1.60472297j -0.29406298-4.2610506j ] ForkPoolWorker-7
10 140 [-4.18590405+2.22237928j  6.30407379-1.82503758j] ForkPoolWorker-2
50 180 [ 0.13048503+0.63320905j -3.57709427+0.81487121j] ForkPoolWorker-7
10 150 [-2.45165378+1.27723991j 11.50037565-6.52369885j] ForkPoolWorker-2
50 190 [ 3.27634353+0.87711485j -2.92126493-2.13283704j] ForkPoolWorker-7
10 160 [ 0.38159953 +2.22092696j -1.52844605-12.84691948j] ForkPoolWorker-2
10 170 [-4.82240171-1.12606256j 19.07636377-0.35471249j] ForkPoolWorker-2
50 200 [-0.85558666+4.50774433j  1.32680666-5.41485613j] ForkPoolWorker-7
10 180 [-3.90167808-0.09404829j 19.27603095-0.04293142j] ForkPoolWorker-2
50 210 [-6.33298193-0.74254711j 18.02676201-0.2659894j ] ForkPoolWorker-7
10 190 [-4.97630558 -3.33486062j  8.93452856+13.51607728j] ForkPoolWorker-2
50 220 [-1.1786814 +2.33981955j  1.08638911-7.37755365j] ForkPoolWorker-7
10 200 [-5.70423054+1.27275847j 18.2096864 +0.48347163j] ForkPoolWorker-2
50 230 [-3.87793832-0.38452003j 19.24064535-0.11146496j] ForkPoolWorker-7
10 210 [-1.46056073+0.80668712j 19.97628185+0.11372022j] ForkPoolWorker-2
50 240 [ 0.5692679  -3.63886988j -0.29981536+18.07632173j] ForkPoolWorker-7
10 220 [-0.14547086+1.04856113j  7.93047108-2.45374314j] ForkPoolWorker-2
50 250 [-7.74786547-0.97424399j 14.02741629+2.10365593j] ForkPoolWorker-7
10 230 [-3.8582    +3.47092154j 11.83076455-7.07650208j] ForkPoolWorker-2
```

```
10 240 [-2.30272021-1.01760015j 19.68131895-0.11631581j] ForkPoolWorker-2
10 250 [ 0.+0.j 20.+0.j] ForkPoolWorker-2
20 0 [ 0.87318683+0.14509962j 20.32680696-0.01053581j] ForkPoolWorker-3
60 0 [-4.82975947-0.40368408j 18.61239558-0.29583198j] ForkPoolWorker-8
20 10 [-10.1336794 +0.09139933j  12.14379312-0.1805146j ] ForkPoolWorker-3
20 20 [-2.10725728-0.64925754j 19.99430462-0.10363535j] ForkPoolWorker-3
60 10 [-3.79791371+0.67272336j 19.46610039-0.00258677j] ForkPoolWorker-8
20 30 [-4.53355305-0.23272627j 18.6446305 +0.05858636j] ForkPoolWorker-3
60 20 [-4.69528936-0.39454359j 18.18554139-0.09398467j] ForkPoolWorker-8
20 40 [-4.92884991-0.1547301j  18.92036402-0.13922413j] ForkPoolWorker-3
60 30 [-1.89530938-0.62865878j 19.76222436-0.10201414j] ForkPoolWorker-8
20 50 [-3.78928723-0.38926492j 18.85031526-0.03013009j] ForkPoolWorker-3
60 40 [-5.66481366-0.17792571j 18.04576658+0.05426577j] ForkPoolWorker-8
20 60 [-6.42379257+0.11182225j 17.70903339+0.02147245j] ForkPoolWorker-3
60 50 [-3.12477486-0.06555388j  7.81497332-2.57521688j] ForkPoolWorker-8
20 70 [-1.34555086+1.7477231j   1.15565268-5.37138646j] ForkPoolWorker-3
60 60 [-0.97255082+0.26141017j 20.06449452+0.04183393j] ForkPoolWorker-8
60 70 [-2.61345193+1.74944919j 13.01005516-9.05211154j] ForkPoolWorker-8
20 80 [-5.56223099+0.38087721j 18.55938906+0.29440535j] ForkPoolWorker-3
60 80 [-3.04371847 +2.65080708j 10.505232  -10.52013675j] ForkPoolWorker-8
20 90 [-3.37557697-2.28529195j  7.84478267+3.49054488j] ForkPoolWorker-3
20 100 [-2.92823255+0.64350834j 19.24831446+0.07955649j] ForkPoolWorker-3
60 90 [0.93196236-1.36828353j 0.78287595+4.43962063j] ForkPoolWorker-8
20 110 [-6.48184195+0.55893767j 18.301447  -0.0892682j ] ForkPoolWorker-3
60 100 [ 0.00376723+3.08262089j -1.28974947-5.40686639j] ForkPoolWorker-8
20 120 [-3.75892987+0.15757849j 19.1397092 +0.0662085j ] ForkPoolWorker-3
60 110 [ 1.92495893 -3.38899295j -3.84145266+26.82399636j] ForkPoolWorker-8
20 130 [-0.60229723-0.09743979j -0.10631394-0.09552296j] ForkPoolWorker-3
60 120 [-1.31673409+1.36062767j  2.7497988 -0.26881992j] ForkPoolWorker-8
20 140 [-5.61879342-3.29140785j 13.24732193+3.54094769j] ForkPoolWorker-3
60 130 [-0.99756945-0.51670612j  0.72084253-3.01628076j] ForkPoolWorker-8
20 150 [ 0.67578042-0.81492487j -3.57926643+0.50303004j] ForkPoolWorker-3
60 140 [-0.19623054-2.007872j   20.03206958-0.09651511j] ForkPoolWorker-8
20 160 [-2.52902116 +1.86105419j  3.01561834-10.74562506j] ForkPoolWorker-3
60 150 [-0.90890328+1.12119834j  0.4659714 -2.82239382j] ForkPoolWorker-8
20 170 [-1.45939149-0.01422312j 20.11491595-0.08155701j] ForkPoolWorker-3
60 160 [-4.08349069-0.95110576j 19.13694848-0.1498082j ] ForkPoolWorker-8
20 180 [-2.43678563-0.60355816j 20.08423797-0.02781656j] ForkPoolWorker-3
60 170 [-2.36321893+2.39231796j  7.10112423-8.68775784j] ForkPoolWorker-8
20 190 [-6.78663917-0.3681684j  17.37354257-0.15903851j] ForkPoolWorker-3
60 180 [-6.06460462-1.30017665j 18.05155887-0.28075524j] ForkPoolWorker-8
20 200 [-5.80454351+1.73700018j 18.49274537+0.39833184j] ForkPoolWorker-3
60 190 [-0.23785867 +3.01487291j  8.1368865 -14.59861403j] ForkPoolWorker-8
20 210 [-3.49070696+2.349379j   19.10035304+0.43983368j] ForkPoolWorker-3
60 200 [ 0.48512398-0.34701836j -1.3176813 -2.50260052j] ForkPoolWorker-8
20 220 [-1.39478109+0.31508321j 20.16558665-0.03006873j] ForkPoolWorker-3
60 210 [-1.91178142+1.19270887j 19.70019309+0.26396475j] ForkPoolWorker-8
20 230 [-5.52977975-0.79890674j 18.98466459-0.03373782j] ForkPoolWorker-3
```

```
60 220 [-2.80386793-0.4773124j  19.77908151-0.08426445j] ForkPoolWorker-8
20 240 [-2.01220338+0.15174284j 19.92156156-0.04752966j] ForkPoolWorker-3
60 230 [0.19227065 -2.59134084j 2.93663769+10.14176111j] ForkPoolWorker-8
20 250 [-3.68338149-0.96707193j 19.64141482-0.11461808j] ForkPoolWorker-3
60 240 [0.12322319 -3.75530718j 0.15539912+10.27921023j] ForkPoolWorker-8
60 250 [-3.78618174-1.95324586j 19.72068176-0.32174405j] ForkPoolWorker-8
30 0 [-4.02418855+0.96446991j 20.93403811+0.16324148j] ForkPoolWorker-4
30 10 [-3.64719853+0.15627044j 11.65043908+2.61844943j] ForkPoolWorker-4
30 20 [-5.20679721+0.13297431j 18.64631509+0.07947556j] ForkPoolWorker-4
30 30 [-7.70102317+0.38052022j 17.38353228+0.24727818j] ForkPoolWorker-4
30 40 [-2.01466808-0.63896383j 20.04381043-0.01109082j] ForkPoolWorker-4
30 50 [-3.77459698 +0.24564878j 14.56962173+10.8355309j ] ForkPoolWorker-4
30 60 [-4.02834703+1.67798583j 19.59759694+0.32272708j] ForkPoolWorker-4
30 70 [-4.46975142-0.32181521j 18.88484654-0.08413333j] ForkPoolWorker-4
30 80 [-4.23095615+0.40886587j 19.00130036-0.08604493j] ForkPoolWorker-4
30 90 [-2.61764719-0.28628837j 19.78426562-0.00792724j] ForkPoolWorker-4
30 100 [-3.6764791 +1.70215255j 19.29055248+0.41573318j] ForkPoolWorker-4
30 110 [-3.77126703-2.2770508j   9.24013174+5.89803455j] ForkPoolWorker-4
30 120 [-3.75067281-2.4801865j   6.99558851+2.69719459j] ForkPoolWorker-4
30 130 [ 1.46209699-3.20480342j -3.72929328+9.91051796j] ForkPoolWorker-4
30 140 [-0.87158998-1.41136823j 19.95825418+0.28051087j] ForkPoolWorker-4
30 150 [-4.59914652+0.4724871j  19.19120114+0.12531703j] ForkPoolWorker-4
30 160 [-1.86886535+2.93991821j  6.48809325-6.55569488j] ForkPoolWorker-4
30 170 [-4.87616048+0.71295535j 18.9238306 +0.24313186j] ForkPoolWorker-4
30 180 [-0.35628903 -3.50291942j -0.95647866+11.85383803j] ForkPoolWorker-4
30 190 [-2.11653289-0.36343569j 19.48872749-0.02562136j] ForkPoolWorker-4
30 200 [-6.30632526-1.17517515j 16.5790541 +0.44554721j] ForkPoolWorker-4
30 210 [-4.76252356-1.57077743j 19.45852785-0.24563293j] ForkPoolWorker-4
30 220 [-3.75642752+1.36956368j 19.00825329+0.23172316j] ForkPoolWorker-4
30 230 [-3.99803711-0.51064086j 19.3853574 -0.09497896j] ForkPoolWorker-4
30 240 [-3.56157771+0.64121324j 19.62753111+0.03505795j] ForkPoolWorker-4
30 250 [-4.53644112-0.63206463j 19.26988173+0.05009579j] ForkPoolWorker-4
80 0 [-0.08689115+0.60015195j  5.86002276-0.56640541j] ForkPoolWorker-7
80 10 [-1.89257809+2.17590724j 20.04377285+0.21212164j] ForkPoolWorker-7
80 20 [-2.73991383-1.12736591j 19.2928661 -0.18446413j] ForkPoolWorker-7
80 30 [-4.01628111-1.3528896j  19.15350002-0.16600961j] ForkPoolWorker-7
80 40 [-3.06422491+0.07318917j 19.8395093 +0.10651519j] ForkPoolWorker-7
80 50 [-2.53602789-2.9285992j   8.64872618+9.84620172j] ForkPoolWorker-7
80 60 [-1.21244849 +3.81495748j -2.04204917-15.01990137j] ForkPoolWorker-7
120 0 [-2.5356491 -0.11900857j 19.76544937-0.04463865j] ForkPoolWorker-2
80 70 [-0.30745177+0.63957839j  1.23982494-3.91172469j] ForkPoolWorker-7
120 10 [-4.95959717-2.10891331j 13.33544252+8.21370909j] ForkPoolWorker-2
80 80 [ 1.11391608-0.57832735j 20.11821734-0.00385631j] ForkPoolWorker-7
120 20 [-3.02487169+2.37973037j 19.17640672+0.55118463j] ForkPoolWorker-2
80 90 [-1.84711687+0.60792795j  4.0173303 -2.28376928j] ForkPoolWorker-7
120 30 [-11.00861608-1.65936733j  12.07167552-0.59700216j] ForkPoolWorker-2
80 100 [-0.02229938-1.0960233j  -3.08211035+1.86909027j] ForkPoolWorker-7
120 40 [-3.56311107-0.81146546j 19.58004154-0.13572184j] ForkPoolWorker-2
```

```
80 110 [-0.20913775+0.77717572j  4.21383711-3.29297215j] ForkPoolWorker-7
120 50 [-2.18215188+0.68428961j  3.6015548 -1.80351681j] ForkPoolWorker-2
80 120 [0.23365358-0.57701223j 2.43538845+0.4175602j ] ForkPoolWorker-7
120 60 [-0.0805072 -0.24878773j  0.72250144+1.57259989j] ForkPoolWorker-2
80 130 [ 1.54542127-1.55834955j -5.56436748+3.28433332j] ForkPoolWorker-7
120 70 [ 1.44731265-0.35820957j -1.46154256+1.34621839j] ForkPoolWorker-2
80 140 [ 0.74313546+0.28993347j -0.15983295+4.08360383j] ForkPoolWorker-7
120 80 [ 2.09515765-1.09826246j -5.39760262+0.80611055j] ForkPoolWorker-2
80 150 [-0.02725071+0.03465012j -0.53710961-0.50976411j] ForkPoolWorker-7
120 90 [ 0.91049948-1.01273619j -0.3524036 +1.55410989j] ForkPoolWorker-2
80 160 [-1.80978095-0.46284655j  6.73005257-2.4910701j ] ForkPoolWorker-7
120 100 [0.56560421+1.12889857j 2.68615696-5.0830866j ] ForkPoolWorker-2
80 170 [0.20216782+0.67207137j 8.69865971-3.64112294j] ForkPoolWorker-7
120 110 [-0.79607216-0.62135896j -0.01990586+1.20828661j] ForkPoolWorker-2
80 180 [ 1.19084614 -2.77271683j -5.56077104+11.89910569j] ForkPoolWorker-7
120 120 [-1.73804313+0.36291606j  1.17779124-2.41745458j] ForkPoolWorker-2
80 190 [-2.80390267-0.70432269j  3.09982347-1.87918736j] ForkPoolWorker-7
120 130 [0.22503846-0.58456792j 2.24754131+3.25372421j] ForkPoolWorker-2
80 200 [-2.3712885 +0.18582932j 19.85457051-0.06931543j] ForkPoolWorker-7
120 140 [0.19369258-1.01617519j 0.27837329+0.34791609j] ForkPoolWorker-2
80 210 [0.72452766+1.33684002j 1.91259479-3.47506379j] ForkPoolWorker-7
120 150 [ 0.19462178+1.79993498j -3.23924931-2.54090264j] ForkPoolWorker-2
80 220 [ 0.94873031-0.24632239j 20.24066673-0.03762439j] ForkPoolWorker-7
120 160 [-1.20834068+0.63237825j  1.03386627-1.90198525j] ForkPoolWorker-2
80 230 [-0.93516255-0.34551287j 20.15716317+0.03862132j] ForkPoolWorker-7
120 170 [-0.95809274+0.49402689j 19.98586126+0.03103428j] ForkPoolWorker-2
80 240 [-2.48696727-1.07216338j 19.98863411-0.13307889j] ForkPoolWorker-7
120 180 [-2.12345919-0.86105458j 19.9017704 -0.0566472j ] ForkPoolWorker-2
120 190 [-2.41526554-1.83902272j  6.14981094+1.64214532j] ForkPoolWorker-2
80 250 [-0.45747905+0.75438699j 20.24536863-0.04945165j] ForkPoolWorker-7
120 200 [-2.87318725-0.61729419j 19.44881089-0.14276562j] ForkPoolWorker-2
120 210 [  3.82823929+2.72213727j -20.24518724-9.88718406j] ForkPoolWorker-2
120 220 [ 0.22865569-0.39593199j -1.04233768+1.44649208j] ForkPoolWorker-2
120 230 [-5.12834166-3.29880088j 10.22682434+3.50862426j] ForkPoolWorker-2
120 240 [ 0.40093153-0.41747099j -4.52896567+2.58900635j] ForkPoolWorker-2
120 250 [-2.61429706-0.50604664j  9.9722218 -7.93096103j] ForkPoolWorker-2
90 0 [-4.92847789+0.9068046j  19.38256305+0.09530326j] ForkPoolWorker-4
90 10 [-2.33198021-2.2321001j   2.30111378+2.72296541j] ForkPoolWorker-4
90 20 [-3.30188304-0.36539269j 10.16813785+0.07536728j] ForkPoolWorker-4
90 30 [-2.45880588+0.48881268j 19.74658701+0.04880545j] ForkPoolWorker-4
90 40 [-2.54918436+1.02978309j 19.72369308+0.10602739j] ForkPoolWorker-4
90 50 [ 1.20433928-0.88020649j -1.14303841+1.05938609j] ForkPoolWorker-4
90 60 [0.0097823 +1.69676952j 1.70343529-4.98386089j] ForkPoolWorker-4
90 70 [-2.29537856+0.80283522j 19.72343671+0.09843268j] ForkPoolWorker-4
90 80 [-0.87991677+0.04438511j -1.99460406-1.92122386j] ForkPoolWorker-4
90 90 [ 0.13775659-0.02883648j -1.26145119+0.90847469j] ForkPoolWorker-4
90 100 [1.26657607-0.09833195j 0.20002519+2.65690991j] ForkPoolWorker-4
90 110 [ 0.45256506-0.81799144j -2.31951691+4.07385906j] ForkPoolWorker-4
```

```
90 120 [-0.36217115+1.06785395j  2.54180482-2.77818496j] ForkPoolWorker-4
90 130 [ 0.57420993-0.25073328j -0.42377996+0.45205237j] ForkPoolWorker-4
90 140 [-1.78673557+0.32981114j 19.86832175+0.01243412j] ForkPoolWorker-4
90 150 [-0.76996065-0.20590009j -0.09179181+0.35023707j] ForkPoolWorker-4
90 160 [0.38945493-0.30784856j 0.02051162+0.51145804j] ForkPoolWorker-4
90 170 [0.43860287-0.71991027j 0.5371744 -2.95133056j] ForkPoolWorker-4
90 180 [-0.53543278+0.8438205j   0.14905372-2.57192733j] ForkPoolWorker-4
90 190 [-2.68588529-0.73602944j 19.74793917-0.02897788j] ForkPoolWorker-4
90 200 [-1.97519674+2.64005401j  3.43286813-3.98431802j] ForkPoolWorker-4
90 210 [-1.05688805+0.40331704j  7.60916201-5.06110247j] ForkPoolWorker-4
90 220 [-1.82495946+1.62284627j 19.76008968+0.24976027j] ForkPoolWorker-4
90 230 [-1.75338343-3.35602887j  7.83940216+9.425559j  ] ForkPoolWorker-4
90 240 [-0.58048588-0.03486392j  2.54662504-0.08905414j] ForkPoolWorker-4
90 250 [-4.23451277-0.65008152j 19.53458003-0.22383821j] ForkPoolWorker-4
100 0 [ 3.65546952-1.26290256j -5.75768133+3.0479046j ] ForkPoolWorker-1
100 10 [-3.02301649+2.39301946j  8.82070957-2.89136501j] ForkPoolWorker-1
100 20 [-3.39181177+0.63473099j 19.38015803+0.14774641j] ForkPoolWorker-1
100 30 [-3.31520392+1.15514364j 19.57315512+0.11864337j] ForkPoolWorker-1
100 40 [ 1.12020205+3.01236892j -1.66862373-5.66154882j] ForkPoolWorker-1
100 50 [-1.73967181+2.35506111j  0.98619733-8.67354165j] ForkPoolWorker-1
100 60 [-0.88751791+1.90080776j  6.28133061+3.67562082j] ForkPoolWorker-1
100 70 [ 0.60055569+1.88934657j -5.30547052-4.62663847j] ForkPoolWorker-1
100 80 [0.34421216-2.30994879j 0.70583724+5.9273455j ] ForkPoolWorker-1
100 90 [0.96120877-1.82520075j 0.57733334+6.66468488j] ForkPoolWorker-1
100 100 [-0.51877027 +1.62648117j -0.87897415-10.60293375j] ForkPoolWorker-1
100 110 [-1.98263325-0.18558897j 14.16237769+1.71297308j] ForkPoolWorker-1
100 120 [-1.5309305 -0.57182012j 13.08614504+7.26884936j] ForkPoolWorker-1
100 130 [ 1.38962115+0.34198735j -9.582247  -1.13916786j] ForkPoolWorker-1
100 140 [-0.56667834+0.75801362j  4.24274801+5.1158016j ] ForkPoolWorker-1
100 150 [-0.23773206+0.73178767j  3.76126709-2.44093451j] ForkPoolWorker-1
100 160 [-0.41052175+1.41486142j  1.02165991-4.43609757j] ForkPoolWorker-1
100 170 [-1.89000911-1.95507885j  6.23807117+6.36512484j] ForkPoolWorker-1
100 180 [-2.19406195-0.33132944j 19.61297899-0.02425866j] ForkPoolWorker-1
100 190 [0.58640036+0.97454298j 0.11663465-1.18591179j] ForkPoolWorker-1
100 200 [-2.38065542-0.75966165j 19.84135365-0.01981993j] ForkPoolWorker-1
100 210 [-0.04477827 -4.23267264j  1.62692926+11.56870777j] ForkPoolWorker-1
100 220 [-1.0083369 +0.01594899j 20.05549112-0.00480314j] ForkPoolWorker-1
100 230 [ 1.87653011+0.63882276j -3.23917415+0.93499412j] ForkPoolWorker-1
100 240 [-0.09220728-0.63553263j  5.44767326+1.2856604j ] ForkPoolWorker-1
100 250 [-5.79653141+0.93421137j 18.59767393+0.43437217j] ForkPoolWorker-1
70 0 [-0.9448802+0.83908598j 20.2347158+0.00421695j] ForkPoolWorker-5
110 0 [-4.5436789 +4.38249487j  9.36127105-6.35389364j] ForkPoolWorker-8
70 10 [-4.34170483-0.46603283j 19.4557117 -0.03412126j] ForkPoolWorker-5
110 10 [-2.94503784+0.7536881j  8.26858241-4.718237j ] ForkPoolWorker-8
70 20 [-1.85913108-1.8455976j   6.33546508+5.24674631j] ForkPoolWorker-5
110 20 [-5.90760021-1.82537785j 18.6572512 -0.69759787j] ForkPoolWorker-8
70 30 [-0.28903826 +4.372528j    1.83947006-12.34579905j] ForkPoolWorker-5
110 30 [-0.96856205-2.25214025j  3.68013159+6.21460988j] ForkPoolWorker-8
```

```
70 40 [-2.76615839-1.13405422j 19.99996167-0.08450125j] ForkPoolWorker-5
110 40 [-3.48630875+0.10963932j 10.54621557+2.3501285j ] ForkPoolWorker-8
70 50 [-2.25061724+1.68317149j 19.47624414+0.27649799j] ForkPoolWorker-5
110 50 [-0.01909618-3.47423713j -0.54398935+5.82341717j] ForkPoolWorker-8
70 60 [0.36171586 +2.68492136j 1.556159  -10.50968918j] ForkPoolWorker-5
110 60 [-3.78624669+1.31946255j 19.35930615+0.24877714j] ForkPoolWorker-8
70 70 [-0.59007511-1.09303285j  1.14851443+1.41706172j] ForkPoolWorker-5
110 70 [ 0.60642808 -1.472799j   -1.33981204+11.00424268j] ForkPoolWorker-8
70 80 [-1.46917936+0.83282014j  6.06085793-6.63952478j] ForkPoolWorker-5
110 80 [-2.76544694-0.72684474j  3.97656054+5.61744759j] ForkPoolWorker-8
70 90 [-1.9279995 +0.54158665j 17.95468086-4.18394005j] ForkPoolWorker-5
110 90 [0.14928151-0.86504949j 0.39036368+3.29930045j] ForkPoolWorker-8
70 100 [-2.2342822+1.27348587j  5.7825384-2.52915805j] ForkPoolWorker-5
110 100 [-2.56464951-1.61593638j  9.03812465+6.82955038j] ForkPoolWorker-8
110 110 [-0.51788682 +0.49544915j  5.39945275+11.67314019j] ForkPoolWorker-8
70 110 [1.1191261 -0.07066033j 1.99440162-3.2489163j ] ForkPoolWorker-5
110 120 [ 1.54612458-0.18354754j -1.30007454+1.81409837j] ForkPoolWorker-8
70 120 [-0.59220679+1.60605986j  3.89655225-4.1597588j ] ForkPoolWorker-5
110 130 [0.46600079-0.09322882j 0.61311891-0.99059175j] ForkPoolWorker-8
70 130 [-0.89992313+2.66161488j  0.5355845 -9.66646297j] ForkPoolWorker-5
110 140 [1.34944658+0.48212167j 1.70126471+5.21693956j] ForkPoolWorker-8
70 140 [-1.52195837-1.98097898j -0.00180249+3.6799114j ] ForkPoolWorker-5
110 150 [-0.59909347+1.44423266j  5.36247471-0.10325671j] ForkPoolWorker-8
110 160 [-0.31700351+0.59140586j -0.57726341-3.24917802j] ForkPoolWorker-8
70 150 [  1.9587798  +1.1424352j  -10.94710245-10.26261478j] ForkPoolWorker-5
110 170 [ 0.11901936 +1.30952369j 15.23488058-11.35320734j] ForkPoolWorker-8
70 160 [ 0.62312376-0.87937028j -2.72460211+1.13778147j] ForkPoolWorker-5
110 180 [0.54317997-0.47171768j 9.04968271+0.92410354j] ForkPoolWorker-8
110 190 [-3.13586853-1.36724015j  9.49717961+5.91788267j] ForkPoolWorker-8
70 170 [-0.92241448+0.77483035j 19.99212042+0.05780341j] ForkPoolWorker-5
110 200 [-0.68421307+3.23956363j  4.17147045-8.92336531j] ForkPoolWorker-8
70 180 [1.09644483-0.2068232j  2.19533262-3.19440636j] ForkPoolWorker-5
110 210 [-2.22023287-2.95567608j 10.18074416+6.40931989j] ForkPoolWorker-8
70 190 [-1.40661977-2.40817611j  4.41865994+8.30884709j] ForkPoolWorker-5
110 220 [0.79098735-1.31566358j 0.34071492+6.66715988j] ForkPoolWorker-8
70 200 [-4.44290194 +7.14597953j 10.35538607-16.8654599j ] ForkPoolWorker-5
110 230 [-4.14426734-0.77060614j 18.69491046-0.12557308j] ForkPoolWorker-8
70 210 [-2.36631226+0.88263864j  6.25769269-6.34774984j] ForkPoolWorker-5
110 240 [-4.45335181 +4.40193645j 11.50606995-12.9265494j ] ForkPoolWorker-8
110 250 [-1.6918928+0.59270987j 20.0934321+0.15590831j] ForkPoolWorker-8
70 220 [-2.1003287  -2.8627976j   7.33383501+11.44479771j] ForkPoolWorker-5
70 230 [-2.85941237-1.27741789j 19.79728156-0.12594765j] ForkPoolWorker-5
70 240 [-7.00235547+1.29443108j 16.89181517+0.73667084j] ForkPoolWorker-5
70 250 [-5.36522362-0.97388812j 18.22465344-0.37270287j] ForkPoolWorker-5
160 0 [-9.28877201-0.10359925j 15.37743182+0.00140057j] ForkPoolWorker-7
160 10 [-0.32112572-0.02044811j  6.3942584 +2.58246761j] ForkPoolWorker-7
160 20 [-1.84412088+0.12271717j  2.32637522+6.53169464j] ForkPoolWorker-7
160 30 [-3.72008657 -6.87568709j  7.80620132+17.08051748j] ForkPoolWorker-7
```

```
160 40 [-2.10502457-0.25472063j 19.55672282+0.12391048j] ForkPoolWorker-7
160 50 [-0.38777797 +2.74758512j  2.42421691-12.08014408j] ForkPoolWorker-7
160 60 [-2.020865  +1.82921099j  7.01240577-0.70936413j] ForkPoolWorker-7
160 70 [1.20077959-0.19732088j 1.44948016+2.2147886j ] ForkPoolWorker-7
160 80 [-2.45218428+0.76602516j  6.09140112-7.78130161j] ForkPoolWorker-7
160 90 [-1.02262556-0.95391237j  0.52973905+1.56103494j] ForkPoolWorker-7
160 100 [-0.94916206+0.90237163j  5.4161789 -0.76989186j] ForkPoolWorker-7
160 110 [0.06278242-0.11723398j 0.08700236-0.5531377j ] ForkPoolWorker-7
160 120 [-0.37105535-0.30092541j  4.0249424 +1.29875639j] ForkPoolWorker-7
160 130 [-0.21059953+0.74492185j -1.53080314-0.39213822j] ForkPoolWorker-7
160 140 [ 1.85819407+0.18588968j -1.36814535+0.42828915j] ForkPoolWorker-7
160 150 [ 0.06525296+0.60389346j -0.68801084-0.76992251j] ForkPoolWorker-7
160 160 [-0.92033369+0.25957852j  1.5117514 +0.85521709j] ForkPoolWorker-7
160 170 [ 0.28622193-1.03990255j -6.70437061+6.63008953j] ForkPoolWorker-7
160 180 [-0.51996237+0.47257701j  6.16616964-7.90316477j] ForkPoolWorker-7
160 190 [1.2166967 +0.46004477j 1.64050302-3.83981536j] ForkPoolWorker-7
160 200 [-0.98106352-0.79197485j -0.84941812+2.28267893j] ForkPoolWorker-7
160 210 [-1.55528725+0.31419658j  7.25146626-0.97408566j] ForkPoolWorker-7
160 220 [-1.53473544+1.89038067j  2.40232446-7.92130439j] ForkPoolWorker-7
160 230 [-4.55455544+0.96429969j 19.57230383+0.07391571j] ForkPoolWorker-7
160 240 [-2.41574935+0.11684109j 19.71013994+0.05892835j] ForkPoolWorker-7
160 250 [-9.09584449-1.01617318j 16.53222892-0.5470606j ] ForkPoolWorker-7
130 0 [-4.67111495+0.45558784j 13.72750647-4.49579752j] ForkPoolWorker-8
130 10 [-6.18644759+0.0045969j  17.57820367+0.01793595j] ForkPoolWorker-8
130 20 [-3.34929634-2.48176911j  4.20604153+5.0105248j ] ForkPoolWorker-8
170 0 [-4.47860484 -3.46401679j 11.16800971+11.11700845j] ForkPoolWorker-1
130 30 [-3.95590578 -3.74970872j  5.76044315+14.71560458j] ForkPoolWorker-8
170 10 [-4.28089164+0.81458534j 19.16916461+0.26678808j] ForkPoolWorker-1
130 40 [ 0.53603459-1.43903322j -1.29680481+5.99922238j] ForkPoolWorker-8
170 20 [-0.48947581+0.4332665j 20.3558678 +0.1180215j] ForkPoolWorker-1
130 50 [-0.98145039-0.98709133j  1.58153644+2.20046951j] ForkPoolWorker-8
170 30 [-4.991912  +0.13135752j 19.25169396-0.07808864j] ForkPoolWorker-1
130 60 [ 0.36270505+1.41680544j -0.89202122-0.80590278j] ForkPoolWorker-8
170 40 [-7.38379167+0.11677772j 16.21131674-0.4280727j ] ForkPoolWorker-1
170 50 [-3.46467859+1.3751825j  13.69672287-6.57727798j] ForkPoolWorker-1
130 70 [-0.17726128-0.84557597j  1.07129203+2.82107504j] ForkPoolWorker-8
170 60 [-2.90273372+1.18148813j  7.35112353-1.5093968j ] ForkPoolWorker-1
130 80 [-0.02055419-0.17804299j -0.11552141+1.49909434j] ForkPoolWorker-8
170 70 [-1.53266045+1.66803512j 19.91539888+0.23717727j] ForkPoolWorker-1
130 90 [-0.88103401-1.47516867j  1.28889072+3.12490912j] ForkPoolWorker-8
170 80 [-3.29033458-1.01556907j 19.66064538-0.09179289j] ForkPoolWorker-1
130 100 [ 0.56407306+0.80062915j -3.88252857+3.15759848j] ForkPoolWorker-8
170 90 [-0.64756884+1.86372719j  3.75187273-1.94502472j] ForkPoolWorker-1
130 110 [-1.00655476-1.17631831j  2.54541948+0.35166937j] ForkPoolWorker-8
170 100 [ 0.08807141-1.03442071j -2.25487257+2.53079675j] ForkPoolWorker-1
130 120 [-0.54896194-1.13887125j  6.7877626 +6.50052986j] ForkPoolWorker-8
170 110 [-1.22106279 +0.55038906j 10.75944734-10.9178871j ] ForkPoolWorker-1
130 130 [-0.36154553-1.28815083j -0.06613389-0.80043579j] ForkPoolWorker-8
```

```
170 120 [ 0.0483392 +0.6107436j  20.00716861+0.02492743j] ForkPoolWorker-1
130 140 [-0.93469189-1.1368609j   5.13779554+3.10203722j] ForkPoolWorker-8
170 130 [-0.26022811+0.07495671j -1.61645192+0.88486254j] ForkPoolWorker-1
130 150 [-0.07604281+0.03745251j  3.79480518-0.90843856j] ForkPoolWorker-8
170 140 [ 0.85019942-1.47609477j -2.23754355+7.24807403j] ForkPoolWorker-1
130 160 [-1.68530427-1.47230574j  0.70645534+0.78573867j] ForkPoolWorker-8
170 150 [ 0.10396632-0.41249238j -1.72599907-1.62974801j] ForkPoolWorker-1
130 170 [ 1.89610218-0.30687042j -5.3202908 +6.34801708j] ForkPoolWorker-8
170 160 [0.14178245-1.81006209j 1.02966882+3.44062684j] ForkPoolWorker-1
170 170 [-0.46540655+0.44532611j -2.46164255-2.58626918j] ForkPoolWorker-1
130 180 [-0.93846977-2.65248134j  0.79433861+4.35501926j] ForkPoolWorker-8
170 180 [ 1.16724186-1.22725369j -1.82960137+3.84540129j] ForkPoolWorker-1
130 190 [-2.49255773+1.76510949j  9.59411499-8.90532596j] ForkPoolWorker-8
170 190 [-1.78103603+0.16115543j 19.79771201+0.04794723j] ForkPoolWorker-1
130 200 [-2.35875692+0.10458519j  8.2223198 +0.52327633j] ForkPoolWorker-8
170 200 [-1.55812444+3.29929149j  6.73085231-7.25539843j] ForkPoolWorker-1
130 210 [-0.121341  -0.3090832j   0.43883171+2.87357979j] ForkPoolWorker-8
170 210 [-4.87332515+0.18244451j 19.12952601+0.1121389j ] ForkPoolWorker-1
130 220 [-4.39255163-1.78901489j 19.21490875-0.37801466j] ForkPoolWorker-8
170 220 [-1.15670577-1.05027148j 20.05010446-0.07283883j] ForkPoolWorker-1
130 230 [-1.78959704+0.02749279j 19.99701185+0.00235896j] ForkPoolWorker-8
170 230 [-3.183584  -0.62885661j 19.54068901-0.1147632j ] ForkPoolWorker-1
130 240 [-6.97811215+0.61768176j 16.80856022+0.25150568j] ForkPoolWorker-8
170 240 [-1.91730075+0.22122742j 19.99915206+0.049234j  ] ForkPoolWorker-1
130 250 [-5.28703518-0.82263117j 19.33103018-0.25026407j] ForkPoolWorker-8
170 250 [-0.06315744-1.67952411j 20.33790262-0.15249333j] ForkPoolWorker-1
140 0 [-5.90427587+3.27063573j 15.7923326 -4.23043514j] ForkPoolWorker-4
180 0 [ 0.72187634-0.76977489j 20.60308011-0.0175759j ] ForkPoolWorker-5
140 10 [-0.82390463 +6.1998549j  -0.02344448-11.13235425j] ForkPoolWorker-4
180 10 [-3.58275391+0.90139852j 19.4364094 +0.1924182j ] ForkPoolWorker-5
140 20 [ 4.20351605+11.19668102j -8.58349237-17.85881242j] ForkPoolWorker-4
180 20 [-0.86391288-2.55065612j 20.15747558-0.20155171j] ForkPoolWorker-5
140 30 [-1.62751959 +0.93113579j 12.87422053+10.20322298j] ForkPoolWorker-4
180 30 [-0.80568037-1.61432771j  2.51018605+1.9818655j ] ForkPoolWorker-5
140 40 [-1.23017246-0.6119765j   6.23176223+3.24431857j] ForkPoolWorker-4
180 40 [-2.30150335+1.52387304j 19.75891169+0.20345079j] ForkPoolWorker-5
140 50 [-0.24825761+2.23377083j  0.09114228-4.44047128j] ForkPoolWorker-4
180 50 [-0.81011844+2.46483081j  6.60084252-9.33516677j] ForkPoolWorker-5
140 60 [-0.79367052+0.14360637j 19.97077109+0.02787076j] ForkPoolWorker-4
180 60 [-3.60029613 -6.21747711j  6.67946778+16.53007967j] ForkPoolWorker-5
140 70 [-1.24075023-2.27151057j  4.35672322+6.30136895j] ForkPoolWorker-4
180 70 [-0.17849965+0.03472714j  2.63733169-0.73040192j] ForkPoolWorker-5
140 80 [ 1.60558995+0.54617899j -2.60580318-1.2300185j ] ForkPoolWorker-4
180 80 [-2.04321007-1.69416634j  3.83607696+4.51749072j] ForkPoolWorker-5
140 90 [-0.13683832-1.01928981j  7.75950014+9.8877702j ] ForkPoolWorker-4
180 90 [ 0.51708874+0.84398049j -2.46746794+2.18698813j] ForkPoolWorker-5
140 100 [1.10188165-0.68460068j 1.09692031+4.61218876j] ForkPoolWorker-4
180 100 [-0.44198243-0.19661706j  9.71510399+2.10507145j] ForkPoolWorker-5
```

```
140 110 [ 0.12855973-0.0314314j  -2.14237448-0.15731821j] ForkPoolWorker-4
180 110 [-0.20003855+0.42898923j  8.63218655-2.39280158j] ForkPoolWorker-5
140 120 [0.92266488+1.07548449j 1.45106564+3.16720288j] ForkPoolWorker-4
180 120 [-0.89815312-0.03157588j 19.94590639+0.00729464j] ForkPoolWorker-5
140 130 [-0.15644562+0.91542024j  0.34622632-4.81607768j] ForkPoolWorker-4
180 130 [-2.93172727+3.74230242j  7.37545357-5.09468001j] ForkPoolWorker-5
140 140 [ 0.02999485+0.99526783j 20.00119057+0.00386957j] ForkPoolWorker-4
180 140 [-2.0371674 -0.3535569j   5.04552437+1.56052141j] ForkPoolWorker-5
140 150 [-0.7738684 -0.90426944j -0.72138971+8.34848028j] ForkPoolWorker-4
140 160 [0.66479554+0.28573694j 0.96622965-0.15468455j] ForkPoolWorker-4
180 150 [  3.81702416-1.40867694j -20.22455094+1.17803725j] ForkPoolWorker-5
140 170 [-0.88325392-0.02081871j -4.38740001+2.20365484j] ForkPoolWorker-4
180 160 [ 0.75135494-0.26613793j -1.39402318+4.65404025j] ForkPoolWorker-5
140 180 [ 3.0116476 +1.6021779j  -5.22300729-9.39674493j] ForkPoolWorker-4
180 170 [-0.68100929-1.12350244j  5.45463542+7.11392091j] ForkPoolWorker-5
140 190 [ 1.01255006-2.89258442j -5.5028904 +9.31252459j] ForkPoolWorker-4
180 180 [-1.05742835-0.98063662j  2.60978574+6.61847811j] ForkPoolWorker-5
140 200 [ 1.37091023+0.32521879j -1.08883748+3.69180814j] ForkPoolWorker-4
180 190 [0.62180477+0.15010637j 2.96982851-5.97909894j] ForkPoolWorker-5
140 210 [-0.04718425-3.37103006j  7.52928582+5.52298964j] ForkPoolWorker-4
180 200 [ 1.11389857+2.79916571j -3.1712583 -5.88131974j] ForkPoolWorker-5
140 220 [-0.33975531 -6.79664236j -0.54371356+27.05494011j] ForkPoolWorker-4
180 210 [-3.59751564-2.25511949j 19.26315397-0.07764182j] ForkPoolWorker-5
140 230 [-0.49595947+0.18468137j  1.70938548+0.07711273j] ForkPoolWorker-4
180 220 [-1.78378521 -0.77685607j  6.9415515 -14.16759773j] ForkPoolWorker-5
140 240 [-0.69337122+1.48014178j 10.26335506-4.18123317j] ForkPoolWorker-4
180 230 [-1.88573493-1.85317938j  3.36343684+2.49356734j] ForkPoolWorker-5
140 250 [-4.43169402+0.26417447j  9.52780247+5.64224569j] ForkPoolWorker-4
180 240 [-2.17876745+1.08783264j 20.04495694+0.06429652j] ForkPoolWorker-5
180 250 [-4.70751708-1.20194266j 18.97100645-0.22659089j] ForkPoolWorker-5
150 0 [0.86858735-2.61245201j 3.63974553+5.08306332j] ForkPoolWorker-6
150 10 [-2.1120154 -0.03040996j  3.33909505-3.79718812j] ForkPoolWorker-6
150 20 [-3.49846503+1.57730504j  4.596278  +1.93312501j] ForkPoolWorker-6
150 30 [-3.70575346-0.37732493j 19.27666567-0.11357597j] ForkPoolWorker-6
150 40 [-0.12939666+2.56680893j  2.60589631-5.19258697j] ForkPoolWorker-6
150 50 [-0.49281222-0.73221999j  3.80034962-0.78566109j] ForkPoolWorker-6
150 60 [-0.40674177+1.04096287j  2.80491349+3.46775605j] ForkPoolWorker-6
150 70 [ 1.75203845-1.37343091j -5.66775771+4.46669178j] ForkPoolWorker-6
150 80 [ 0.8107574 -0.01472958j -5.20509666+0.42259312j] ForkPoolWorker-6
150 90 [-2.39360252-0.46149629j  8.2788258 +3.24689004j] ForkPoolWorker-6
150 100 [-0.41463186+1.43651911j -0.18898182-2.94124261j] ForkPoolWorker-6
150 110 [0.75399806+0.97990062j 1.55950275-3.85402255j] ForkPoolWorker-6
150 120 [0.49718873-0.02861894j 0.29570149-0.55943819j] ForkPoolWorker-6
150 130 [-0.5884188 +0.02948897j -0.73760764-0.29604032j] ForkPoolWorker-6
150 140 [0.93207262+0.72783441j 4.62274797-5.12393624j] ForkPoolWorker-6
190 0 [-1.2208886 +1.09527679j 20.39957868+0.13850886j] ForkPoolWorker-2
190 10 [-7.14075713+0.47411126j 16.82178115-0.09493611j] ForkPoolWorker-2
150 150 [ 0.57300005+0.6145875j  -0.51589294+1.89166526j] ForkPoolWorker-6
```

```
190 20 [-7.20211973-1.66506645j 17.45249518-0.67945714j] ForkPoolWorker-2
150 160 [-1.00636083-0.14651257j -0.23705559-2.84338148j] ForkPoolWorker-6
190 30 [-6.23843249-0.97018294j 17.66751087-0.39346443j] ForkPoolWorker-2
150 170 [ 0.96384604-1.16514154j -2.86503423+0.72846047j] ForkPoolWorker-6
190 40 [-1.52652721+1.57312518j 20.00647425+0.17221724j] ForkPoolWorker-2
150 180 [-1.7109691 +0.76179152j  4.71253676-9.01937614j] ForkPoolWorker-6
190 50 [-0.51902446-1.55428284j  2.96354665-1.34870284j] ForkPoolWorker-2
150 190 [-0.92885712-0.6355165j   4.7336939 +4.16696468j] ForkPoolWorker-6
190 60 [-1.65785742+0.39819667j  5.16924317-2.7152929j ] ForkPoolWorker-2
150 200 [-3.7101849  -2.42280176j  9.10659276+10.40999463j] ForkPoolWorker-6
150 210 [ 0.35507478-1.27007961j -3.32927997+2.97688548j] ForkPoolWorker-6
190 70 [-0.66719199-1.30127217j  5.07814924+4.5769827j ] ForkPoolWorker-2
150 220 [-2.47855953-0.72418278j  4.36418918+0.79011579j] ForkPoolWorker-6
190 80 [-3.94558531-1.86154628j  4.31683737+2.80108938j] ForkPoolWorker-2
150 230 [-1.69647925+2.16328807j 19.88413931+0.25866888j] ForkPoolWorker-6
190 90 [-1.3137635 -0.23591857j 19.88794267+0.00911633j] ForkPoolWorker-2
150 240 [-0.44616613-0.42520809j 19.95374439-0.02521527j] ForkPoolWorker-6
190 100 [-1.27842934+1.44297509j  1.88802838-1.33843069j] ForkPoolWorker-2
150 250 [-6.14430525-0.85832212j 12.36208014+6.79713738j] ForkPoolWorker-6
190 110 [-0.8009283+1.12082819j  3.3792839+2.54175744j] ForkPoolWorker-2
190 120 [ 1.73219205-1.48528001j -1.88650208+2.15072083j] ForkPoolWorker-2
190 130 [-0.56067659+0.75981829j  0.9733217 -2.07291165j] ForkPoolWorker-2
190 140 [ 1.47837231-1.68353474j -7.15534045+5.49141495j] ForkPoolWorker-2
190 150 [-2.28264467+0.86316484j  4.97369275-3.0023138j ] ForkPoolWorker-2
190 160 [ 0.53642563-1.43837149j -0.34355542+2.91157583j] ForkPoolWorker-2
190 170 [-2.43964615+0.46002542j 19.73885254+0.04507869j] ForkPoolWorker-2
190 180 [-4.81566932-0.60053228j  8.96549649+3.11547672j] ForkPoolWorker-2
190 190 [-0.93795375+0.96752215j 20.07324993+0.05201041j] ForkPoolWorker-2
190 200 [-2.77028577-7.27679352j -0.22097727+8.01981931j] ForkPoolWorker-2
190 210 [-1.89505925-3.13973794j 10.72644203+7.4239044j ] ForkPoolWorker-2
190 220 [-4.16890356+1.96845855j 14.56503087+0.19671603j] ForkPoolWorker-2
190 230 [-4.61777227 +5.36946771j 10.62480026-16.97771688j] ForkPoolWorker-2
190 240 [-3.59697651+1.35050958j  6.20159135-4.13765414j] ForkPoolWorker-2
190 250 [-3.52252501+0.6260199j  19.51753777+0.03220078j] ForkPoolWorker-2
200 0 [-1.93189948-0.49204183j 20.13928954-0.03369868j] ForkPoolWorker-4
200 10 [-3.11197337+2.69499167j  3.12298048-8.92284283j] ForkPoolWorker-4
200 20 [-6.15874916-0.44585757j 18.21089383-0.100748j  ] ForkPoolWorker-4
200 30 [-1.10358818-1.34442632j  1.74406003-2.74385143j] ForkPoolWorker-4
200 40 [-2.54173284+0.34721842j 19.66584045+0.01769073j] ForkPoolWorker-4
200 50 [-5.74223619+0.90022864j  8.54078008-2.07402191j] ForkPoolWorker-4
200 60 [-1.2268816 -2.46902945j  7.42514446+2.4567113j ] ForkPoolWorker-4
240 0 [-3.44381195+2.10291278j 20.04347034+0.19397784j] ForkPoolWorker-5
200 70 [ 3.9707389 +0.60085278j -5.70618879-3.32513349j] ForkPoolWorker-4
240 10 [-3.03704116+0.00618411j 19.49317708+0.08951973j] ForkPoolWorker-5
200 80 [-2.27817504+0.9355904j  20.00221955+0.09472809j] ForkPoolWorker-4
240 20 [-3.63956871-1.01471553j 19.72259351-0.11797885j] ForkPoolWorker-5
200 90 [-1.77420376-1.59921157j  1.32477447+0.9347202j ] ForkPoolWorker-4
240 30 [-3.49391872+1.99920046j 19.74033872+0.22339541j] ForkPoolWorker-5
```

```
200 100 [-2.1055209 +1.12439979j 19.86355851+0.08504441j] ForkPoolWorker-4
240 40 [-1.40100764-1.29513899j 20.13800534-0.17612831j] ForkPoolWorker-5
200 110 [-1.43537007+0.68928778j  7.43325144-4.39503419j] ForkPoolWorker-4
240 50 [-0.1612845 -2.36621865j  3.07535001+8.38669306j] ForkPoolWorker-5
200 120 [-1.0545661 -0.72973197j  5.12149069+9.04032118j] ForkPoolWorker-4
240 60 [-0.49612522+1.21755774j  2.79003546-2.21683815j] ForkPoolWorker-5
200 130 [ 0.20262676-0.53291416j -1.15592696+0.85480454j] ForkPoolWorker-4
240 70 [-5.49556885+1.59131545j 17.78400192+0.98537446j] ForkPoolWorker-5
200 140 [-0.23938033-2.3963222j   1.58003699+2.21620531j] ForkPoolWorker-4
240 80 [-3.79012103-0.13327275j 19.68862129-0.02347965j] ForkPoolWorker-5
240 90 [ 1.16636395+3.25441819j -5.58700984-5.24816477j] ForkPoolWorker-5
200 150 [-2.29485648-1.01738128j  5.8224629 +1.57864658j] ForkPoolWorker-4
240 100 [0.24845391 -3.42507391j 2.69684865+10.62811555j] ForkPoolWorker-5
200 160 [-0.29107138-3.52484161j -0.42196568+7.6531774j ] ForkPoolWorker-4
240 110 [-1.61406649+3.59534409j  6.10185787-6.20707768j] ForkPoolWorker-5
200 170 [-3.31845702+1.20690406j 19.50388956+0.18122511j] ForkPoolWorker-4
240 120 [ 0.4772919 -4.22656191j -0.4224435+12.26461042j] ForkPoolWorker-5
200 180 [-2.24251036-0.08090558j  6.68165348+2.66312685j] ForkPoolWorker-4
240 130 [-8.4957773 -0.264128j   15.79134994-0.41869452j] ForkPoolWorker-5
240 140 [-5.41655934-0.51066109j 18.08504603-0.39654627j] ForkPoolWorker-5
200 190 [-3.30385405-3.12850004j  4.99914781+2.08492872j] ForkPoolWorker-4
240 150 [-1.13261858-0.5292507j  19.85223915-0.22434917j] ForkPoolWorker-5
200 200 [-2.05129983-0.07724742j 19.96743893-0.01436345j] ForkPoolWorker-4
240 160 [-3.60935848-0.06402566j 19.51060584-0.01672657j] ForkPoolWorker-5
200 210 [-1.13412953 -1.81797619j  7.93377779+12.5767789j ] ForkPoolWorker-4
240 170 [-2.82111988-0.66262864j 19.94684343-0.03221309j] ForkPoolWorker-5
200 220 [-0.47634635 +3.49578364j  0.49388424-23.47210681j] ForkPoolWorker-4
240 180 [-0.39826746-0.51105399j 20.24301347+0.03532448j] ForkPoolWorker-5
200 230 [-5.00452679+0.35267644j 19.45893975+0.20021172j] ForkPoolWorker-4
240 190 [ 1.30381963-1.16086849j -1.0440377 +4.61501109j] ForkPoolWorker-5
200 240 [ 1.15515855-0.02731251j -4.46863378-0.79884374j] ForkPoolWorker-4
240 200 [ 1.43123742-1.37841854j -0.15713049+5.33917673j] ForkPoolWorker-5
200 250 [-7.92058941+0.14055903j 17.03094852+0.14420502j] ForkPoolWorker-4
240 210 [-2.4496907 +0.78599897j 19.77284777+0.09788266j] ForkPoolWorker-5
240 220 [-4.16403198-0.22606513j 18.73550958-0.02602553j] ForkPoolWorker-5
240 230 [-7.43999973-0.17235519j 14.97387801-0.35226108j] ForkPoolWorker-5
240 240 [-6.69703745+1.55708443j 17.54414277-0.62890396j] ForkPoolWorker-5
240 250 [ 2.16768555 -3.58525308j -7.10868202+12.54965346j] ForkPoolWorker-5
210 0 [-5.80635453+1.6623381j  17.97904938+1.93519578j] ForkPoolWorker-7
210 10 [-2.01392617+1.19746051j 19.78143455+0.09485742j] ForkPoolWorker-7
210 20 [-5.94861873-0.35151465j 18.63192398-0.1969761j ] ForkPoolWorker-7
210 30 [-3.01704855-0.28989788j 19.74848336-0.11481516j] ForkPoolWorker-7
210 40 [-0.90854825+1.11299064j 20.25739007+0.29969645j] ForkPoolWorker-7
210 50 [-5.36475682-1.08601207j 17.93770551-0.19391358j] ForkPoolWorker-7
210 60 [-4.32863153+0.72076047j 19.09818782+0.2561863j ] ForkPoolWorker-7
210 70 [-3.88147917+0.37654384j 19.32468744-0.06713826j] ForkPoolWorker-7
210 80 [-0.60045623-0.01470793j -2.71604833+1.128343j  ] ForkPoolWorker-7
210 90 [-4.02581074 +2.93322419j 15.06184916-12.48554047j] ForkPoolWorker-7
```

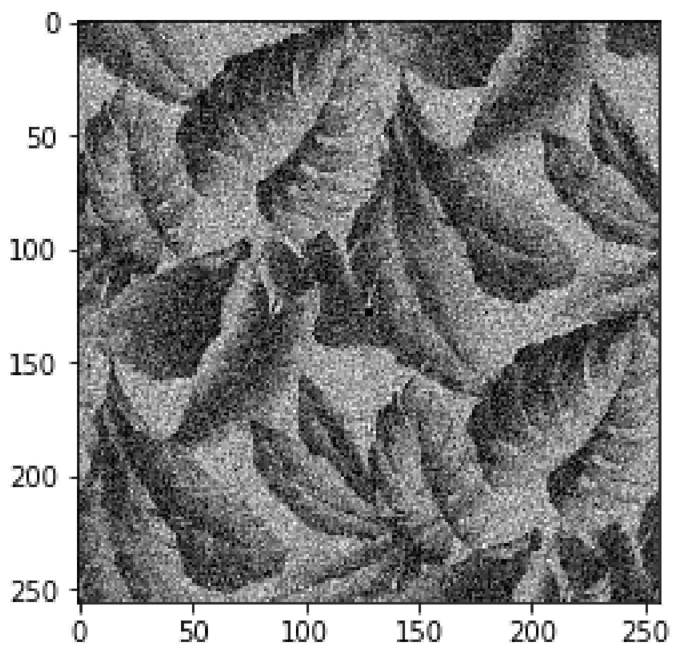
```
210 100 [-0.82730884-1.79657197j  3.71985505+9.00311725j] ForkPoolWorker-7
210 110 [-0.95929811-0.13270959j  1.79381987+3.1125263j ] ForkPoolWorker-7
210 120 [-0.0919827 +1.0497469j  -0.89767822-4.00514213j] ForkPoolWorker-7
210 130 [ 1.61254409+1.64157147j -2.19064492-0.0871245j ] ForkPoolWorker-7
210 140 [-2.62286687-2.26475694j 10.80805602+5.75230605j] ForkPoolWorker-7
210 150 [-0.53012918-0.12175887j  0.99250418-0.05437489j] ForkPoolWorker-7
250 0 [-2.91564187-0.66949017j 19.61231839-0.02122216j] ForkPoolWorker-2
210 160 [ 0.79589491+1.05663754j -0.94930427-1.04566691j] ForkPoolWorker-7
250 10 [-3.57307846+1.4901995j 20.7030863 +0.2749027j] ForkPoolWorker-2
210 170 [-3.22391474+0.02943144j 19.36818781-0.01240699j] ForkPoolWorker-7
250 20 [-4.0168032 -1.15890911j 19.61695439-0.25023816j] ForkPoolWorker-2
210 180 [-4.23612416+2.57892313j 19.29362103+0.63869892j] ForkPoolWorker-7
250 30 [-2.37189913-0.5626498j  19.65551082-0.02183319j] ForkPoolWorker-2
210 190 [-0.57292934+1.71237545j  2.76674745-7.3758947j ] ForkPoolWorker-7
250 40 [-3.80735084-3.24344732j 19.76990962-0.65854951j] ForkPoolWorker-2
210 200 [-1.62521981 -3.62082343j  4.57007237+19.85356297j] ForkPoolWorker-7
250 50 [-0.50423826+1.46252023j  0.16374957-3.57709675j] ForkPoolWorker-2
210 210 [ 1.06152057 +5.67076545j -9.05416216-27.45006358j] ForkPoolWorker-7
210 220 [-4.05966712+1.05309835j 18.71659194+0.35327432j] ForkPoolWorker-7
250 60 [-4.01876717-0.19924499j 19.63697527-0.09199171j] ForkPoolWorker-2
250 70 [-7.28694861-3.09825941j 16.38761942+1.70539635j] ForkPoolWorker-2
210 230 [-4.84347248+0.02535928j 19.25883743-0.02365099j] ForkPoolWorker-7
250 80 [-0.20365532+1.39784667j 20.3439067 +0.03595761j] ForkPoolWorker-2
210 240 [-1.55341411-0.31846603j 19.98812476+0.01130565j] ForkPoolWorker-7
250 90 [-3.33538515+1.55206647j 19.67468143+0.15760131j] ForkPoolWorker-2
210 250 [-5.00818316+0.51330267j 19.63153549+0.14613588j] ForkPoolWorker-7
250 100 [-5.11183172+0.74452645j 18.53044594-0.14436074j] ForkPoolWorker-2
250 110 [-2.52963074+0.67403819j 19.98352548+0.03881576j] ForkPoolWorker-2
250 120 [-1.32193557-1.97071843j -0.73205953+4.78846599j] ForkPoolWorker-2
250 130 [-1.46454019+0.45795189j 20.01919638+0.11655885j] ForkPoolWorker-2
250 140 [-2.75864224+1.39725189j  8.42404791-3.50789321j] ForkPoolWorker-2
250 150 [ 0.3336856 -2.00200941j -1.18127955+0.90326669j] ForkPoolWorker-2
250 160 [-6.93799235-0.57576033j 17.21693274-0.04761023j] ForkPoolWorker-2
250 170 [-2.3533311 -0.43924575j 19.97763478-0.05229938j] ForkPoolWorker-2
250 180 [-3.717985  +0.04064306j 19.07115924-0.09395621j] ForkPoolWorker-2
250 190 [-4.35200289-0.24717597j 19.41413883-0.08262178j] ForkPoolWorker-2
250 200 [-5.21014613+0.03867558j 18.27275709-0.08537084j] ForkPoolWorker-2
250 210 [-4.27402067-0.88584285j 19.50956022-0.23349038j] ForkPoolWorker-2
250 220 [-3.59332554-0.04748008j 19.22911621+0.00324983j] ForkPoolWorker-2
250 230 [-3.99198081-0.66497274j 18.41113249-0.50465264j] ForkPoolWorker-2
250 240 [  1.09107466 +4.42923441j -11.80477557-11.53262836j] ForkPoolWorker-2
250 250 [-1.17871437-2.27869549j 20.14293311-0.59350606j] ForkPoolWorker-2
220 0 [-9.61676471-1.40778138j 16.36186503-0.50988822j] ForkPoolWorker-1
220 10 [-4.65220086+2.62470063j 13.06761882-8.65648131j] ForkPoolWorker-1
220 20 [-3.07034623-0.27851443j 19.91118636-0.01834575j] ForkPoolWorker-1
220 30 [-3.27222413-0.31632866j 19.42758559-0.16783737j] ForkPoolWorker-1
220 40 [-3.98866427-0.3216438j  19.11482035-0.10424898j] ForkPoolWorker-1
220 50 [-2.02720356-3.41069636j  2.90250632+8.78692012j] ForkPoolWorker-1
```

```
220 60 [-2.55311269-1.04590929j 19.72608562-0.07188345j] ForkPoolWorker-1
220 70 [-3.73606565-1.25243998j 17.69646561-0.45711702j] ForkPoolWorker-1
220 80 [-1.25040086-1.306414j   20.0816166 -0.16005951j] ForkPoolWorker-1
220 90 [-3.34783453-0.00018465j 19.66606382+0.03023257j] ForkPoolWorker-1
220 100 [-1.98474435-0.36836728j 19.92616863+0.00665738j] ForkPoolWorker-1
220 110 [0.91046417+1.20407639j 1.27002546-2.50770374j] ForkPoolWorker-1
220 120 [-1.25239305+1.48016203j -1.14356943-0.34388801j] ForkPoolWorker-1
220 130 [-2.44812105-0.98382583j 19.55982952-0.17091246j] ForkPoolWorker-1
220 140 [-0.53453695+0.34683796j  2.35485541+1.72534287j] ForkPoolWorker-1
220 150 [-2.80647119-1.07951058j  9.16169131+4.31363983j] ForkPoolWorker-1
220 160 [  5.32510083+2.77066611j -11.75057872-9.83100862j] ForkPoolWorker-1
220 170 [-3.03248469-0.08825242j 19.84430693-0.09931j   ] ForkPoolWorker-1
220 180 [-2.14236931-1.29418569j 19.8202055 -0.31783011j] ForkPoolWorker-1
220 190 [ 1.20134554 -3.76715779j -0.74046012+10.13063387j] ForkPoolWorker-1
220 200 [-1.53552944 -3.70455773j  6.67499731+12.74388501j] ForkPoolWorker-1
220 210 [-0.47721494 +1.88154628j  3.1983584 -10.40770562j] ForkPoolWorker-1
220 220 [-0.26463339-0.95369169j 20.17026101-0.10748263j] ForkPoolWorker-1
220 230 [-4.26835505-2.11131292j 19.64363419-0.29261784j] ForkPoolWorker-1
220 240 [-5.50353705-1.3034633j  18.78917898-0.15453517j] ForkPoolWorker-1
220 250 [-6.81224029-0.63615734j 18.7911051 -0.32781847j] ForkPoolWorker-1
230 0 [-2.3049607 -3.67432794j  3.53720459+4.27138743j] ForkPoolWorker-8
230 10 [-3.31566823-0.87517297j  6.15310572+1.8034356j ] ForkPoolWorker-8
230 20 [-5.39589603+0.3129711j 18.8669307 -0.1024875j] ForkPoolWorker-8
230 30 [-3.67701301+1.14395045j 19.44520318+0.30869959j] ForkPoolWorker-8
230 40 [-0.30443637+0.51041432j  2.56800547-3.93067287j] ForkPoolWorker-8
230 50 [-4.18213442-0.14904493j 19.16896198-0.13220119j] ForkPoolWorker-8
230 60 [ 2.14044513 +4.68378573j -4.5757395 -21.78454278j] ForkPoolWorker-8
230 70 [-2.50246641+3.12439242j 19.77486739+0.49673855j] ForkPoolWorker-8
230 80 [-1.53057818+0.30732443j 20.06984552+0.07868727j] ForkPoolWorker-8
230 90 [-2.63299651-1.64202167j  7.74633489+6.18672316j] ForkPoolWorker-8
230 100 [-5.2005803 +1.51560932j 18.30027071+0.61605636j] ForkPoolWorker-8
230 110 [-4.51847174-1.95525887j 18.91475569-0.29175446j] ForkPoolWorker-8
230 120 [-1.9504862 +0.1448654j   3.07823011+1.28516979j] ForkPoolWorker-8
230 130 [-0.30645518-0.35395537j 20.05131246-0.01973703j] ForkPoolWorker-8
230 140 [ 1.60703115+0.78148733j -4.23284332+7.28738287j] ForkPoolWorker-8
230 150 [-2.90672628-1.62602954j 19.75134276-0.17707103j] ForkPoolWorker-8
230 160 [-1.64923101-0.73906969j 19.93008472+0.03131513j] ForkPoolWorker-8
230 170 [-5.01183357+0.08237013j 19.38829178+0.05524926j] ForkPoolWorker-8
230 180 [-3.86919334+2.19120855j  3.88929406-5.60585222j] ForkPoolWorker-8
230 190 [-1.6880559 +2.73055285j  5.82092254-4.92619346j] ForkPoolWorker-8
230 200 [-0.9481975 +0.02396639j 20.06753887-0.07100921j] ForkPoolWorker-8
230 210 [-3.1190641 -0.38807201j 19.53048331-0.08722703j] ForkPoolWorker-8
230 220 [-1.7528765 +0.16927005j 20.02619591-0.02748771j] ForkPoolWorker-8
230 230 [-5.43474608+0.50737984j 19.16629152+0.28993224j] ForkPoolWorker-8
230 240 [-8.57947138+1.01463481j 14.68914236+0.58926524j] ForkPoolWorker-8
230 250 [-6.96849702+0.4486217j 15.7019015 -1.8154532j] ForkPoolWorker-8
Multi-threaded minimization took  258.01894187927246  sec
```

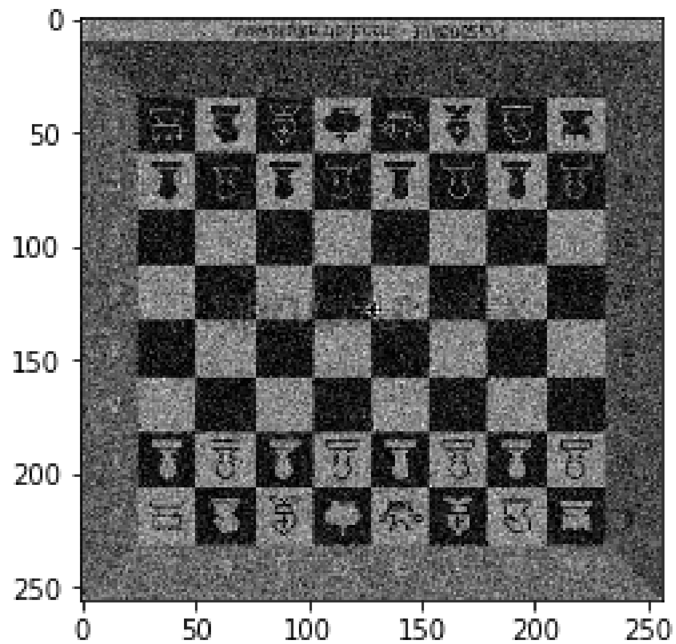
In [75]:
```
# Restore visible part
comprecfft = fm.hidtovis(icf,bins,comprec)
comprecmaps = np.fft.ifft2(comprecfft)
comprecmaps = np.fft.ifftshift(comprecmaps, axes=(1,2))
comprecmaps = np.abs(comprecmaps)
```

In [76]:
```
# Show reconstructed maps with excluded outliers
for i in range(nc):
    print(i)
    imc = comprecmaps[i]
    stdc = np.std(imc)
    super_threshold_indices = np.abs(imc) > 5*stdc
    imc[super_threshold_indices] = 0.0
    plt.imshow(imc, cmap='gray')
    plt.show()
```

0



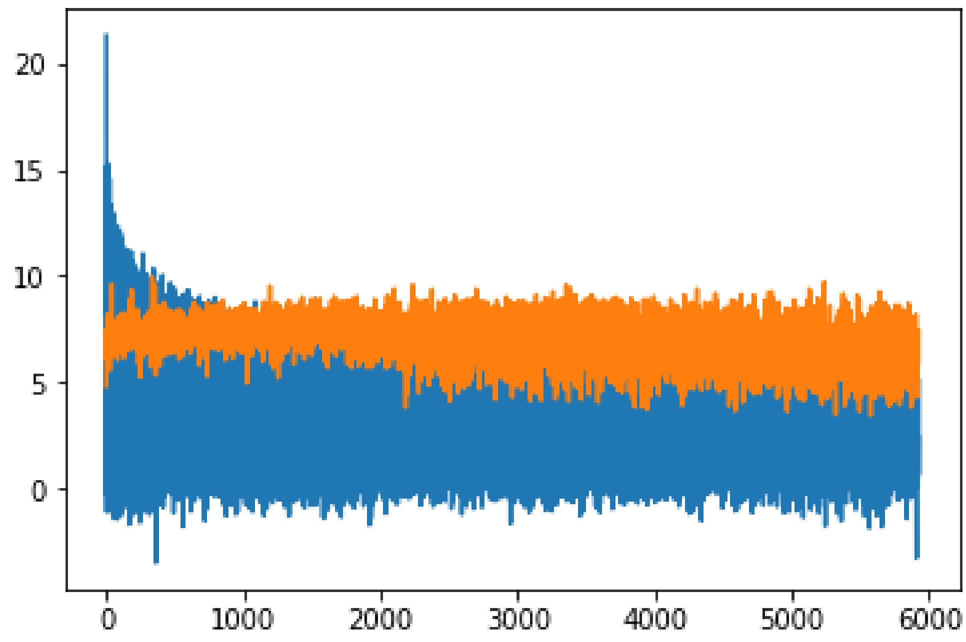
1



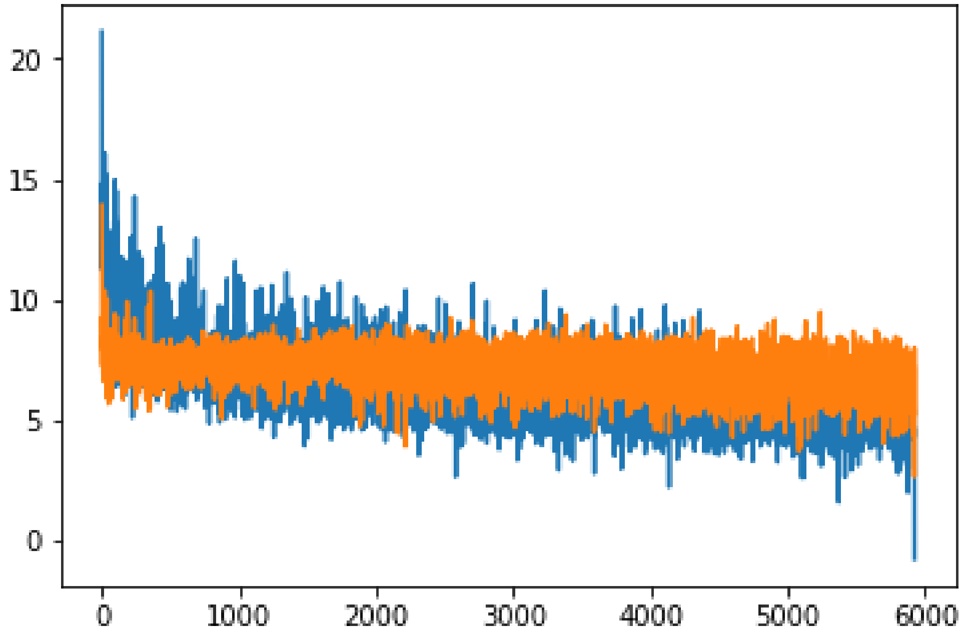
```
In [70]: # Export reconstructions in a FITS file
         hdu = fits.PrimaryHDU(data=comprecmaps, header=components[0].header)
         hdu.writeto(reconfile, overwrite=True)

In [71]: # Find residuals and export them in a FITS file
         residuals = components_data - comprecmaps
         hdu = fits.PrimaryHDU(data=residuals, header=components[0].header)
         hdu.writeto(errorfile, overwrite=True)

In [72]: # Find residual power spectra
         residfft = np.fft.fft2(residuals)
         residfft = np.fft.fftshift(residfft, axes=(1,2))
         residps = np.zeros((nc, topbin+1))
         for i in range(nc):
             workspace = np.abs(residfft[i,:,:]**2)
             summ, num = fm.azav(workspace, nx, ny, topbin, bins)
             residps[i,:] = summ

In [73]: # Find component power spectra
         compps = np.zeros((nc, topbin+1))
         for i in range(nc):
             workspace = np.abs(compfft[i,:,:]**2)
             summ, num = fm.azav(workspace, nx, ny, topbin, bins)
             compps[i,:] = summ

In [74]: # Show logpower spectra for components and residuals
         for i in range(nc):
```

```
print("Component ", i)
plt.plot(np.log(compps[i,:])), plt.plot(np.log(residps[i,:]))
plt.show()
```

```
Component  0
```




```
Component  1
```




```
In [ ]:
```