

eht-imaging tutorial

eht-imaging github and documentation [here](#)

Antonio Fuentes (IAA-CSIC)

EHT 2021 Winter Virtual Collaboration Meeting

1. Load and inspect the data

+ oifits, txt

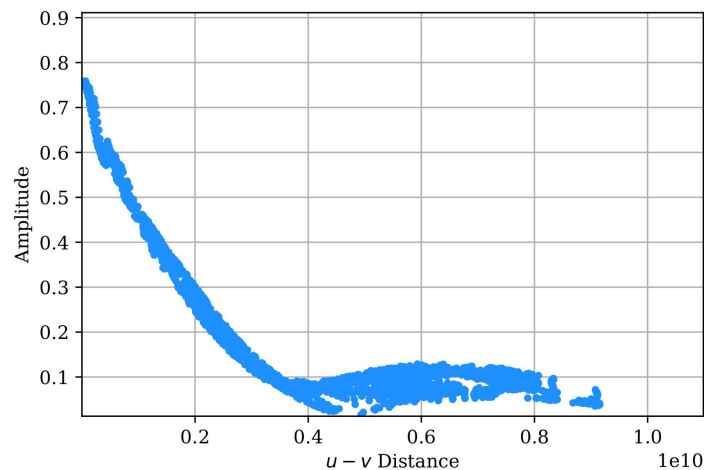
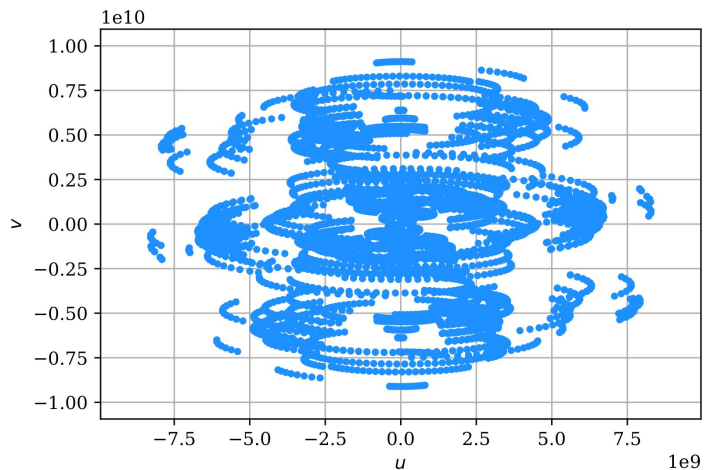
Load the data

```
inobs = './M87_ngeht_ref1_230_thnoise.uvfits'  
obs = eh.obsdata.load_uvfits(inobs)
```

Inspect the data

```
obs.plotall('u', 'v', conj=True)  
obs.plotall('uvdist', 'amp')
```

+ phases, cl. quants.,
single baselines, etc.



2. Data preparation

Do some time averaging

```
obs.add_scans()
```

```
obs = obs.avg_coherent(0., scan_avg=True)
```

60s, scan_avg=False

Zero baseline flux (as measured by ALMA-APEX)

```
zbl = np.median(obs.unpack_b1('AA', 'AP', 'amp')['amp'])
```

Array nominal resolution

```
res = obs.res()
```

+ deconvolution, etc.

Non-closing error budget (1%)

```
obs = obs.add_fractional_noise(0.01)
```

Make a copy of the initial data

```
obs_init = obs.copy()
```

3. Define a prior / init image

```
# Image parameters
```

```
npix      = 100  
fov       = 700*eh.RADPERUAS  
prior_fwhm = 100*eh.RADPERUAS  
shift     = 170*eh.RADPERUAS
```

```
# Gaussian prior image
```

```
gaussprior = eh.image.make_square(obs, npix, fov)  
gaussprior = gaussprior.add_gauss(zbl, (prior_fwhm, prior_fwhm, 0,  
                                       shift, -shift))
```

shift from img center

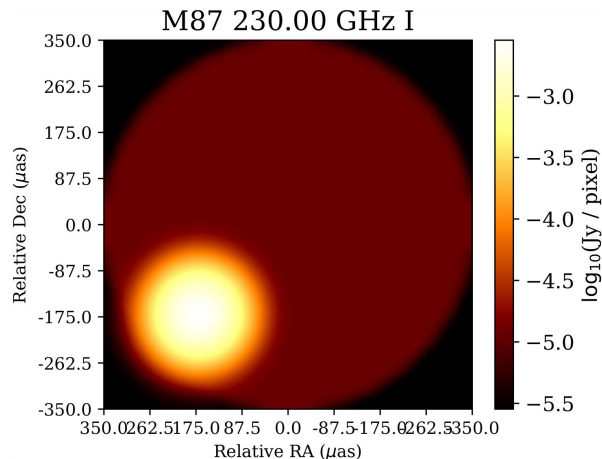
```
# Large disk to capture extended emission
```

```
gaussprior = gaussprior.add_tophat(zbl*0.1, fov/2).blur_circ(res)
```

```
# To avoid gradient singularities in the first step,
```

```
# add an additional small Gaussian
```

```
gaussprior = gaussprior.add_gauss(zbl*1e-3, (prior_fwhm,  
                                             prior_fwhm, 0, 0, 0))
```



4. Image reconstruction (1st round)

Data terms and regularizers

```
data_term = {'amp': 1, 'cphase': 1, 'logcamp': 1}
reg_term   = {'simple': 0, 'tv': 1, 'tv2': 0, 'l1': 1}
```

Initialize imaging with a Gaussian image

```
imgr = eh.imager.Imager(obs, gaussprior, prior_im=gaussprior,
                        flux=zbl, data_term=data_term, maxit=350,
                        norm_reg=True, reg_term=reg_term, ttype='nfft',
                        stop=1e-6)
```

Imaging

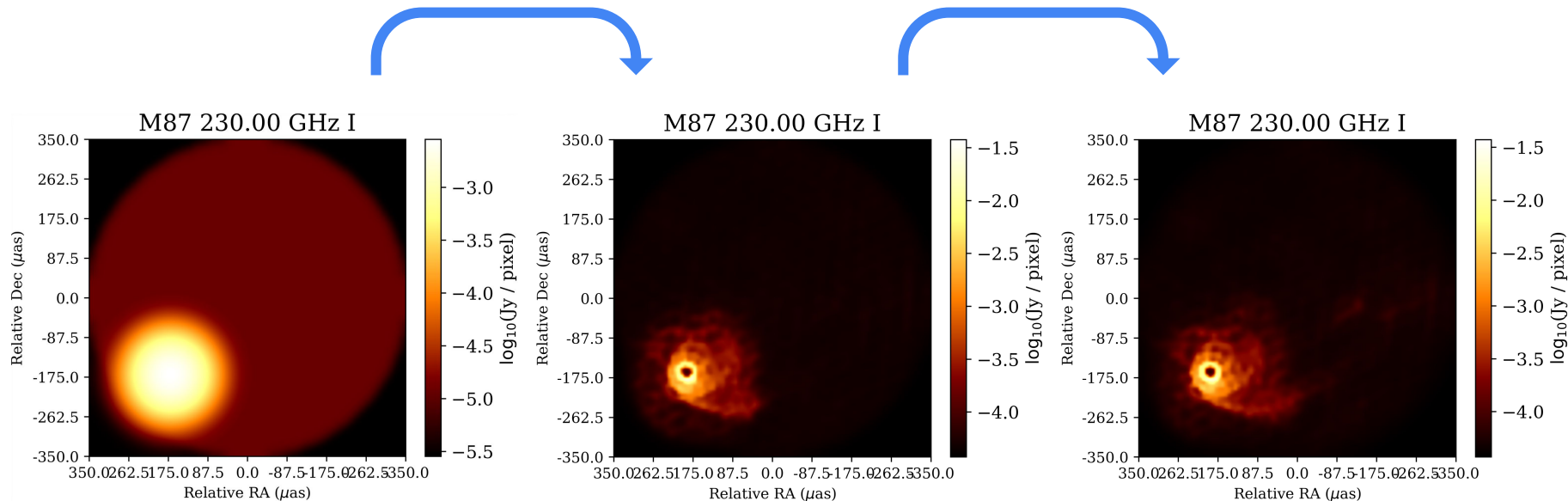
```
imgr.make_image_I(show_updates=False)
for repeat in range(4):
    imgr.init_next = imgr.out_last().blur_circ(res)
    imgr.make_image_I(show_updates=False)
```

blur prev img & use as
init for next iter

Self-calibrate to the previous model

```
obs_sc = eh.selfcal(obs, imgr.out_last(), method='both', ttype='nfft',
                   solution_interval=0.0, processes=0)
```

4. Image reconstruction (1st round)



5. Image reconstruction (2nd round)

```
# Update init image and data terms
```

```
init      = imgr.out_last().blur_circ(res)  
data_term = {'vis': 10, 'cphase': 10, 'logcamp': 10}
```

```
# Reinitialize imaging
```

```
imgr = eh.imager.Imager(obs_sc, init, prior_im=gaussprior,  
                        flux=zbl, data_term=data_term, maxit=350,  
                        norm_reg=True, reg_term=reg_term, ttype='nfft',  
                        stop=1e-6)
```

```
# Imaging
```

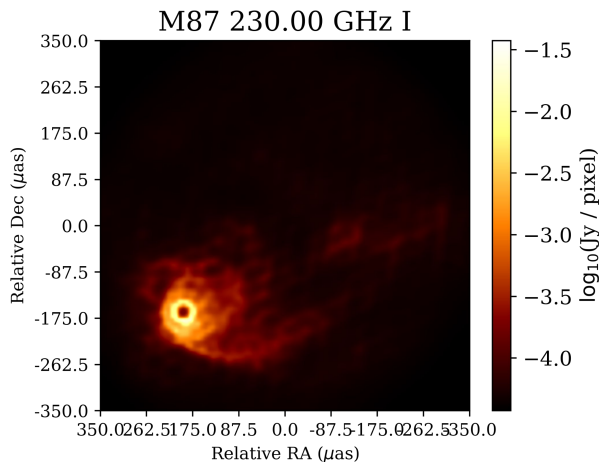
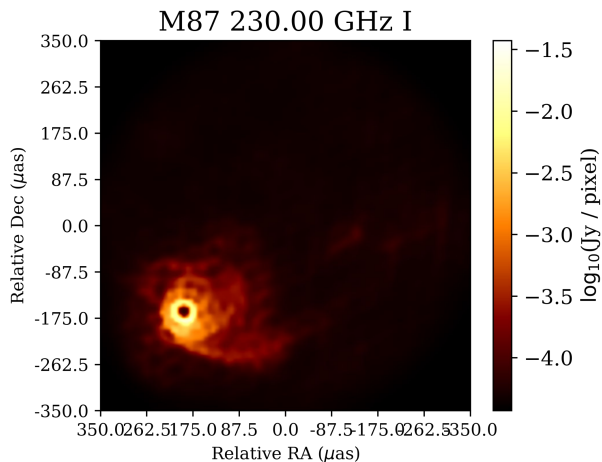
```
imgr.make_image_I(show_updates=False)  
for repeat in range(4):  
    imgr.init_next = imgr.out_last().blur_circ(res)  
    imgr.make_image_I(show_updates=False)
```

blur prev img & use as
init for next iter

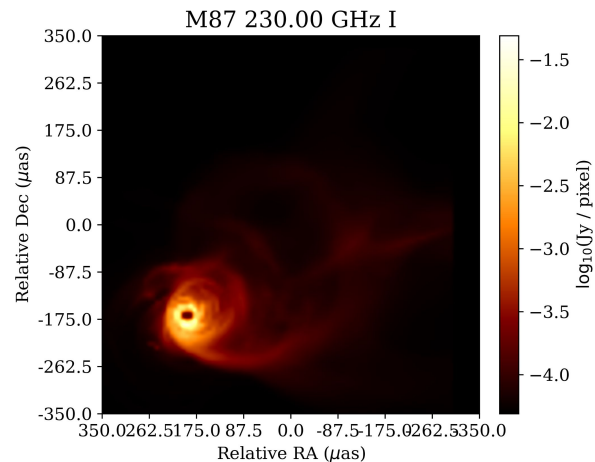
```
# Self-calibrate to the previous model
```

```
obs_sc = eh.selfcal(obs_sc, imgr.out_last(), method='both', ttype='nfft',  
                  solution_interval=0.0, processes=0)
```

5. Image reconstruction (2nd round)



ground truth image



6. Save output

Save the final image and self-calibrated data

```
imgr.out_last().save_fits('final_image.fits')
```

```
obs_sc.save_uvfits('selfcal_data.uvfits')
```

+ txt

Save a png of the final image

```
imgr.out_last().display(cbar_unit=['Tb'], label_type='scale',  
                        export_pdf='final_image.png')
```

```
imgr.out_last().display(cbar_unit=['Tb'], label_type='scale', scale='log',  
                        export_pdf='final_image_logscale.png')
```

+ oifits, txt

Save summary of the final image and associated chi2

```
eh.imgsum(imgr.out_last(), obs_sc, obs_init, 'imgsum.pdf')
```

7. Check image fit to data

+ phases, cl. quants.,
single baselines, etc.

```
# Plot image model fit to visibility amplitudes
```

```
eh.plotall_obs_im_compare(obs_sc, imgr.out_last(), 'uvdist', 'amp')
```

```
# Plot image model fit to closure phases
```

```
eh.plot_cphase_obs_im_compare(obs_sc, imgr.out_last(), 'AA', 'SM', 'LM')
```

