(Some) future plans for GalSim

Josh Meyers

- galsim.ChromaticRealGalaxy
- galsim.PhaseScreenPSF

The LSST PSF is chromatic

- Atmosphere
  - Differential chromatic refraction
  - Chromatic seeing
- Optics (diffraction/refraction)
- Sensors (absorption length)

- Primary concern for weak lensing is difference between PSF star SED and gal SED.

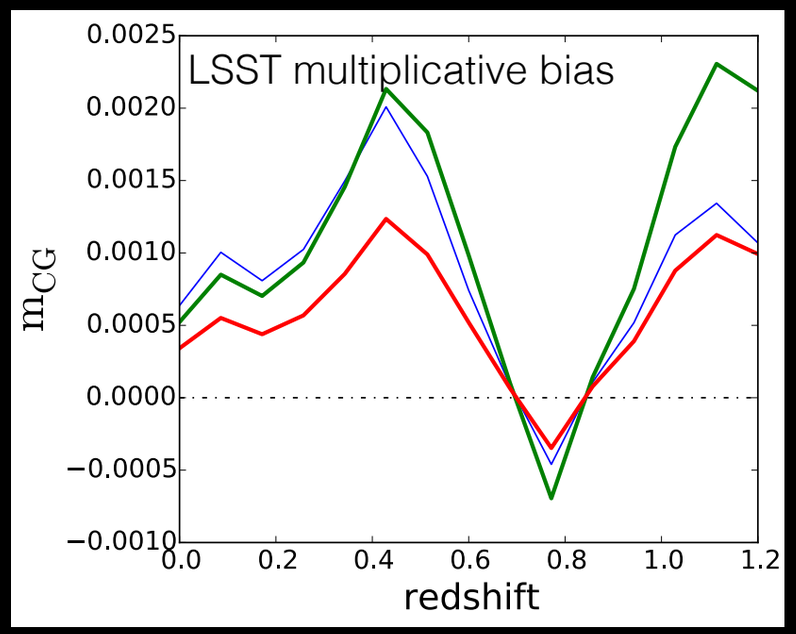
Color gradients

- SED varies with position in real galaxies; so does the PSF.
- Definitely a problem for Euclid (Voigt++12, Semboloni++13)
- Bias due to color gradient scales like

$$\frac{\mathrm{dPSF}}{\mathrm{d}\lambda} \times (\text{filter width})^2 \times \frac{\text{PSF area}}{\text{gal area}}$$

- LSST/Euclid ~ (0.3) x $(0.4)^2$ x 3^2 ~ 0.4
- At least this bad in actual simulations (S. Kamath)
- However, real galaxies are not bulge+disk! Especially at high redshift.


NGC 2442



Kamath++ in prep.

galsim.RealGalaxy

- Use HST images to model realistic galaxies.

Observed image HST PSF galaxy

$$I(\vec{x}) = \Pi(\vec{x}) * f(\vec{x})$$

galsim.RealGalaxy

- Use HST images to model realistic galaxies.

Observed image HST PSF galaxy

$$I(\vec{x}) = \Pi(\vec{x}) * f(\vec{x})$$

- Deconvolve HST PSF. Apply affine transformation (e.g., shear, dilate, rotate). Convolve by desired (larger) PSF.

galsim.RealGalaxy

- Use HST images to model realistic galaxies.

Observed image HST PSF galaxy

$$I(\vec{x}) = \Pi(\vec{x}) * f(\vec{x})$$

- Deconvolve HST PSF. Apply affine transformation (e.g., shear, dilate, rotate). Convolve by desired (larger) PSF.

$$\tilde{I}(\vec{k}) = \tilde{\Pi}(\vec{k})\tilde{f}(\vec{k}) \xrightarrow{\text{deconvolve}} \tilde{f}(\vec{k}) = \frac{\tilde{I}(\vec{k})}{\tilde{\Pi}(\vec{k})}$$

galsim.RealGalaxy

- Use HST images to model realistic galaxies.

Observed image, HST PSF, galaxy

$$I(\vec{x}) = \Pi(\vec{x}) * f(\vec{x})$$

- Deconvolve HST PSF. Apply affine transformation (e.g., shear, dilate, rotate). Convolve by desired (larger) PSF.

$$\tilde{I}(\vec{k}) = \tilde{\Pi}(\vec{k})\tilde{f}(\vec{k}) \xrightarrow{\text{deconvolve}} \tilde{f}(\vec{k}) = \frac{\tilde{I}(\vec{k})}{\tilde{\Pi}(\vec{k})}$$

$$\xrightarrow{\text{transform}} \tilde{f}'(\vec{k}) = \tilde{f}(A\vec{k})$$

galsim.RealGalaxy

- Use HST images to model realistic galaxies.

Observed image ↘ HST PSF ↓ galaxy ↙

$$I(\vec{x}) = \Pi(\vec{x}) * f(\vec{x})$$

- Deconvolve HST PSF. Apply affine transformation (e.g., shear, dilate, rotate). Convolve by desired (larger) PSF.

$$\tilde{I}(\vec{k}) = \tilde{\Pi}(\vec{k})\tilde{f}(\vec{k}) \xrightarrow{\text{deconvolve}} \tilde{f}(\vec{k}) = \frac{\tilde{I}(\vec{k})}{\tilde{\Pi}(\vec{k})}$$

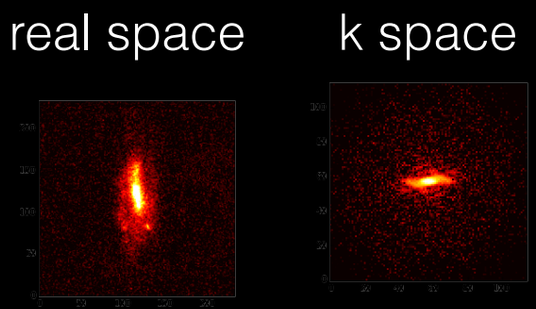
$$\xrightarrow{\text{transform}} \tilde{f}'(\vec{k}) = \tilde{f}(A\vec{k})$$

$$\xrightarrow{\text{reconvolve}} \tilde{I'}_{\text{target}}(\vec{k}) = \tilde{\Pi}_{\text{target}}(\vec{k})\tilde{f}'(\vec{k})$$

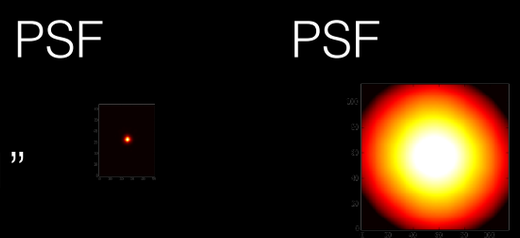
galsim.RealGalaxy

- Use HST images to model realistic galaxies.

Observed image ← $I(\vec{x})$ = HST PSF $\Pi(\vec{x})$ $*$ galaxy $f(\vec{x})$



- Deconvolve HST PSF. Apply affine transformation (e.g., shear, dilate, rotate). Convolve by desired (larger) PSF.



$$\tilde{I}(\vec{k}) = \tilde{\Pi}(\vec{k})\tilde{f}(\vec{k}) \xrightarrow{\text{deconvolve}} \tilde{f}(\vec{k}) = \frac{\tilde{I}(\vec{k})}{\tilde{\Pi}(\vec{k})}$$

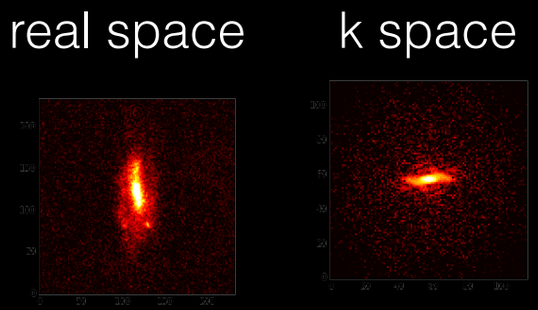
$$\xrightarrow{\text{transform}} \tilde{f}'(\vec{k}) = \tilde{f}(A\vec{k})$$

$$\xrightarrow{\text{reconvolve}} \tilde{I'}_{\text{target}}(\vec{k}) = \tilde{\Pi}_{\text{target}}(\vec{k})\tilde{f}'(\vec{k})$$

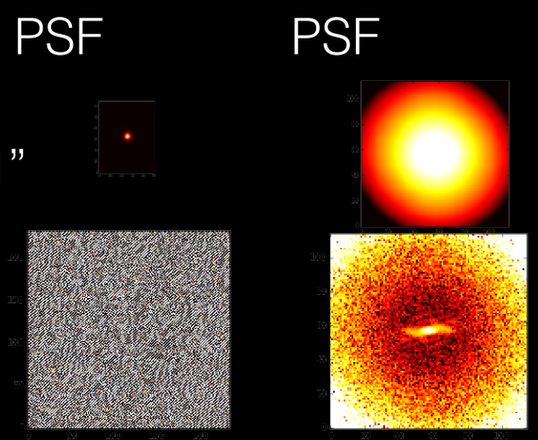
galsim.RealGalaxy

- Use HST images to model realistic galaxies.

Observed image ← $I(\vec{x})$ = HST PSF $\Pi(\vec{x})$ $*$ galaxy $f(\vec{x})$



- Deconvolve HST PSF. Apply affine transformation (e.g., shear, dilate, rotate). Convolve by desired (larger) PSF.



$$\tilde{I}(\vec{k}) = \tilde{\Pi}(\vec{k})\tilde{f}(\vec{k}) \xrightarrow{\text{deconvolve}} \tilde{f}(\vec{k}) = \frac{\tilde{I}(\vec{k})}{\tilde{\Pi}(\vec{k})}$$

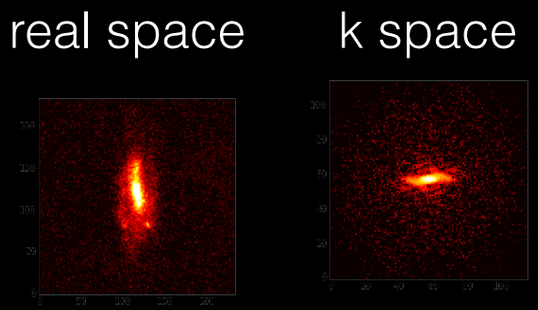
$$\xrightarrow{\text{transform}} \tilde{f}'(\vec{k}) = \tilde{f}(A\vec{k})$$

$$\xrightarrow{\text{reconvolve}} \tilde{I'}_{\text{target}}(\vec{k}) = \tilde{\Pi}_{\text{target}}(\vec{k})\tilde{f}'(\vec{k})$$

galsim.RealGalaxy

- Use HST images to model realistic galaxies.

Observed image ← $I(\vec{x})$ = HST PSF $\Pi(\vec{x})$ $*$ galaxy $f(\vec{x})$

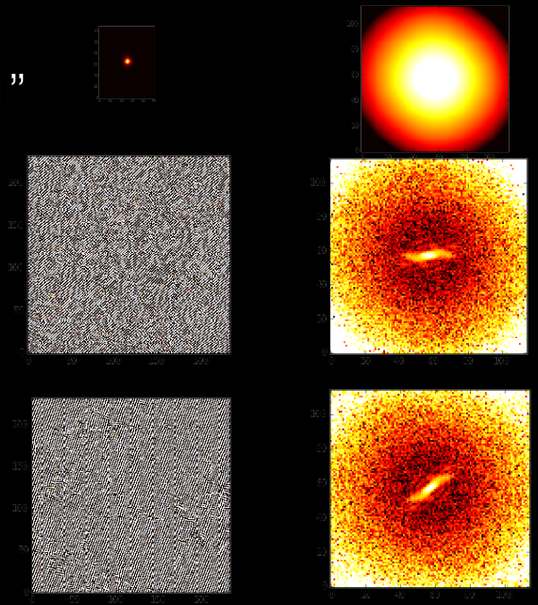


- Deconvolve HST PSF. Apply affine transformation (e.g., shear, dilate, rotate). Convolve by desired (larger) PSF.

$$\tilde{I}(\vec{k}) = \tilde{\Pi}(\vec{k})\tilde{f}(\vec{k}) \xrightarrow{\text{deconvolve}} \tilde{f}(\vec{k}) = \frac{\tilde{I}(\vec{k})}{\tilde{\Pi}(\vec{k})}$$

$$\xrightarrow{\text{transform}} \tilde{f}'(\vec{k}) = \tilde{f}(A\vec{k})$$

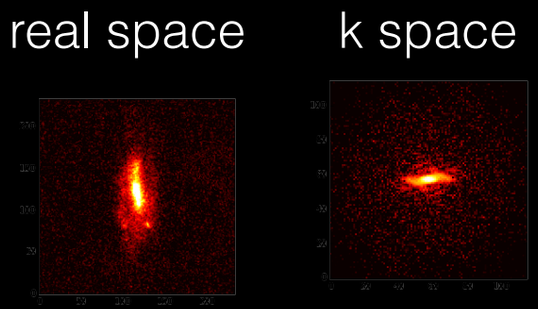
$$\xrightarrow{\text{reconvolve}} \tilde{I'}_{\text{target}}(\vec{k}) = \tilde{\Pi}_{\text{target}}(\vec{k})\tilde{f}'(\vec{k})$$



galsim.RealGalaxy

- Use HST images to model realistic galaxies.

Observed image ← $I(\vec{x})$ = HST PSF $\Pi(\vec{x})$ $*$ galaxy $f(\vec{x})$

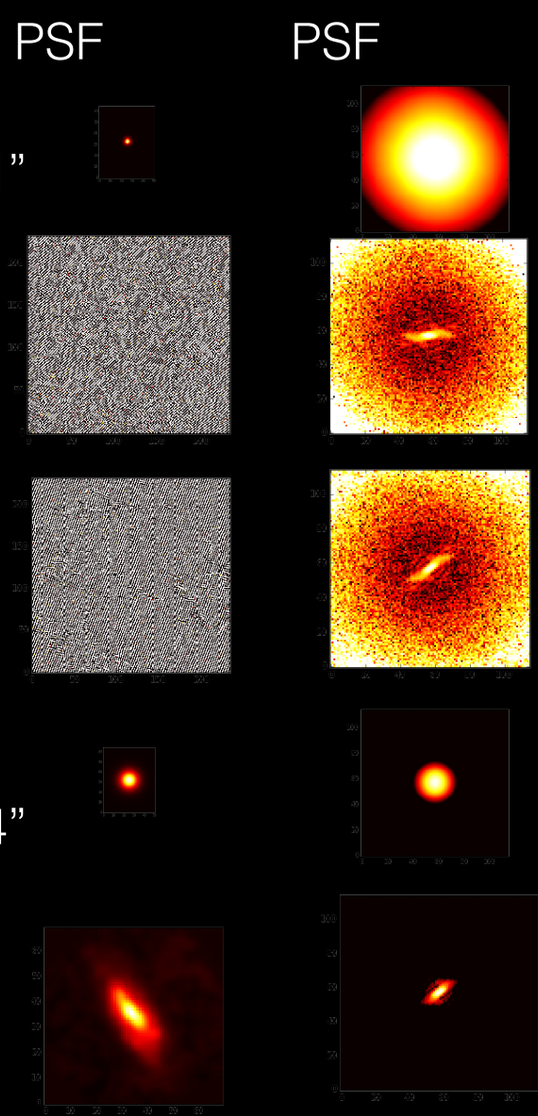


- Deconvolve HST PSF. Apply affine transformation (e.g., shear, dilate, rotate). Convolve by desired (larger) PSF.

$$\tilde{I}(\vec{k}) = \tilde{\Pi}(\vec{k})\tilde{f}(\vec{k}) \xrightarrow{\text{deconvolve}} \tilde{f}(\vec{k}) = \frac{\tilde{I}(\vec{k})}{\tilde{\Pi}(\vec{k})}$$

$$\xrightarrow{\text{transform}} \tilde{f}'(\vec{k}) = \tilde{f}(A\vec{k})$$

$$\xrightarrow{\text{reconvolve}} \tilde{I}'_{\text{target}}(\vec{k}) = \tilde{\Pi}_{\text{target}}(\vec{k})\tilde{f}'(\vec{k})$$



galsim.RealChromaticGalaxy

- Use multi-band HST images to model realistic galaxies with color-gradients.

Images

Filters

HST PSF

galaxy

$$I_i(\vec{x}) = \int T_i(\lambda) \left[\Pi(\vec{x}, \lambda) * f(\vec{x}, \lambda)\right] \mathrm{d}\lambda$$

filter index

galsim.RealChromaticGalaxy

- Use multi-band HST images to model realistic galaxies with color-gradients.

Images → $I_i(\vec{x})$ = ∫ Filters → $T_i(\lambda)$ [HST PSF → $\Pi(\vec{x}, \lambda)$ ∗ galaxy → $f(\vec{x}, \lambda)$] $\mathrm{d}\lambda$

filter index → i

- Assert that galaxy is sum of separable profiles with particular SEDs; solve for spatial components.

$$f(\vec{x}, \lambda) = \sum_j S_j(\lambda) a_j(\vec{x})$$

SED index → j; $S_j(\lambda)$: SEDs; $a_j(\vec{x})$: spatial terms

RealChromaticGalaxy

- Again, go to Fourier space:

Ansatz ⟶ $$\tilde{I}_i(\vec{k}) = \int T_i(\lambda)\tilde{\Pi}(\vec{k},\lambda)\sum_j S_j(\lambda)\tilde{a}_j(\vec{k})\,\mathrm{d}\lambda$$

Rearrange ⟶ $$= \sum_j \left[\int T_i(\lambda)S_j(\lambda)\tilde{\Pi}(\vec{k},\lambda)\,\mathrm{d}\lambda\right]\tilde{a}_j(\vec{k})$$

Relabel ⟶ $$= \sum_j \tilde{\Pi}^{\mathrm{eff}}_{ij}(\vec{k})\tilde{a}_j(\vec{k})$$

j

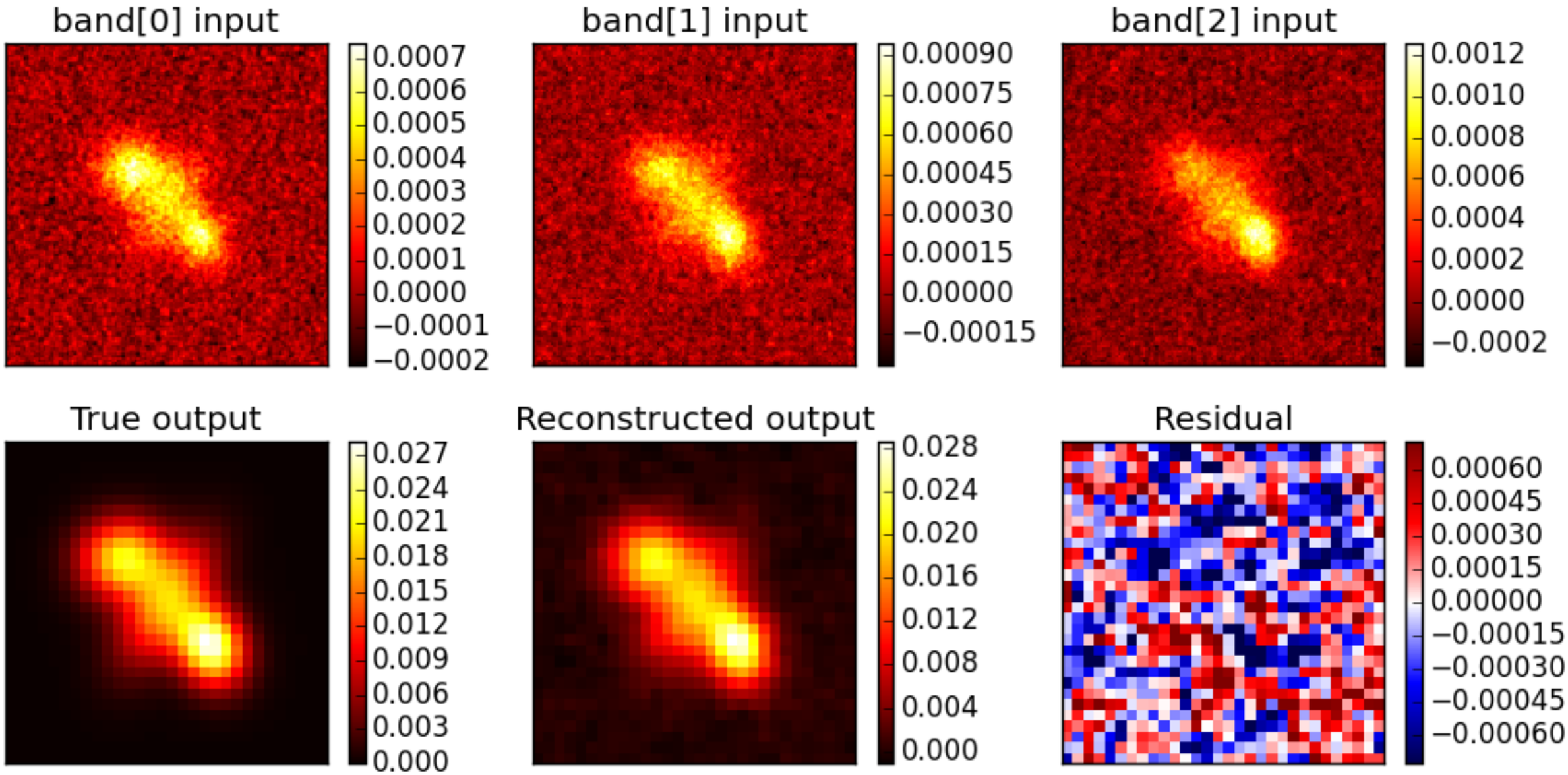
Solving for these.

Effective PSF of jth SED through the ith filter.

- Left with a matrix equation for each k-mode. So solve it!
- Can also do this properly accounting for correlated noise in each image: see https://github.com/GalSim-developers/GalSim/blob/%23640/devel/modules/CGNotes.pdf for details

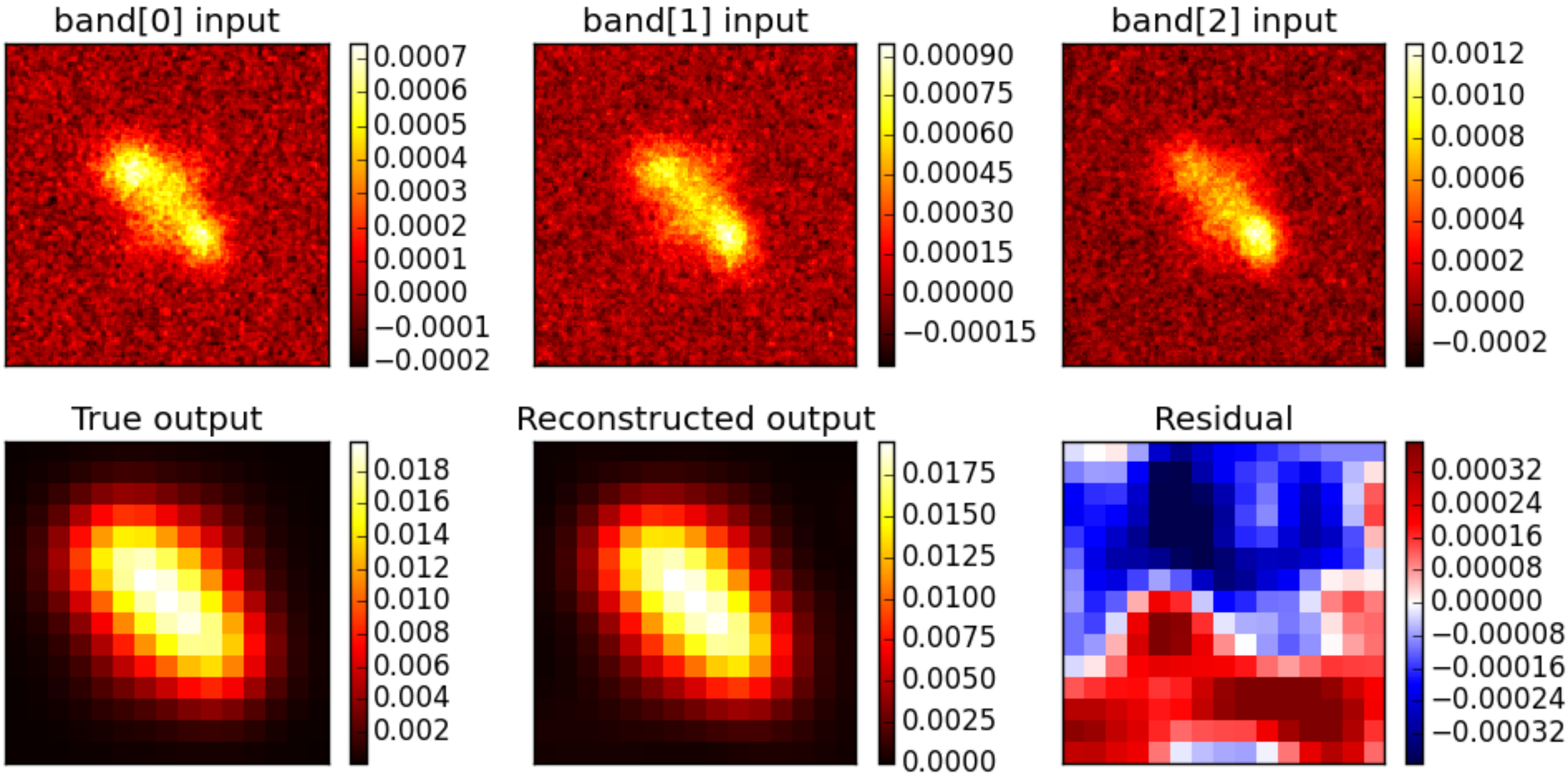
(Toy) example

- r-, i-, z-like input filters, Euclid-like output filter, PSF, and pixels.



(Toy) example

- r-, i-, z-like input filters, out-of-phase output filter and LSST-like PSF, pixel scale.

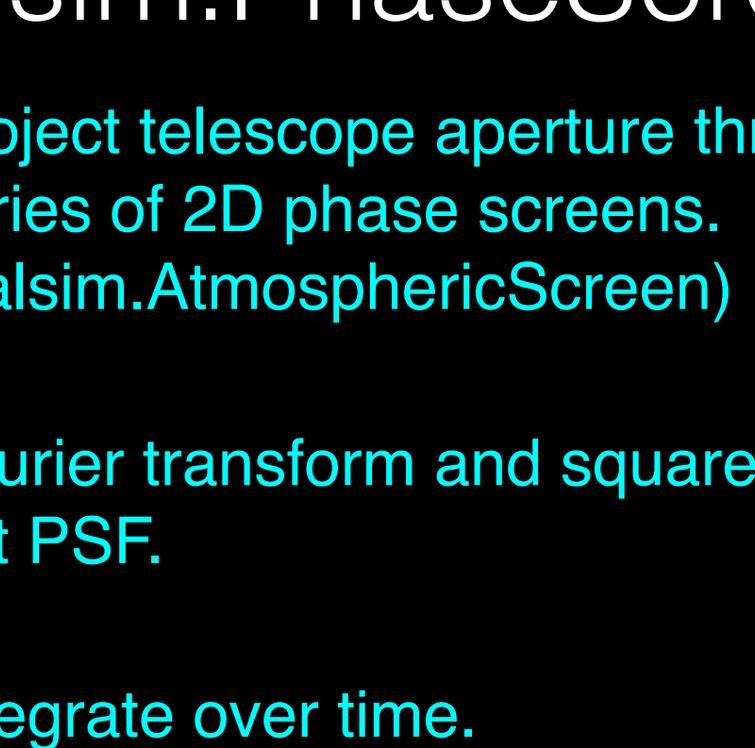


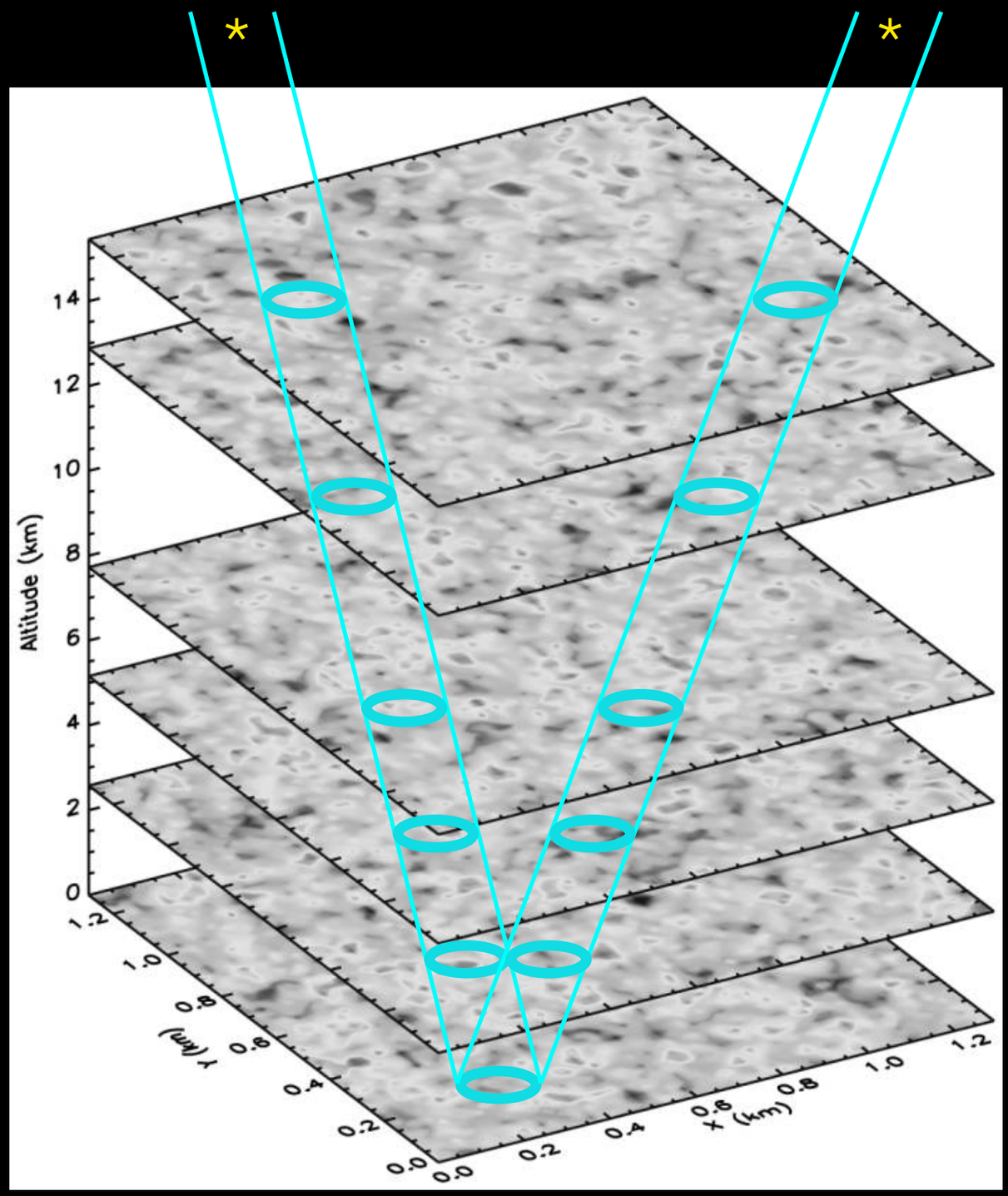
Next steps

- AEGIS catalog of ~29000 galaxies in V- and I-band.
  - Sowmya Kamath currently processing these (continuing work done by Bradley Emi, Jason Rhodes, Andres Plazas, and Rachel Mandelbaum.)
- CANDELS - similar area/depth to AEGIS, but 5-8 filters.
- HST Frontier Fields (parallels) - smaller but deeper, ~7 filters.
- (Chromatic)RealGalaxies from hydro sims (Issue #669)
- Tests: assess the importance of asserted SED mismatch.

galsim.PhaseScreenPSF

- Project telescope aperture through series of 2D phase screens. (galsim.AtmosphericScreen)
- Fourier transform and square to get PSF.
- Integrate over time. (galsim.PhaseScreenPSF)

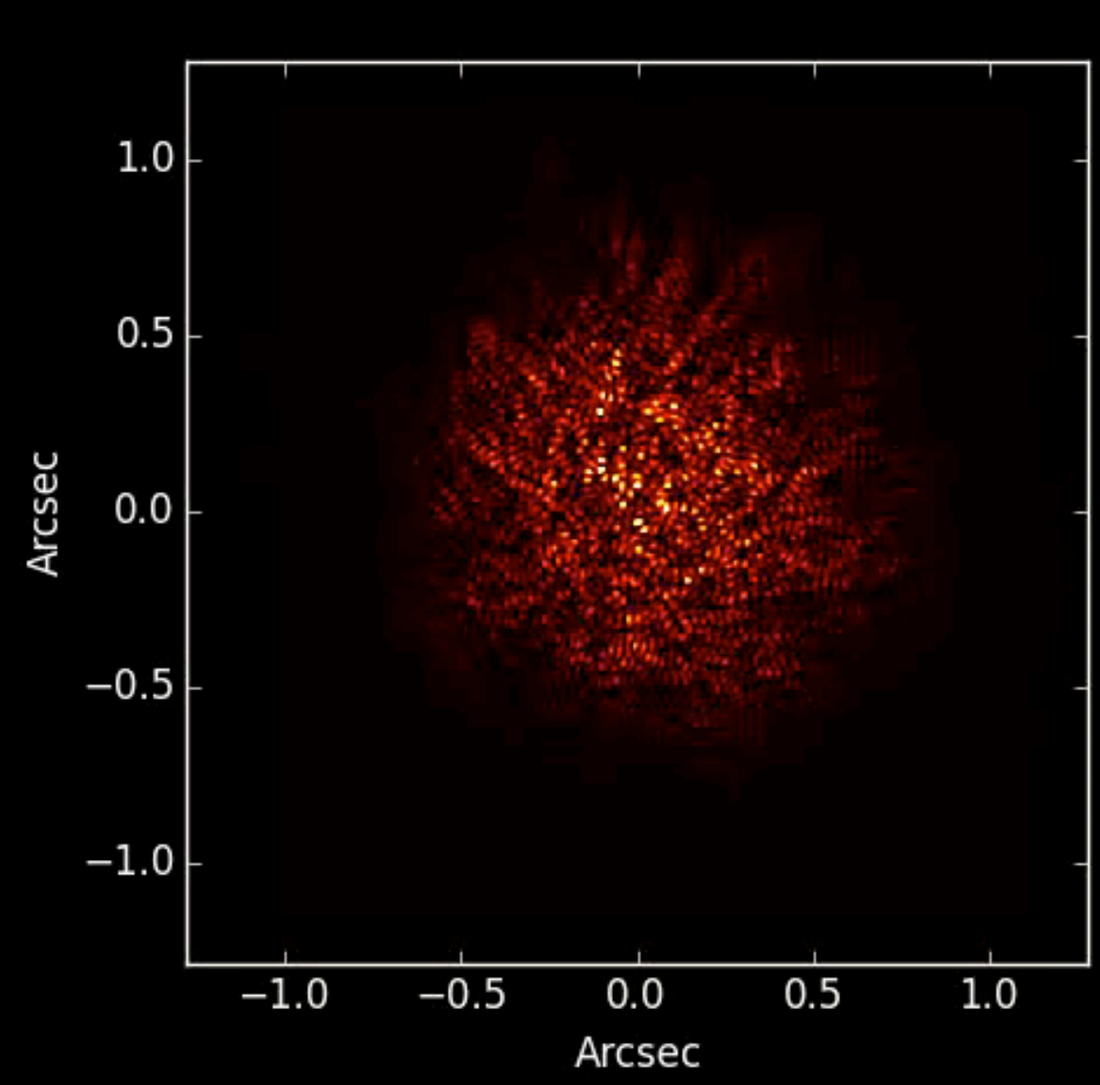


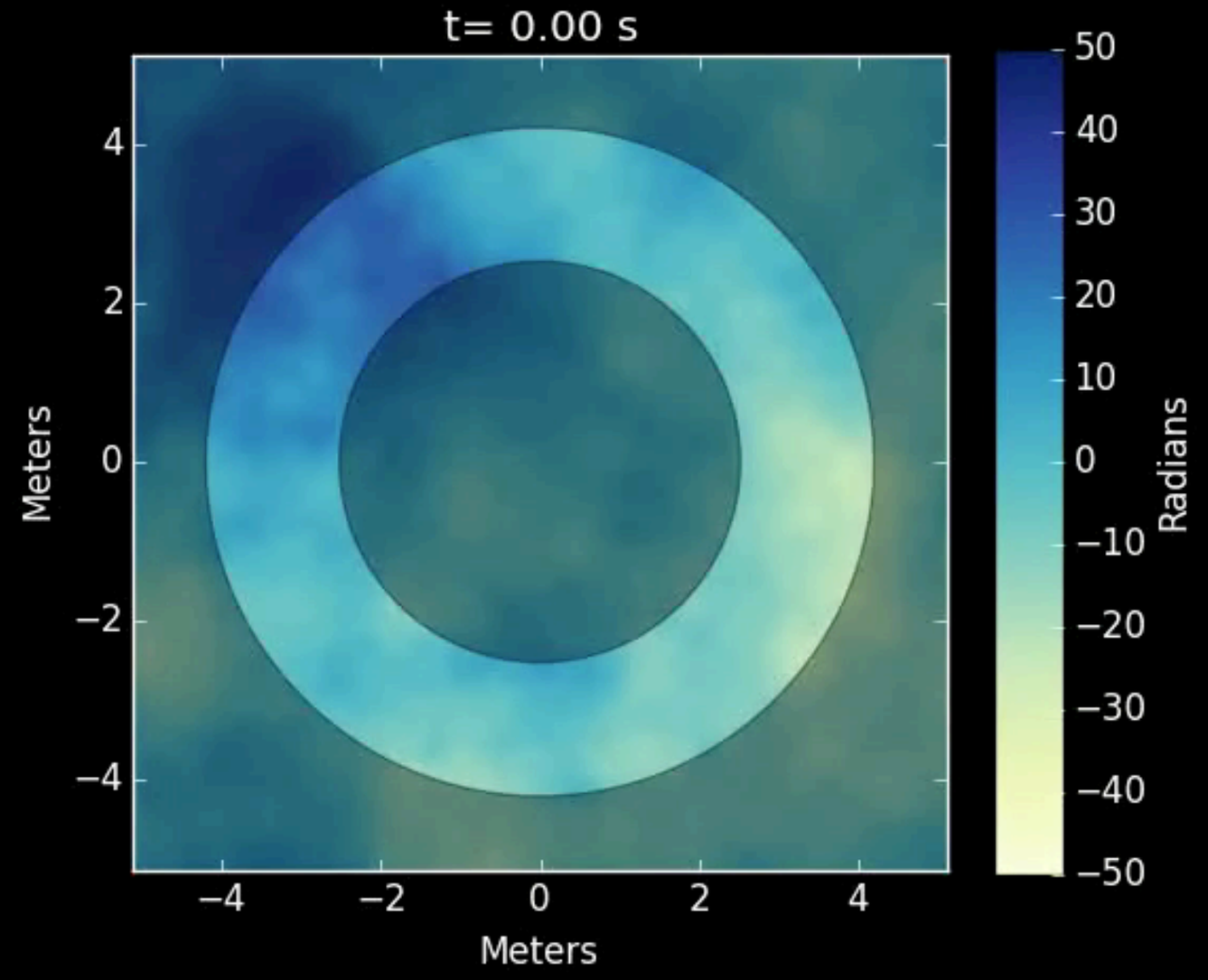


Jee+Tyson11

Instantaneous PSF

$$I(x,y) \propto \left| \mathcal{F}\left[P(u,v) \exp\left(\frac{-2\pi i}{\lambda} W(u,v) \right) \right] \right|^2$$

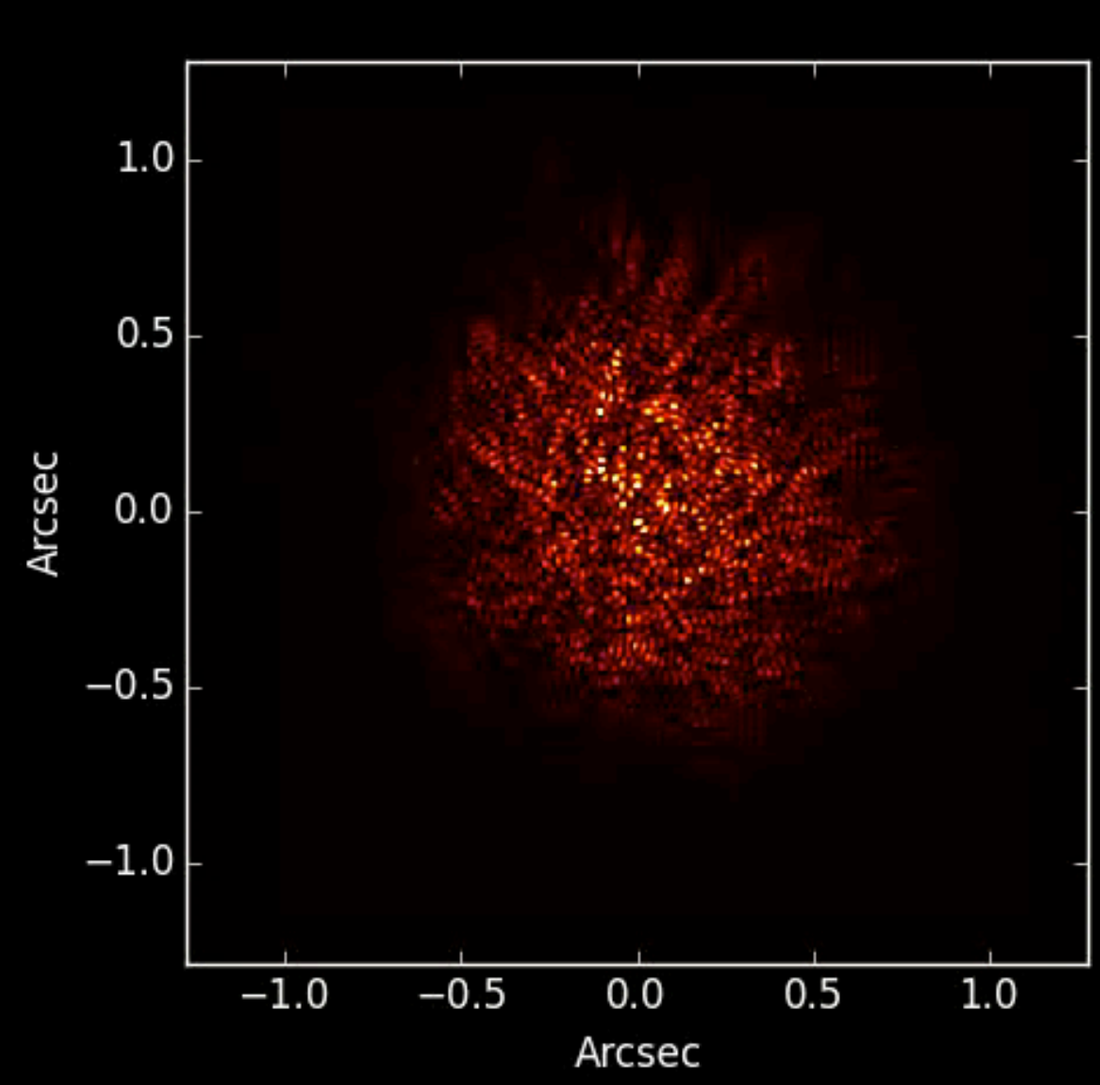


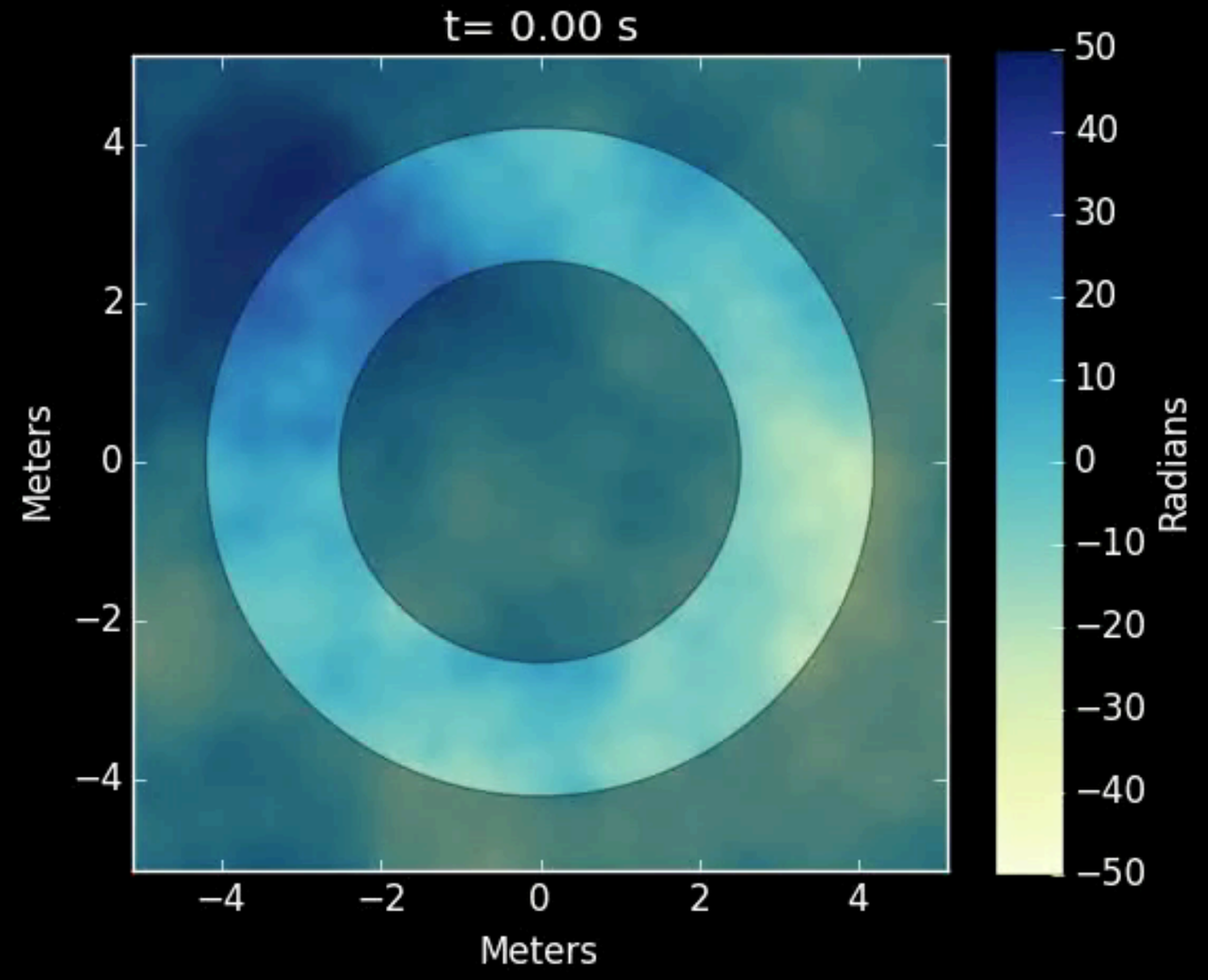


https://youtu.be/kwxKvWAxOb0

Cumulative PSF

$$I(x,y) \propto \left| \mathcal{F}\left[P(u,v) \exp\left(\frac{-2\pi i}{\lambda} W(u,v) \right) \right] \right|^2$$





https://youtu.be/1IN3Ub4lESA

Interface

```
aper = galsim.Aperture(diam=8.4, obscuration=0.6, nstruts=4, strut_thick=0.02)

screen1 = galsim.AtmosphericScreen(screen_size=100.0, screen_scale=0.1, altitude=1.,
                                   time_step=0.03, r0_500=0.15, L0=25.0, vx=1.2, vy=2.3)

screen10 = galsim.AtmosphericScreen(screen_size=100.0, screen_scale=0.1, altitude=10.,
                                    time_step=0.03, r0_500=0.3, L0=25.0, vx=-12.0, vy=-1.)

#                    Z1   Z2   Z3   Z4   Z5   Z6   Z7 and so on...
aberrations = [0.0, 0.0, 0.0, 0.0, 0.1, 0.2, 0.3, 0.4]

optics = galsim.OpticalScreen(aberrations=aberrations, lam_0=500.0)
```

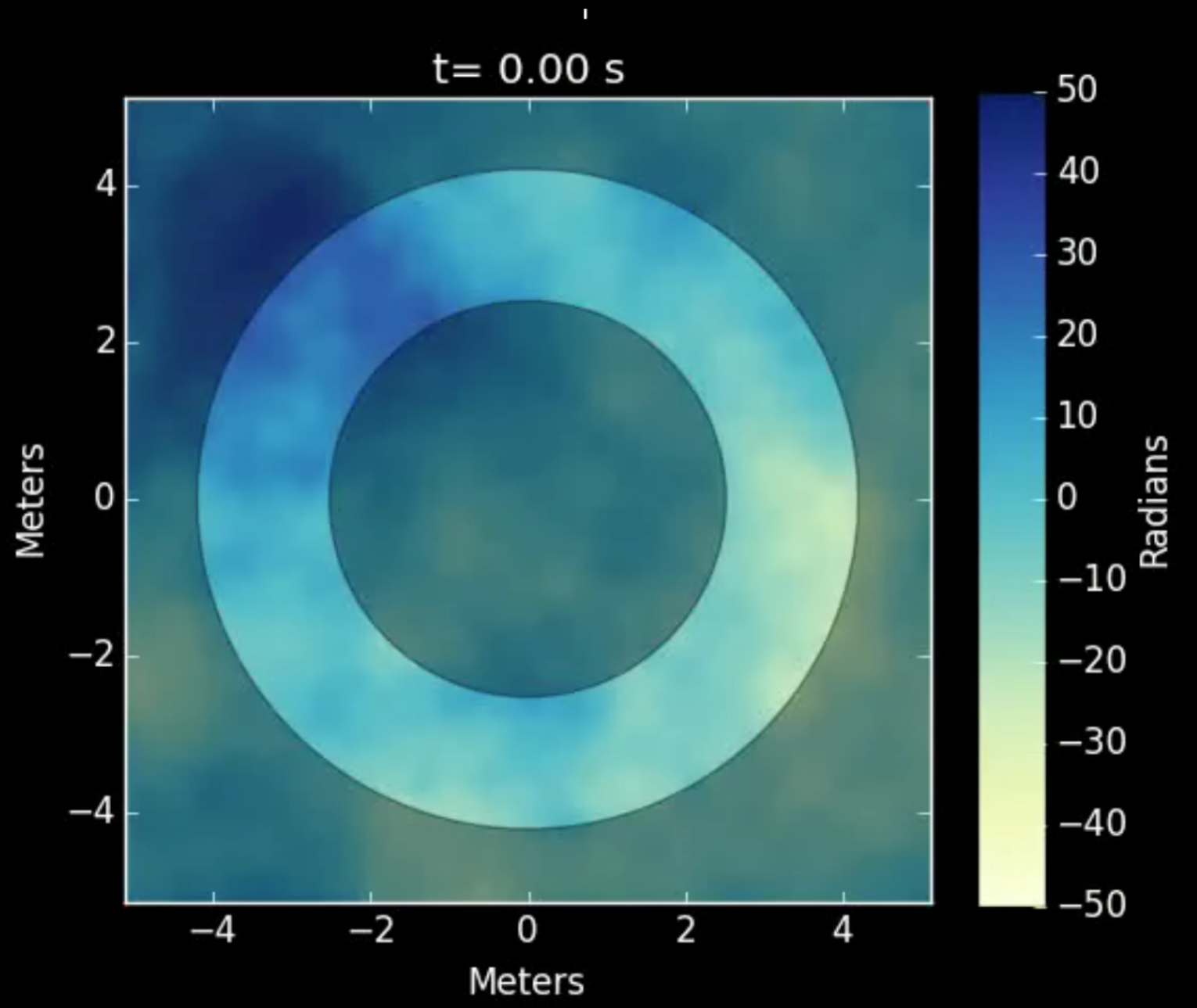
```
screens = galsim.PhaseScreenList([screen1, screen10, optics])

wavefront = screens.wavefront(aper, theta_x=0.1*galsim.degrees, theta_y=0.2*galsim.degrees)

# Get 5x5 array of PSFs
thx = thy = numpy.linspace(-0.5, 0.5, 5)
thx, thy = numpy.meshgrid(thx, thy)

PSFs = screens.makePSF(lam=625., aper=aper, exptime=30.0,
                       theta_x=thx*galsim.degrees, theta_y=thy*galsim.degrees)
```

Speed and Accuracy

- Both under assessment
  - especially accuracy
    - especially PSF correlations

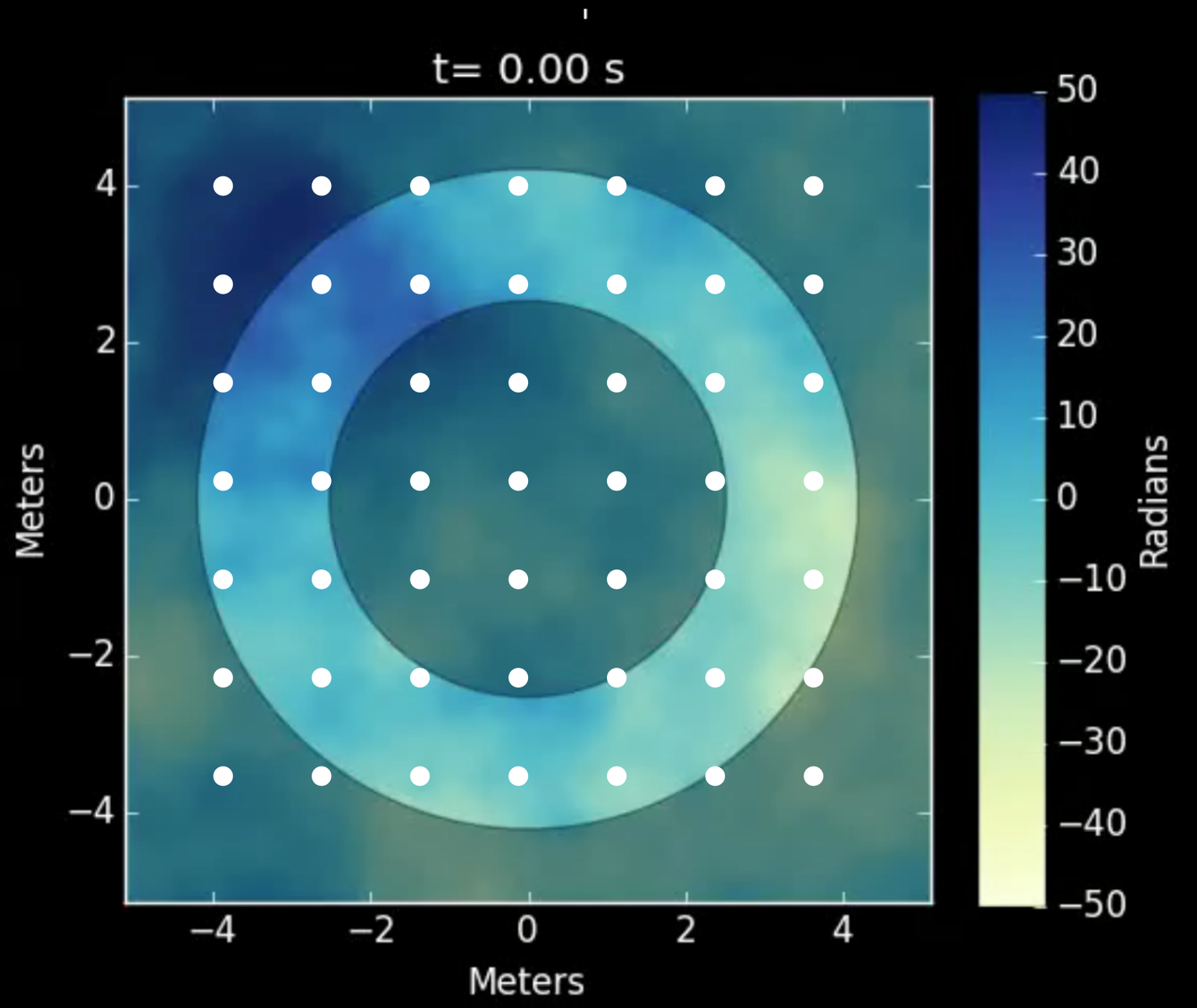
- Speed for 6-layer atmosphere with default settings:
  - 2.5h per 30s monochromatic PSF.
- Cautiously optimistic potential speed:
  - ~minute per monochromatic PSF.

Knobs


t= 0.00 s
Meters
Meters
Radians
4
2
0
−2
−4
−4
−2
0
2
4
50
40
30
20
10
0
−10
−20
−30
−40
−50

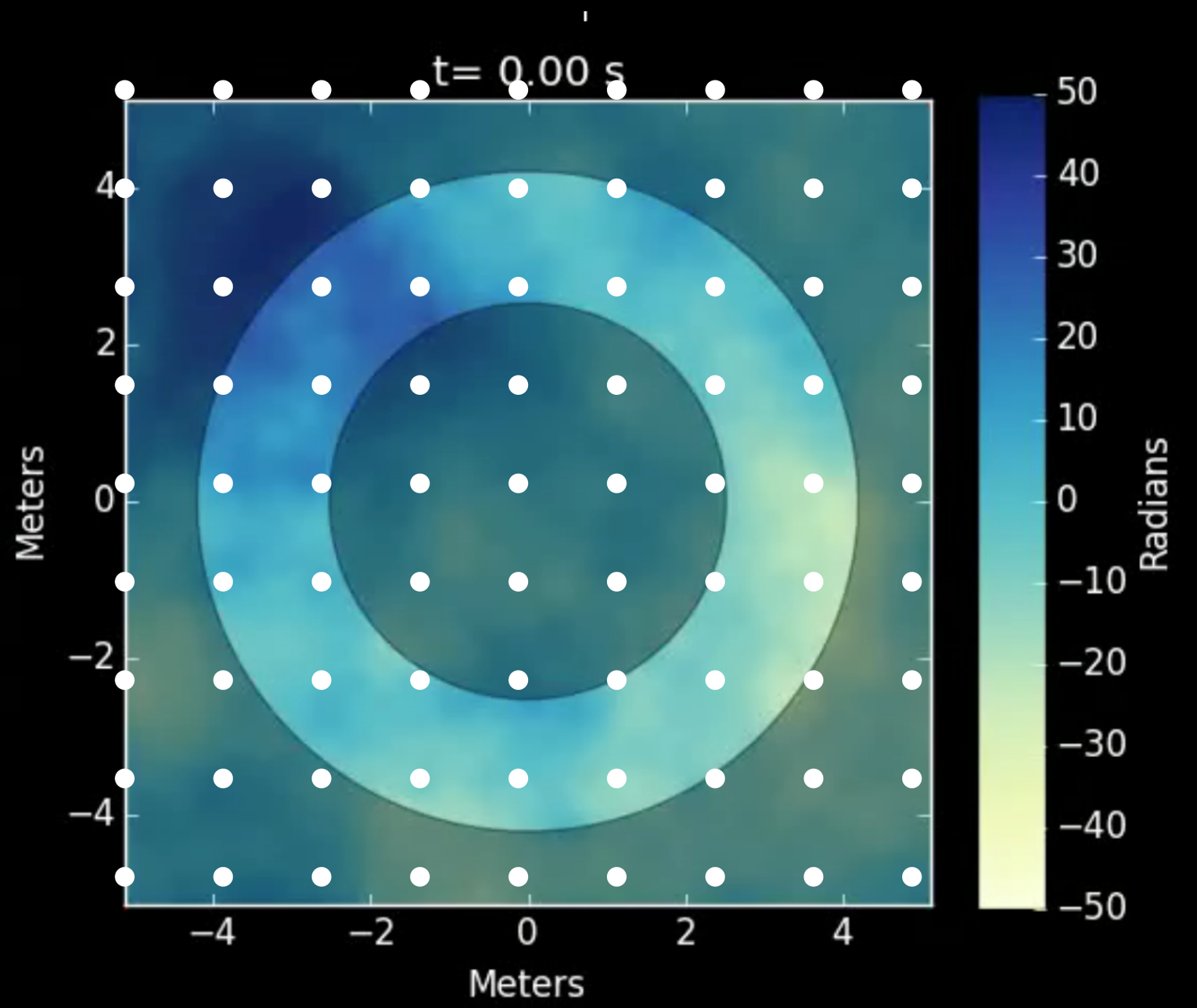
Knobs

- Pupil sampling



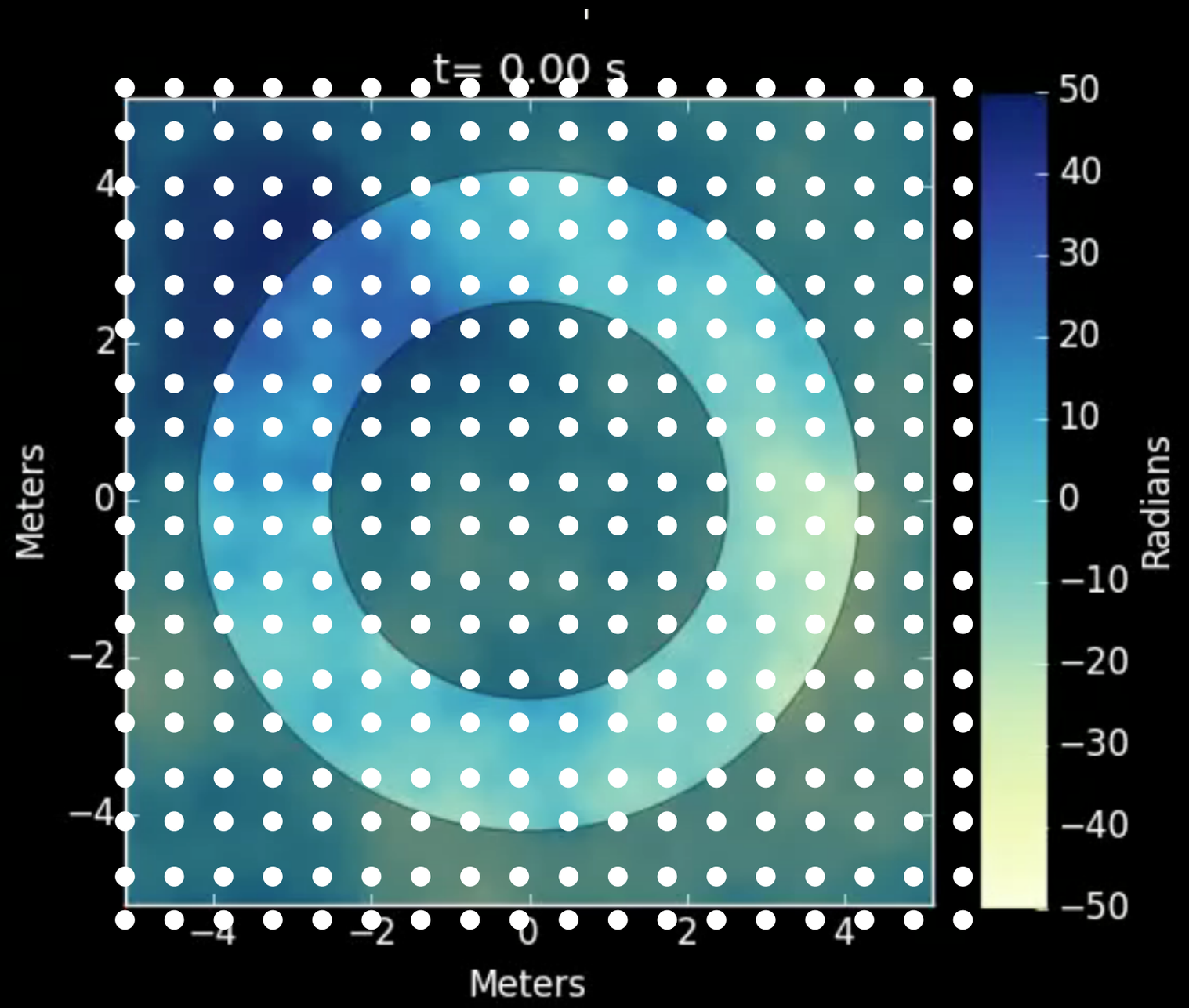
Knobs

- Pupil sampling
  - zero-padding



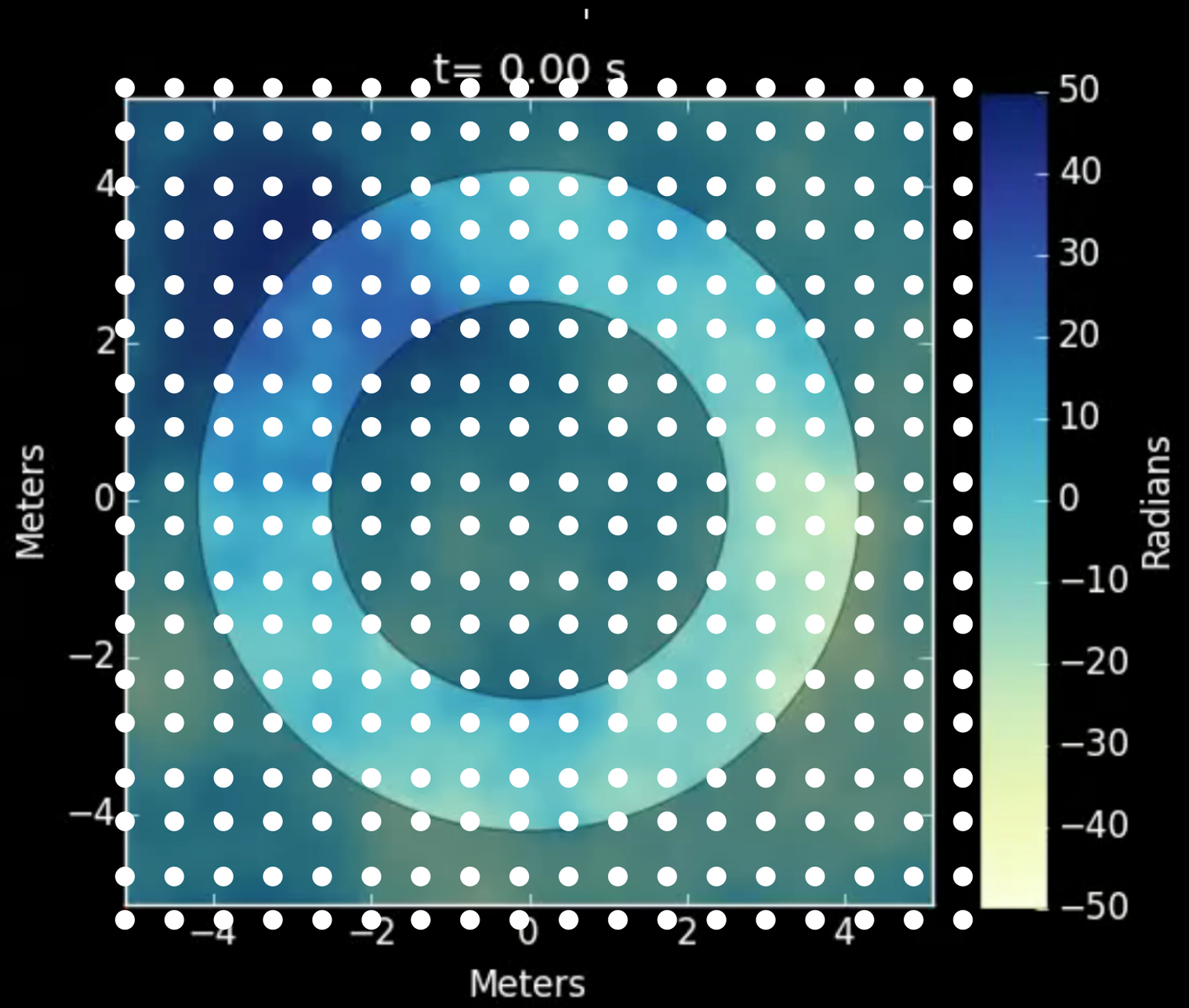
Knobs

- Pupil sampling
  - zero-padding
  - sampling frequency



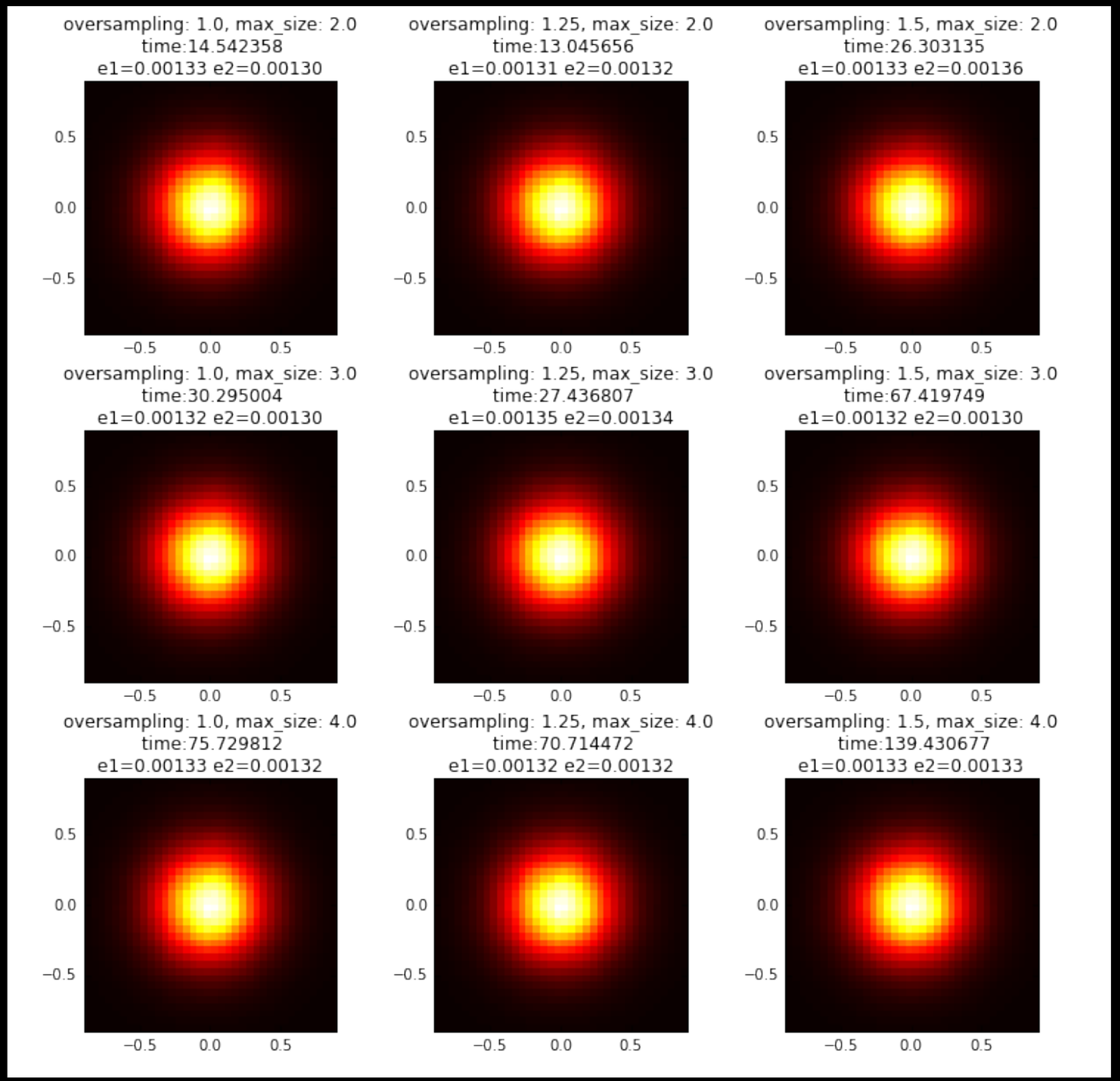
Knobs

- Pupil sampling
  - zero-padding
  - sampling frequency
- Time step interval



Direct comparison of sampling settings

max_size ~ stepK

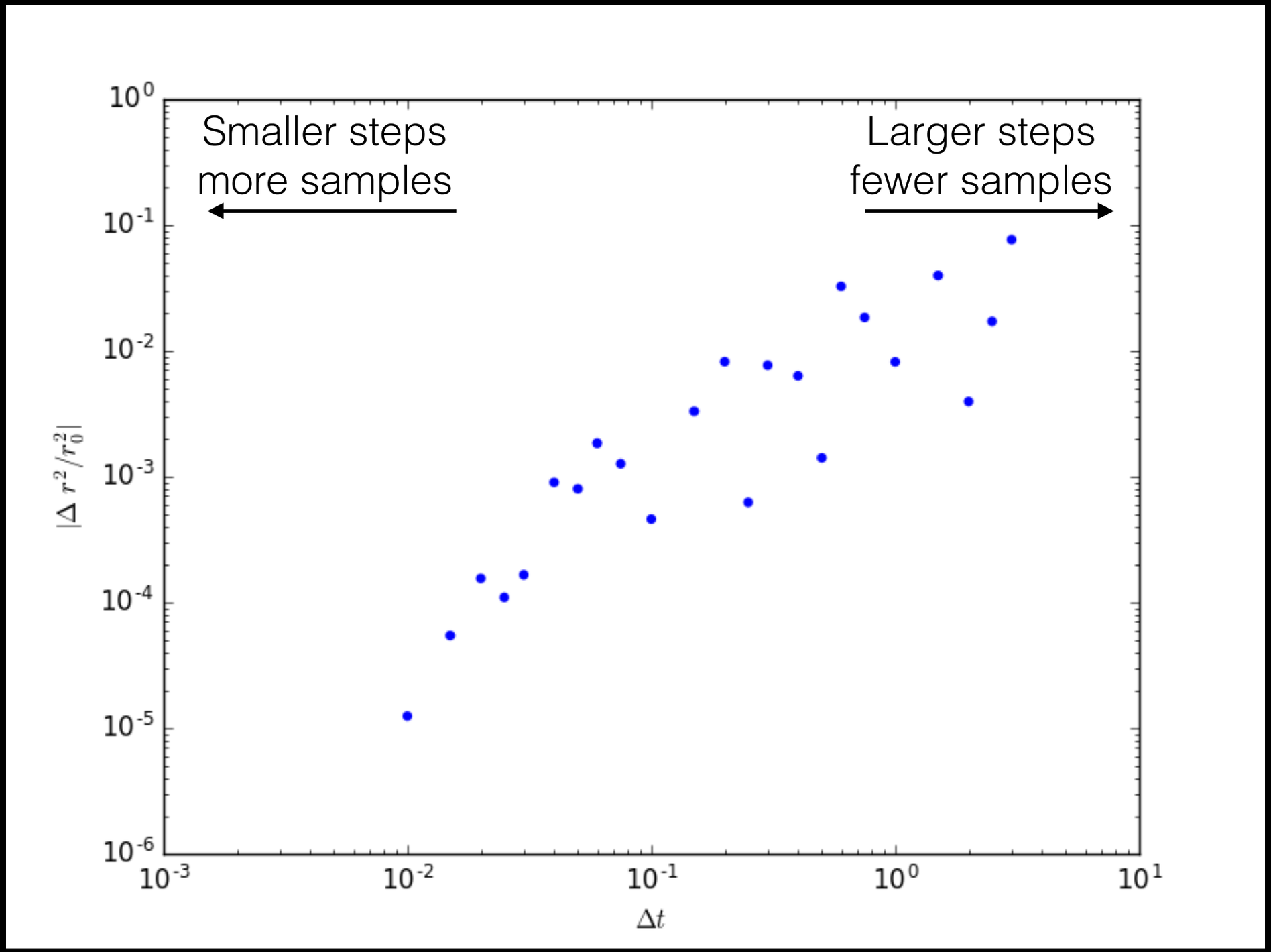


Oversampling ~ maxK

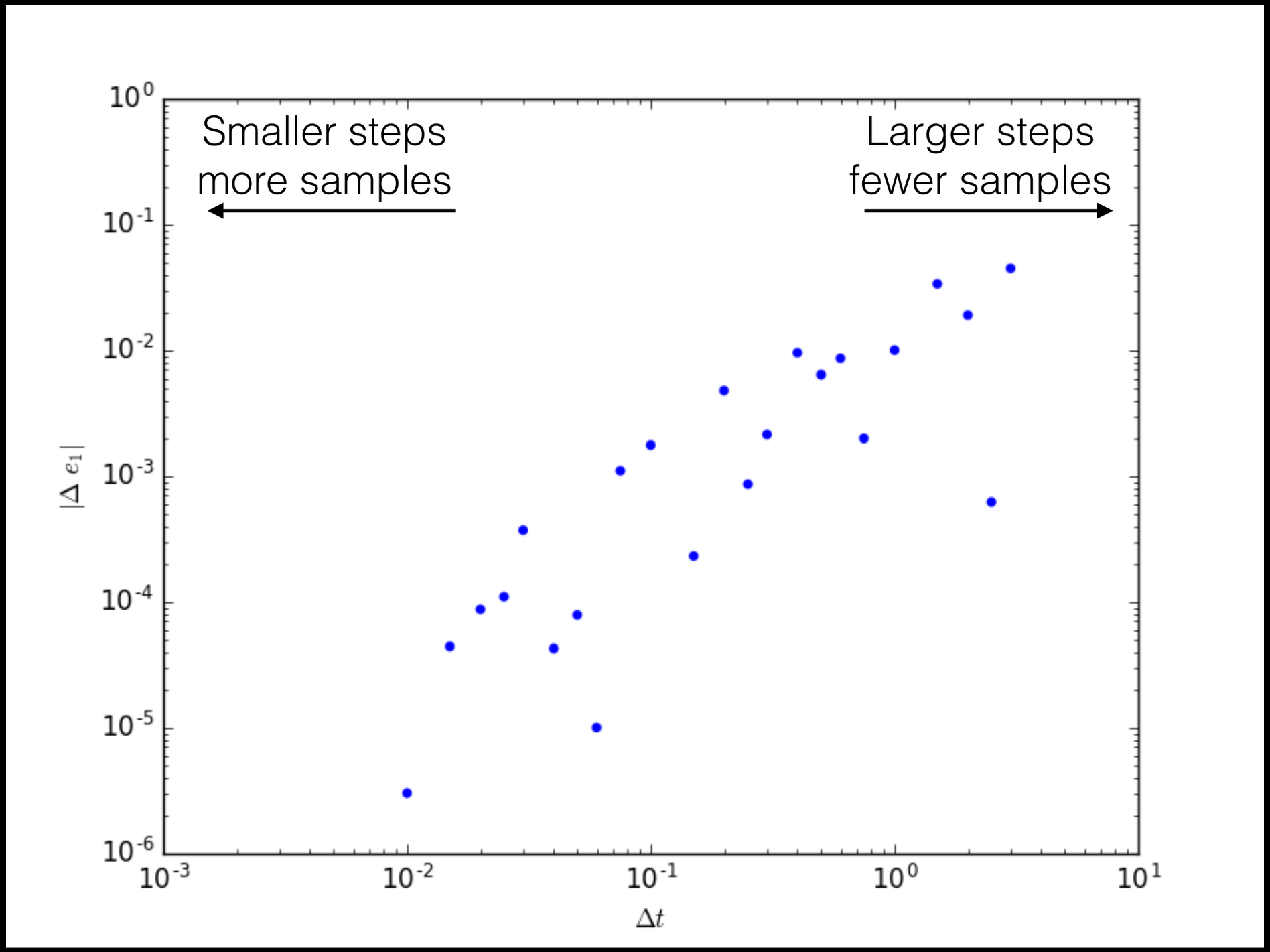
Time step

- Default (5 ms) follows Jee+Tyson (2011)
  - Roughly half the ratio of the Fried parameter (0.16 m) to the maximum wind speed (20 m/s).
  - (I think) this is mostly important for getting PSF correlations right.

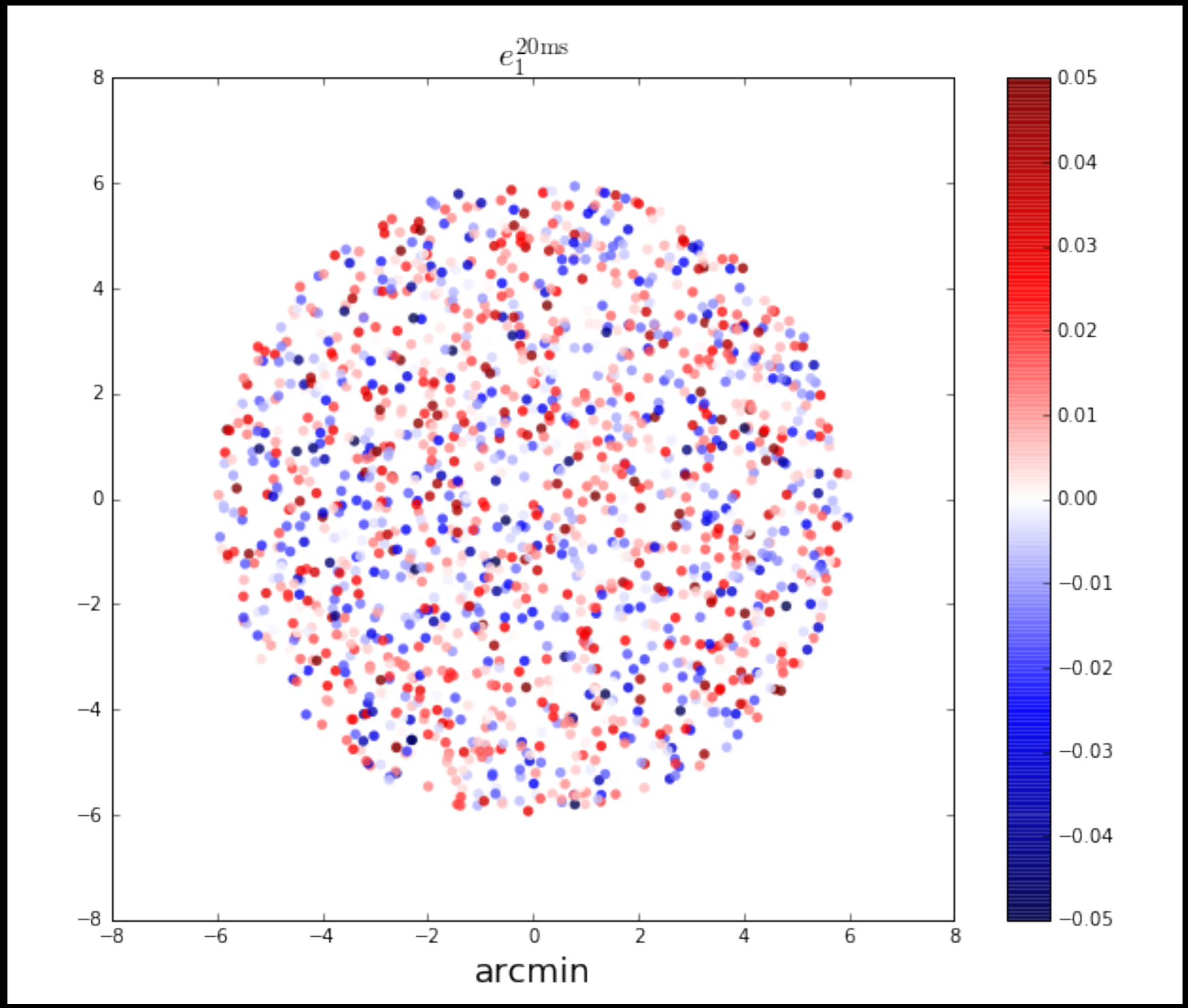
Single 30 s PSFs with different temporal samplings to PSF with default sampling



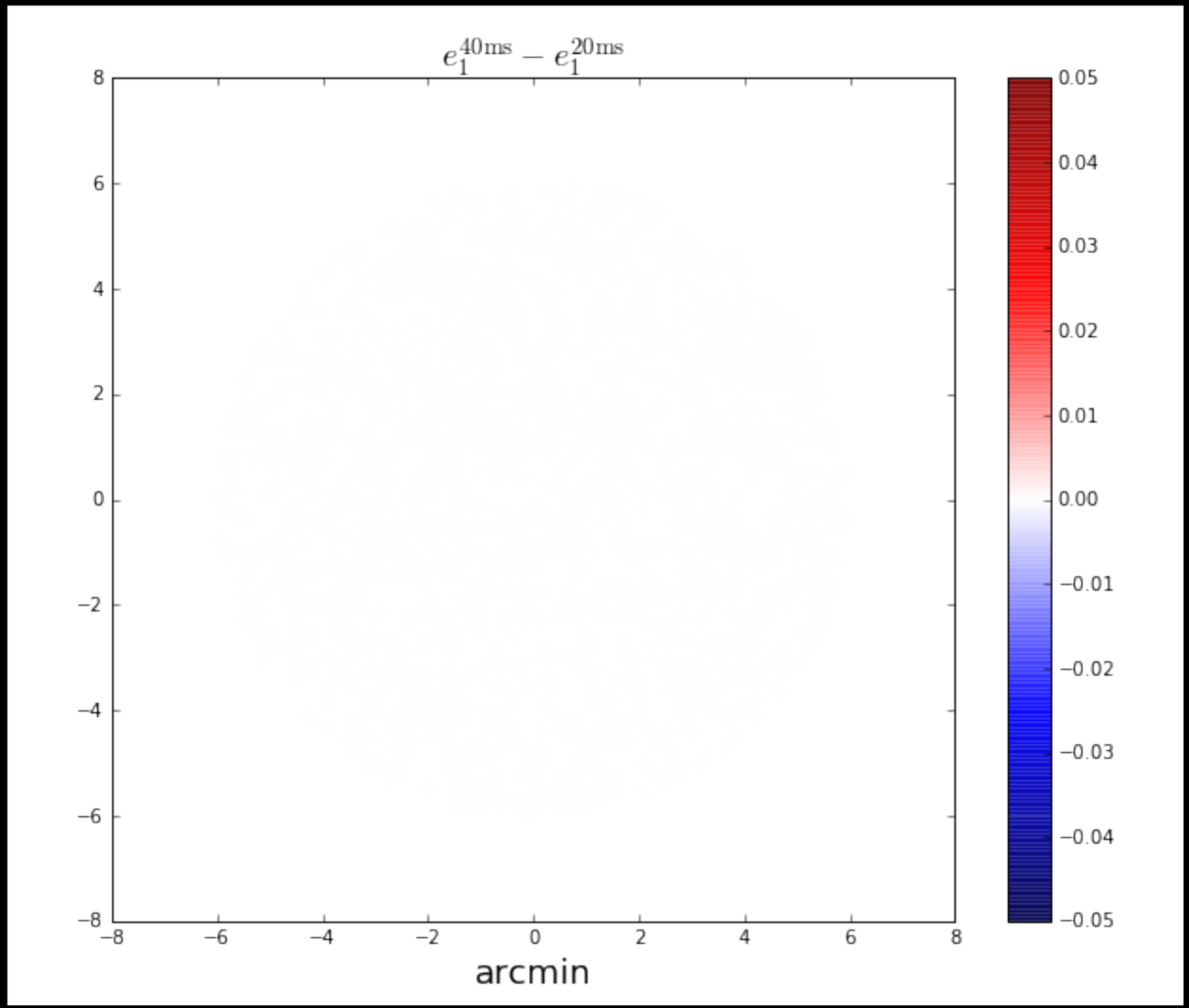
Single 30 s PSFs with different temporal samplings to PSF with default sampling



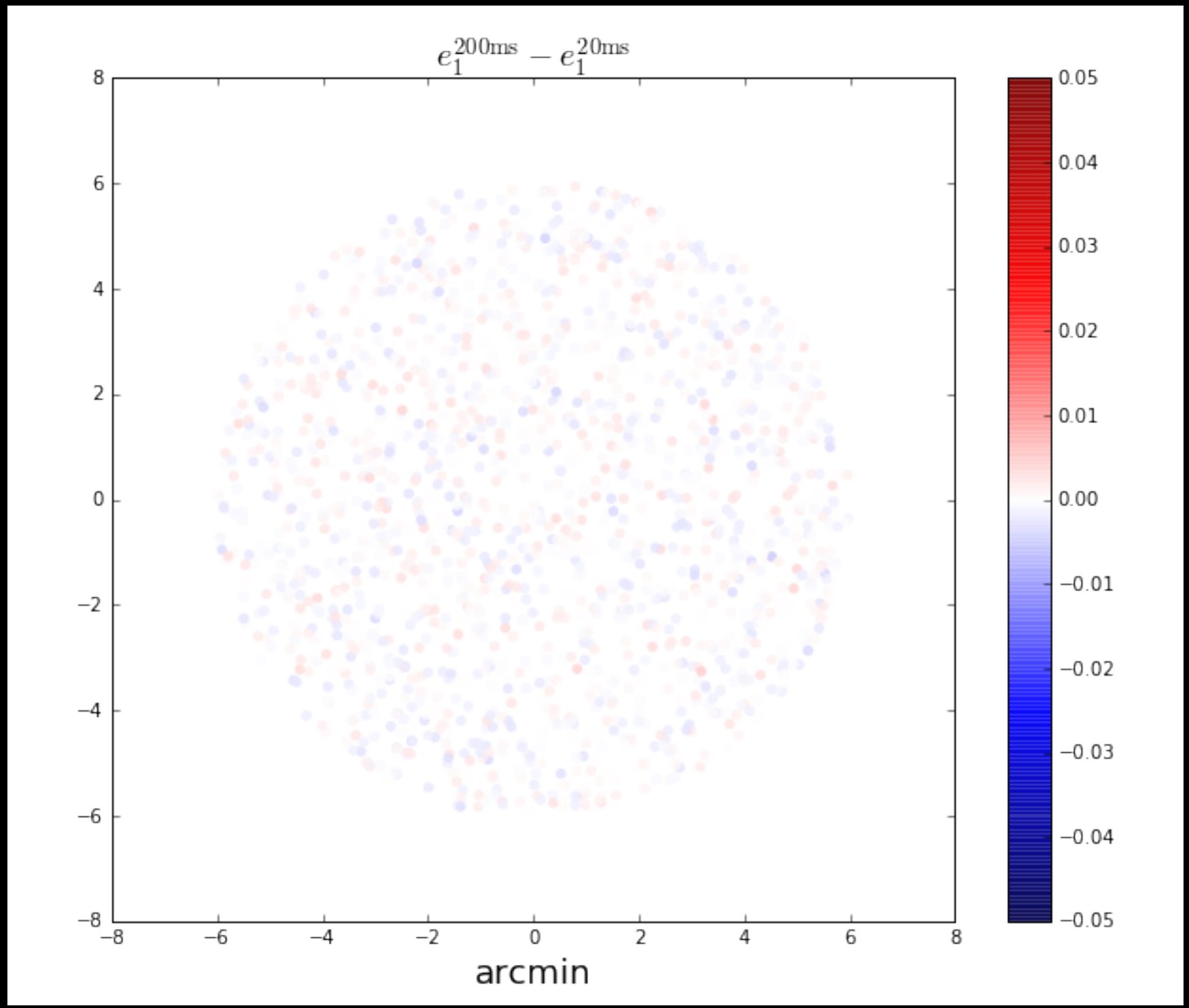
PSFs across the field of view



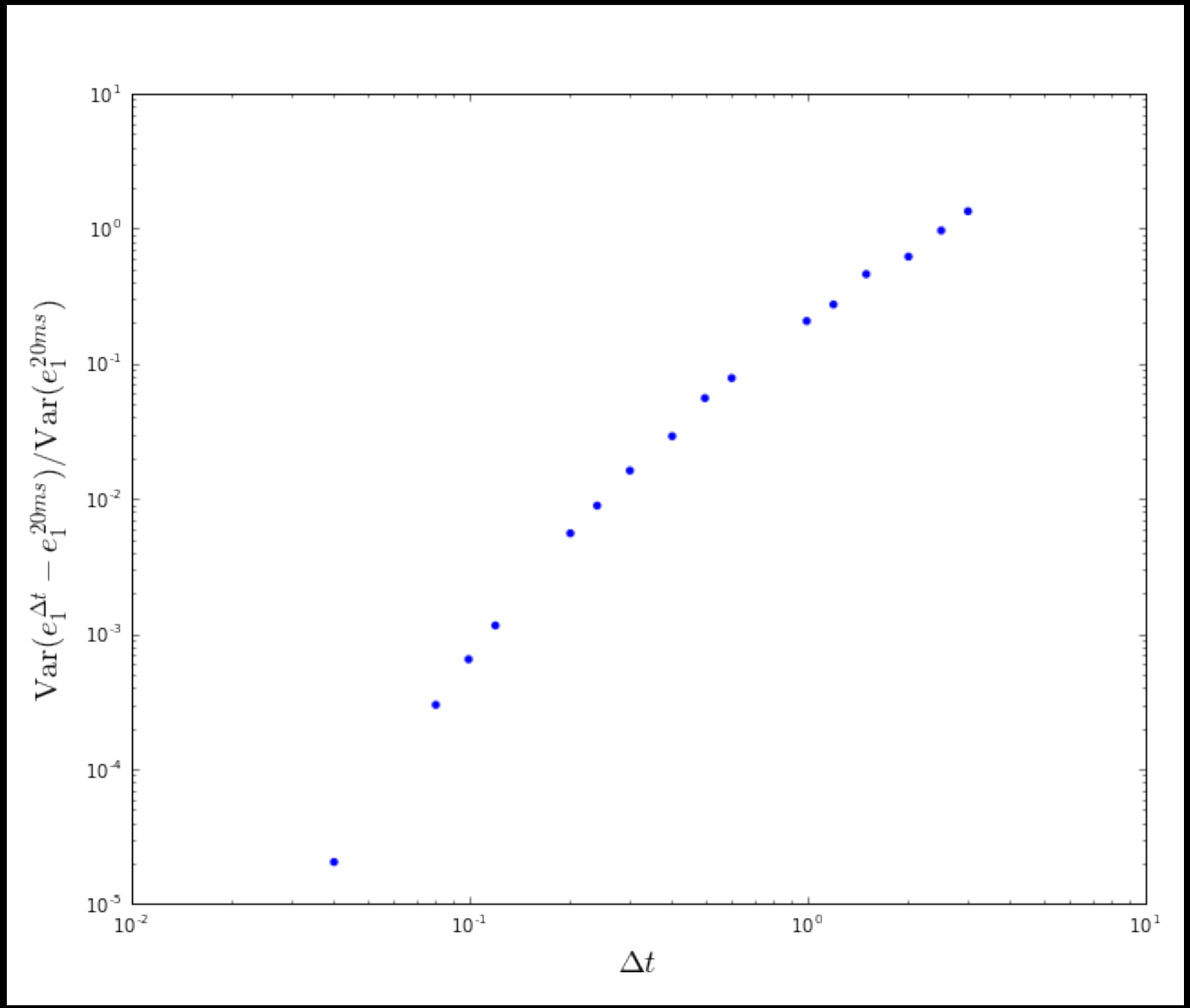
PSFs across the field of view



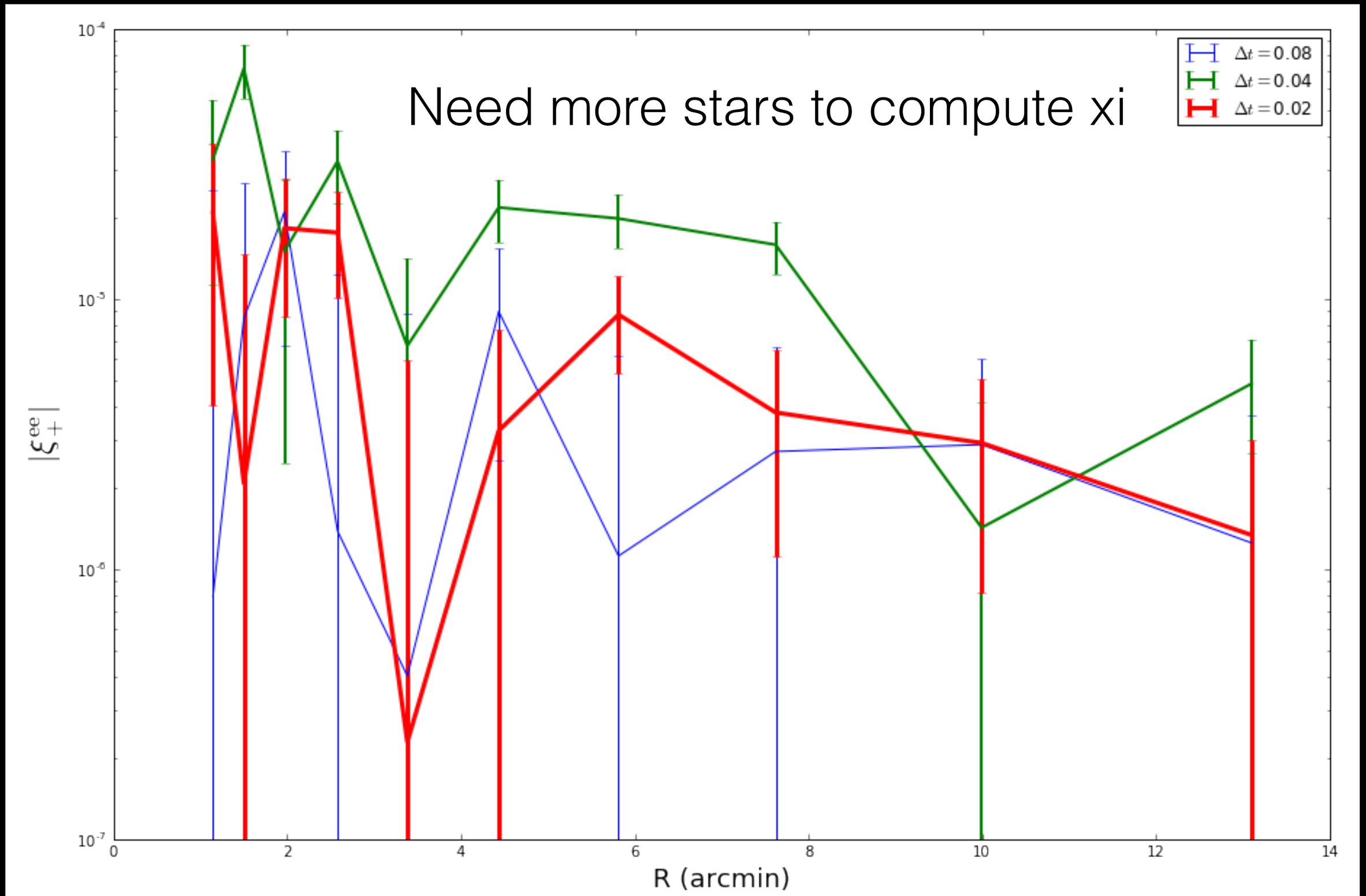
PSFs across the field of view



PSFs across the field of view



PSFs across the field of view



Questions.

- Appropriate atmospheric screen parameters? Number of layers?
- What are requirements for “realism?”
- What are requirements for “realistic level of complexity?”

Backup slides

Conclusions & todo

- Cautiously optimistic that can reduce running time of single monochromatic atmospheric PSF from ~few hours to ~few minutes while maintaining a “realistic level of complexity.”

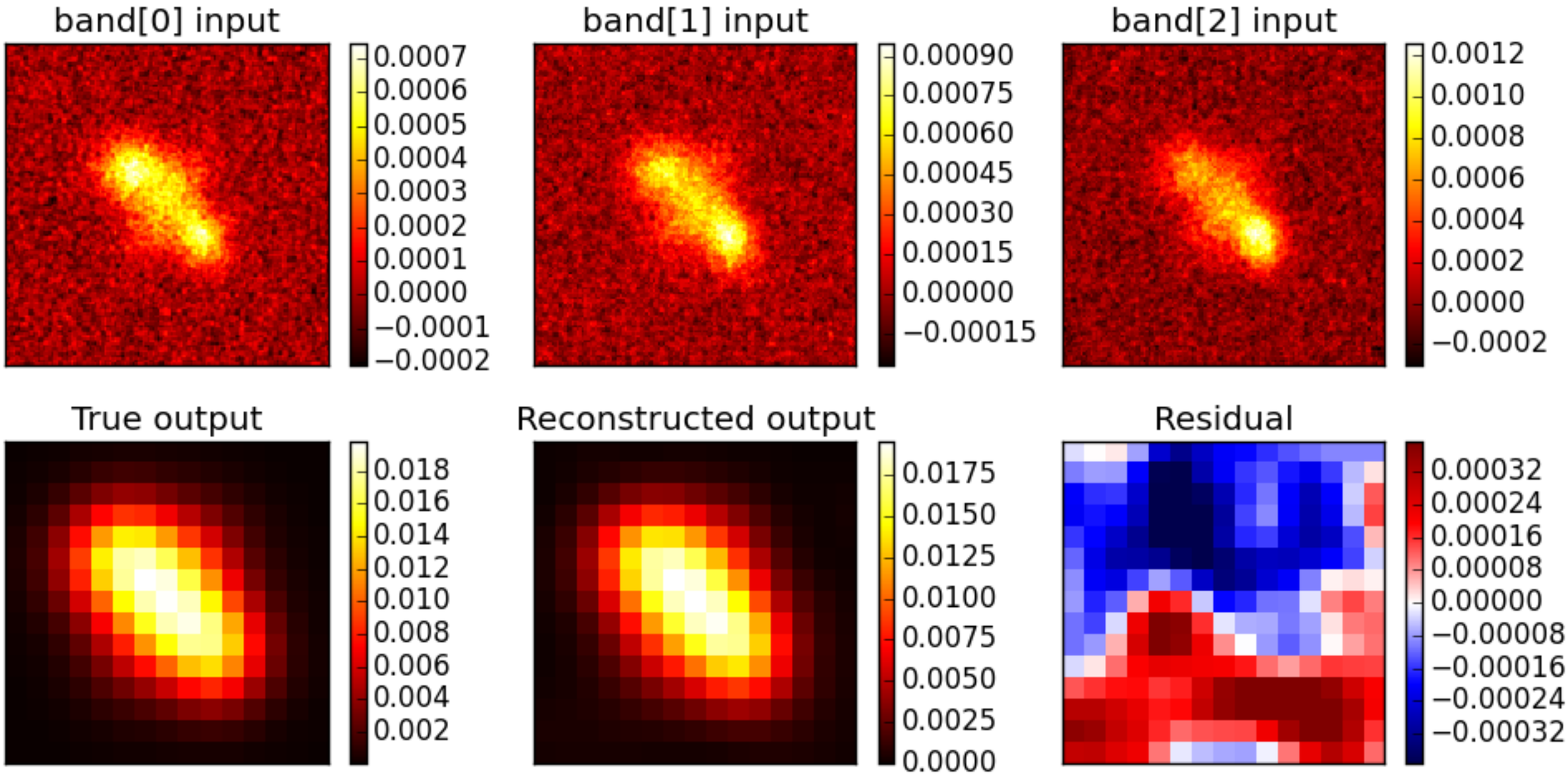
- Exact level of realism is tricky.

    - In particular, need to verify that atmospheric PSF correlation function is “realistic.”

    - What are are the requirements?

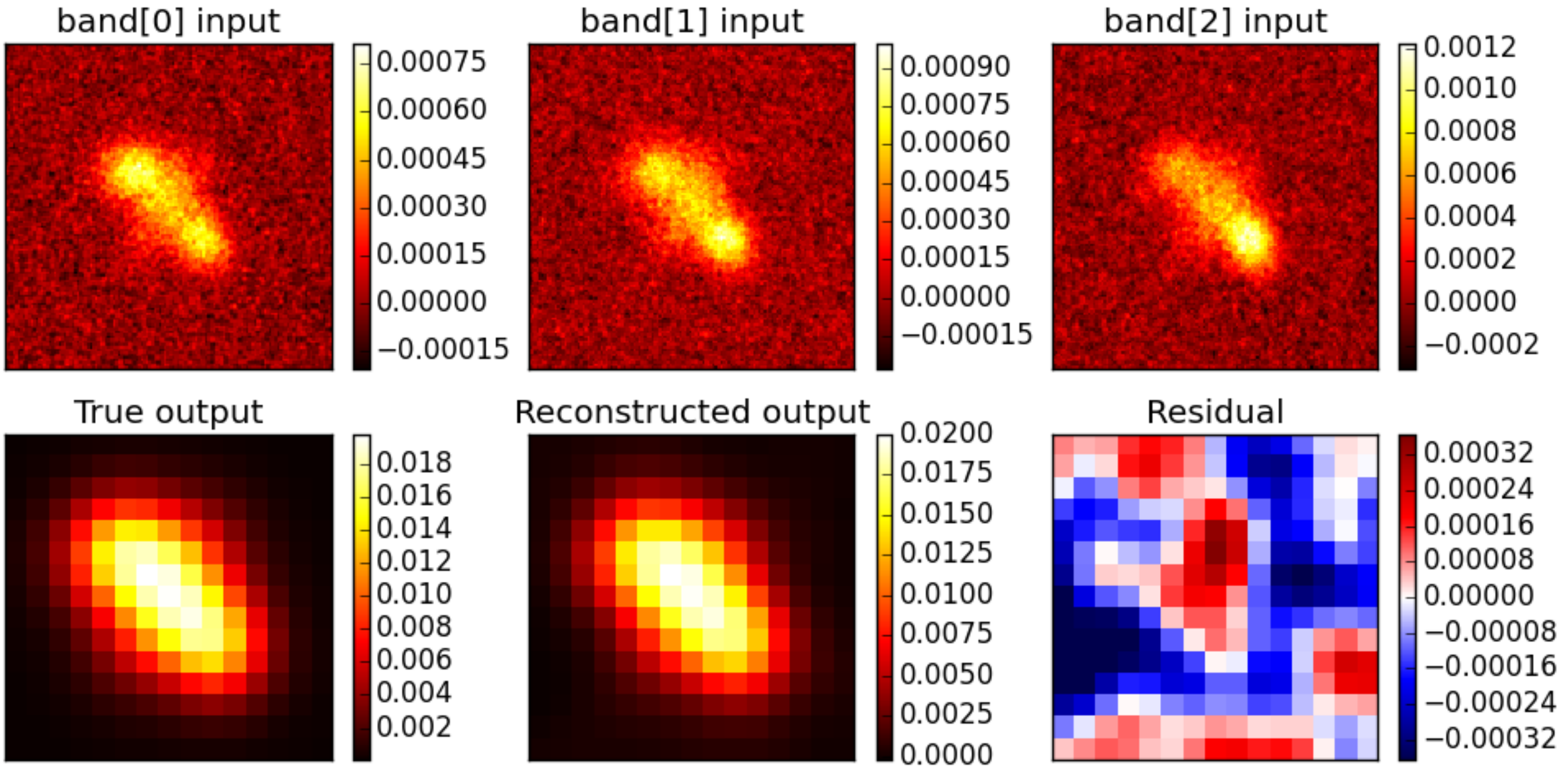
(Toy) example

- r-, i-, z-like input filters, out-of-phase output filter and LSST-like PSF, pixel scale.



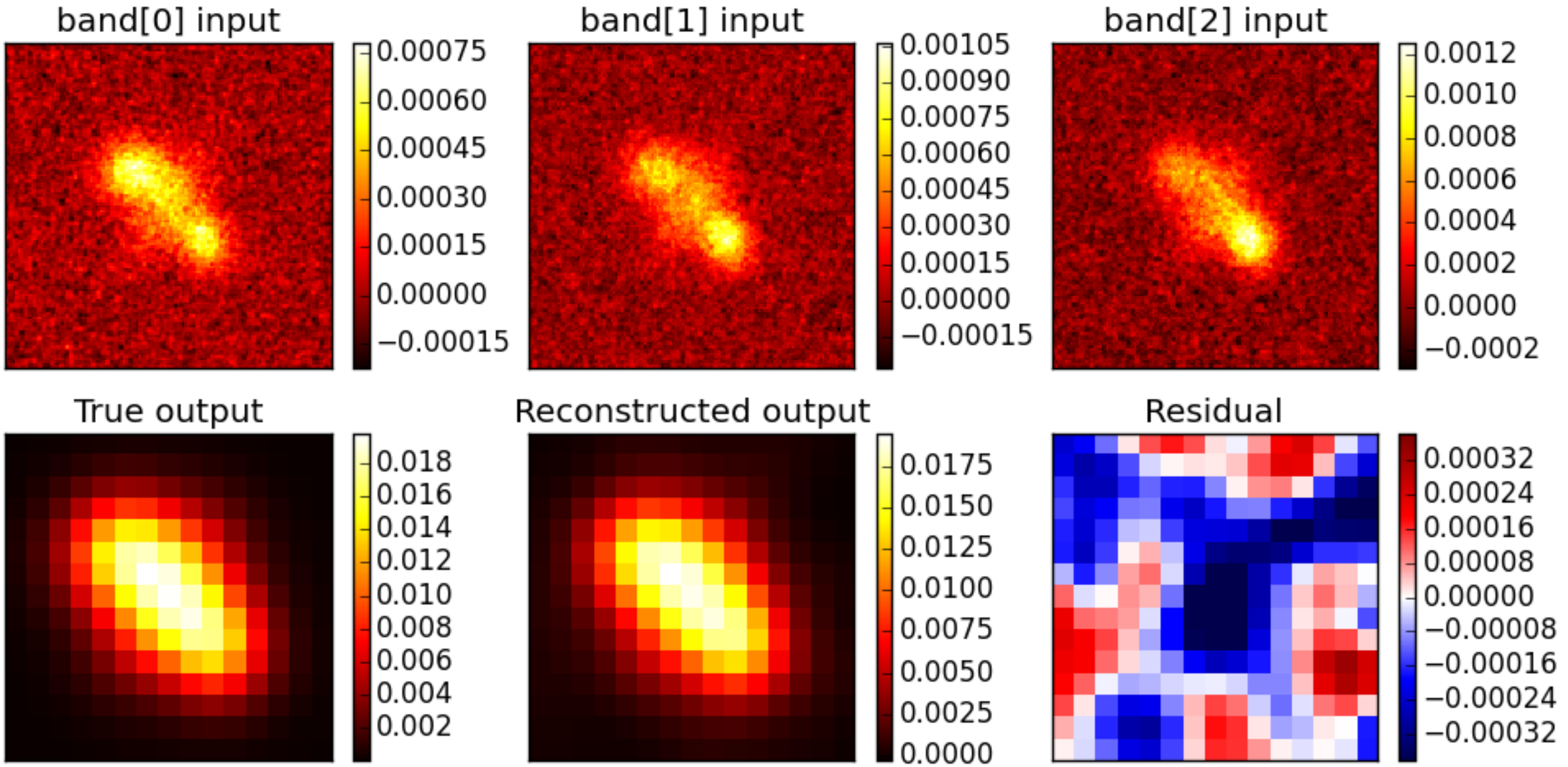
(Toy) example

- r-, i-, z-like input filters, out-of-phase output filter and LSST-like PSF, pixel scale. Different random seed.



(Toy) example

- r-, i-, z-like input filters, out-of-phase output filter and LSST-like PSF, pixel scale. Yet another different random seed.



Planes of interest and relations required for FFTs

$$P(u,v)\exp\left(\frac{-2\pi i}{\lambda}W(u,v)\right)$$

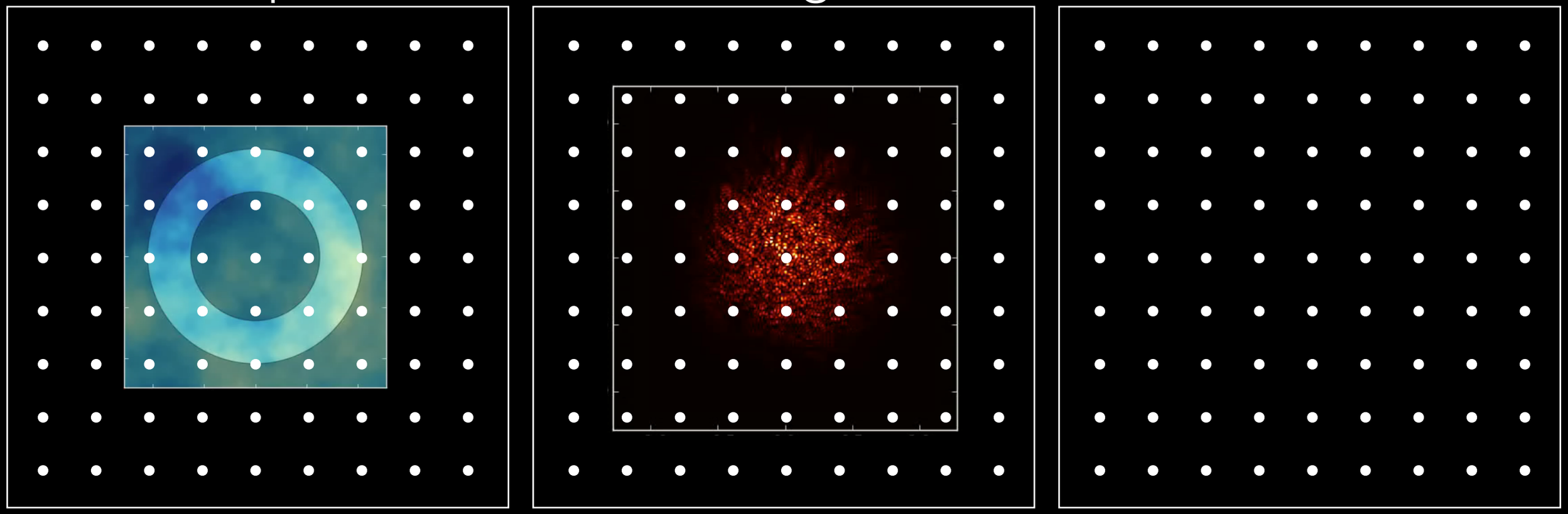
$$\left|\mathcal{F}\left[P(u,v)\exp\left(\frac{-2\pi i}{\lambda}W(u,v)\right)\right]\right|^2$$

$$\mathcal{F}^{-1}\left\{\left|\mathcal{F}\left[P(u,v)\exp\left(\frac{-2\pi i}{\lambda}W(u,v)\right)\right]\right|^2\right\}$$

Pupil

Image

Fourier



$\leftrightarrow \Delta L$

L

$\leftrightarrow \Delta\theta$

θ

$\leftrightarrow$ stepK

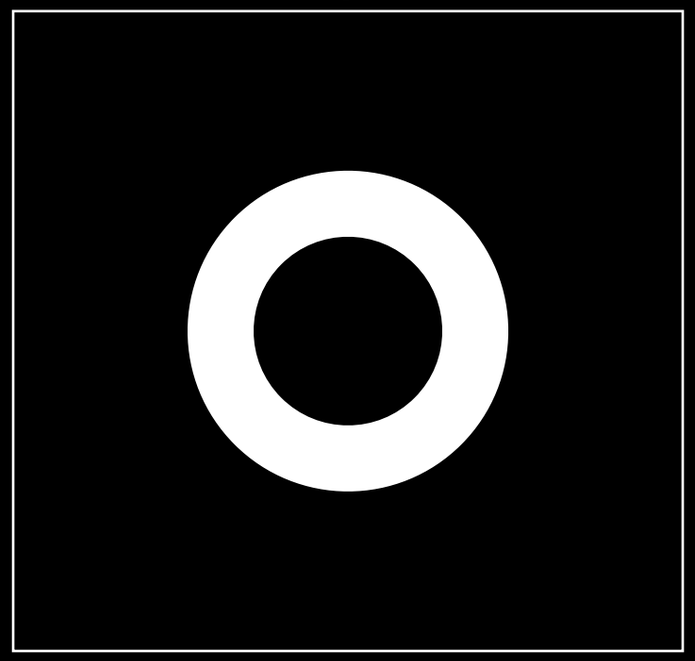
$2 \times \text{maxK}$

$$L = \frac{\lambda}{\Delta\theta} = \frac{\lambda \cdot \text{maxK}}{\pi}$$

$$\Delta L = \frac{\lambda}{\theta} = \frac{\lambda \cdot \text{stepK}}{2\pi}$$

Obscured Airy

Pupil

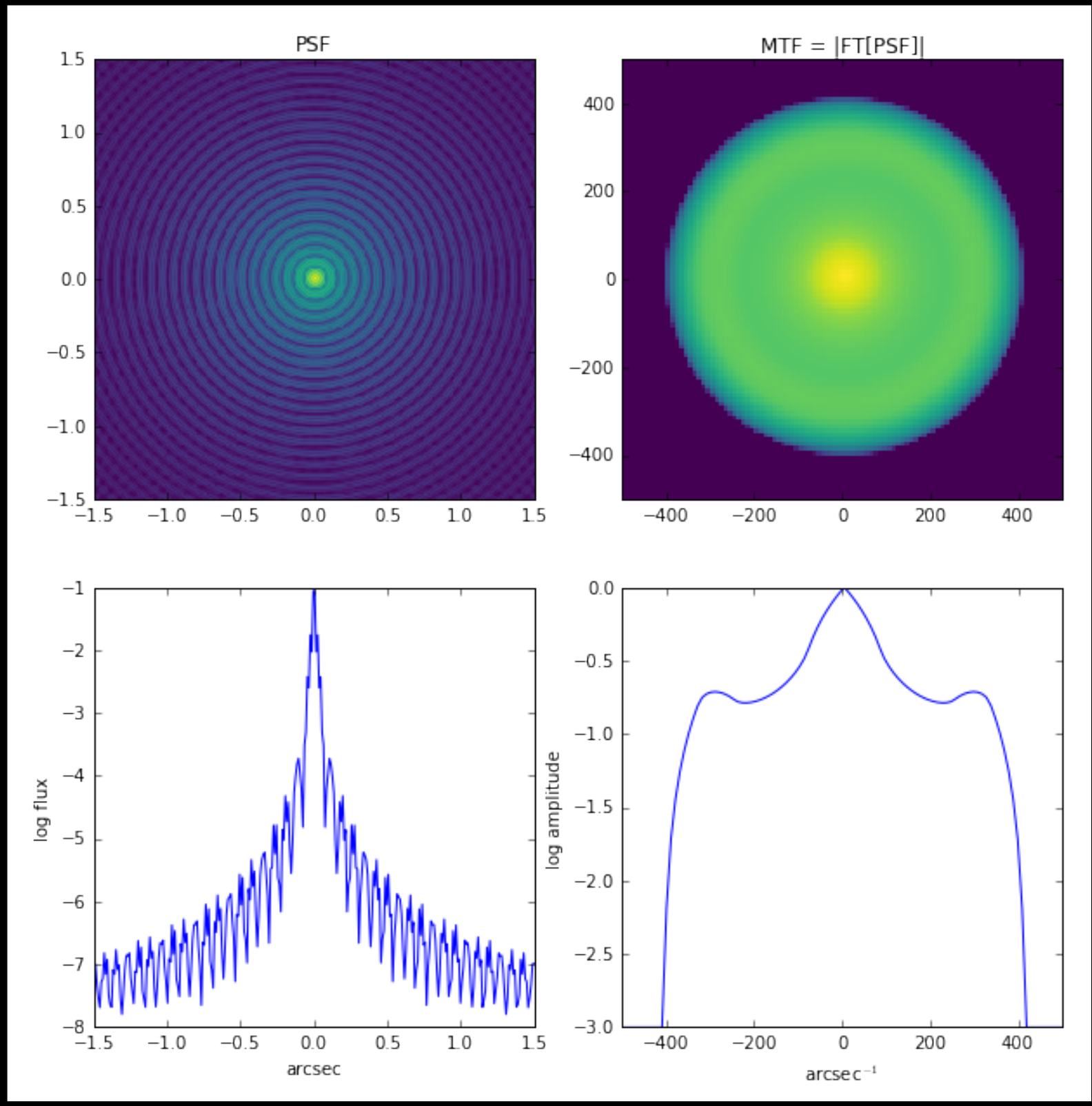


need large maxK

compact PSF

can use large stepK

maxK/stepK ~ 200



Kolmogorov

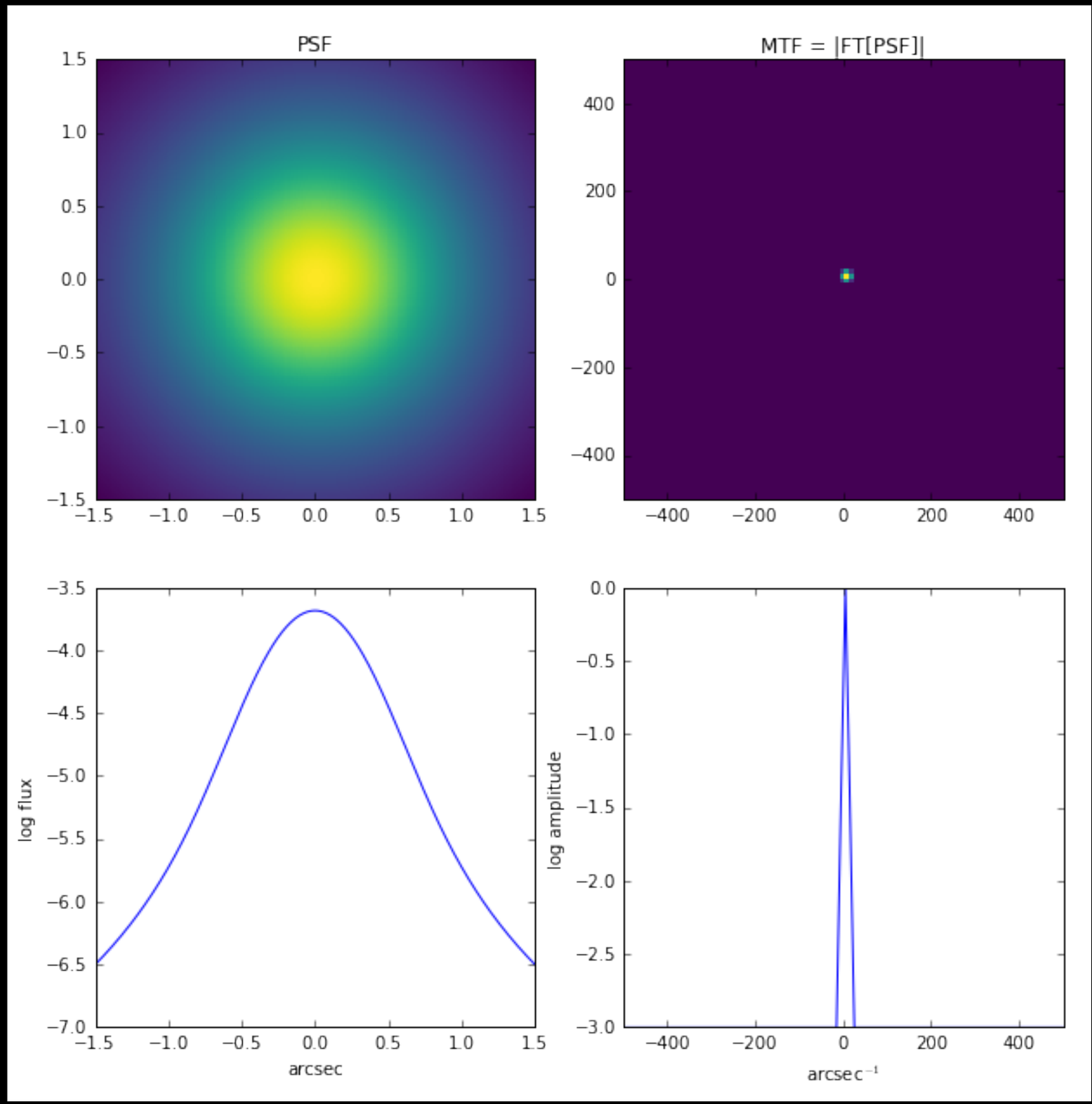
No pupil

can use small maxK

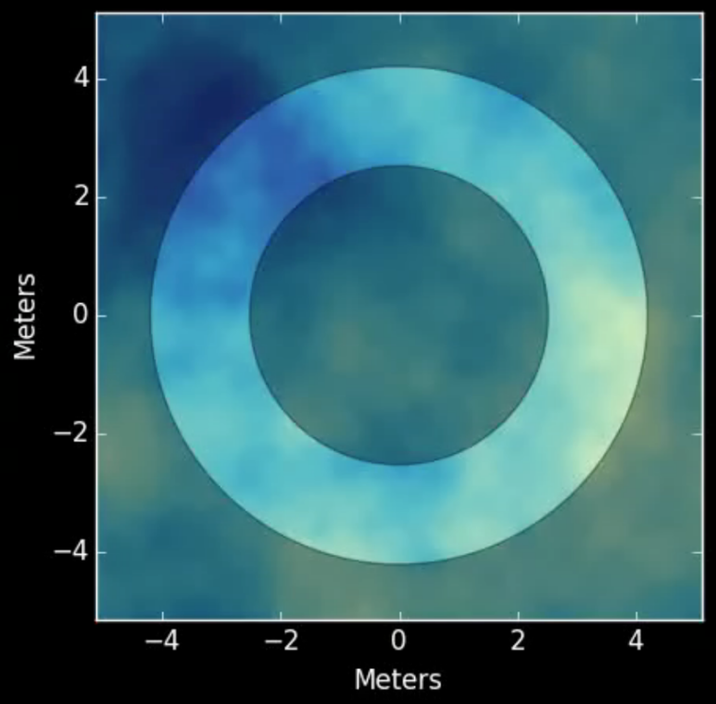
wide PSF

need small stepK

maxK/stepK ~ 25

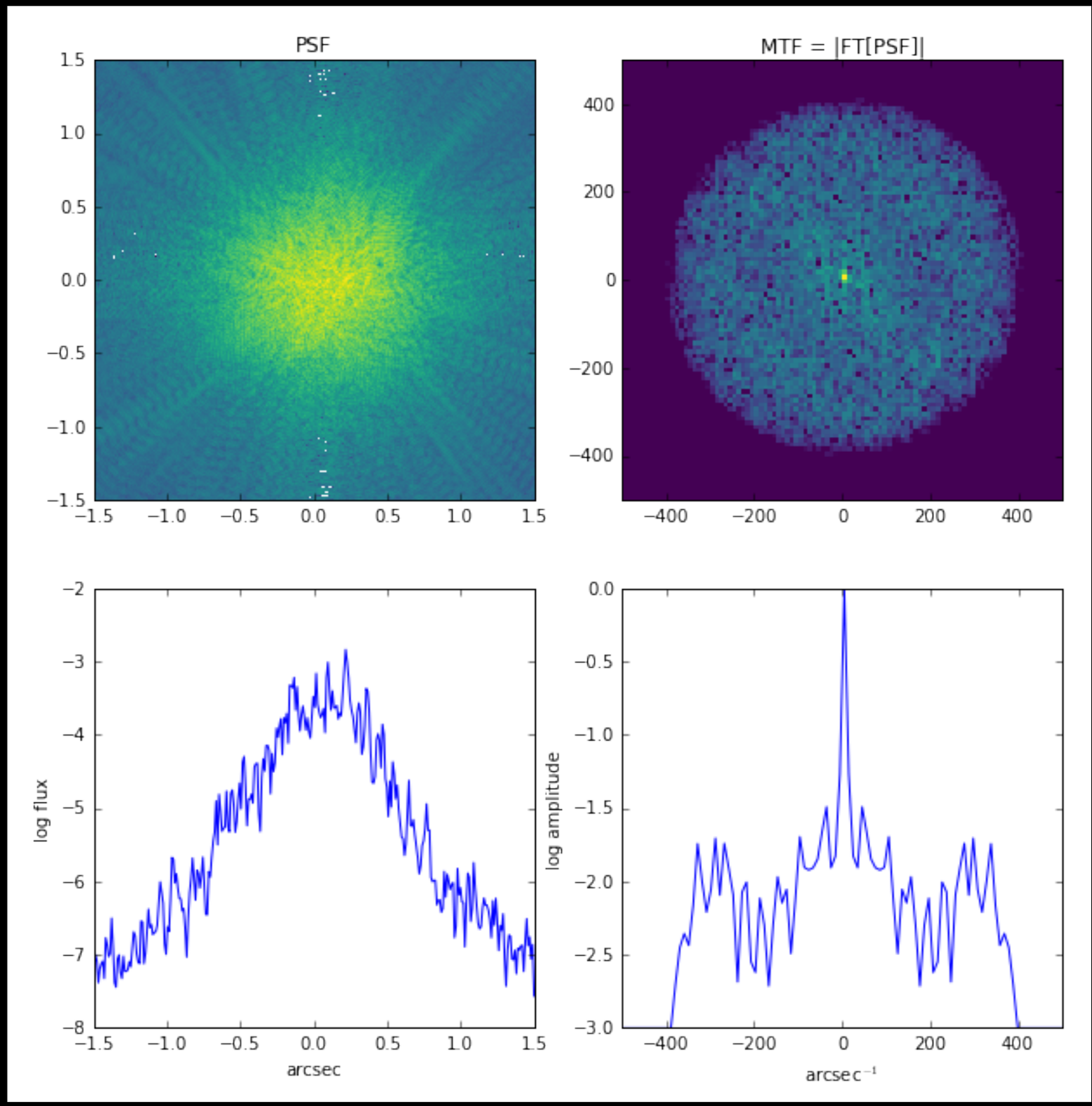


Instantaneous Atmospheric



requires large maxK
wide PSF
need small stepK

GalSim assumed
maxK/stepK ~800
could have
used ~200 (!)

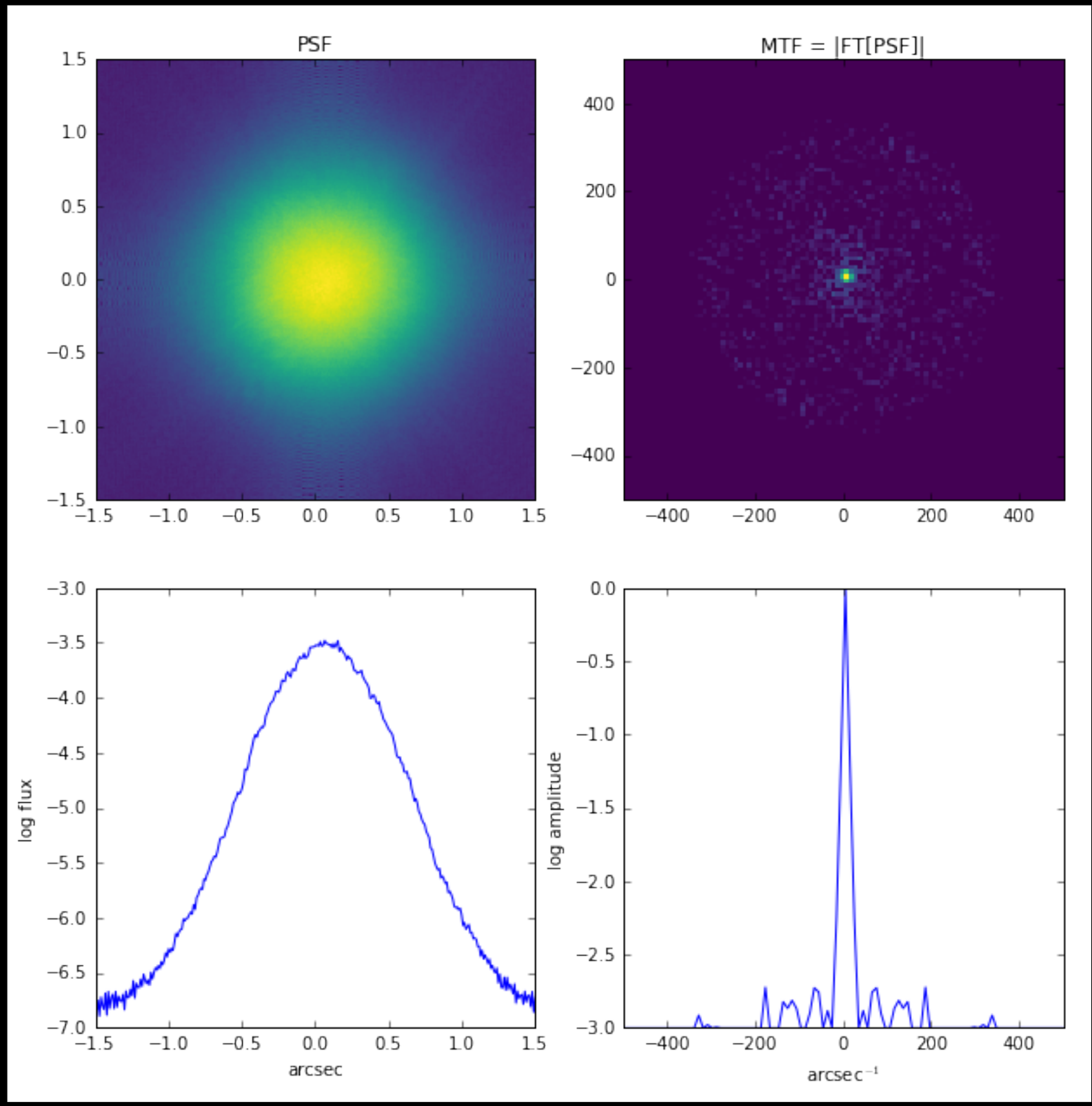


3 seconds cumulative atmospheric

No single equivalent pupil

OTF = autocorrelation of pupil (MTF = |OTF|)

even though cumulative OTF is compact, need whole aperture to compute needed correlations.

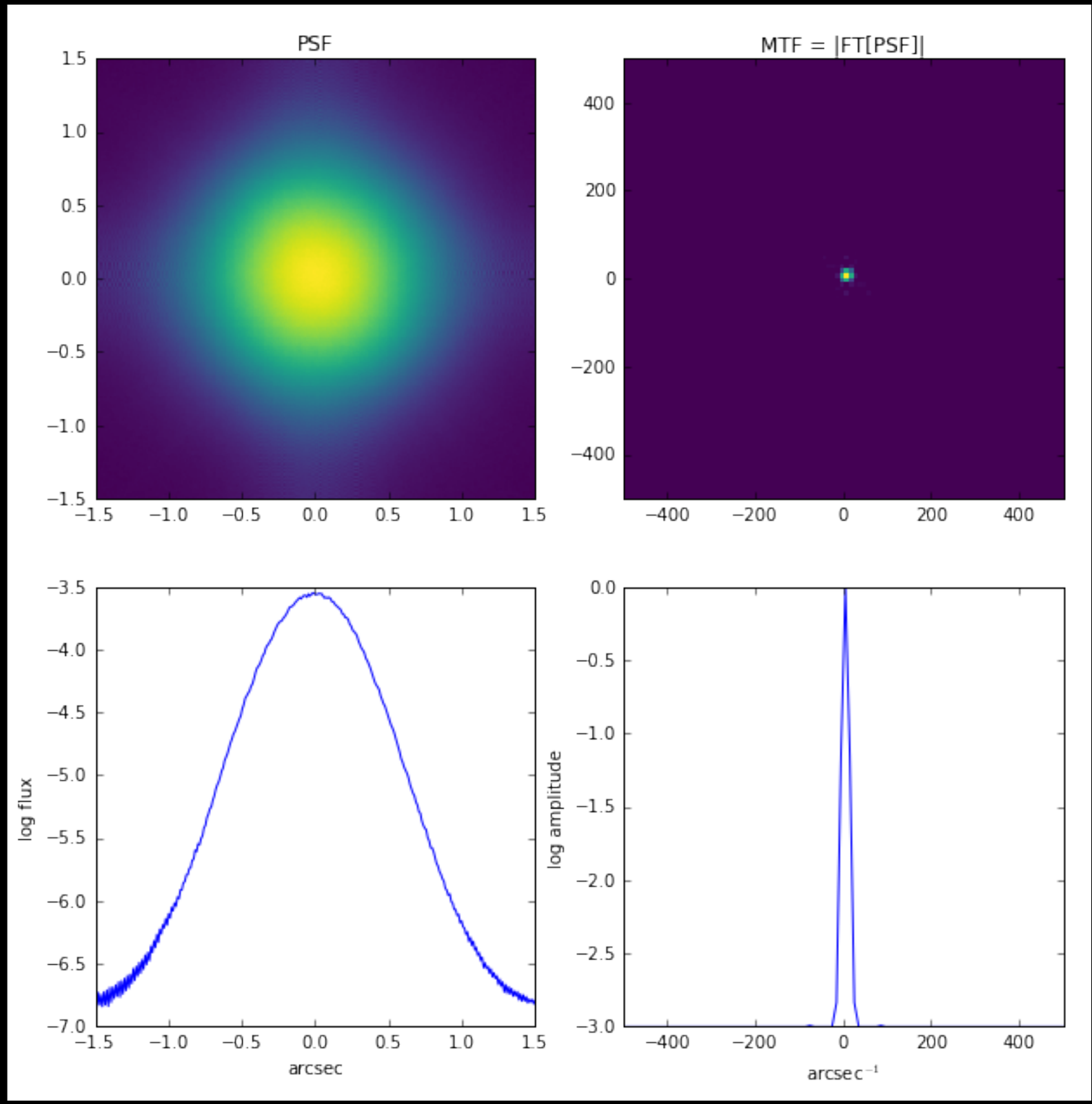


30 seconds cumulative atmospheric

Now similar to Kolmogorov

maxK/stepK ~ 25

Large k modes disappear. No need for oversampling.



Sampling

- Expected atmospheric PSFs would require the worst attributes of Airy (large maxK) and Kolmogorov (small stepK).

- Simulated atmospheric PSF appears to be more compact than Kolmogorov, so possibly don’t need quite as small of stepK as is currently being employed.

- Needs more study.

More Timing

```
[josh@Trogdor ~/sandbox/speed_accuracy]$ python speed.py --oversampling 1.0 --pad_factor 1.0 --max_size 1e6 --exptime 30.0
Making atmosphere
Done making atmosphere
(1536, 1536)
|>------------------------------------------------------------------|  19 /6.0k (  0.32%) ETA 51m39s
```

Restrict output size to 5 arcsec

```
[josh@Trogdor ~/sandbox/speed_accuracy]$ python speed.py --oversampling 1.0 --pad_factor 1.0 --max_size 5 --exptime 30.0
Making atmosphere
Done making atmosphere
(768, 768)
|=>----------------------------------------------------------------|  85 /6.0k (  1.42%) ETA 10m42s
```

Restrict output size to 2 arcsec

```
[josh@Trogdor ~/sandbox/speed_accuracy]$ python speed.py --oversampling 1.0 --pad_factor 1.0 --max_size 2 --exptime 30.0
Making atmosphere
Done making atmosphere
(384, 384)
|===>--------------------------------------------------------------| 263 /6.0k (  4.38%) ETA  1m56s
```