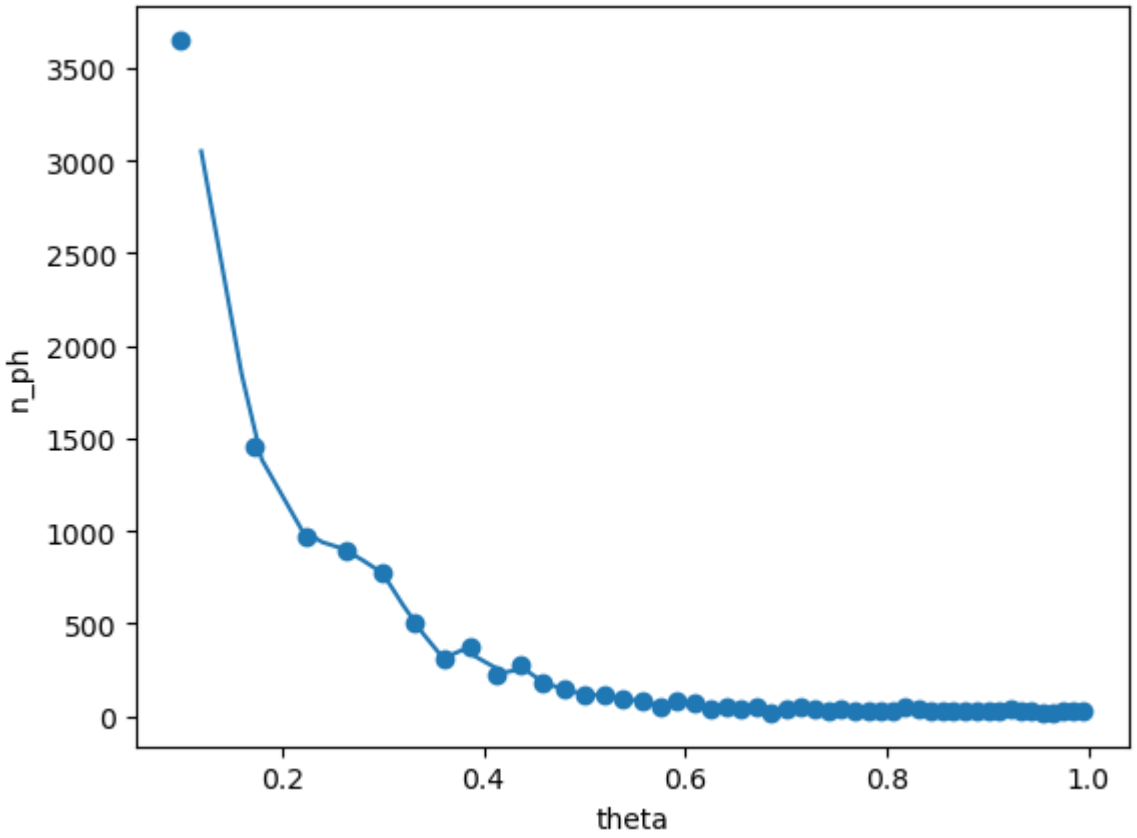
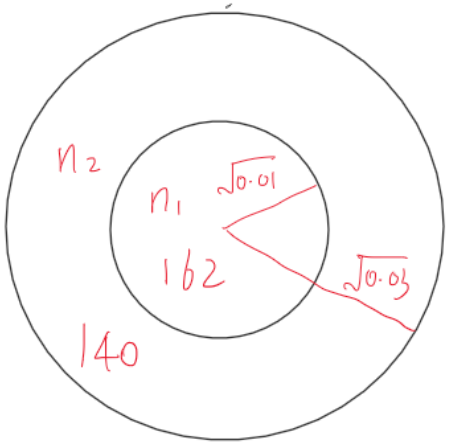
PSF强度-角度分布

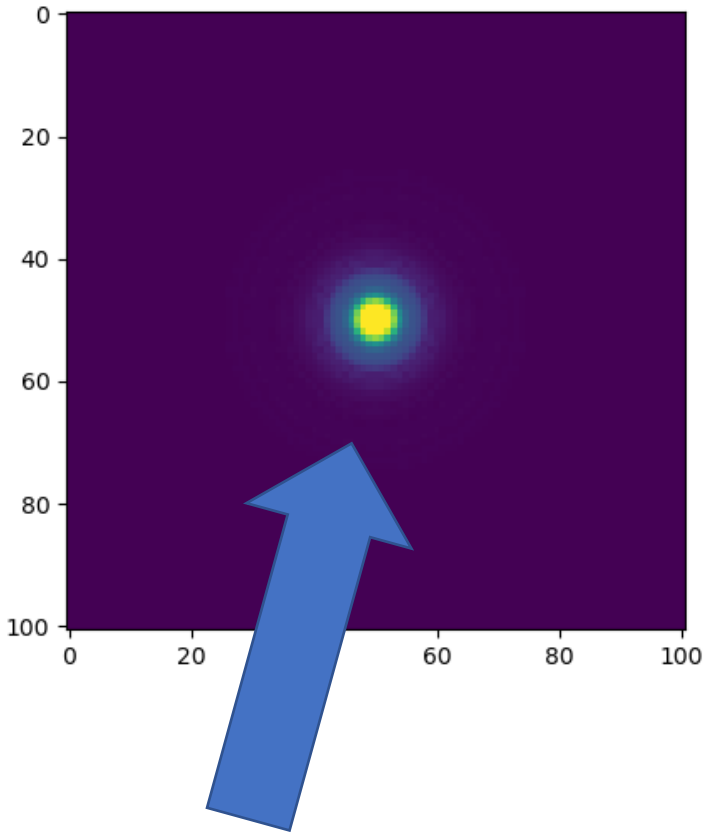



```
theta_2=0.010000    MC_ns: 155.280755   MC: 162.462573   Data: 146.999994
theta_2=0.030000    MC_ns: 132.360034   MC: 140.723670   Data: 147.000004
theta_2=0.050000    MC_ns: 97.112971    MC: 105.658426   Data: 134.999999
theta_2=0.070000    MC_ns: 97.297605    MC: 105.843060   Data: 109.999999
theta_2=0.090000    MC_ns: 91.434258    MC: 99.252439    Data: 101.999998
theta_2=0.110000    MC_ns: 62.771998    MC: 71.044725    Data: 72.999999
theta_2=0.130000    MC_ns: 38.169818    MC: 45.078909    Data: 58.999999
theta_2=0.150000    MC_ns: 52.481231    MC: 59.481231    Data: 67.000000
theta_2=0.170000    MC_ns: 32.371992    MC: 41.917447    Data: 52.999999
theta_2=0.190000    MC_ns: 43.454949    MC: 52.818585    Data: 46.000000
theta_2=0.210000    MC_ns: 26.207660    MC: 33.298569    Data: 50.000001
theta_2=0.230000    MC_ns: 23.150990    MC: 31.423717    Data: 42.999999
theta_2=0.250000    MC_ns: 20.784927    MC: 27.148564    Data: 32.000000
```



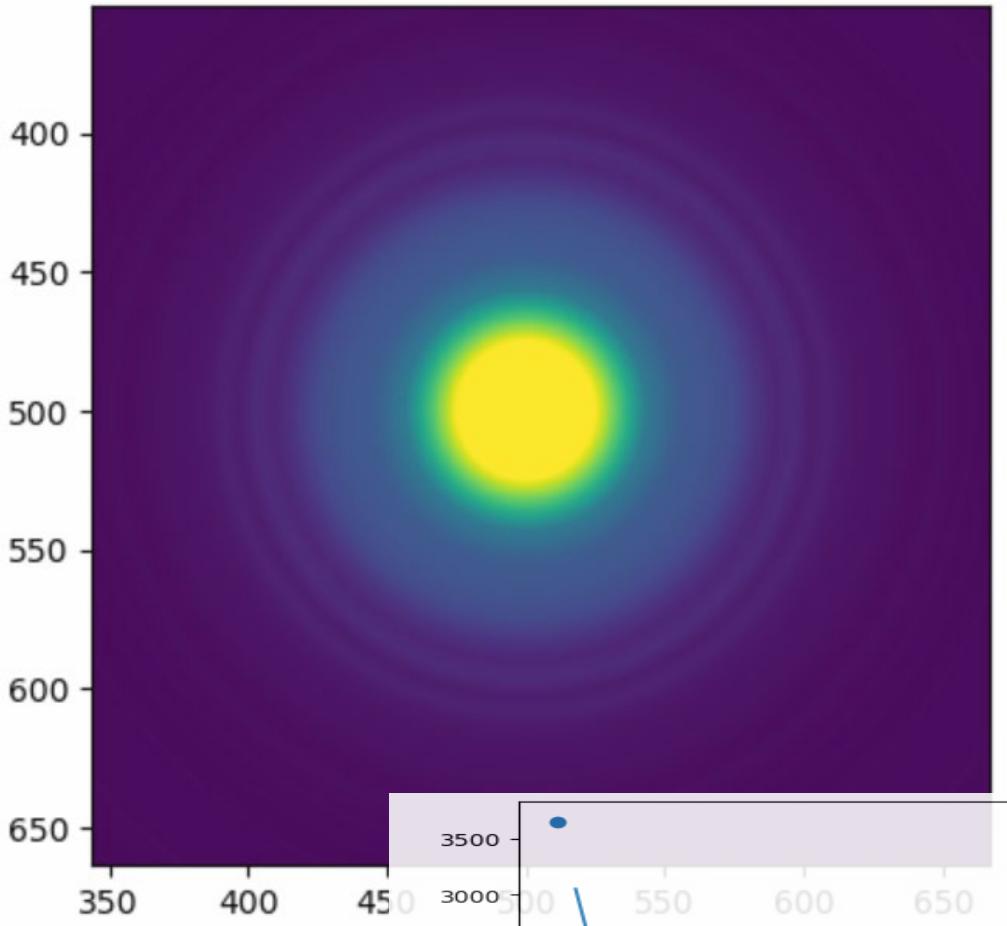
$$\Rightarrow n_1 = \frac{162}{\pi \cdot 0.01}$$

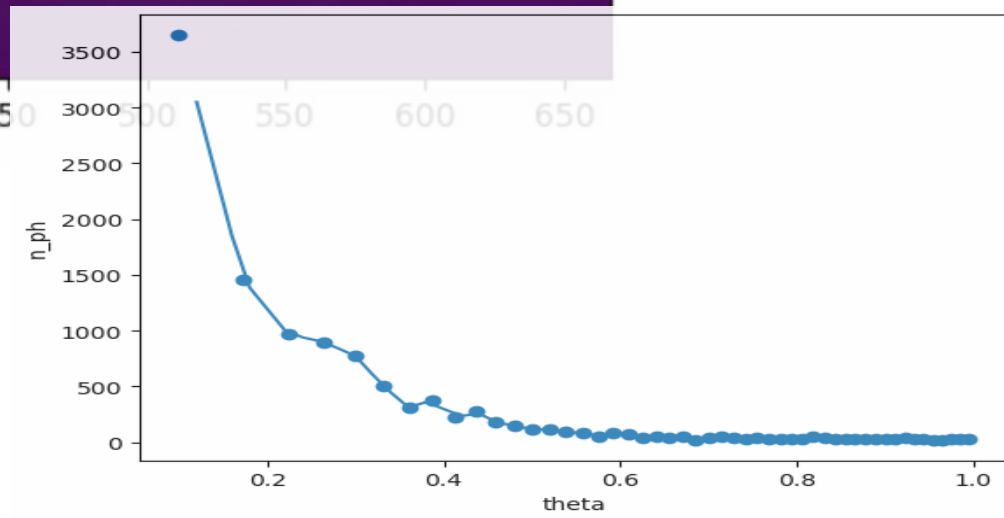
$$n_2 = \frac{140}{\pi \cdot (0.03 - 0.01)}$$



点源模糊后

X/Y:-2~2 deg





如何使用

1. 定义坐标轴 : ra = dec = np.linspace(-10,10,201)

2. 用meshgrid生成格点 xx,yy = np.meshgrid(ra,dec)

3. 使用PSF.py中的interp函数插值，生成一个PSF卷积函数：
   PSFConv = PSF.interp(idx),这里idx对应不同能量的点扩散函数，见注释

如何使用

4. 随便定义一个天区

sky = np.zeros_like(xx); sky[a,b] += 1

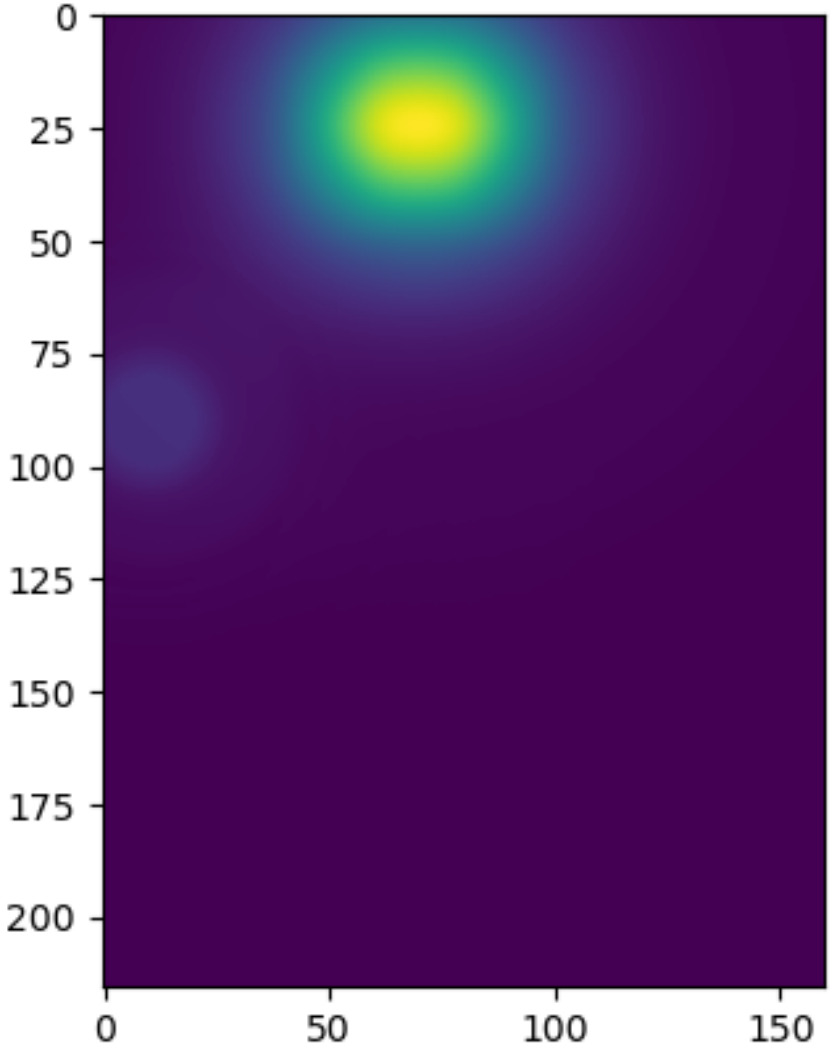
PSFConv(sky)

返回值为卷积后的天区。

得到卷积后总信号等于卷积前总信号

```
print('total signal before conv: %.2f'%z.sum())
print('total signal after  conv: %.2f'%blurz.sum())
```

```
total signal before conv: 1.00
total signal after  conv: 1.00
```




```
total signal before conv: 130000.00000000
total signal after  conv: 109325.03358931
```

由于边界效应，PSF造成的信号可能弥散到图外导致总信号减少。
实际操作可取大些的天区卷积，然后截取中间部分训练