ModelZoo

June 15, 2023

1 Model Zoo

In this notebook you will see every kind of model in AutoPhot. Printed in each cell will also be the list of parameters which the model looks for while fitting. Many models have unique capabilities and features, this will be introduced here, though fully taking advantage of them will be dependent on your science case.

For a family tree of all the AutoPhot models see this link

Note, we will not be covering `Group_Model` here as that requires a dedicated discussion. See the dedicated notebook for that.

[1]:
```
import autophot as ap
import numpy as np
import torch
import matplotlib.pyplot as plt
%matplotlib inline
basic_target = ap.image.Target_Image(np.zeros((100,100)), pixelscale = 1,␣
 ↪zeropoint = 20)
```

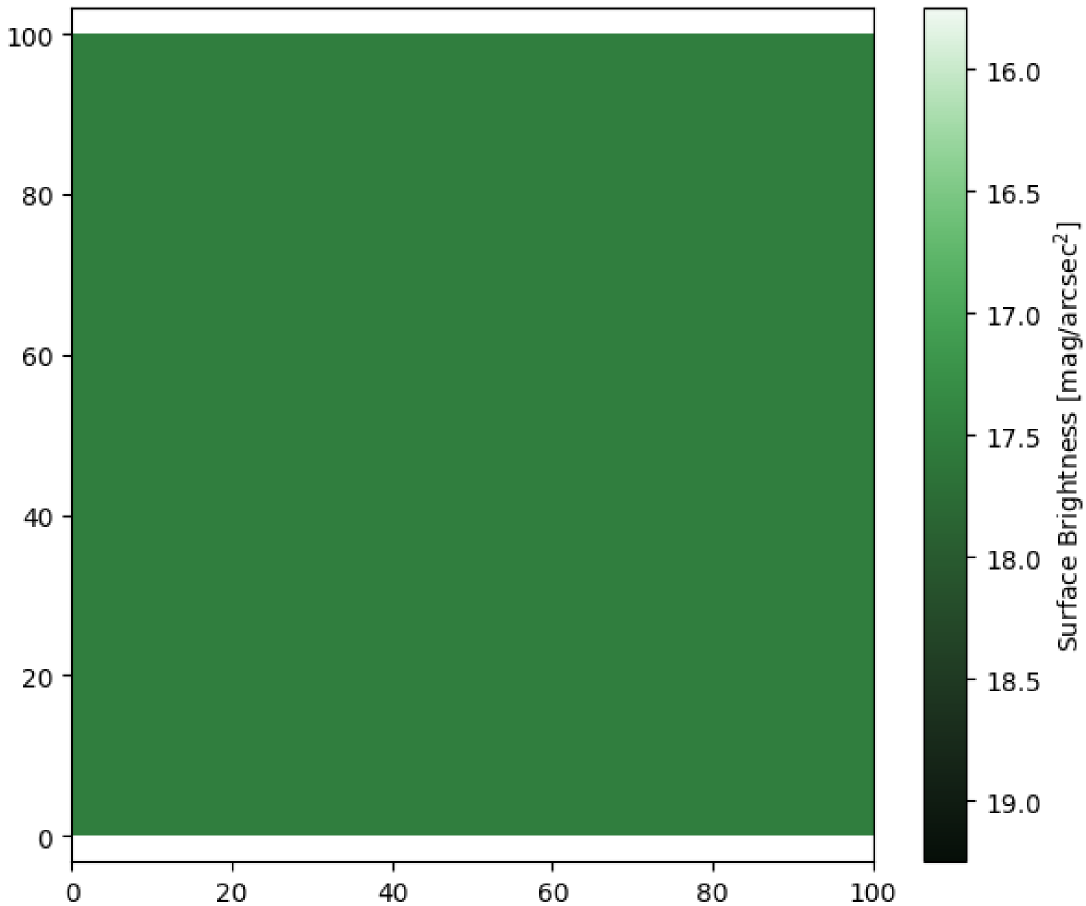
1.1 Sky Models

1.1.1 Flat Sky Model

[2]:
```
M = ap.models.AutoPhot_Model(name = "flat sky", model_type = "flat sky model",␣
 ↪parameters = {"center":[50,50], "sky": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(figsize = (7,6))
ap.plots.model_image(fig, ax, M)
plt.show()
```

```
('sky',)
('log10(flux/arcsec^2)',)
```

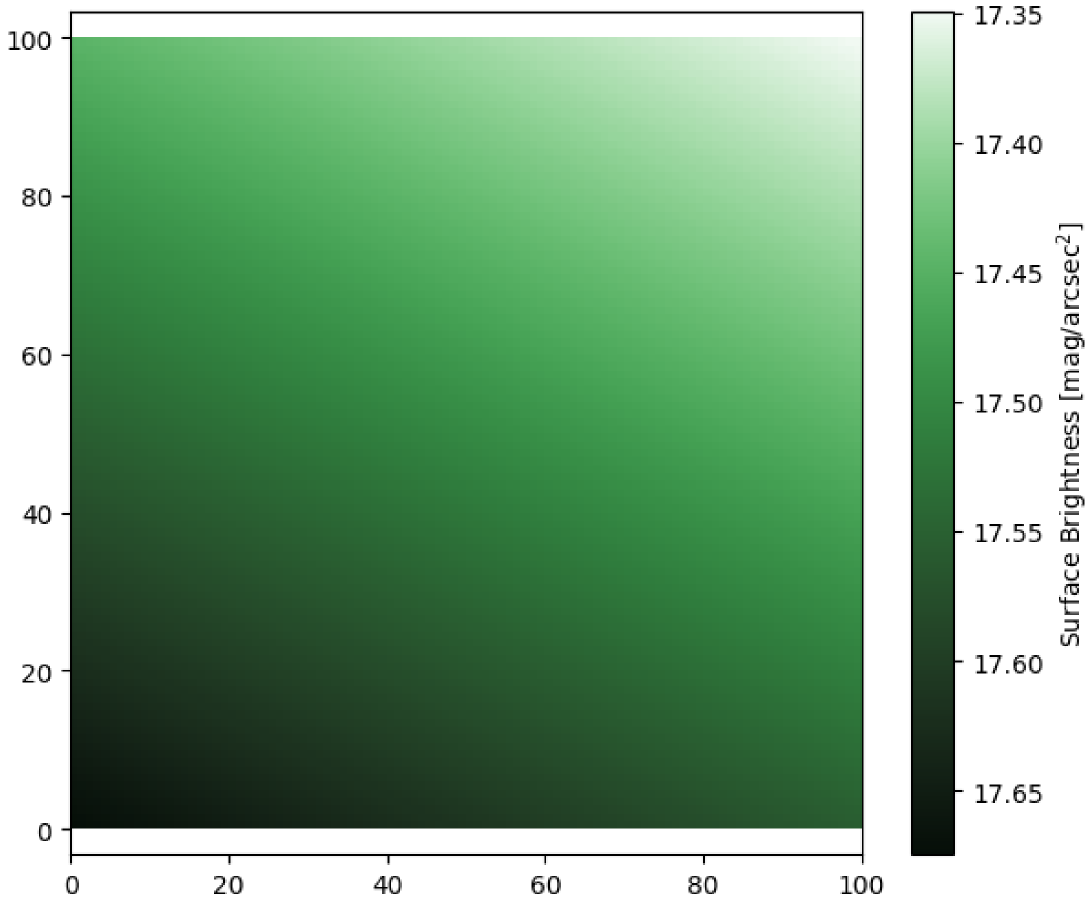


1.1.2 Plane Sky Model

[3]:
```
M = ap.models.AutoPhot_Model(name = "plane sky", model_type = "plane sky␣
↪model", parameters = {"center":[50,50], "sky": 10, "delta": [1e-2,2e-2]},␣
↪target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(figsize = (7,6))
ap.plots.model_image(fig, ax, M)
plt.show()
```

```
('sky', 'delta')
('flux/arcsec^2', 'sky/arcsec')
```



1.2 Star Models

1.2.1 PSF Star

Note that in this model you can define an arbitrary pixel map, for the sake of demonstration we build an Airy disk but you can assign whatever you like to the pixels.

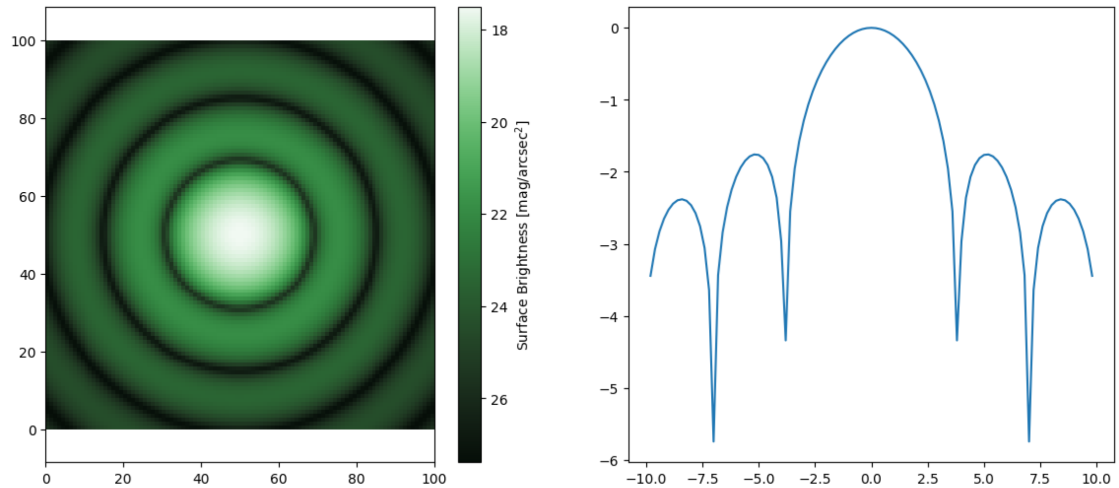
[4]:
```
from scipy.special import jv
xx, yy = np.meshgrid(np.linspace(-50,50,101), np.linspace(-50,50,101))
x = np.sqrt(xx**2 + yy**2)/5 +1e-6
PSF = (2*jv(1, x)/x)**2 + 1e-4 # the PSF can be any image, here we construct an␣
 ↪airy disk
target = ap.image.Target_Image(data = np.zeros((100,100)), pixelscale = 1,␣
 ↪zeropoint = 20, psf = PSF) # the target image holds the PSF for itself

M = ap.models.AutoPhot_Model(name = "psf star", model_type = "psf star model",␣
 ↪target = target, parameters = {"center": [50,50], "flux": 1})
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
```

```
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
x = np.linspace(-49,49,99)/5 + 1e-6
ax[1].plot(x, np.log10((2*jv(1, x)/x)**2))
plt.show()
```

```
('center', 'flux')
('arcsec', 'log10(flux/arcsec^2)')
```



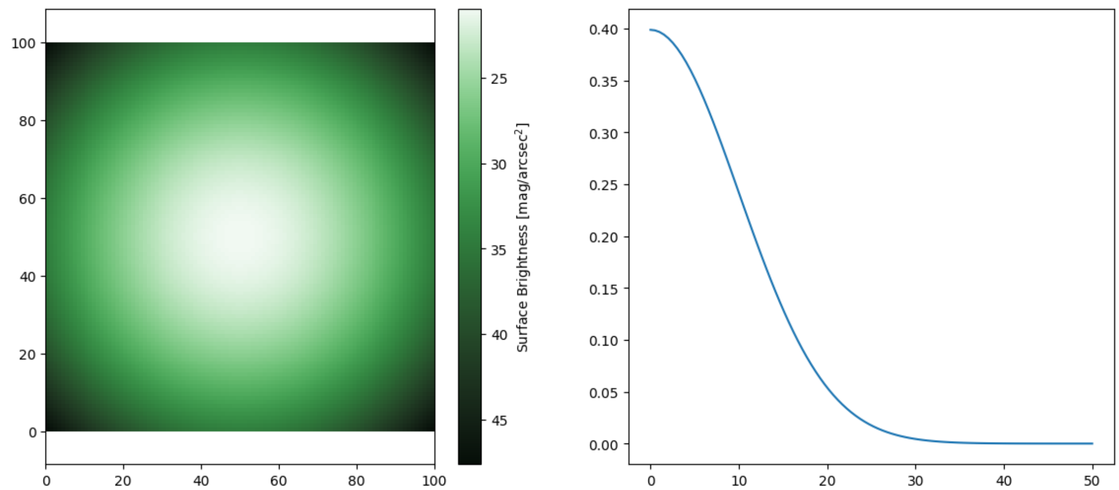
1.2.2 Gaussian Star

Never a great PSF model, but the Gaussian is simple. This makes it a good starting choice to get results before stepping up the complexity level.

[5]:
```
M = ap.models.AutoPhot_Model(name = "gaussian star", model_type = "gaussian␣
↪star model", parameters = {"center":[50,50], "sigma": 10, "flux": 1}, target␣
↪= basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ax[1].plot(np.linspace(0,50,100), M.radial_model(torch.linspace(0,50,100)).
↪detach().cpu().numpy())
plt.show()
```

```
('center', 'sigma', 'flux')
('arcsec', 'arcsec', 'log10(flux)')
```

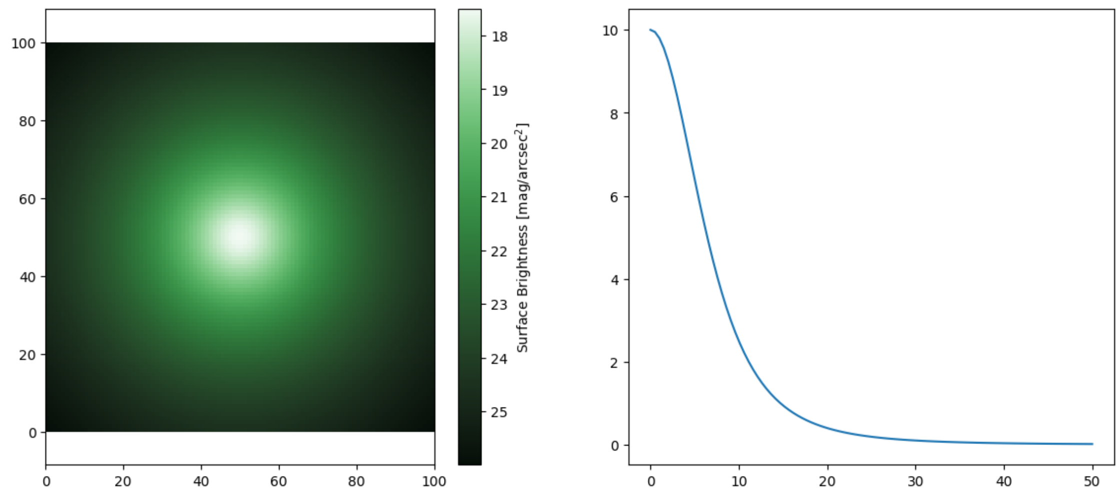


1.2.3 Moffat Star

[6]:
```
M = ap.models.AutoPhot_Model(name = "moffat star", model_type = "moffat star␣
↪model", parameters = {"center":[50,50], "n": 2., "Rd": 10., "I0": 1.},␣
↪target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ax[1].plot(np.linspace(0,50,100), M.radial_model(torch.linspace(0,50,100)).
↪detach().cpu().numpy())
plt.show()
```

```
('center', 'n', 'Rd', 'I0')
('arcsec', 'none', 'arcsec', 'log10(flux/arcsec^2)')
```



2 Galaxy Models

2.0.1 Spline Galaxy Model

This model has a radial surface brightness profile which can take on any function (that can be represented as a spline). This is somewhat like elliptical isophote fitting, though it is more precise in its definition of the SB model.

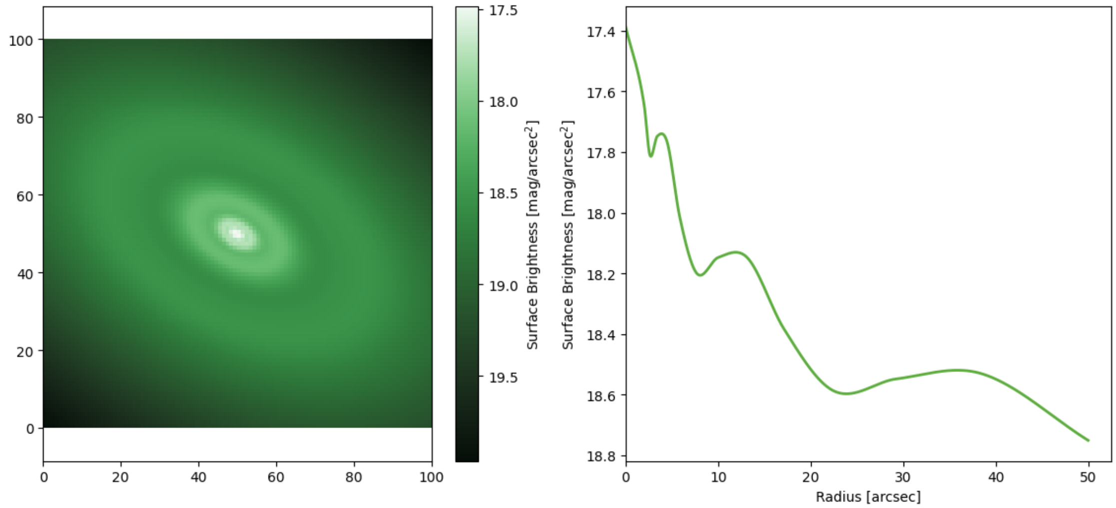
[7]:
```
# Here we make an arbitrary spline profile out of a sine wave and a line
x = np.linspace(0,10,14)
spline_profile = np.sin(x*2+2)/20 + 1 - x/20
# Here we write down some corresponding radii for the points in the␣
↪non-parametric profile. AutoPhot will make
# radii to match an input profile, but it is generally better to manually␣
↪provide values so you have some control
# over their placement. Just note that it is assumed the first point will be at␣
↪R = 0.
NP_prof = [0] + list(np.logspace(np.log10(2),np.log10(50),13))

M = ap.models.AutoPhot_Model(name = "spline galaxy", model_type = "spline␣
↪galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/
↪180, "I(R)": {"value": spline_profile, "prof": NP_prof}}, target =␣
↪basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
```

```
plt.show()
```

```
('center', 'q', 'PA', 'I(R)')
('arcsec', 'b/a', 'radians', 'log10(flux/arcsec^2)')
```

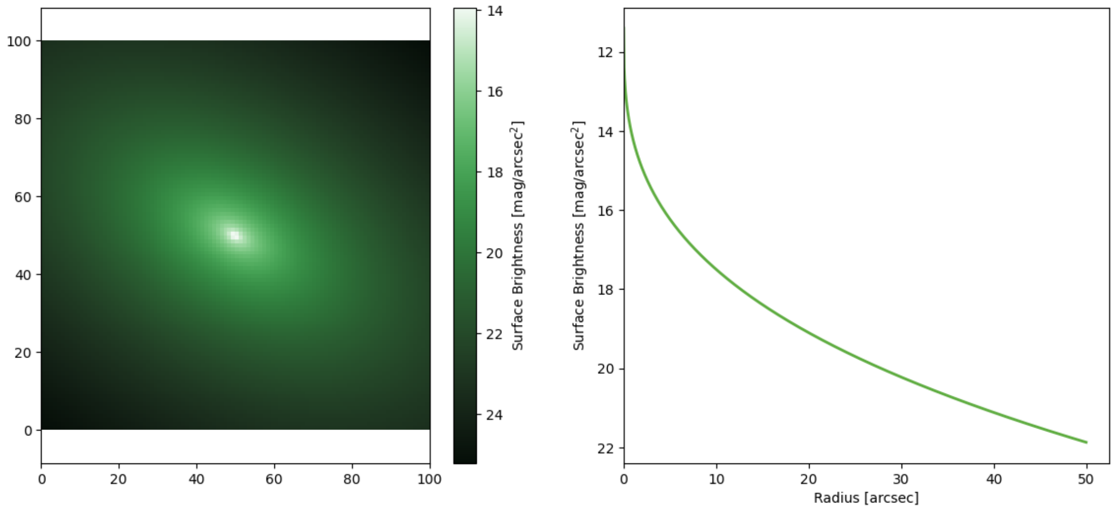


2.0.2 Sersic Galaxy Model

[8]:
```
M = ap.models.AutoPhot_Model(name = "sersic galaxy", model_type = "sersic␣
↪galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/
↪180, "n": 3, "Re": 10, "Ie": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'n', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'none', 'arcsec', 'log10(flux/arcsec^2)')
```

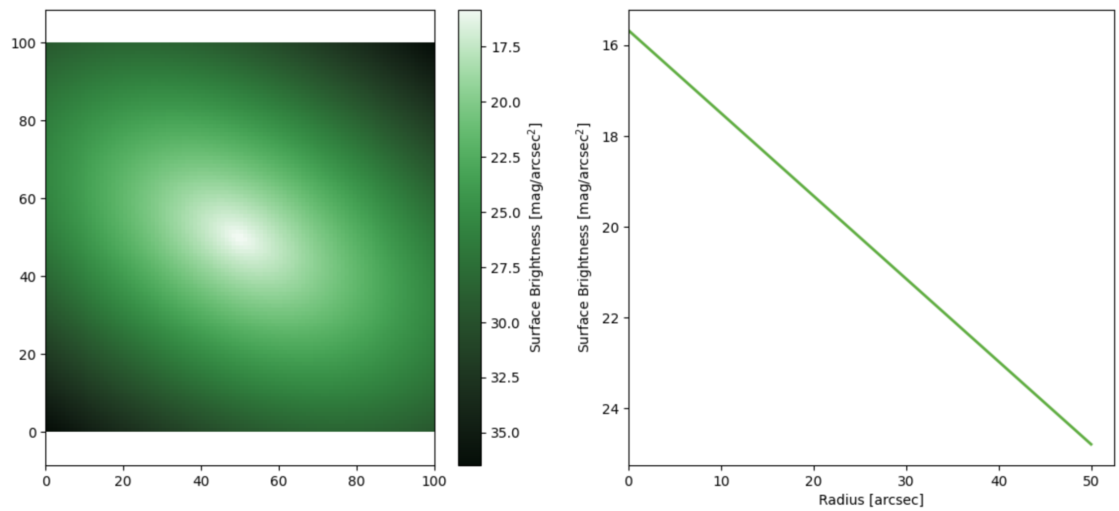


2.0.3 Exponential Galaxy Model

[9]:
```
M = ap.models.AutoPhot_Model(name = "exponential galaxy", model_type =␣
↪"exponential galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA":␣
↪60*np.pi/180, "Re": 10, "Ie": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'arcsec', 'log10(flux/arcsec^2)')
```

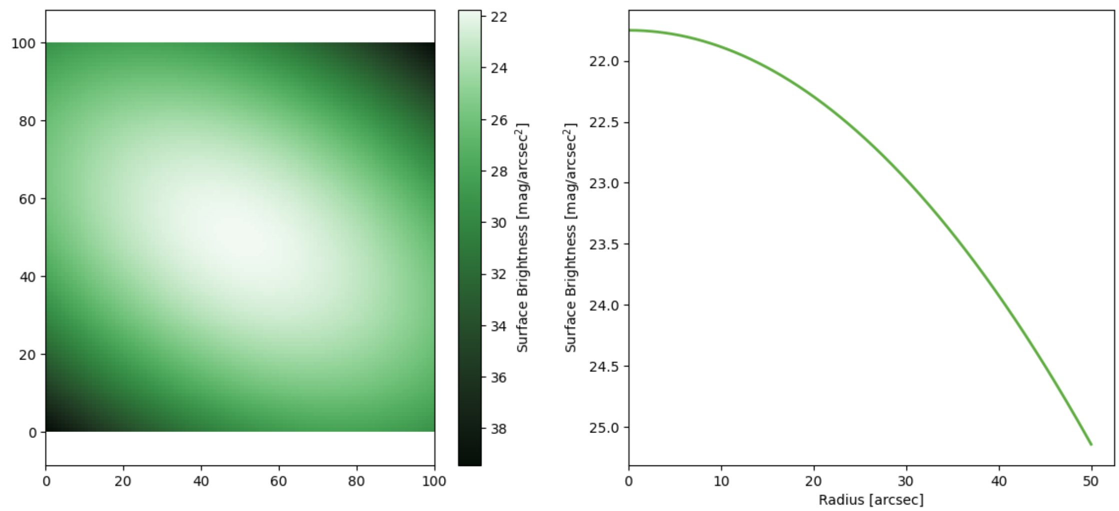


2.0.4 Gaussian Galaxy Model

[10]:
```
M = ap.models.AutoPhot_Model(name = "Gaussian", model_type = "gaussian galaxy␣
↪model", parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/180,␣
↪"sigma": 20, "flux": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'sigma', 'flux')
('arcsec', 'b/a', 'radians', 'arcsec', 'log10(flux)')
```

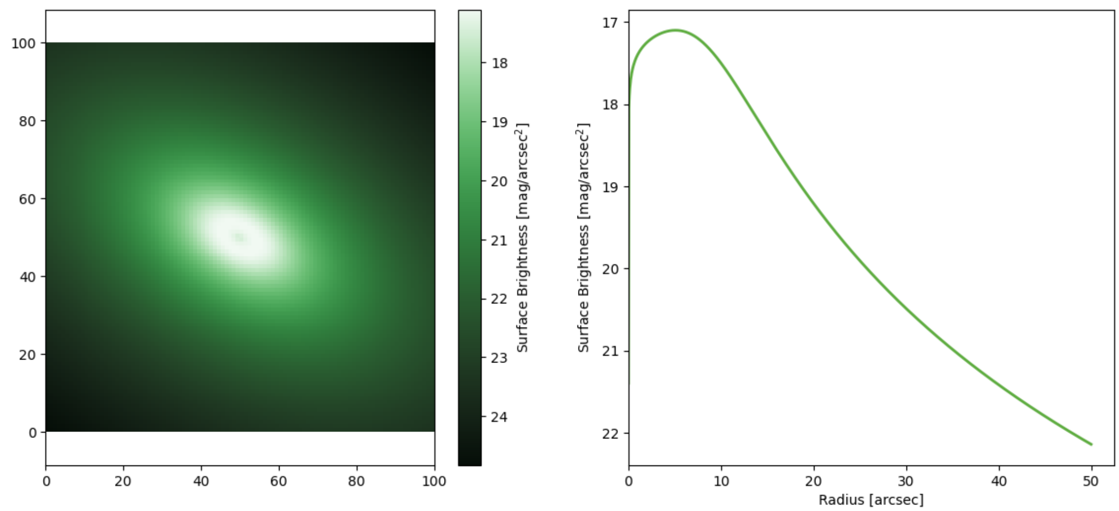


2.0.5 Nuker Galaxy Model

[11]:
```
M = ap.models.AutoPhot_Model(name = "Nuker", model_type = "nuker galaxy model",␣
  ↪parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/180, "Rb": 10.,␣
  ↪"Ib": 1., "alpha": 4., "beta": 3., "gamma": -0.2}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'Rb', 'Ib', 'alpha', 'beta', 'gamma')
('arcsec', 'b/a', 'radians', 'arcsec', 'log10(flux/arcsec^2)', 'none', 'none',
'none')
```



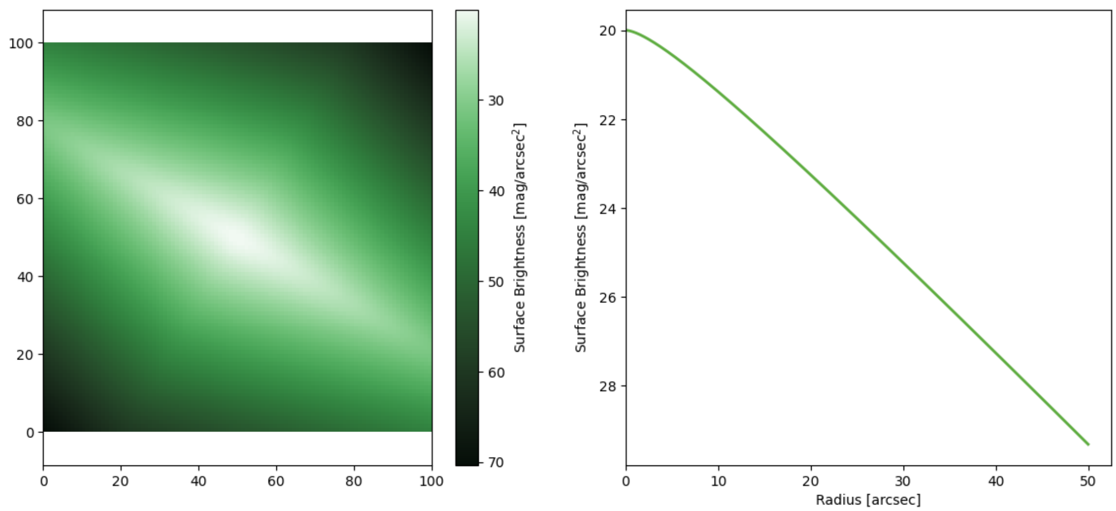
2.1 Edge on model

Currently there is only one dedicared edge on model, the self gravitating isothermal disk from van der Kruit & Searle 1981. If you know of another common edge on model, feel free to let us know and we can add it in!

[12]:
```
M = ap.models.AutoPhot_Model(name = "edgeon", model_type = "isothermal sech2␣
↪edgeon model", parameters = {"center": [50,50], "PA": 60*np.pi/180, "I0": 0.
↪, "hs": 3., "rs": 5.}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'PA', 'I0', 'hs', 'rs')
('arcsec', 'rad', 'log10(flux/arcsec^2)', 'arcsec', 'arcsec')
```



2.2 Super Ellipse Models

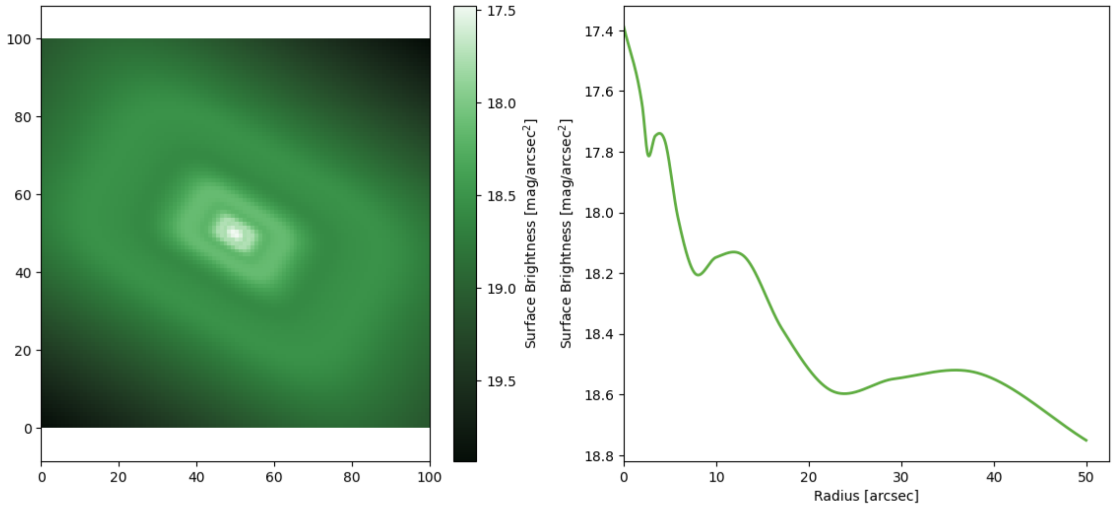
A super ellipse is a regular ellipse, except the radius metric changes from R = sqrt(x^2 + y^2) to the more general: R = (x^C + y$^{C)}$1/C. The parameter C = 2 for a regular ellipse, for 0<C<2 the shape becomes more "disky" and for C > 2 the shape becomes more "boxy." In AutoPhot we use the parameter C0 = C-2 for simplicity.

2.2.1 Spline SuperEllipse

[13]:
```
M = ap.models.AutoPhot_Model(name = "spline superellipse", model_type = "spline␣
↪superellipse galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA":␣
↪60*np.pi/180, "C0": 2, "I(R)": {"value": spline_profile, "prof": NP_prof}},␣
↪target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'C0', 'I(R)')
('arcsec', 'b/a', 'radians', 'C-2', 'log10(flux/arcsec^2)')
```

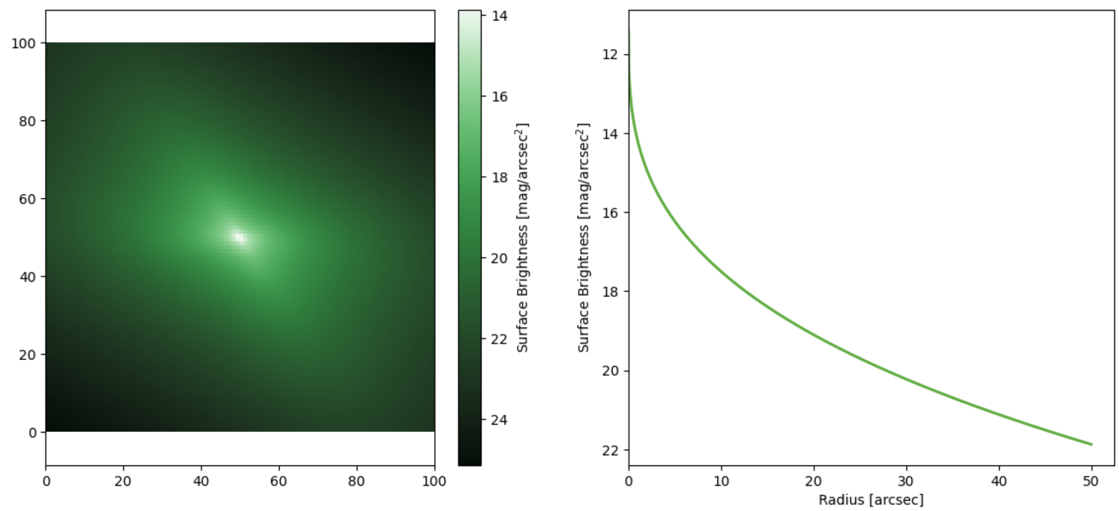


2.2.2 Sersic SuperEllipse

[14]:
```
M = ap.models.AutoPhot_Model(name = "sersic superellipse", model_type = "sersic␣
↪superellipse galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA":␣
↪60*np.pi/180, "C0": 2, "n": 3, "Re": 10, "Ie": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'C0', 'n', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'C-2', 'none', 'arcsec', 'log10(flux/arcsec^2)')
```

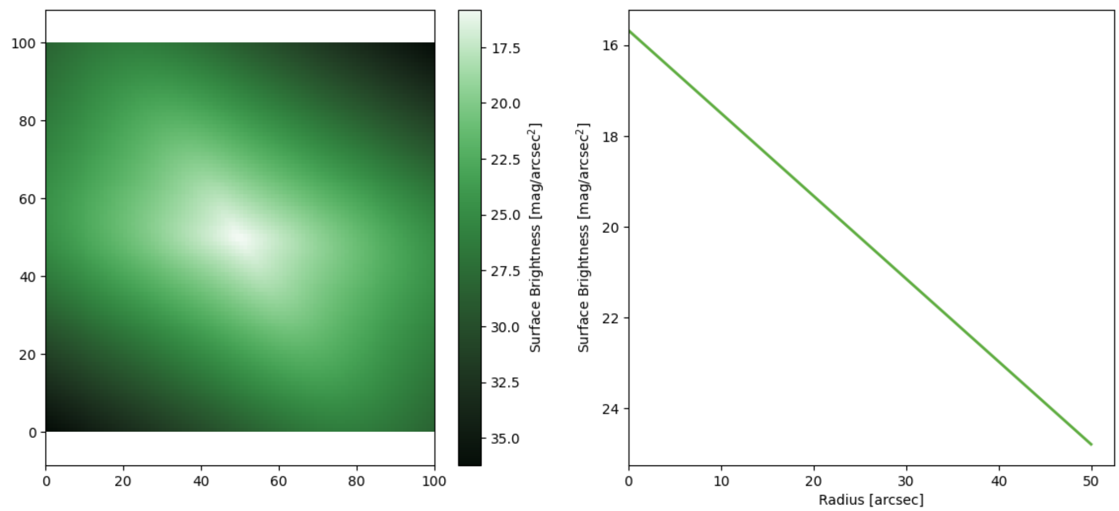


2.2.3 Exponential SuperEllipse

[15]:
```
M = ap.models.AutoPhot_Model(name = "exponential superellipse", model_type =␣
↪"exponential superellipse galaxy model", parameters = {"center": [50,50],␣
↪"q": 0.6, "PA": 60*np.pi/180, "C0": 2, "Re": 10, "Ie": 1}, target =␣
↪basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'C0', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'C-2', 'arcsec', 'log10(flux/arcsec^2)')
```

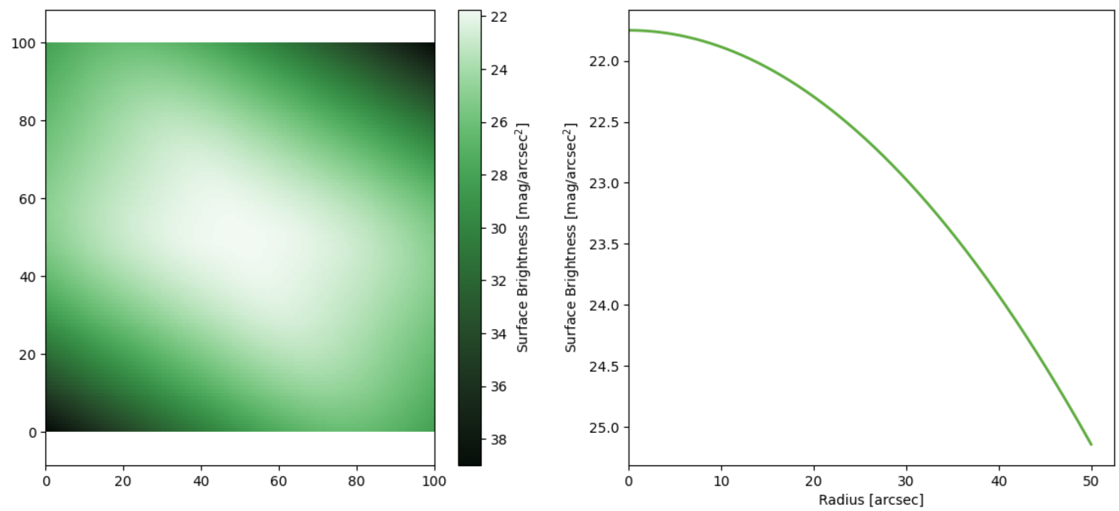


2.2.4 Gaussian SuperEllipse

[16]:
```
M = ap.models.AutoPhot_Model(name = "gaussian superellipse", model_type =␣
↪"gaussian superellipse galaxy model", parameters = {"center": [50,50], "q":␣
↪0.6, "PA": 60*np.pi/180, "C0": 2, "sigma": 20, "flux": 1}, target =␣
↪basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'C0', 'sigma', 'flux')
('arcsec', 'b/a', 'radians', 'C-2', 'arcsec', 'log10(flux)')
```

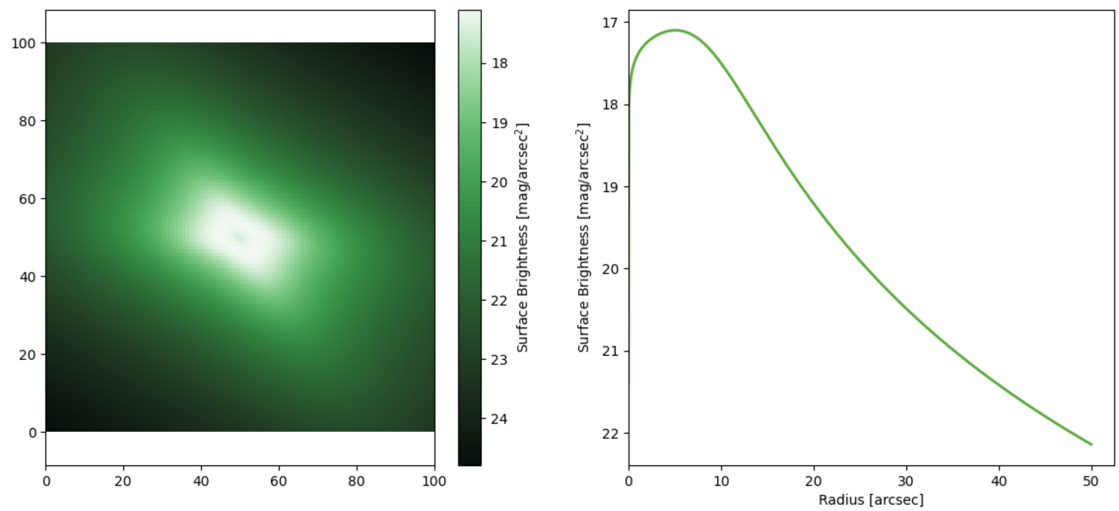


2.2.5 Nuker SuperEllipse

[17]:
```
M = ap.models.AutoPhot_Model(name = "nuker superellipse", model_type = "nuker␣
↪superellipse galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA":␣
↪60*np.pi/180, "C0": 2, "Rb": 10., "Ib": 1., "alpha": 4., "beta": 3., "gamma":
↪ -0.2}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'C0', 'Rb', 'Ib', 'alpha', 'beta', 'gamma')
('arcsec', 'b/a', 'radians', 'C-2', 'arcsec', 'log10(flux/arcsec^2)', 'none',
'none', 'none')
```



2.3 Fourier Ellipse Models

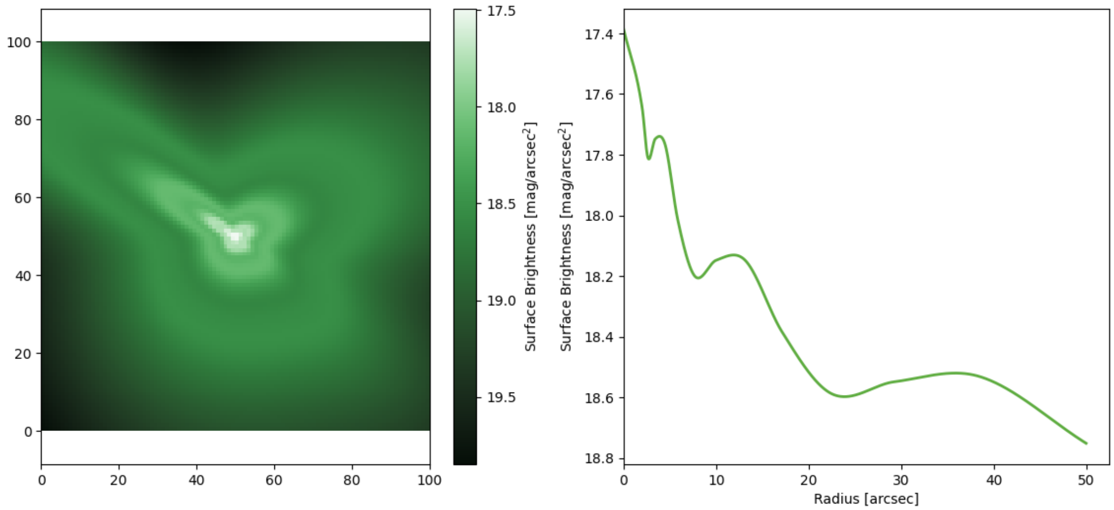
A Fourier ellipse is a scaling on the radius values as a function of theta. It takes the form: $R' = R * exp(\sum_m am * cos(m * theta + phim))$, where am and phim are the parameters which describe the Fourier perturbations. Using the "modes" argument as a tuple, users can select which Fourier modes are used. As a rough intuition: mode 1 acts like a shift of the model; mode 2 acts like ellipticity; mode 3 makes a lopsided model (triangular in the extreme); and mode 4 makes peanut/diamond perturbations.

2.3.1 Spline Fourier

[18]:
```
fourier_am = np.array([0.1, 0.3, -0.2])
fourier_phim = np.array([10*np.pi/180, 0, 40*np.pi/180])
M = ap.models.AutoPhot_Model(name = "spline fourier", model_type = "spline␣
↪fourier galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA":␣
↪60*np.pi/180, "am": fourier_am, "phim": fourier_phim, "I(R)": {"value":␣
↪spline_profile, "prof": NP_prof}}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'am', 'phim', 'I(R)')
('arcsec', 'b/a', 'radians', 'none', 'radians', 'log10(flux/arcsec^2)')
```

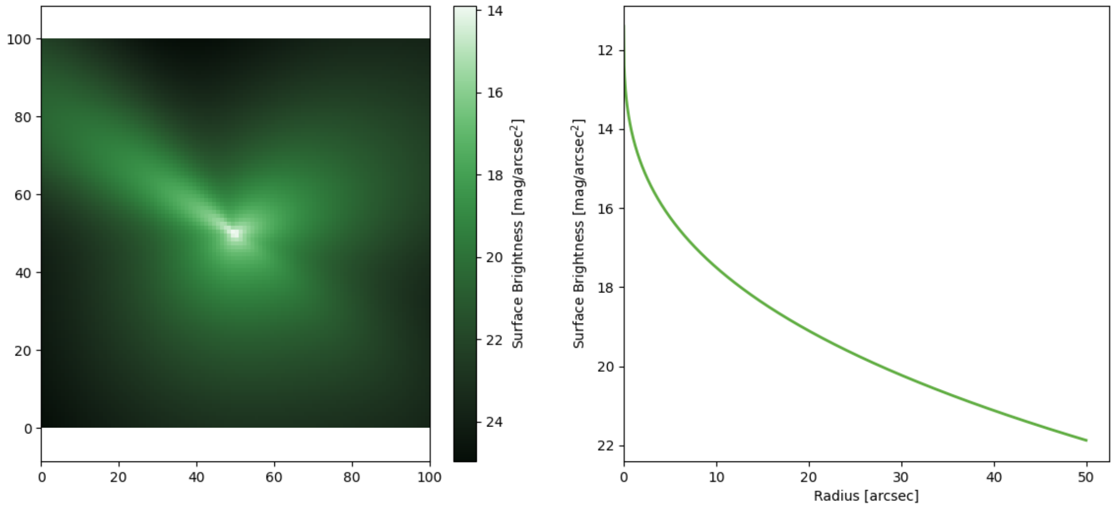


2.3.2 Sersic Fourier

[19]:
```
M = ap.models.AutoPhot_Model(name = "sersic fourier", model_type = "sersic␣
↪fourier galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA":␣
↪60*np.pi/180, "am": fourier_am, "phim": fourier_phim, "n": 3, "Re": 10, "Ie":
↪ 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'am', 'phim', 'n', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'none', 'radians', 'none', 'arcsec',
'log10(flux/arcsec^2)')
```

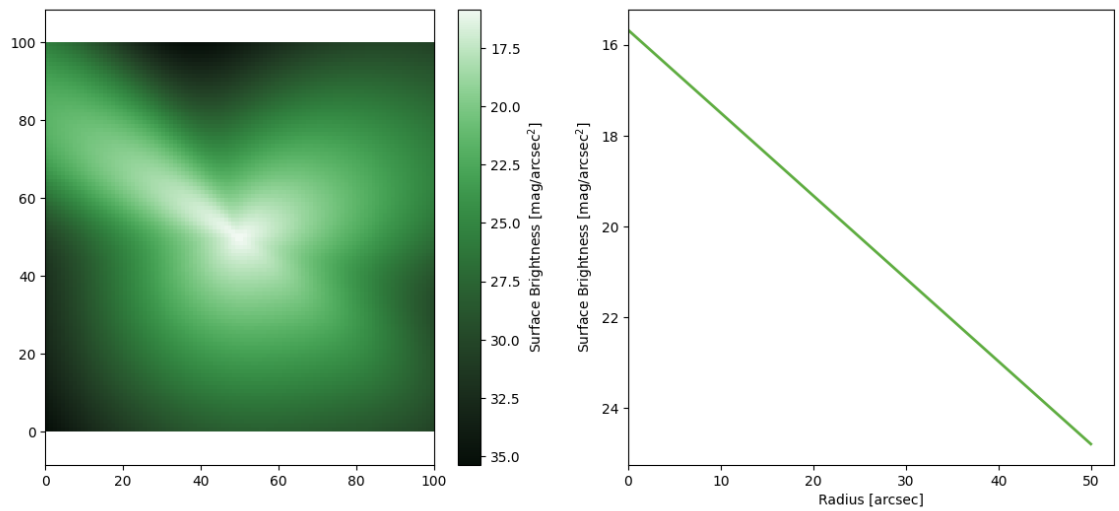


2.3.3 Exponential Fourier

[20]:
```
M = ap.models.AutoPhot_Model(name = "exponential fourier", model_type =␣
↪"exponential fourier galaxy model", parameters = {"center": [50,50], "q": 0.
↪6, "PA": 60*np.pi/180, "am": fourier_am, "phim": fourier_phim, "Re": 10,␣
↪"Ie": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'am', 'phim', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'none', 'radians', 'arcsec',
'log10(flux/arcsec^2)')
```

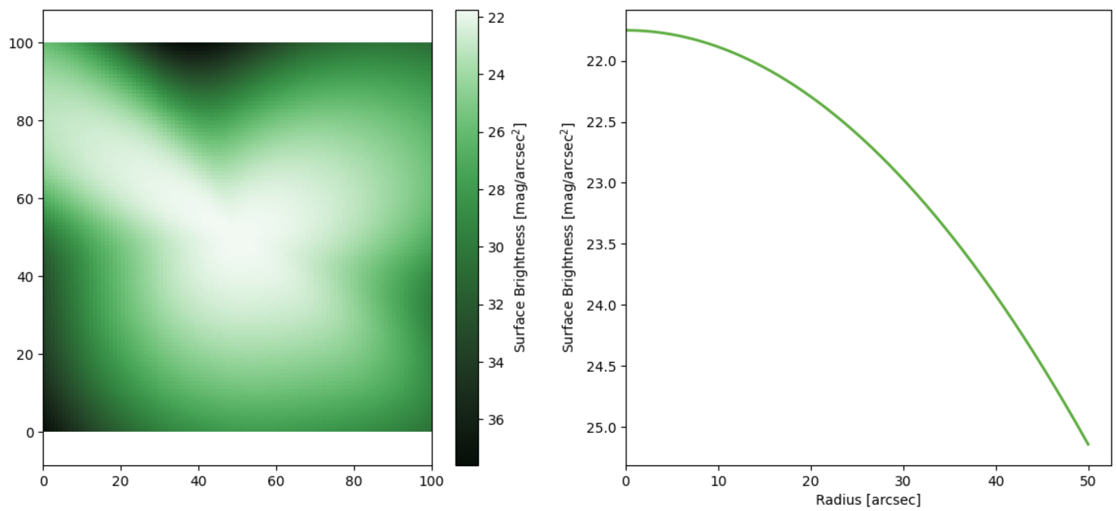


2.3.4 Gaussian Fourier

[21]:
```
M = ap.models.AutoPhot_Model(name = "gaussian fourier", model_type = "gaussian fourier galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/180, "am": fourier_am, "phim": fourier_phim, "sigma": 20, "flux": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'am', 'phim', 'sigma', 'flux')
('arcsec', 'b/a', 'radians', 'none', 'radians', 'arcsec', 'log10(flux)')
```

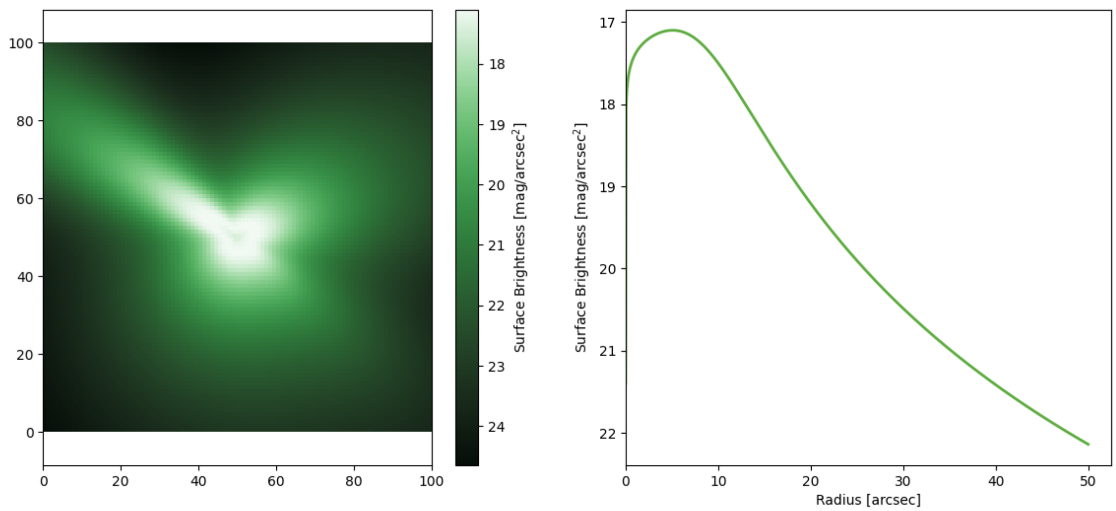


2.3.5 Nuker Fourier

[22]:
```
M = ap.models.AutoPhot_Model(name = "nuker fourier", model_type = "nuker fourier galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/180, "am": fourier_am, "phim": fourier_phim, "Rb": 10., "Ib": 1., "alpha": 4., "beta": 3., "gamma": -0.2}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'am', 'phim', 'Rb', 'Ib', 'alpha', 'beta', 'gamma')
('arcsec', 'b/a', 'radians', 'none', 'radians', 'arcsec',
'log10(flux/arcsec^2)', 'none', 'none', 'none')
```



2.4 Warp Model

A warp model performs a radially varying coordinate transform. Essentially instead of applying a rotation matrix **Rot** on all coordinates X,Y we instead construct a unique rotation matrix for each coordinate pair **Rot(R)** where $R = \sqrt{(}X^2 + Y^2)$. We also apply a radially dependent axis ratio **q(R)** to all the coordinates:

$R = \sqrt{(}X^2 + Y^2)$

$X, Y = Rotate(X, Y, PA(R))$

$Y = Y/q(R)$

The net effect is a radially varying PA and axis ratio which allows the model to represent spiral arms, bulges, or other features that change the apparent shape of a galaxy in a radially varying way.

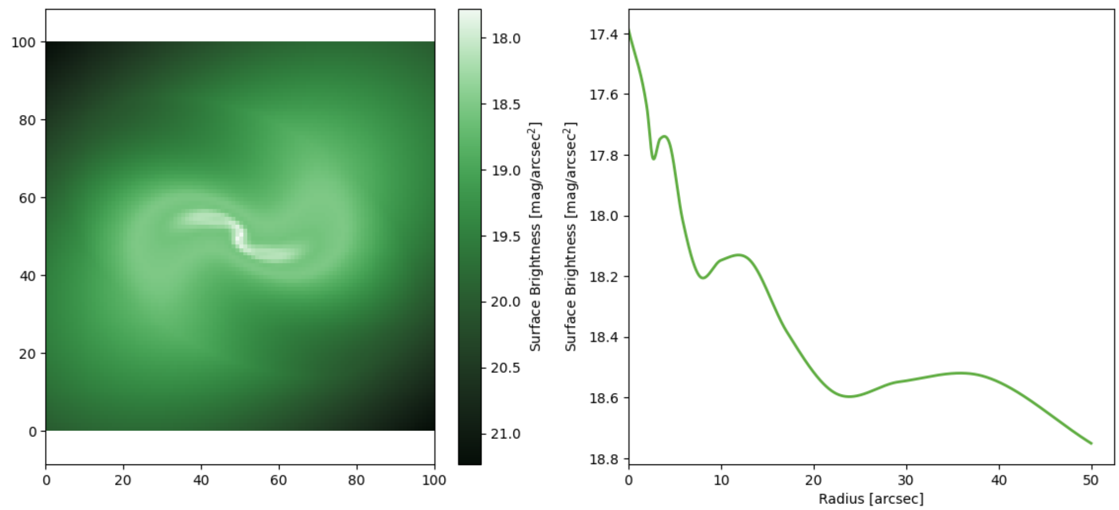
2.4.1 Spline Warp

[23]:
```
warp_q = np.linspace(0.1,0.4,14)
warp_pa = np.linspace(0,np.pi-0.2,14)
M = ap.models.AutoPhot_Model(name = "spline warp", model_type = "spline warp␣
↪galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/
↪180, "q(R)": warp_q, "PA(R)": warp_pa, "I(R)": {"value": spline_profile,␣
↪"prof": NP_prof}}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
```

```
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'I(R)')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'log10(flux/arcsec^2)')
```

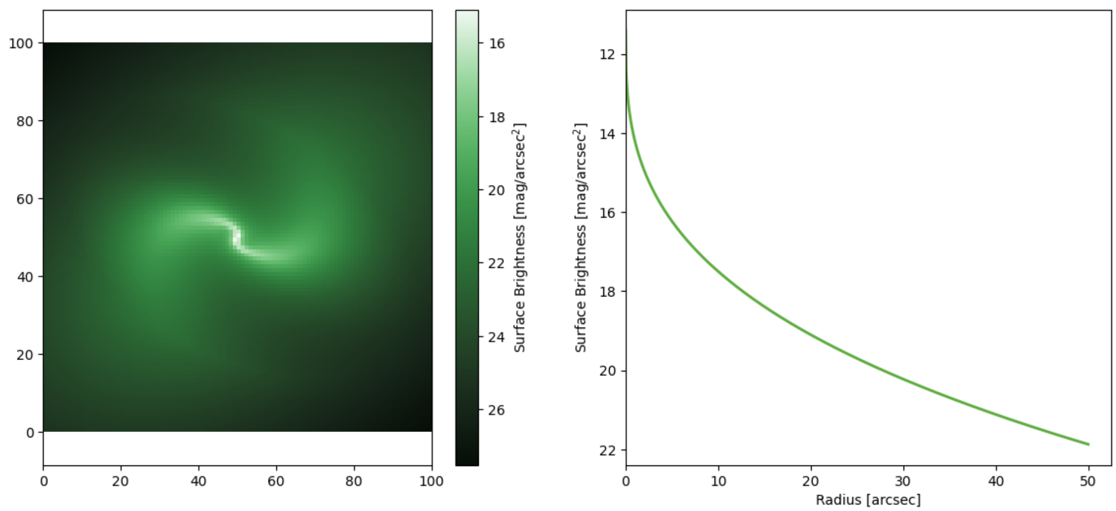


2.4.2 Sersic Warp

[24]:
```
M = ap.models.AutoPhot_Model(name = "sersic warp", model_type = "sersic warp␣
↪galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/
↪180, "q(R)": warp_q, "PA(R)": warp_pa, "n": 3, "Re": 10, "Ie": 1}, target =␣
↪basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'n', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'none', 'arcsec',
'log10(flux/arcsec^2)')
```

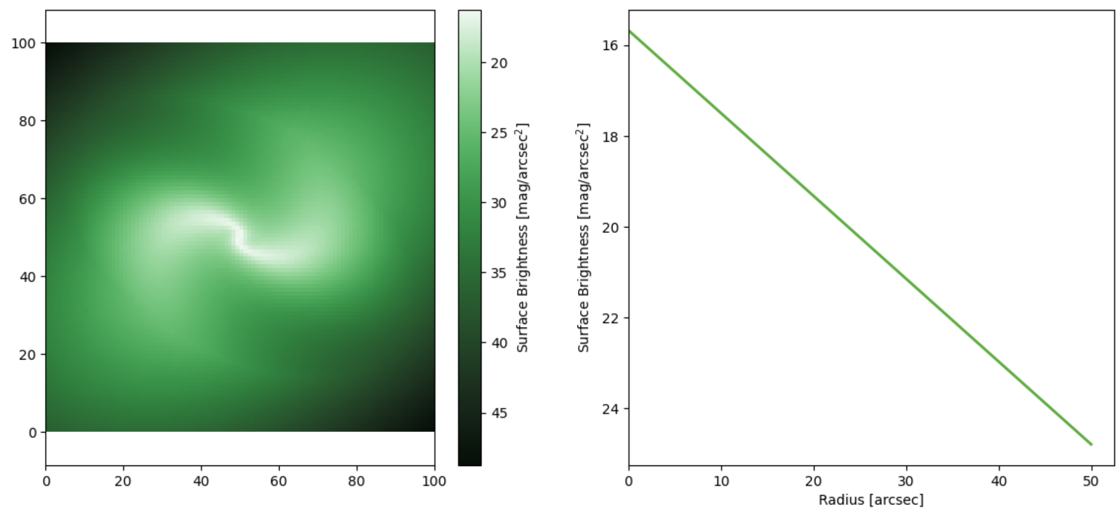


2.4.3 Exponential Warp

[25]:
```
M = ap.models.AutoPhot_Model(name = "exp warp", model_type = "exponential warp␣
↪galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/
↪180, "q(R)": warp_q, "PA(R)": warp_pa, "Re": 10, "Ie": 1}, target =␣
↪basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'arcsec', 'log10(flux/arcsec^2)')
```

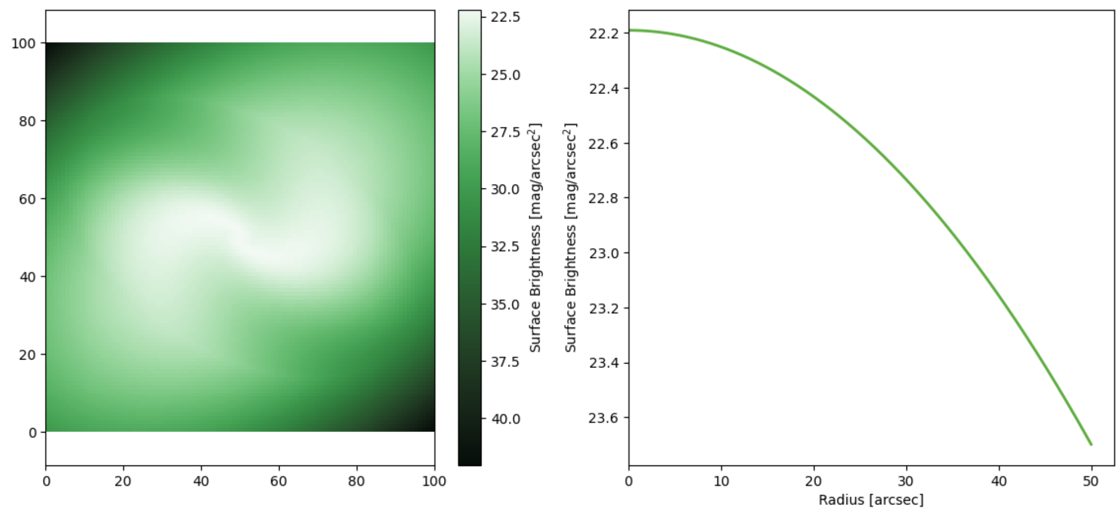


2.4.4 Gaussian Warp

[26]:
```
M = ap.models.AutoPhot_Model(name = "gauss warp", model_type = "gaussian warp␣
↪galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/
↪180, "q(R)": warp_q, "PA(R)": warp_pa, "sigma": 30, "flux": 1}, target =␣
↪basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'sigma', 'flux')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'arcsec', 'log10(flux)')
```

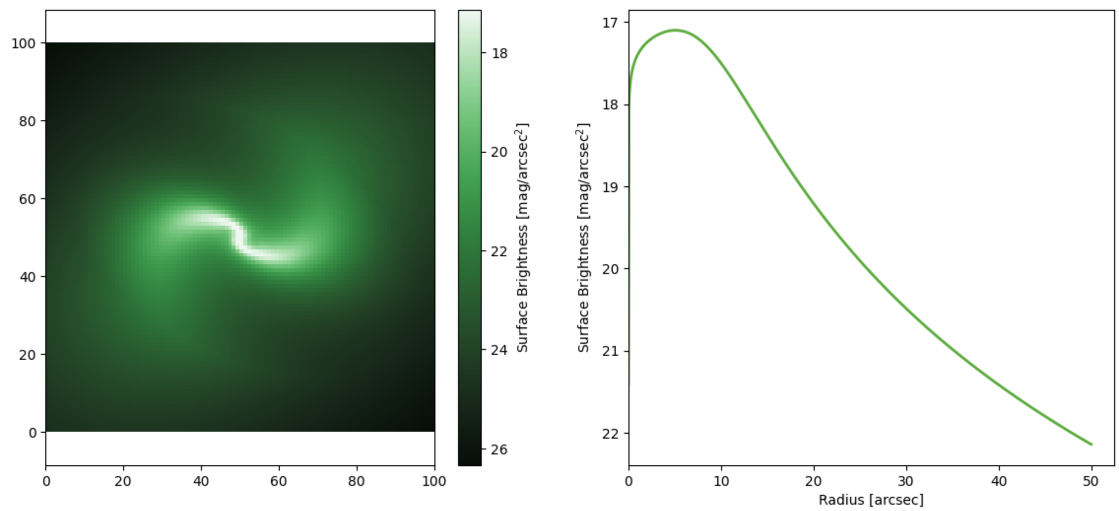


2.4.5 Nuker Warp

[27]:
```
M = ap.models.AutoPhot_Model(name = "nuker warp", model_type = "nuker warp␣
↪galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/
↪180, "q(R)": warp_q, "PA(R)": warp_pa, "Rb": 10., "Ib": 1., "alpha": 4.,␣
↪"beta": 3., "gamma": -0.2}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'Rb', 'Ib', 'alpha', 'beta', 'gamma')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'arcsec', 'log10(flux/arcsec^2)',
'none', 'none', 'none')
```



2.5 Ray Model

A ray model allows the user to break the galaxy up into regions that can be fit separately. There are two basic kinds of ray model: symmetric and asymetric. A symmetric ray model (symmetric_rays = True) assumes 180 degree symmetry of the galaxy and so each ray is reflected through the center. This means that essentially the major axes and the minor axes are being fit separately. For an asymmetric ray model (symmetric_rays = False) each ray is it's own profile to be fit separately.

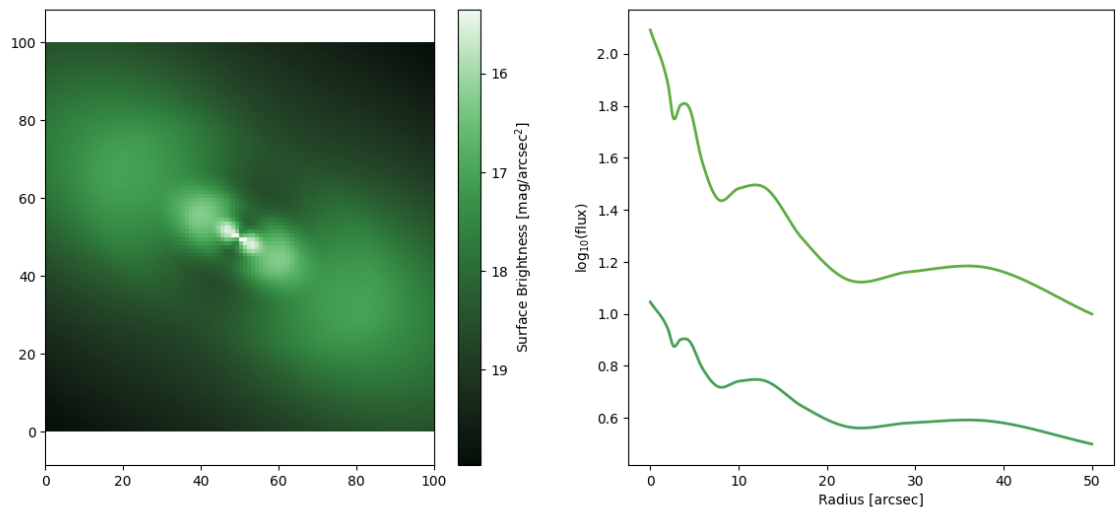
In a ray model there is a smooth boundary between the rays. This smoothness is acomplished by applying a $(\cos(r * theta) + 1)/2$ weight to each profile, where r is dependent on the number of rays and theta is shifted to center on each ray in turn. The exact cosine weighting is dependent on if the rays are symmetric and if there is an even or odd number of rays.

2.5.1 Spline Ray

[28]:
```
M = ap.models.AutoPhot_Model(name = "spline ray", model_type = "spline ray galaxy model", symmetric_rays = True, rays = 2, parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/180, "I(R)": {"value": np.array([spline_profile*2, spline_profile]), "prof": NP_prof}}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.ray_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'I(R)')
('arcsec', 'b/a', 'radians', 'log10(flux/arcsec^2)')
```

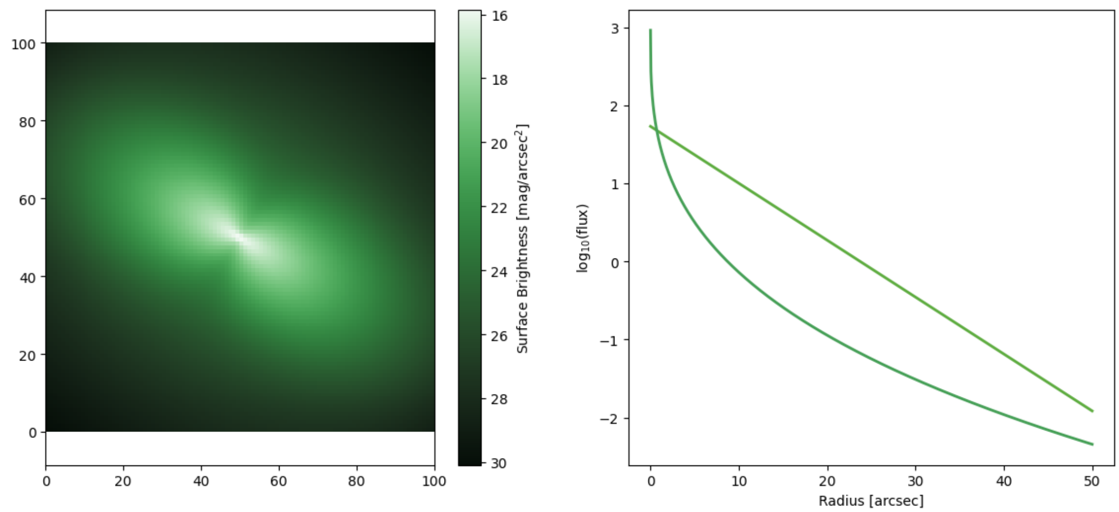


2.5.2 Sersic Ray

[29]:
```
M = ap.models.AutoPhot_Model(name = "sersic ray", model_type = "sersic ray␣
↪galaxy model", symmetric_rays = True, rays = 2, parameters = {"center":␣
↪[50,50], "q": 0.6, "PA": 60*np.pi/180, "n": [1,3], "Re": [10,5],"Ie": [1,0.
↪5]}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.ray_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'n', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'none', 'arcsec', 'log10(flux/arcsec^2)')
```

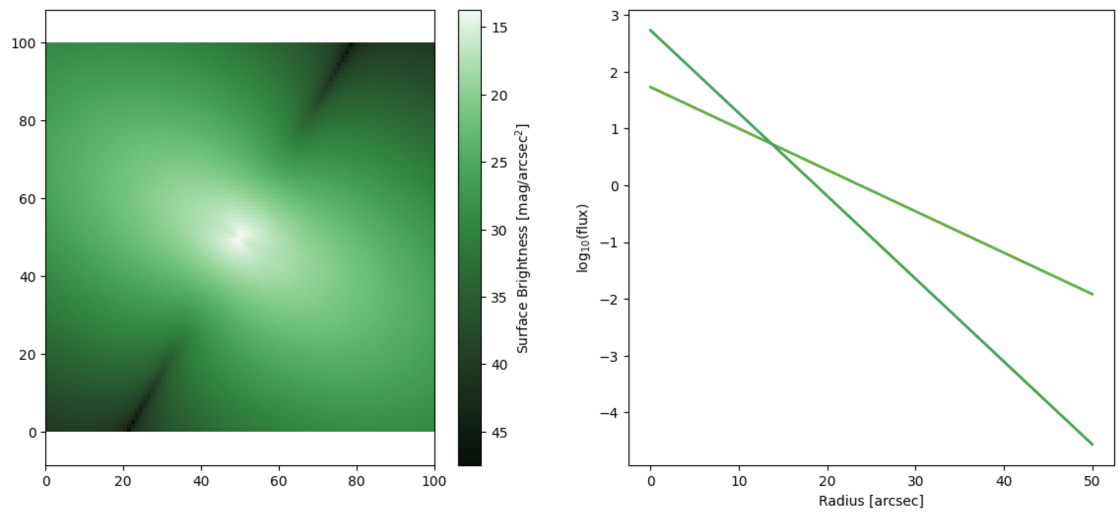


2.5.3 Exponential Ray

[30]:
```
M = ap.models.AutoPhot_Model(name = "exponential ray", model_type =␣
↪"exponential ray galaxy model", symmetric_rays = True, rays = 2, parameters␣
↪= {"center": [50,50], "q": 0.6, "PA": 60*np.pi/180, "Re": [10,5], "Ie":␣
↪[1,2]}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.ray_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'arcsec', 'log10(flux/arcsec^2)')
```

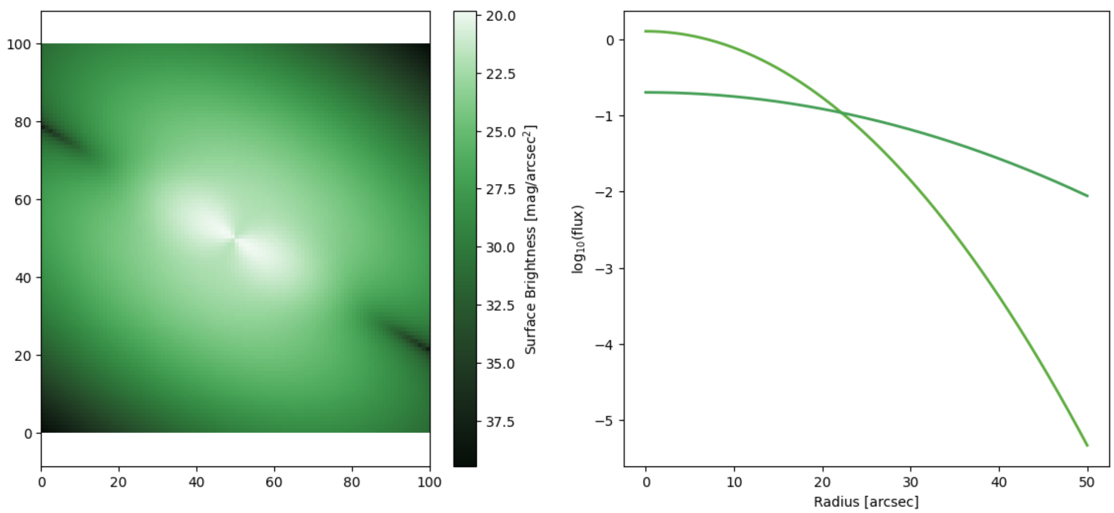


2.5.4 Gaussian Ray

[31]:
```
M = ap.models.AutoPhot_Model(name = "gaussian ray", model_type = "gaussian ray␣
↪galaxy model", symmetric_rays = True, rays = 2, parameters = {"center":␣
↪[50,50], "q": 0.6, "PA": 60*np.pi/180, "sigma": [10,20], "flux": [1.5,1.]},␣
↪target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.ray_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'sigma', 'flux')
('arcsec', 'b/a', 'radians', 'arcsec', 'log10(flux)')
```

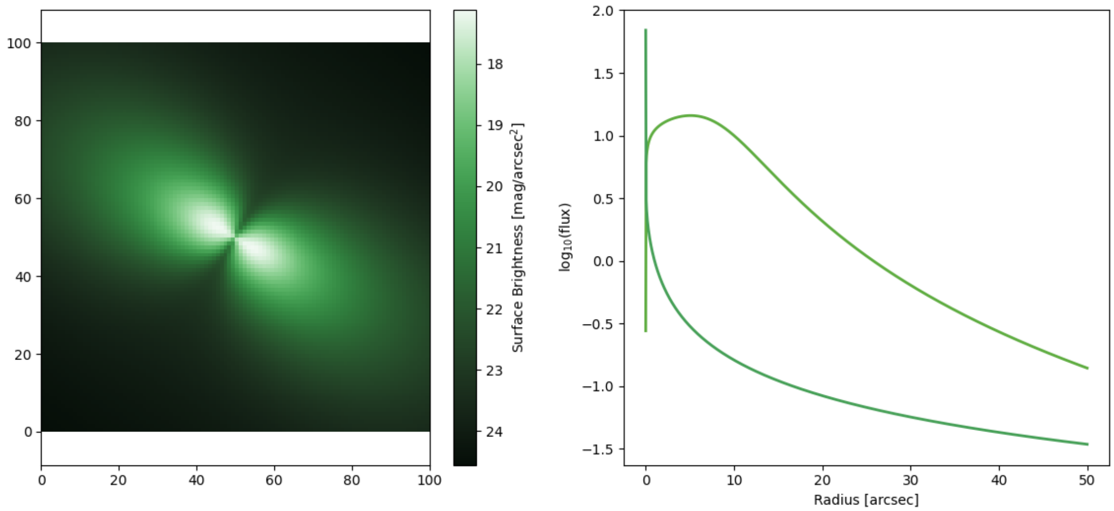


2.5.5 Nuker Ray

[32]:
```python
M = ap.models.AutoPhot_Model(name = "nuker ray", model_type = "nuker ray galaxy␣
↪model", symmetric_rays = True, rays = 2, parameters = {"center": [50,50],␣
↪"q": 0.6, "PA": 60*np.pi/180, "Rb": [10.,1.], "Ib": [1.,0.], "alpha": [4.,1.
↪], "beta": [3.,1.], "gamma": [-0.2,0.2]}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.ray_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'Rb', 'Ib', 'alpha', 'beta', 'gamma')
('arcsec', 'b/a', 'radians', 'arcsec', 'log10(flux/arcsec^2)', 'none', 'none',
'none')
```



2.6 Wedge Model

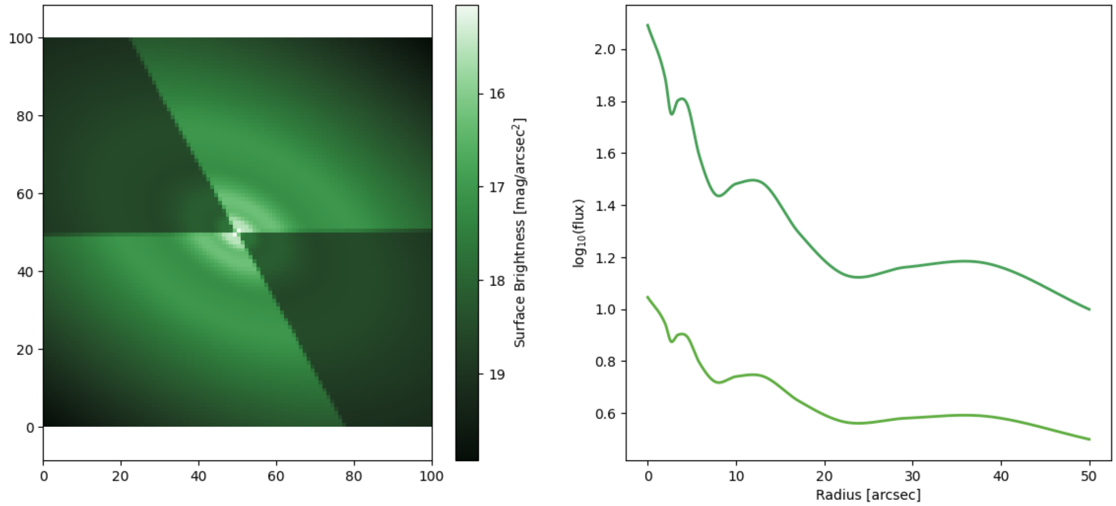
A wedge model behaves just like a ray model, except the boundaries are sharp. This has the advantage that the wedges can be very different in brightness without the "smoothing" from the ray model washing out the dimmer one. It also has the advantage of less "mixing" of information between the rays, each one can be counted on to have fit only the pixels in it's wedge without any influence from a neighbor. However, it has the disadvantage that the discontinuity at the boundary makes fitting behave strangely when a bright spot lays near the boundary.

2.6.1 Spline Wedge

[33]:
```
M = ap.models.AutoPhot_Model(name = "spline wedge", model_type = "spline wedge␣
↪galaxy model", symmetric_wedges = True, wedges = 2, parameters = {"center":␣
↪[50,50], "q": 0.6, "PA": 60*np.pi/180, "I(R)": {"value": np.
↪array([spline_profile, spline_profile*2]), "prof": NP_prof}}, target =␣
↪basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.wedge_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'I(R)')
('arcsec', 'b/a', 'radians', 'log10(flux/arcsec^2)')
```



3 High Order Warp Models

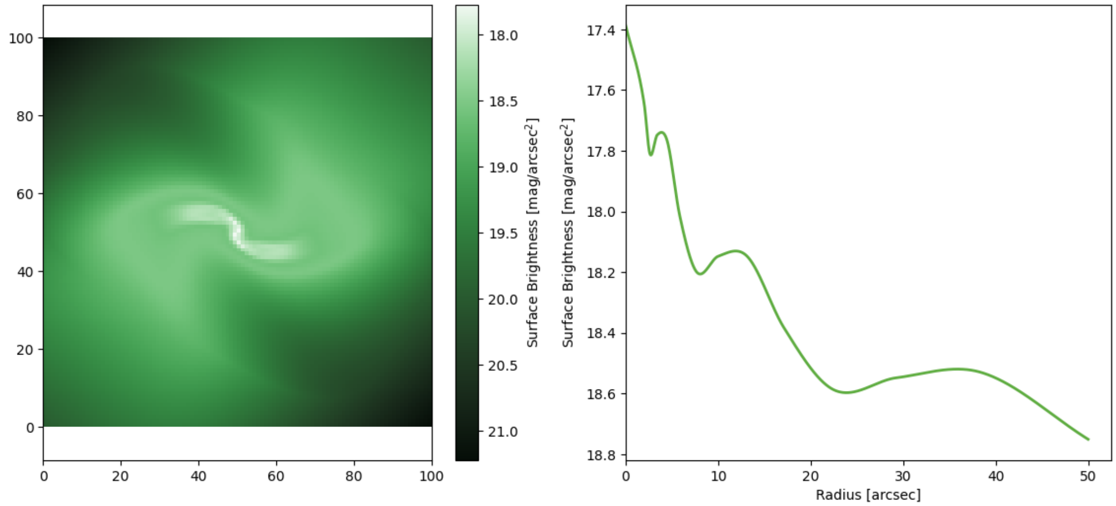
The models below combine the Warp coordinate transform with radial behaviour transforms: SuperEllipse and Fourier. These higher order models can create higly complex shapes, though their scientific use-case is less clear. They are included for completeness as they may be useful in some specific instances. These models are also included to demonstrate the flexibility in making AutoPhot models, in a future tutorial we will discuss how to make your own model types.

3.0.1 Spline SuperEllipse Warp

[34]:
```
M = ap.models.AutoPhot_Model(name = "spline super warp", model_type = "spline␣
↪superellipse warp galaxy model", parameters = {"center": [50,50], "q": 0.6,␣
↪"PA": 60*np.pi/180, "q(R)": warp_q, "PA(R)": warp_pa, "C0": 2, "I(R)":␣
↪{"value": spline_profile, "prof": NP_prof}}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'C0', 'I(R)')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'C-2', 'log10(flux/arcsec^2)')
```



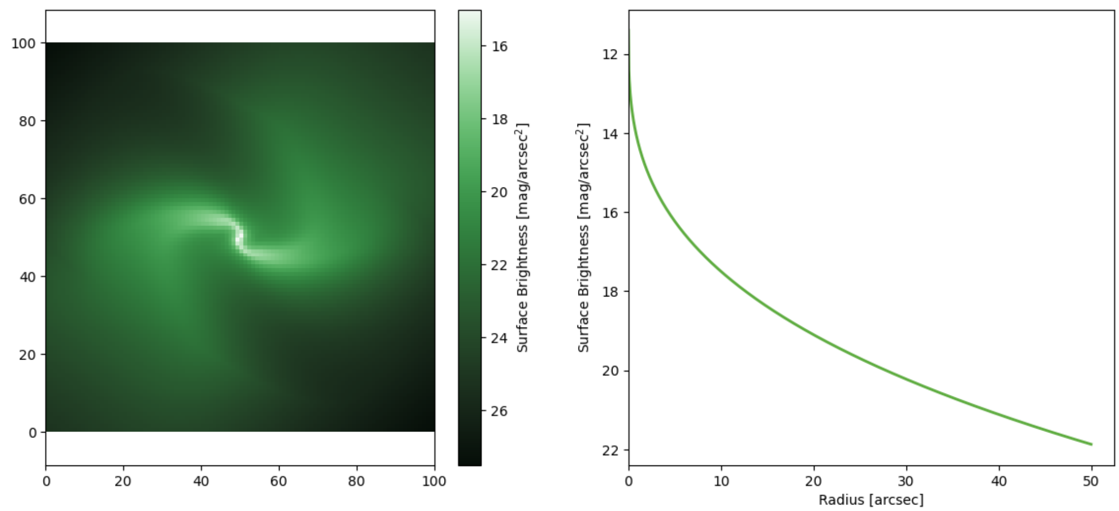
3.0.2 Sersic SuperEllipse Warp

[35]:

```
M = ap.models.AutoPhot_Model(name = "sersic super warp", model_type = "sersic superellipse warp galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA": 60*np.pi/180, "q(R)": warp_q, "PA(R)": warp_pa, "C0": 2, "n": 3, "Re": 10, "Ie": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'C0', 'n', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'C-2', 'none', 'arcsec',
'log10(flux/arcsec^2)')
```

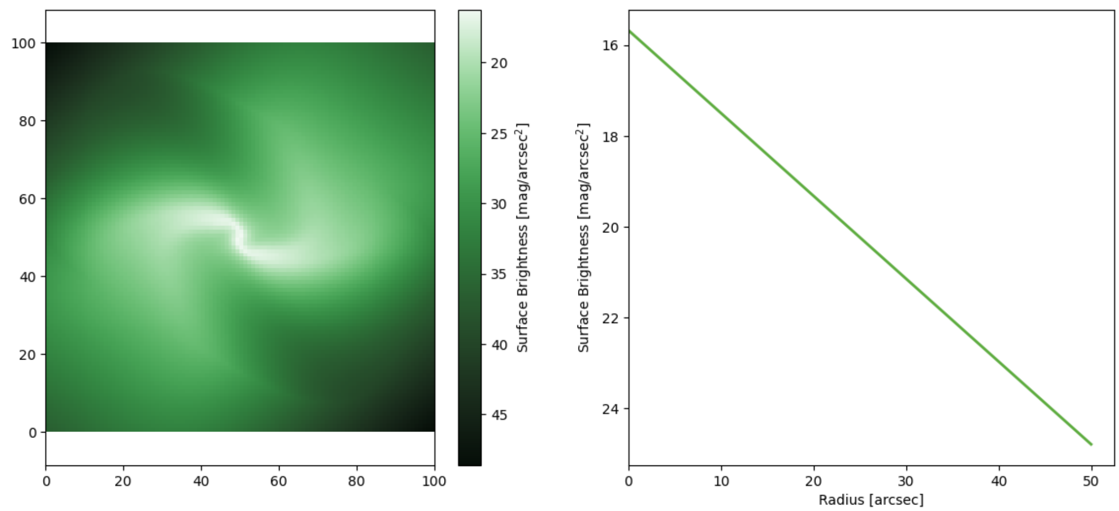


3.0.3 Exponential SuperEllipse Warp

[36]:
```
M = ap.models.AutoPhot_Model(name = "exponential super warp", model_type =␣
↪"exponential superellipse warp galaxy model", parameters = {"center":␣
↪[50,50], "q": 0.6, "PA": 60*np.pi/180, "q(R)": warp_q, "PA(R)": warp_pa,␣
↪"C0": 2, "Re": 10, "Ie": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'C0', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'C-2', 'arcsec',
'log10(flux/arcsec^2)')
```

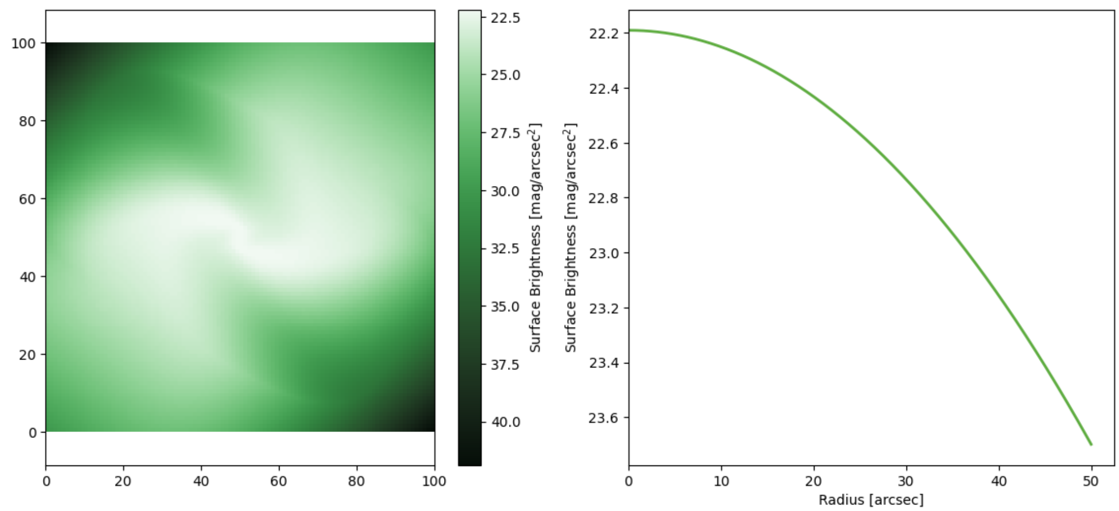


3.0.4 Gaussian SuperEllipse Warp

[37]:
```
M = ap.models.AutoPhot_Model(name = "gauss super warp", model_type = "gaussian␣
↪superellipse warp galaxy model", parameters = {"center": [50,50], "q": 0.6,␣
↪"PA": 60*np.pi/180, "q(R)": warp_q, "PA(R)": warp_pa, "C0": 2, "sigma": 30,␣
↪"flux": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'C0', 'sigma', 'flux')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'C-2', 'arcsec', 'log10(flux)')
```



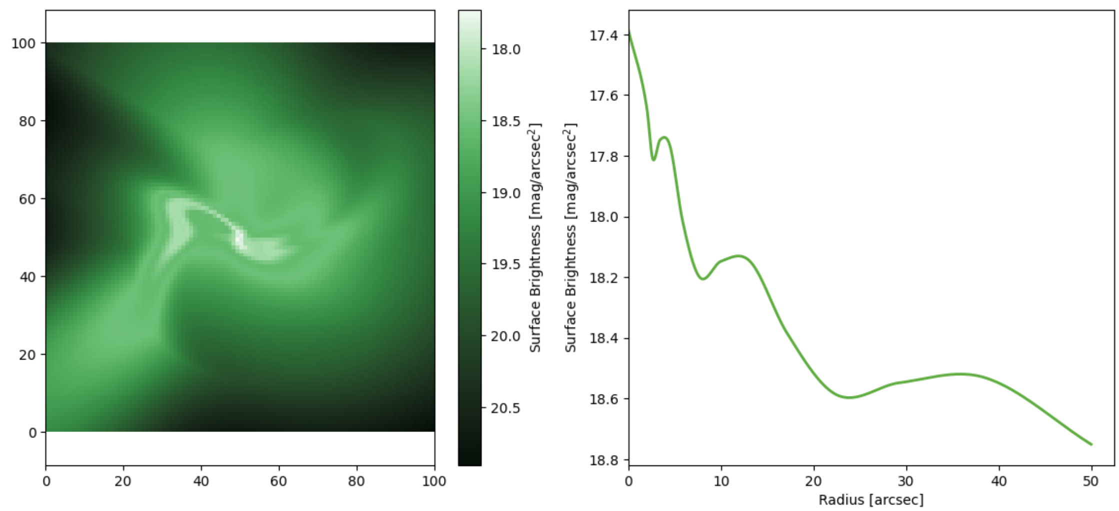
3.0.5 Spline Fourier Warp

not sure how this abomination would fit a galaxy, but you are welcome to try

[38]:
```
M = ap.models.AutoPhot_Model(name = "spline fourier warp", model_type = "spline␣
↪fourier warp galaxy model", modes = (1,3,4), parameters = {"center":␣
↪[50,50], "q": 0.6, "PA": 60*np.pi/180, "q(R)": warp_q, "PA(R)": warp_pa,␣
↪"am": fourier_am, "phim": fourier_phim, "I(R)": {"value": spline_profile,␣
↪"prof": NP_prof}}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'am', 'phim', 'I(R)')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'none', 'radians',
'log10(flux/arcsec^2)')
```

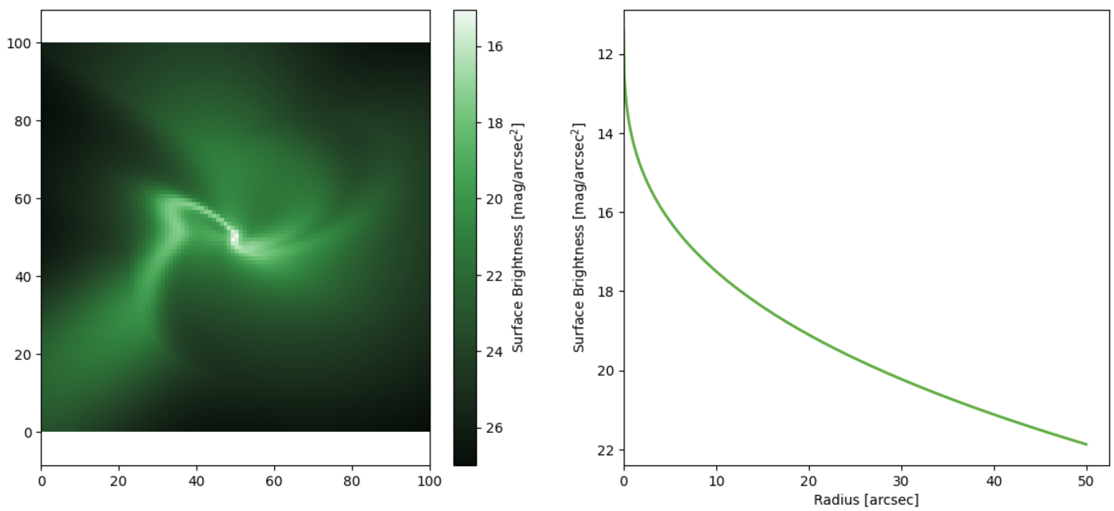


3.0.6 Sersic Fourier Warp

[39]:
```
M = ap.models.AutoPhot_Model(name = "sersic fourier warp", model_type = "sersic␣
↪fourier warp galaxy model", parameters = {"center": [50,50], "q": 0.6, "PA":␣
↪60*np.pi/180, "q(R)": warp_q, "PA(R)": warp_pa, "am": fourier_am, "phim":␣
↪fourier_phim, "n": 3, "Re": 10, "Ie": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'am', 'phim', 'n', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'none', 'radians', 'none', 'arcsec',
'log10(flux/arcsec^2)')
```



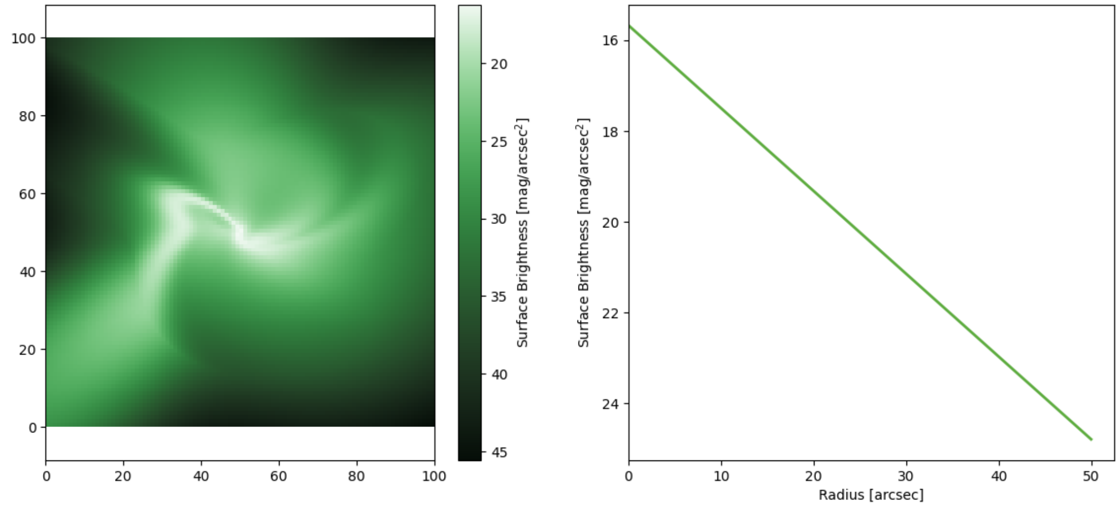
3.0.7 Exponential Fourier Warp

[40]:

```
M = ap.models.AutoPhot_Model(name = "exp fourier warp", model_type =␣
↪"exponential fourier warp galaxy model", parameters = {"center": [50,50],␣
↪"q": 0.6, "PA": 60*np.pi/180, "q(R)": warp_q, "PA(R)": warp_pa, "am":␣
↪fourier_am, "phim": fourier_phim, "Re": 10, "Ie": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'am', 'phim', 'Re', 'Ie')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'none', 'radians', 'arcsec',
'log10(flux/arcsec^2)')
```

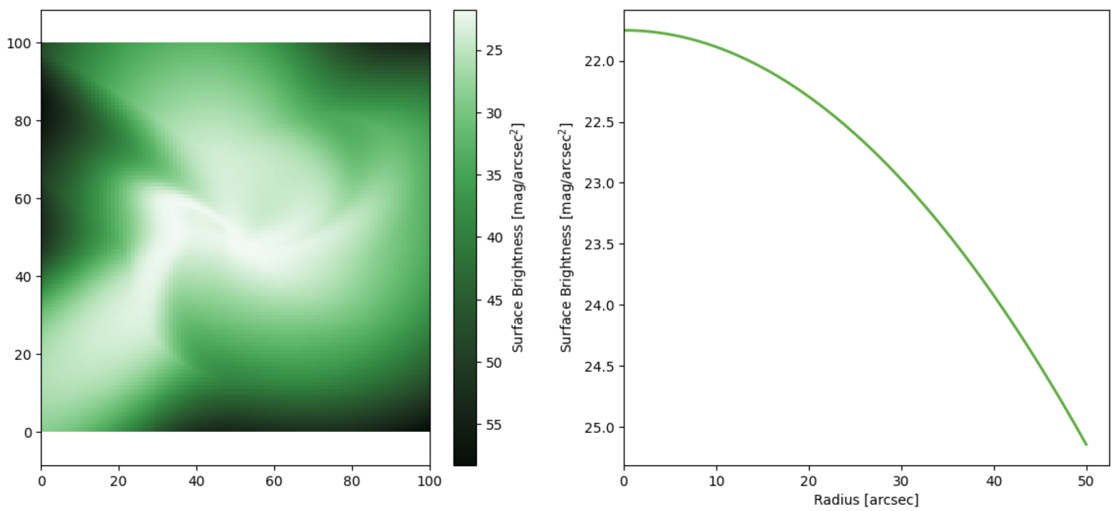


3.0.8 Gassian Fourier Warp

[41]:
```
M = ap.models.AutoPhot_Model(name = "gaussian fourier warp", model_type =␣
↪"gaussian fourier warp galaxy model", parameters = {"center": [50,50], "q":␣
↪0.6, "PA": 60*np.pi/180, "q(R)": warp_q, "PA(R)": warp_pa, "am": fourier_am,␣
↪"phim": fourier_phim, "sigma": 20, "flux": 1}, target = basic_target)
print(M.parameter_order)
print(tuple(P.units for P in M.parameters))
M.initialize()

fig, ax = plt.subplots(1,2, figsize = (14,6))
ap.plots.model_image(fig, ax[0], M)
ap.plots.galaxy_light_profile(fig,ax[1],M)
plt.show()
```

```
('center', 'q', 'PA', 'q(R)', 'PA(R)', 'am', 'phim', 'sigma', 'flux')
('arcsec', 'b/a', 'radians', 'b/a', 'rad', 'none', 'radians', 'arcsec',
'log10(flux)')
```



Surface Brightness [mag/arcsec^2]
Surface Brightness [mag/arcsec^2]
Radius [arcsec]

[ ]: