Problema 2: Perfil de Sérsic

a) Del enunciado, sabemos que el perfil de brillo superficial de Sérsic se define como:

$$I(r) = I_e \exp\left\{ -b_n \left[\left(\frac{r}{r_e}\right)^{1/n} - 1 \right] \right\}$$

Para obtener la luminosidad de un objeto con este perfil, tenemos que integrar el perfil sobre un área circular:

$$L = \iint_S I(r)\, dS$$

Considerando simetría circular, tenemos que:

$$L = \int_0^{2\pi} \int_0^{\infty} I(r)\, r\, dr\, d\theta$$

$$= 2\pi \int_0^{\infty} r\, I_e \cdot \exp\left\{ -b_n \left[\left(\frac{r}{r_e}\right)^{1/n} - 1 \right] \right\} dr$$

Hacemos el cambio de variable:

$$u = (r/r_e)^{1/n}$$

$$u|_0 = 0$$

$$u|^{\infty} \to \infty \quad \text{para el rango típico de índices de Sérsic } (n << r)$$

$$du = \frac{1}{n}\left(\frac{r}{r_e}\right)^{(1/n)-1} \frac{1}{r_e}\, dr = \frac{1}{n} u^{1-n} \frac{1}{r_e}\, dr$$

$\rightarrow r = \mu^n r_e$

$\rightarrow dr = n r_e \mu^{n-1} \, d\mu$

Reemplazando en la integral:

$$\Rightarrow \mathcal{L} = 2\pi I_e \int_0^\infty (\mu^n r_e) \, e^{-b_n(\mu-1)} (n r_e \mu^{n-1}) \, d\mu$$

$$= 2\pi I_e \int_0^\infty \mu^{2n-1} r_e^2 n \, e^{-b_n \mu} e^{b_n} \, d\mu$$

$$= 2\pi I_e r_e^2 n e^{b_n} \int_0^\infty \mu^{2n-1} e^{-b_n \mu} \, d\mu$$

Hacemos otro cambio de variable:

$t = b_n \mu$

$t|_0 = 0$

$t|^\infty \rightarrow \infty$

$dt = b_n \, d\mu$

$\rightarrow \mu = b_n^{-1} t$

$\rightarrow d\mu = b_n^{-1} dt$

Reemplazando,

$$\Rightarrow \mathcal{L} = 2\pi I_e r_e^2 n e^{b_n} \int_0^\infty (b_n^{-1} t)^{2n-1} e^{-t} b_n^{-1} dt$$

$$= 2\pi I_e r_e^2 n e^{b_n} \int_0^\infty \left(b_n^{1-2n} \cdot b_n^{-1}\right) t^{2n-1} e^{-t} dt$$

$$= \frac{2\pi n e^{b_n}}{(b_n)^{2n}} I_e r_e^2 \underbrace{\int_0^\infty t^{2n-1} e^{-t}}_{= \Gamma(2n)}$$

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$$

Función Gamma

$$\Rightarrow \quad L = \frac{2\pi n e^{b_n} \Gamma(2n)}{(b_n)^{2n}} I_e r_e^2$$

b) Queremos demostrar que, si r_e contiene la mitad de la luminosidad, entonces b_n cumple que $2\gamma(2n, b_n) = \Gamma(2n)$.

Primero, podemos obtener la luminosidad contenida dentro de un radio r_e, integrando desde 0 a este radio:

$$L_{r_e} = 2\pi I_e \int_0^{r_e} r \exp\left\{-b_n\left[\left(\frac{r}{r_e}\right)^{1/n} - 1\right]\right\} dr$$

Hacemos el cambio de variable:

$$t = b_n \left(\frac{r}{r_e}\right)^{1/n}$$

$$t|_0 = 0$$

$$t|^{r_e} = b_n$$

$$dt = \frac{b_n}{n}\left(\frac{r}{r_e}\right)^{(1/n)-1} \frac{1}{r_e} dr$$

$$\rightarrow \quad r = \frac{r_e t^n}{(b_n)^n}$$

$$\rightarrow \quad dr = \frac{n r_e}{(b_n)^n} t^{n-1} dt$$

$$\Rightarrow \quad r dr = \frac{n r_e^2}{(b_n)^{2n}} t^{2n-1} dt$$

Reemplazando,

$$\gamma(s,x) = \int_0^x t^{s-1} e^{-t} dt$$

Función gamma incompleta inferior

$$\rightarrow \mathcal{L}_{r_e} = \frac{2\pi I_e n r_e^2 e^{b_n}}{(b_n)^{2n}} \underbrace{\int_0^{b_n} t^{2n-1} e^{-t} dt}_{\gamma(2n,\, b_n)}$$

$$\Rightarrow \mathcal{L}_{r_e} = \frac{2\pi I_e n r_e^2 e^{b_n}}{(b_n)^{2n}} \gamma(2n, b_n)$$

Si r_e contiene la mitad de la luminosidad, entonces debe cumplirse que:

$$\mathcal{L}_{r_e} = \frac{\mathcal{L}_{tot}}{2} \longrightarrow 2\mathcal{L}_{r_e} = \mathcal{L}_{tot}$$

Siendo $\mathcal{L}_{tot} = \frac{2\pi n e^{b_n} \Gamma(2n)}{(b_n)^{2n}} I_e r_e^2$. Luego,

$$2 \cdot \frac{\cancel{2\pi} \cancel{I_e} \cancel{n} \cancel{r_e^2} \cancel{e^{b_n}}}{\cancel{(b_n)^{2n}}} \gamma(2n, b_n) = \frac{\cancel{2\pi n} \cancel{e^{b_n}} \Gamma(2n)}{\cancel{(b_n)^{2n}}} \cancel{I_e} \cancel{r_e^2}$$

$$\longrightarrow \quad \underline{2\gamma(2n, b_n) = \Gamma(2n)}$$

Así, encontramos la expresión que describe a los coeficientes b_n.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import (MultipleLocator, AutoMinorLocator)
from scipy.special import gammaincinv
from astropy.modeling.functional_models import Sersic2D
from astropy.visualization import simple_norm

plt.rcParams.update({
    'text.usetex': False,
    'text.latex.preamble': r'\usepackage{amsmath}',
    'font.family': 'serif',
    'font.weight': 'normal',
    'figure.facecolor': 'lightgray',
    'mathtext.fontset': 'dejavuserif'
})
```

c) Graficar el perfil radial para índices 0.5, 1, 2, 4, y 6.

- Usar I_e y r_e como unidades

$$I(r) = I_e \exp\left\{-b_n\left[\left(\frac{r}{r_e}\right)^{1/n} - 1\right]\right\}$$

para unidades r_e:

$$R = \frac{r}{r_e}$$

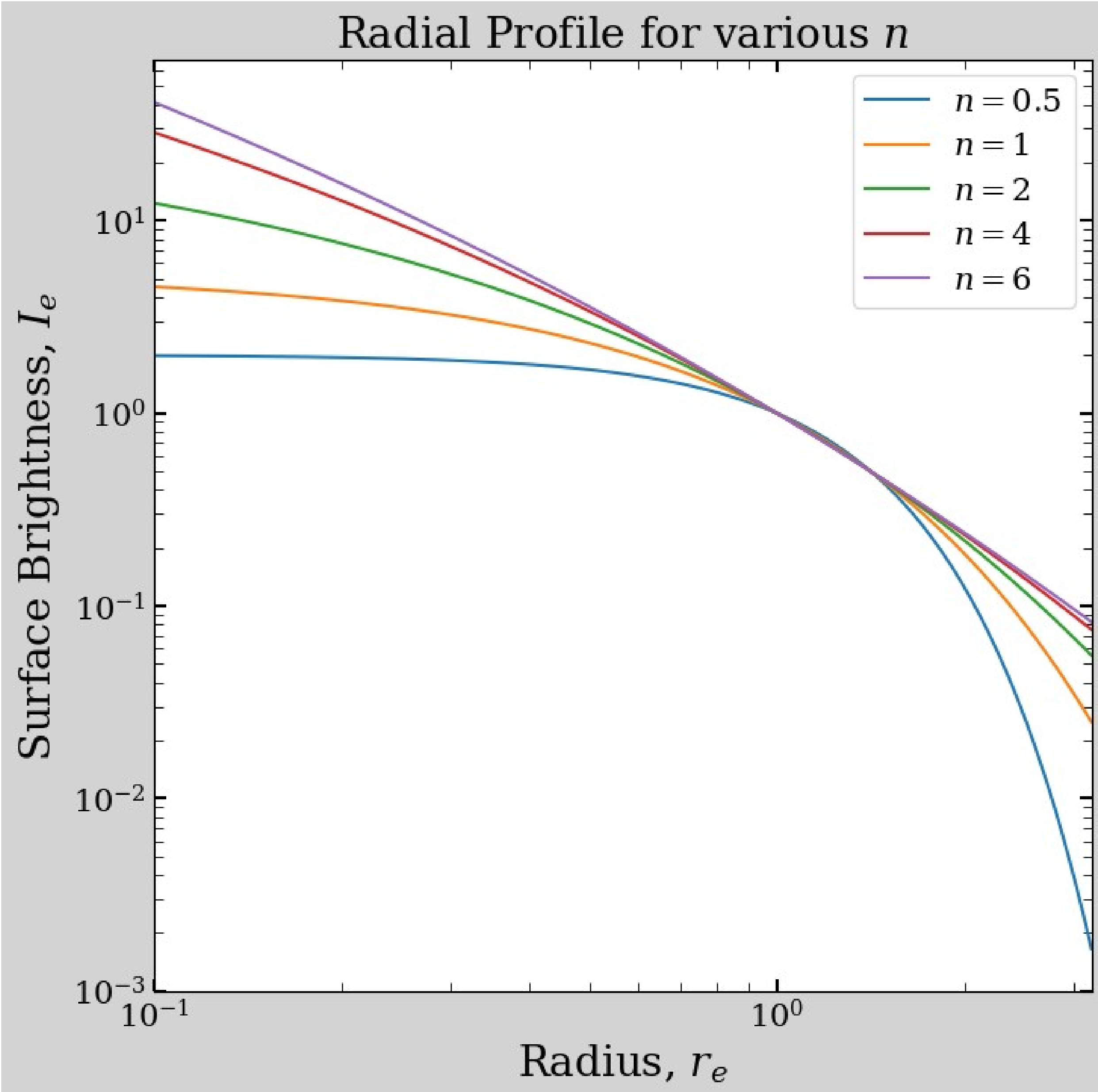
para unidades I_e:

$$I(R) \equiv \frac{I(r)}{I_e} = \exp\{-b_n[R^{1/n} - 1]\}$$

```python
def calculate_b_coefficients(n):
    return gammaincinv(2.0 * n, 0.5)

def radial_profile(r_array, n):
    b = calculate_b_coefficients(n)
    return np.exp(-b * (r_array ** (1/n) - 1))

radial_array = np.linspace(0.1, 3.2, 500)

radial_profile_0_5 = radial_profile(radial_array, 0.5)
radial_profile_1 = radial_profile(radial_array, 1)
radial_profile_2 = radial_profile(radial_array, 2)
radial_profile_4 = radial_profile(radial_array, 4)
radial_profile_6 = radial_profile(radial_array, 6)
```

```python
fig, ax = plt.subplots(figsize=(8, 8))
ax.plot(radial_array, radial_profile_0_5, label=r'$n=0.5$')
ax.plot(radial_array, radial_profile_1, label=r'$n=1$')
ax.plot(radial_array, radial_profile_2, label=r'$n=2$')
ax.plot(radial_array, radial_profile_4, label=r'$n=4$')
ax.plot(radial_array, radial_profile_6, label=r'$n=6$')


ax.set_ylabel(r'Surface Brightness, $I_{e}$', fontsize=20)
ax.set_xlabel(r"Radius, $r_e$", fontsize=20)
ax.set_title(r'Radial Profile for various $n$', fontsize=20)

ax.set_xscale('log')
ax.set_yscale('log')

ax.set_xlim(0.1, 3.2)

ax.tick_params(axis='both', labelsize=15, direction='in', right=True,
top=True,
                length=6, width=1.5, grid_color='black', grid_alpha=1,
grid_linestyle="-",
                grid_linewidth=0.5)

ax.tick_params(which='minor', length=4, color='black', direction='in',
top=True, right=True,
                grid_alpha=0.2, grid_linewidth=0.5,
grid_linestyle="-",grid_color='r')


ax.grid(False, which='both')
ax.legend(fontsize=15, markerscale=1)
```

```
<matplotlib.legend.Legend at 0x7b47ac281c90>
```



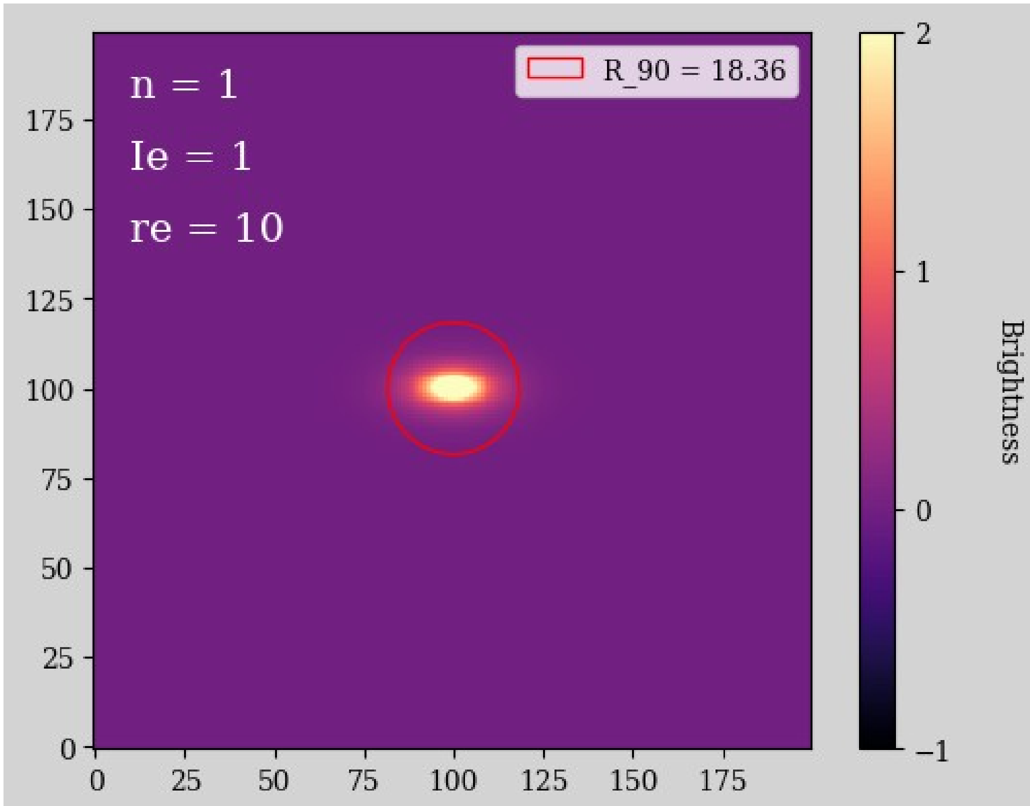
Notamos que a medida que los n aumentan, el brillo superficial decae más lento a medida que nos alejamos del centro del perfil.

d) Usar Sersic2D de astropy para generar imagenes de perfiles de sersic variando I_e, r_e y n

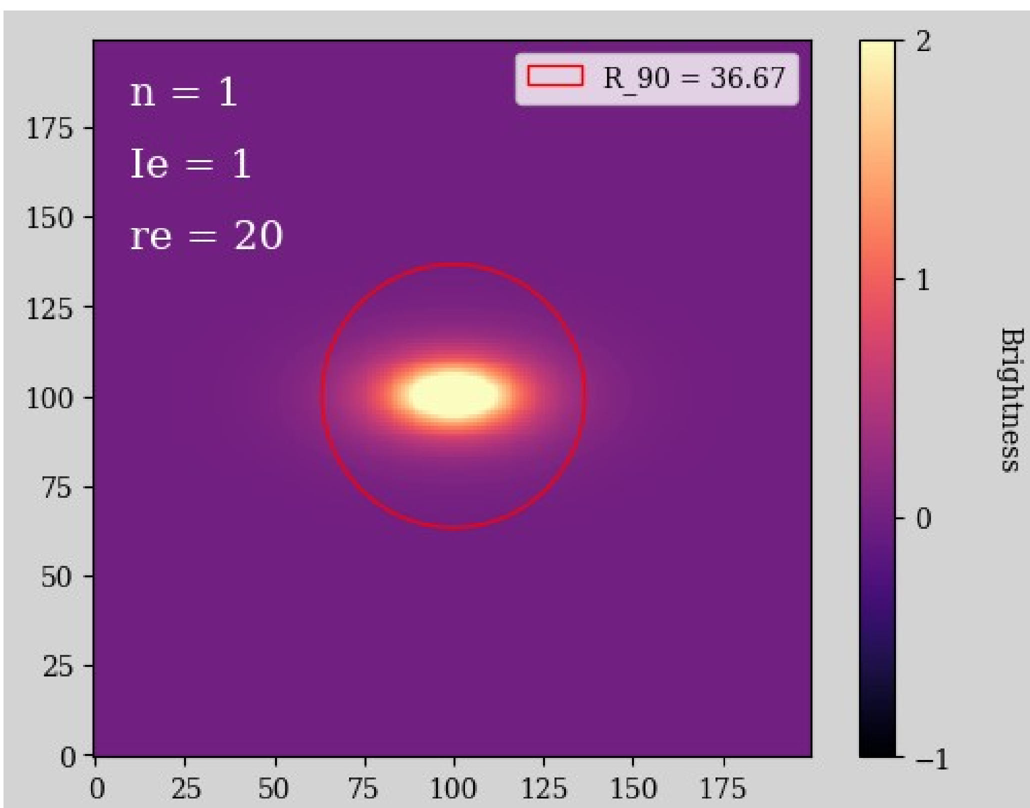
```
def calculate_sersic_profile(intensity, effective_radius, n,
center_x=100, center_y=100, ellipticity=0.5, angle=0):

    # creamos una malla de coordenadas x e y, con un tamaño de 200x200
    x_coords, y_coords = np.meshgrid(np.arange(200), np.arange(200))
```

```
    # definimos el perfil de Sersic2D
    sersic_model = Sersic2D(
        amplitude=intensity,
        r_eff=effective_radius,
        n=n,
        x_0=center_x,
        y_0=center_y,
        ellip=ellipticity,
        theta=angle
    )
    sersic_image = sersic_model(x_coords, y_coords)

    # calculamos la intensidad total sumando todos los pixeles
    total_intensity = sersic_image.sum()

    # calculamos los radios desde el centro de los perfiles
    radii = np.sqrt((x_coords - center_x) ** 2 + (y_coords - center_y)
** 2)

    # ordenamos los radios
    flattened_indices = np.argsort(radii, axis=None)
    # obtenemos un array 1d de radios con ravel()
    sorted_radii = radii.ravel()[flattened_indices]
    # obtenemos un array 1d de intensidades con ravel()
    sorted_intensities = sersic_image.ravel()[flattened_indices]

    # calculamos la intensidad acumulada
    cumulative_intensity = np.cumsum(sorted_intensities)

    # buscamos el índice del radio que contiene el 90% de la
intensidad total
    ninety_percent_index = np.searchsorted(cumulative_intensity, 0.9 *
total_intensity)
    radius_90 = sorted_radii[ninety_percent_index]

    # graficamos
    fig, ax = plt.subplots()
    im = ax.imshow(sersic_image, origin='lower',
interpolation='nearest', vmin=-1, vmax=2, cmap='magma')
    colorbar = fig.colorbar(im, ax=ax)
    colorbar.set_label('Brightness', rotation=270, labelpad=25)
    colorbar.set_ticks([-1, 0, 1, 2])

    # añadimos un círculo indicando el radio del 90%
    circle = plt.Circle((center_x, center_y), radius_90, color='red',
fill=False, label=f'R_90 = {radius_90:.2f}')

    # agregamos textos con los valores de n, Ie y re
    ax.text(0.05, 0.95, f'n = {n}', transform=ax.transAxes,
```

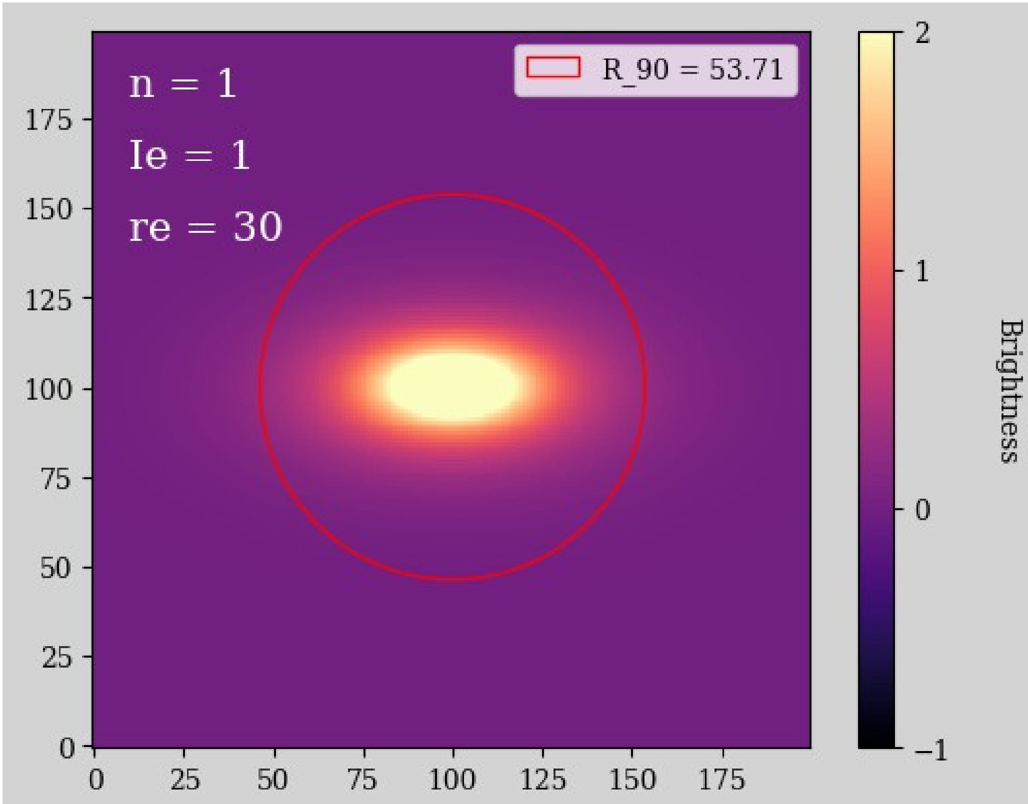
```python
fontsize=15, verticalalignment='top', color='white')
    ax.text(0.05, 0.85, f'Ie = {intensity}', transform=ax.transAxes,
fontsize=15, verticalalignment='top', color='white')
    ax.text(0.05, 0.75, f're = {effective_radius}',
transform=ax.transAxes, fontsize=15, verticalalignment='top',
color='white')

    ax.add_artist(circle)
    ax.legend()

    plt.show()

    return

# Definimos los valores de n, Ie y re que vamos a plotear

Ie_values = [1, 5, 10]
re_values = [10, 20, 30]
n_values = [1, 2, 4]

# Iteramos sobre todos los valores de n, Ie y re

for n in n_values:
    for Ie in Ie_values:
        for re in re_values:
            calculate_sersic_profile(
                intensity=Ie,
                effective_radius=re,
                n=n
            )
```



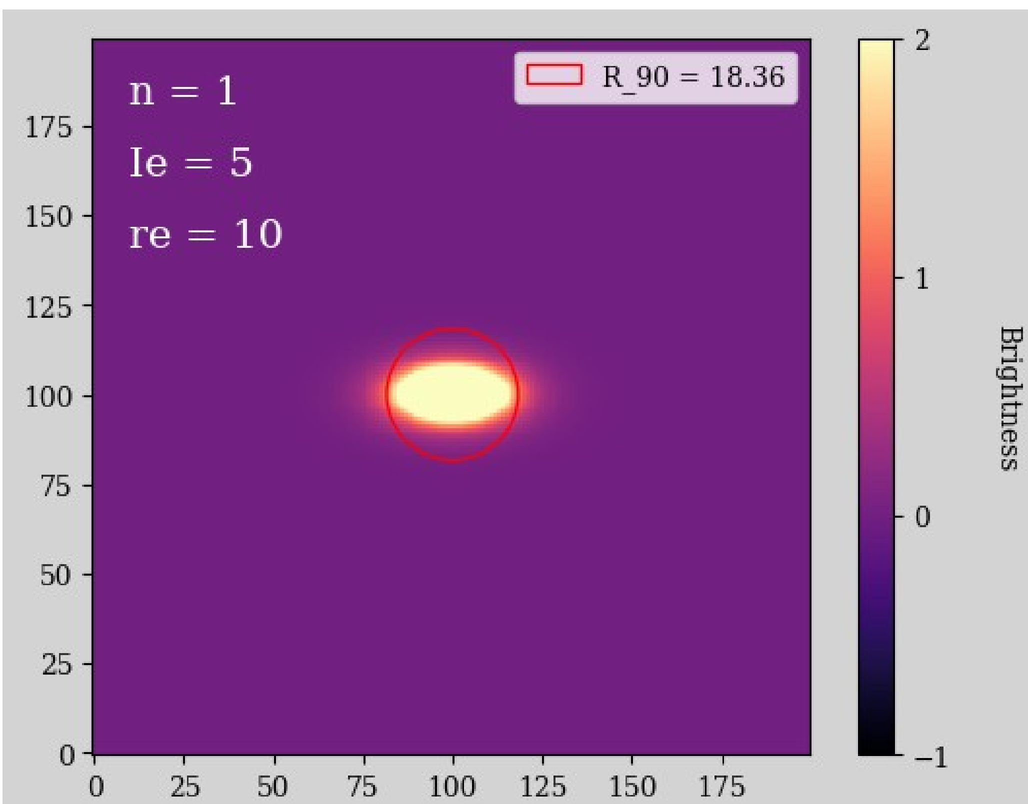
n = 1
Ie = 1
re = 10
R_90 = 18.36
Brightness



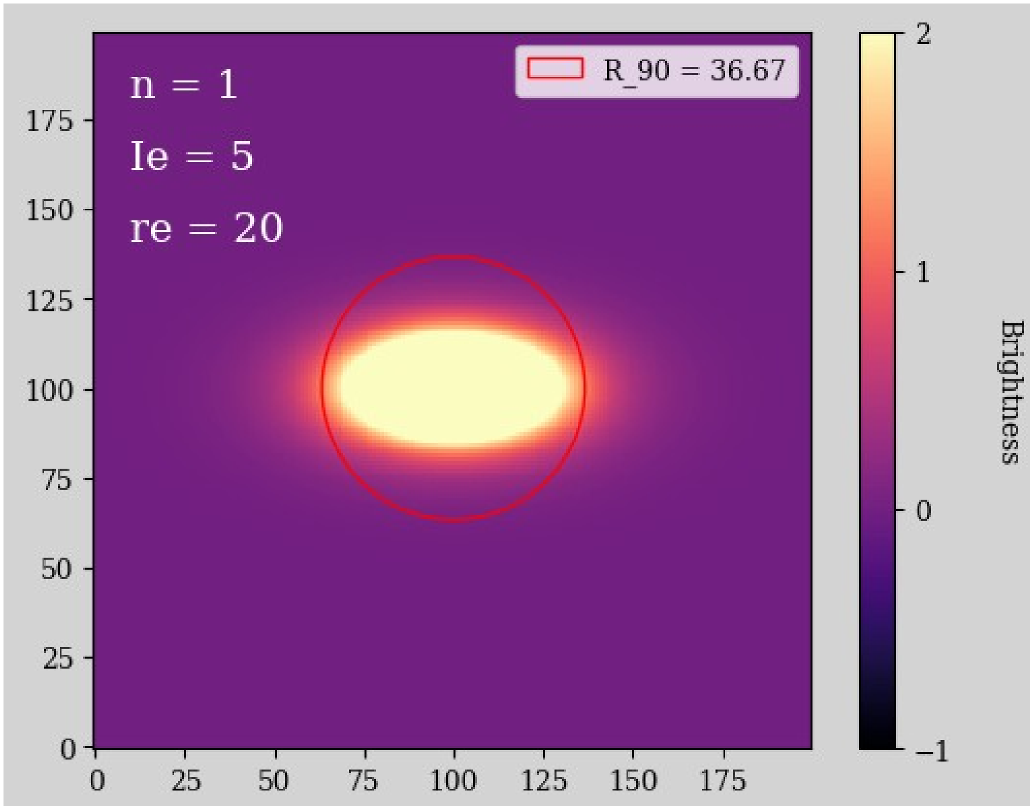
n = 1
Ie = 1
re = 20
R_90 = 36.67
Brightness



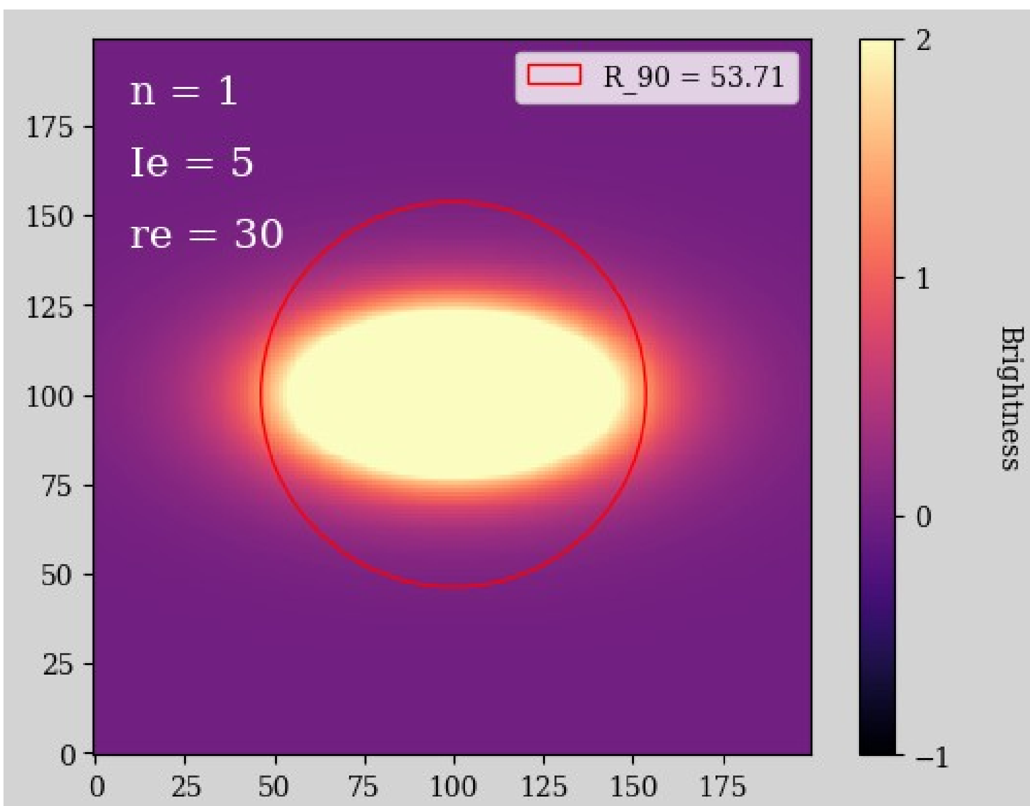
n = 1
Ie = 1
re = 30
R_90 = 53.71
Brightness



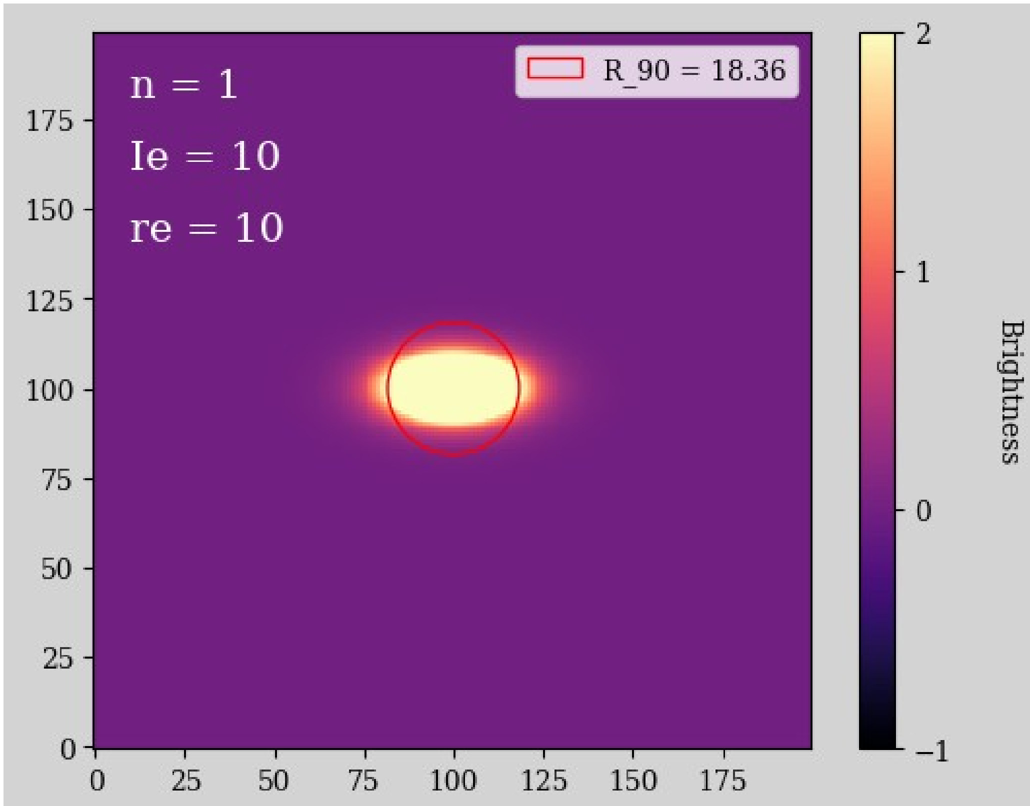
n = 1
Ie = 5
re = 10
R_90 = 18.36
Brightness



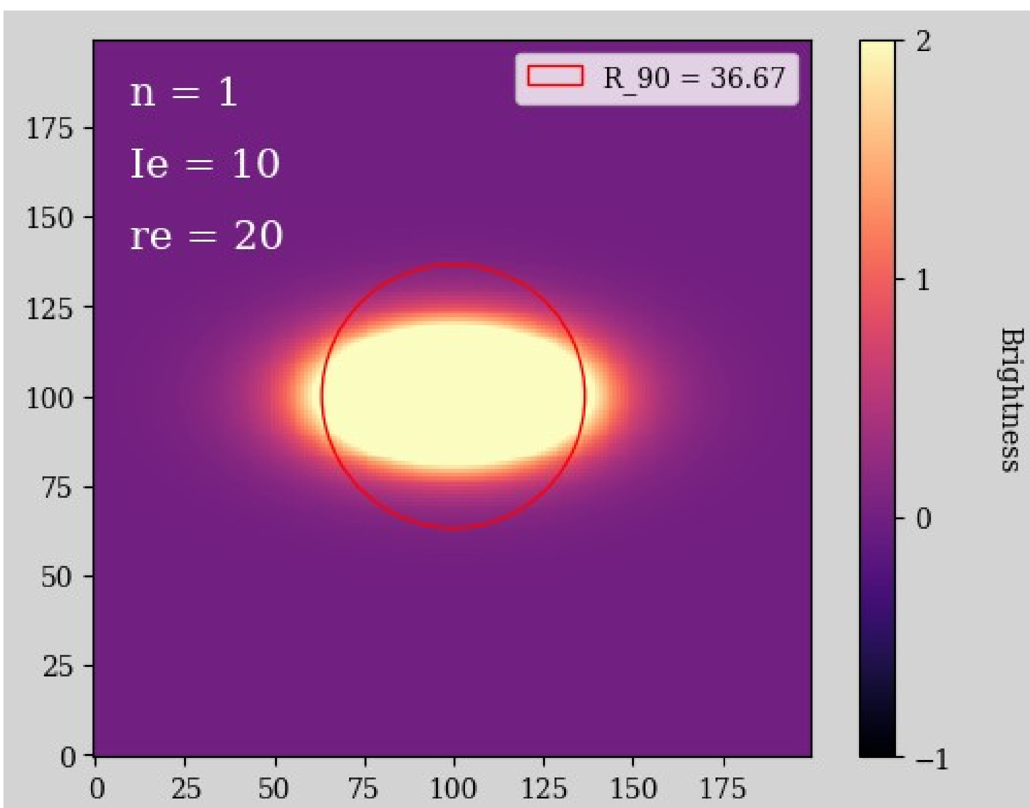
n = 1
Ie = 5
re = 20
R_90 = 36.67
Brightness



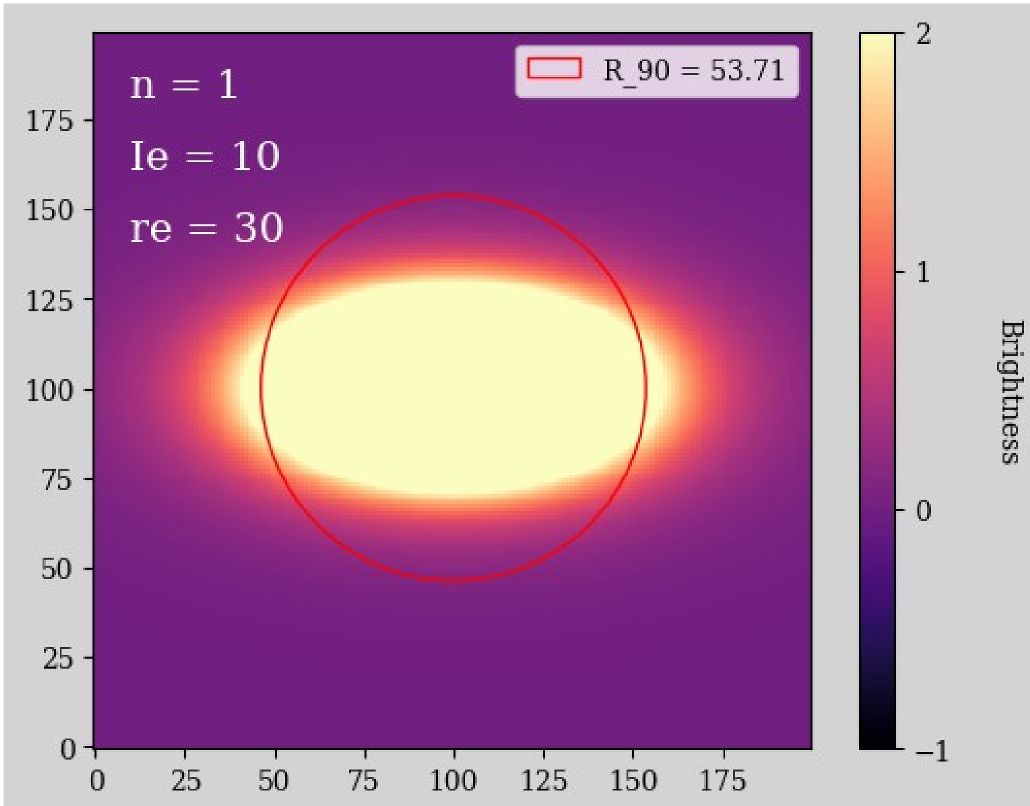
n = 1
Ie = 5
re = 30
R_90 = 53.71
Brightness



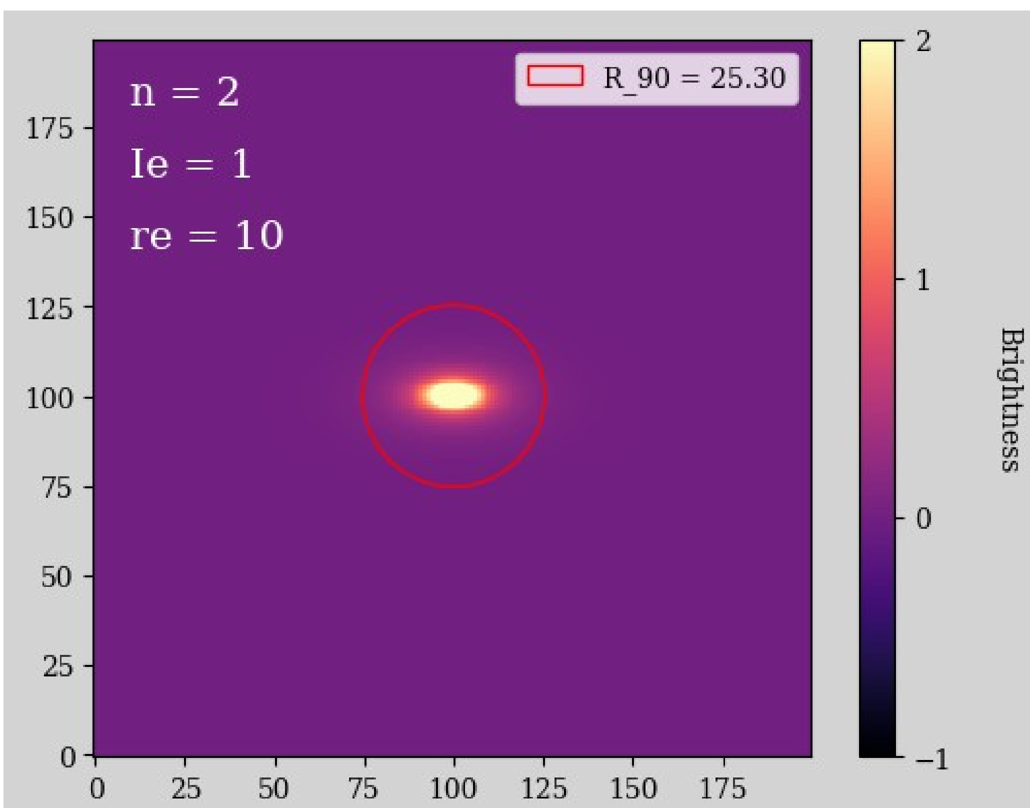
n = 1
Ie = 10
re = 10
R_90 = 18.36
Brightness



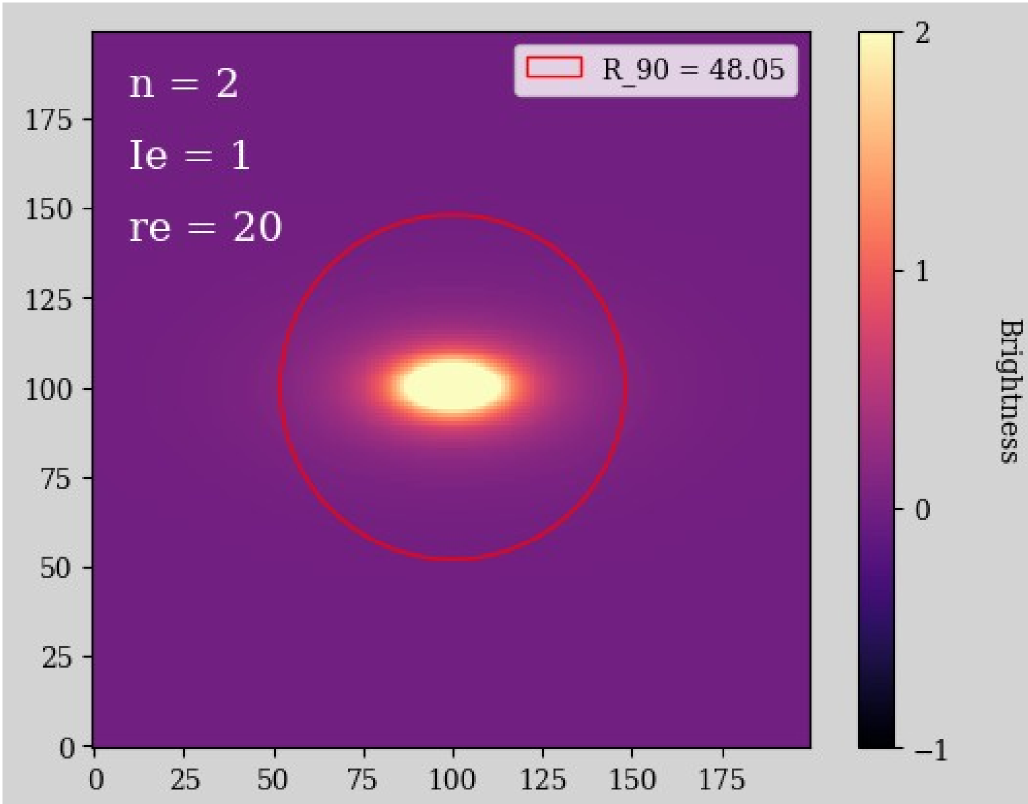
n = 1
Ie = 10
re = 20
R_90 = 36.67
Brightness

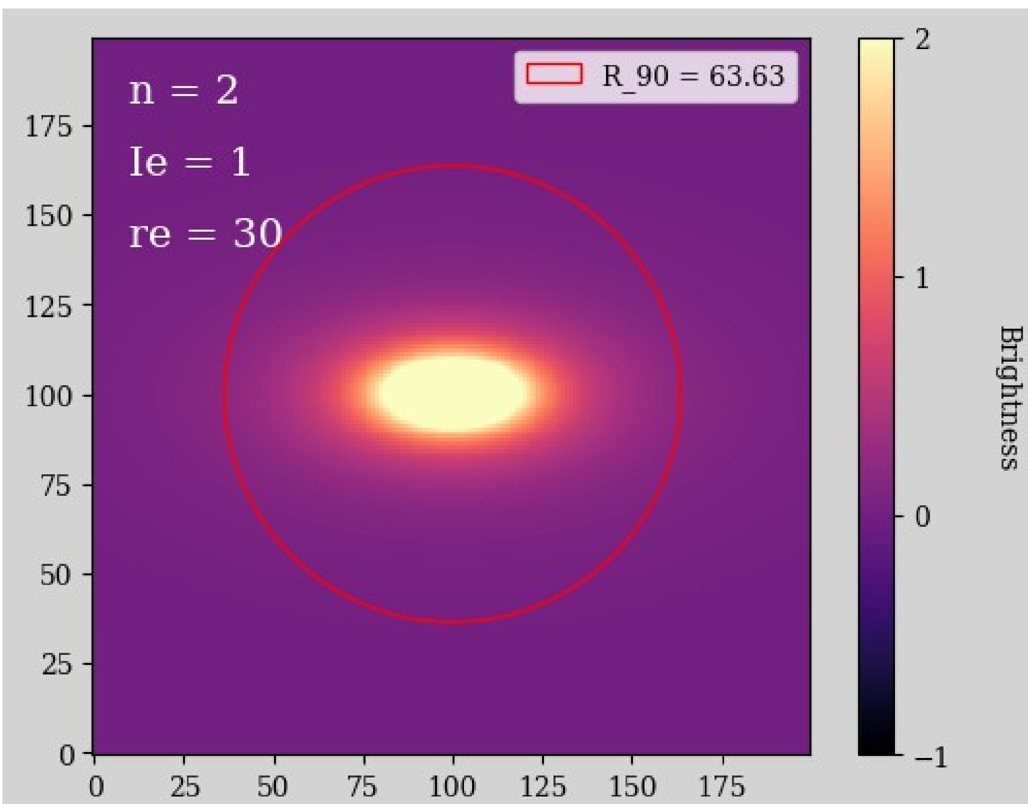


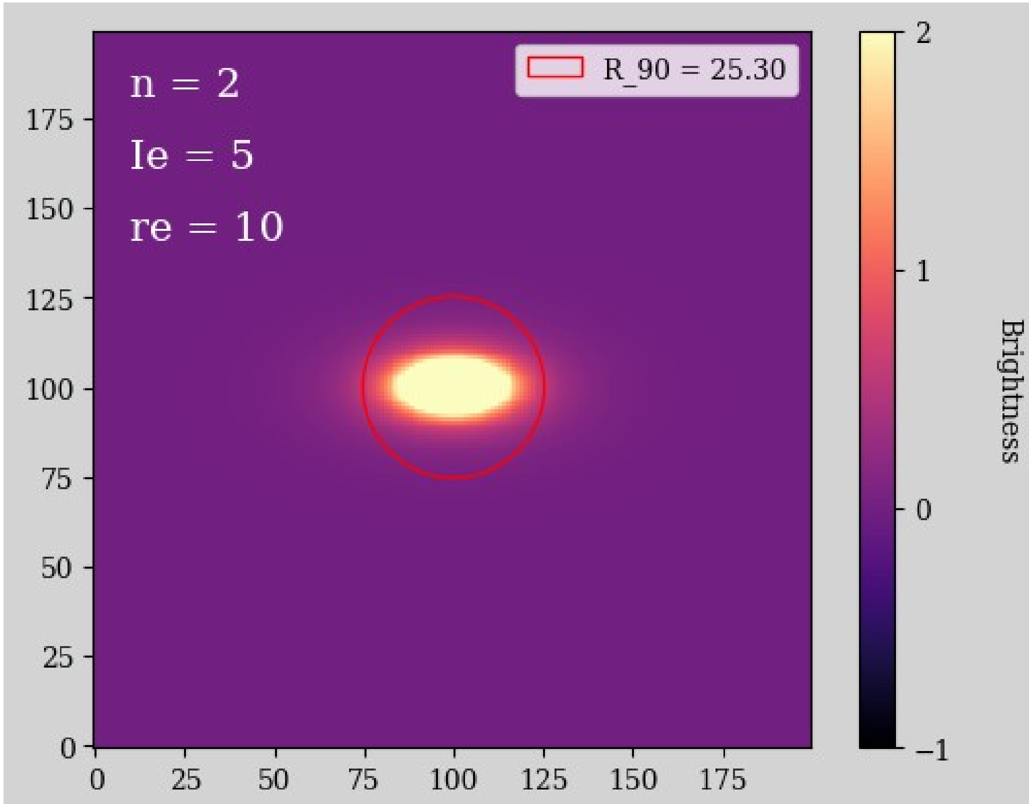
n = 1
Ie = 10
re = 30
R_90 = 53.71
Brightness



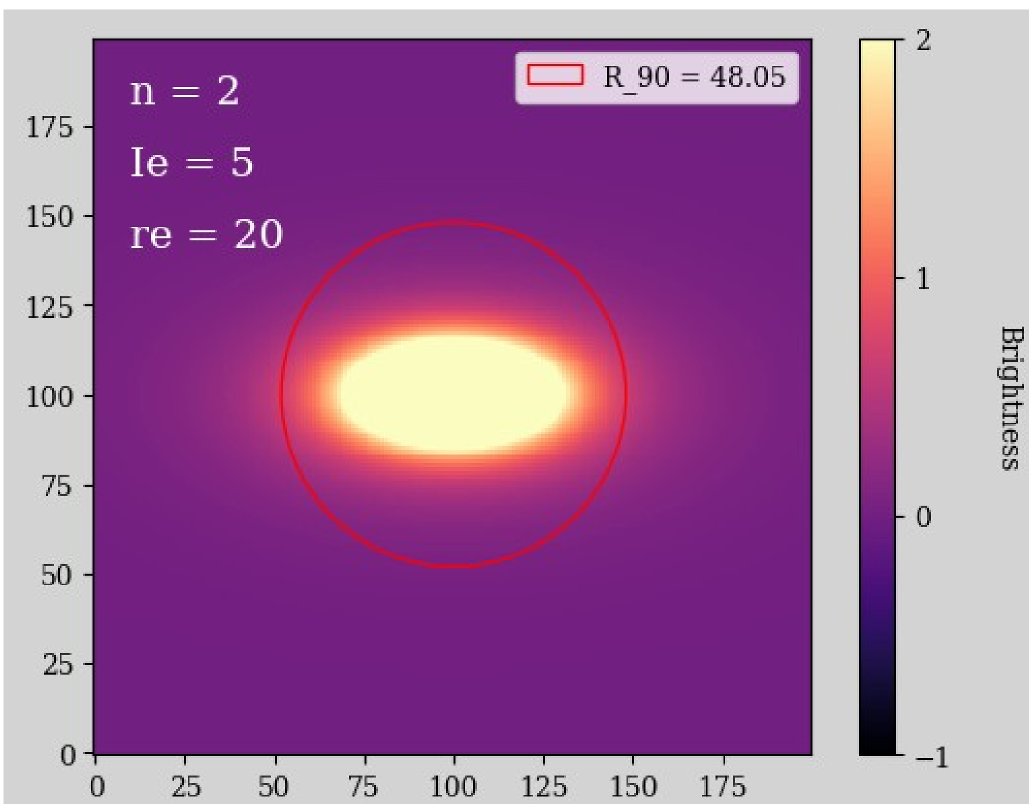
n = 2
Ie = 1
re = 10
R_90 = 25.30
Brightness


n = 2
Ie = 1
re = 20
R_90 = 48.05
Brightness

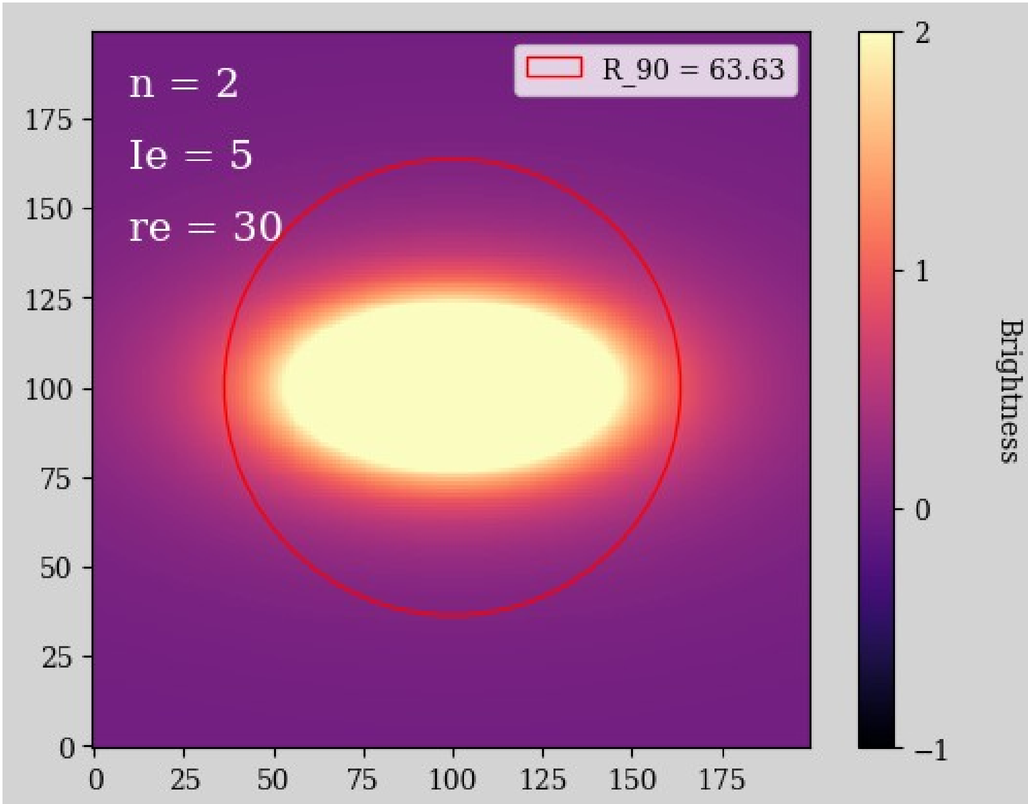

n = 2
Ie = 1
re = 30
R_90 = 63.63
Brightness



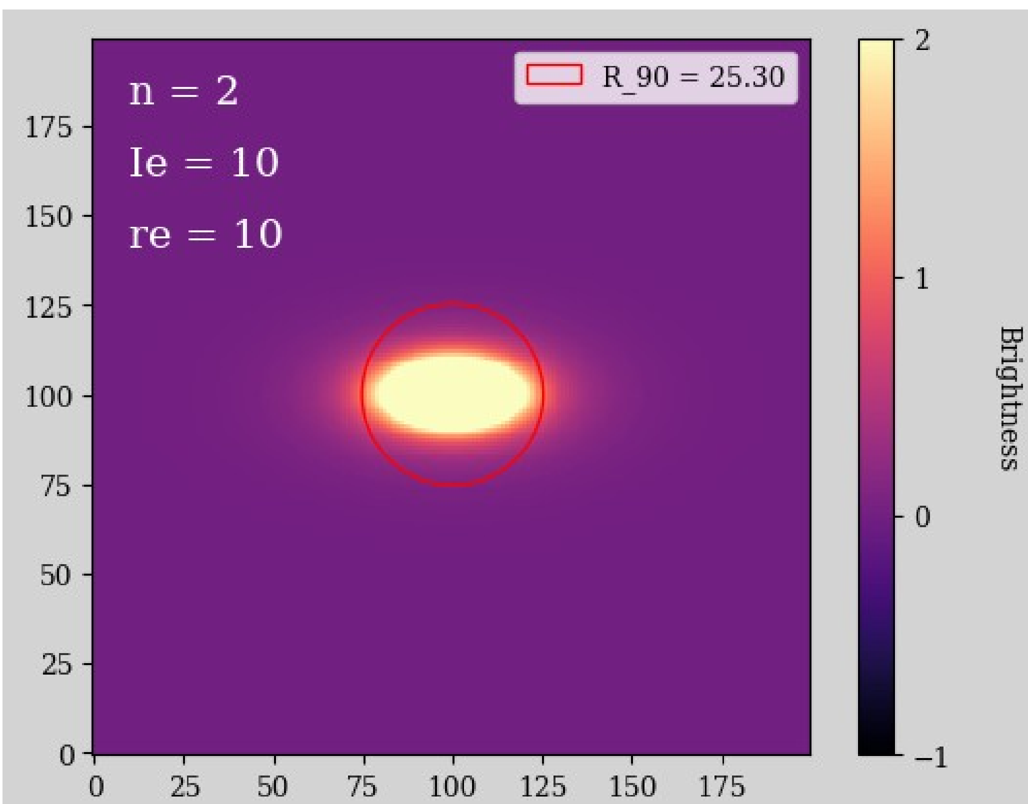
n = 2
Ie = 5
re = 10
R_90 = 25.30
Brightness



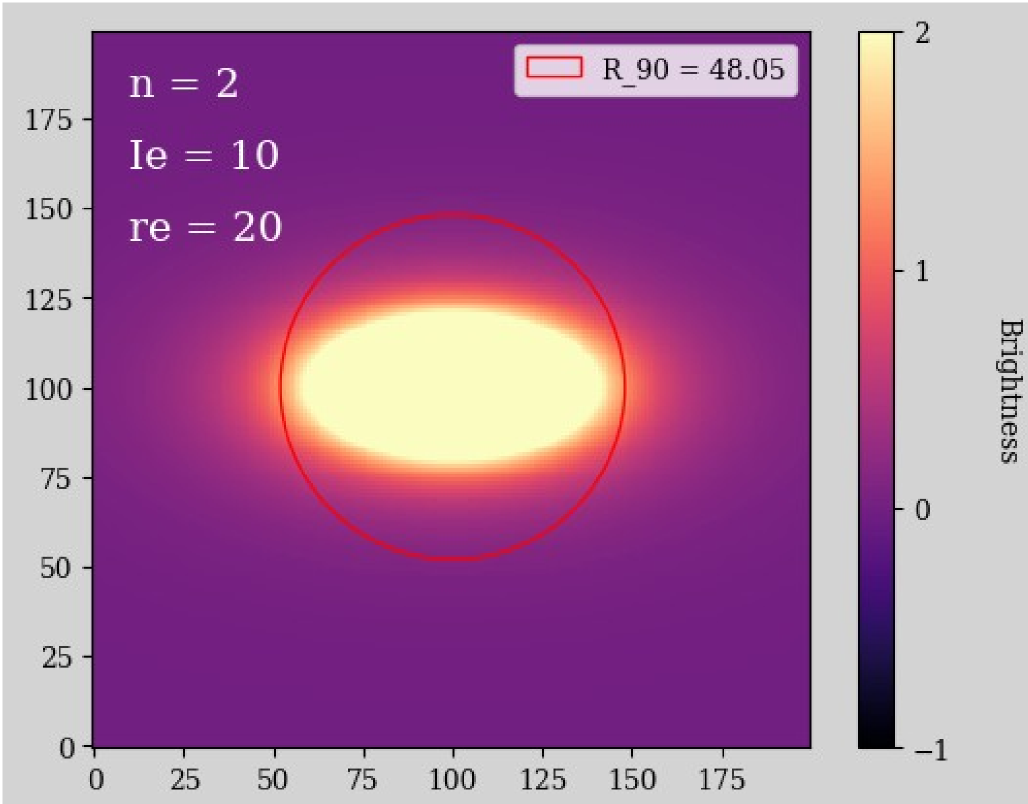
n = 2
Ie = 5
re = 20
R_90 = 48.05
Brightness

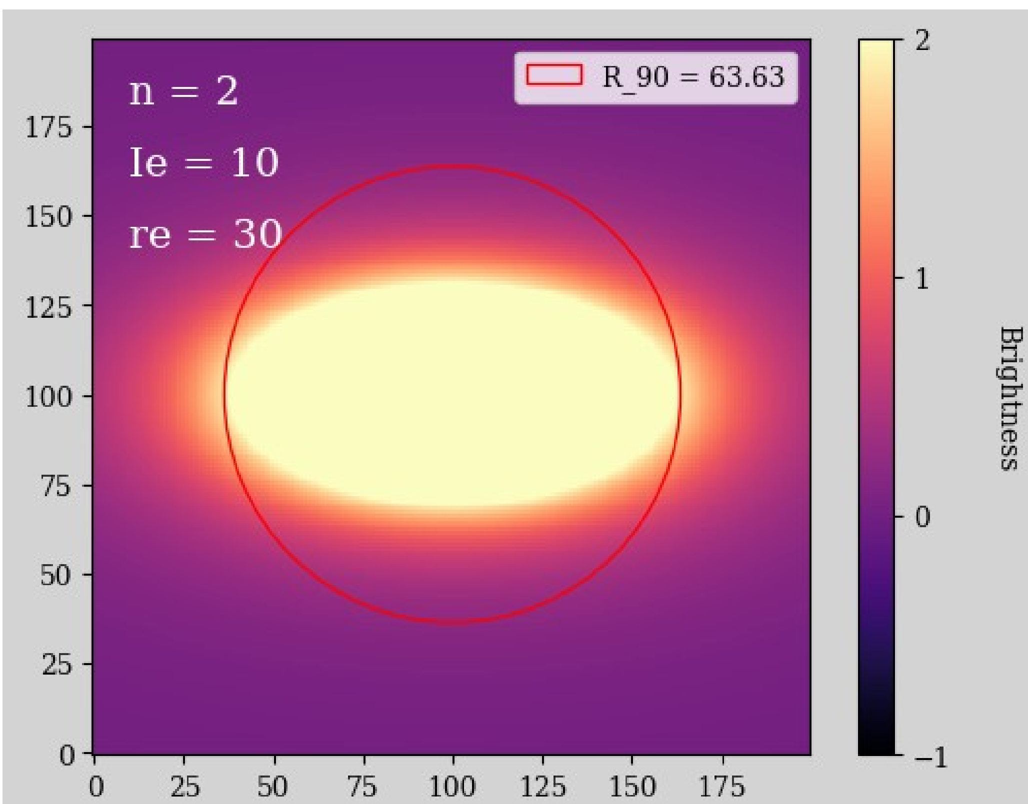


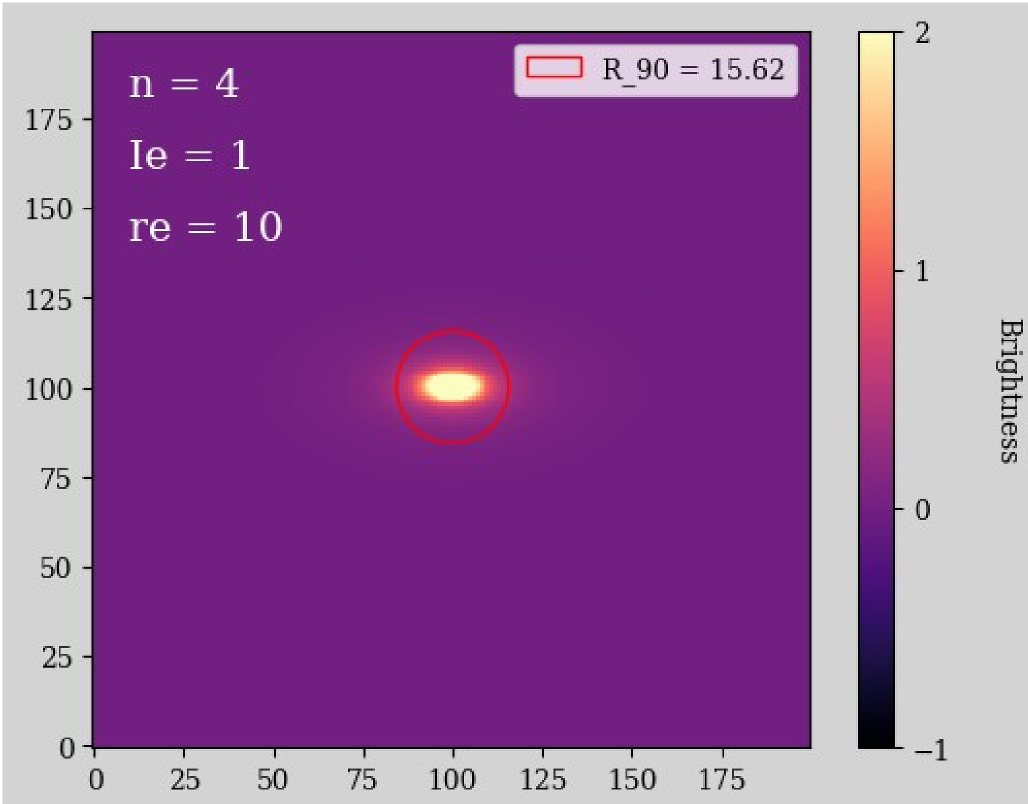
R_90 = 63.63
n = 2
Ie = 5
re = 30
Brightness

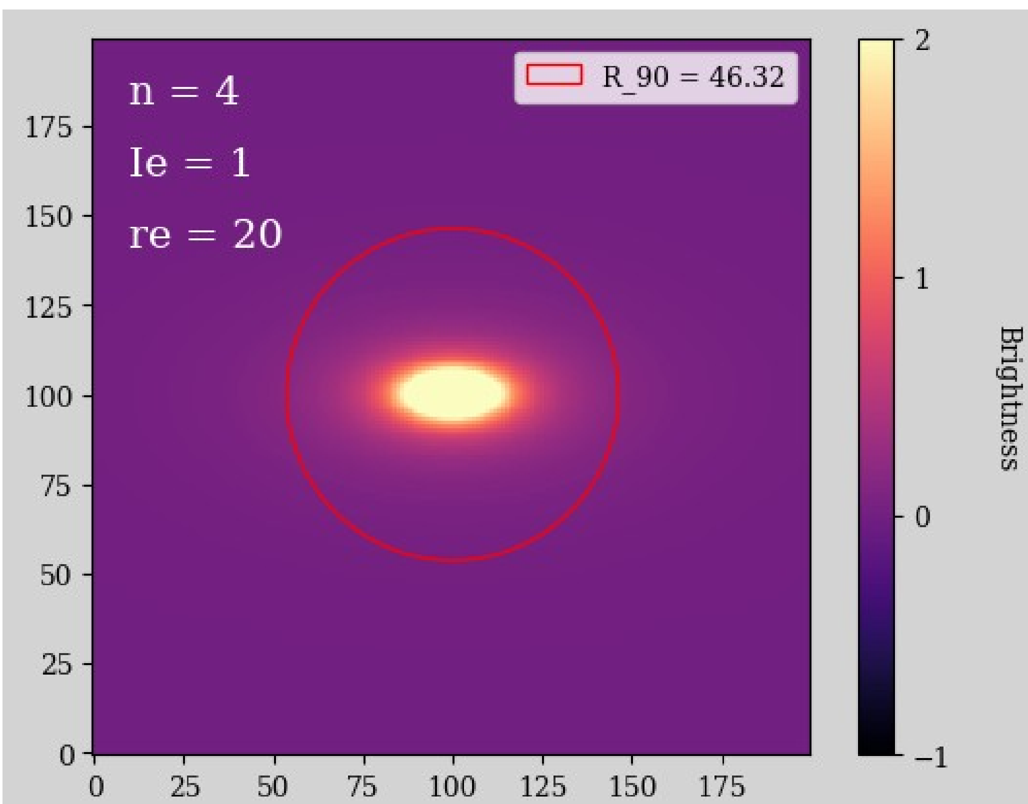


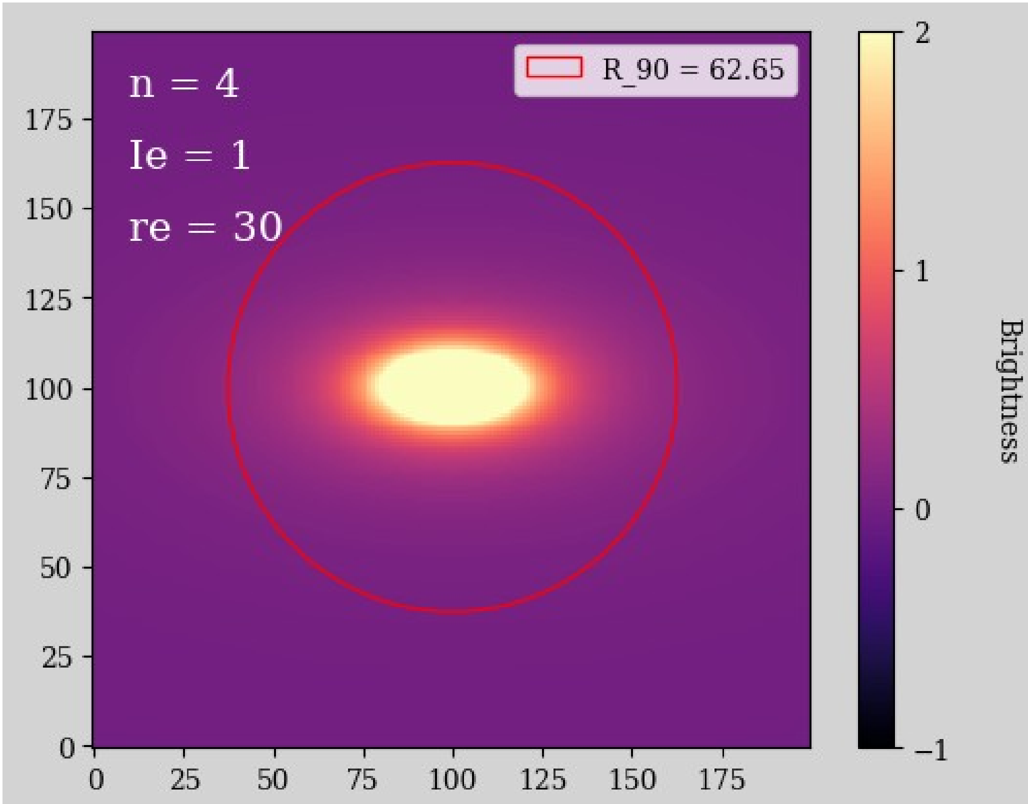
R_90 = 25.30
n = 2
Ie = 10
re = 10
Brightness


n = 2
Ie = 10
re = 20
R_90 = 48.05
Brightness

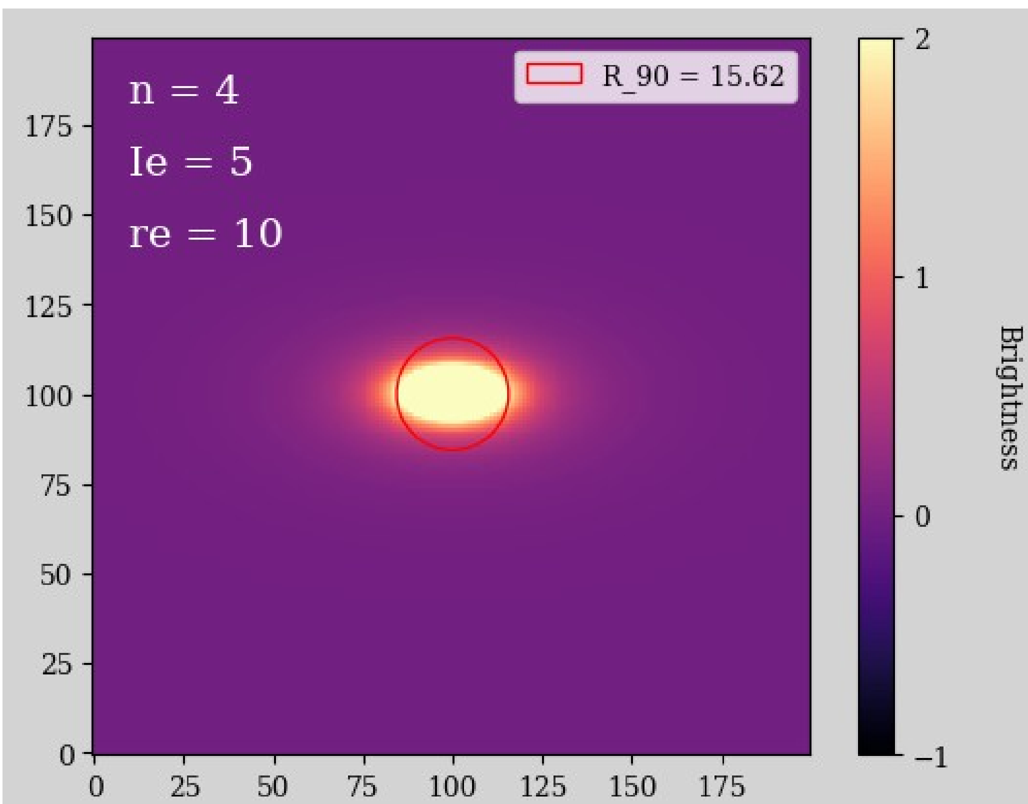

n = 2
Ie = 10
re = 30
R_90 = 63.63
Brightness


n = 4
Ie = 1
re = 10
R_90 = 15.62
Brightness
2
1
0
−1
175
150
125
100
75
50
25
0
0
25
50
75
100
125
150
175

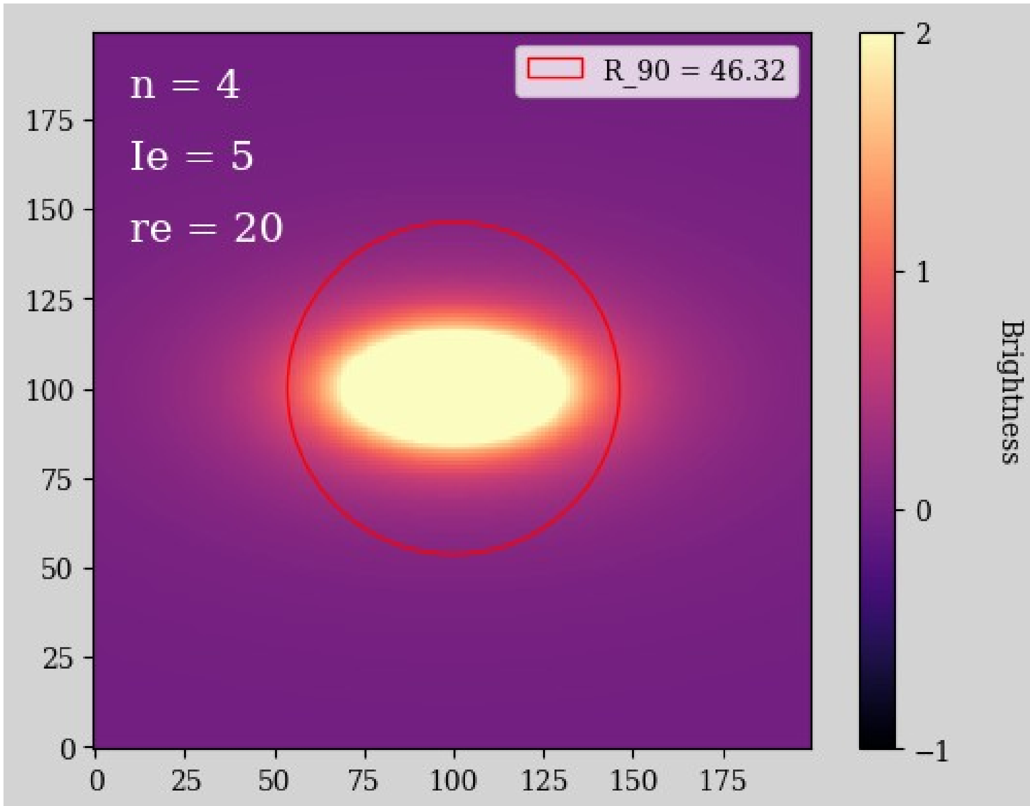

n = 4
Ie = 1
re = 20
R_90 = 46.32
Brightness
2
1
0
−1
175
150
125
100
75
50
25
0
0
25
50
75
100
125
150
175

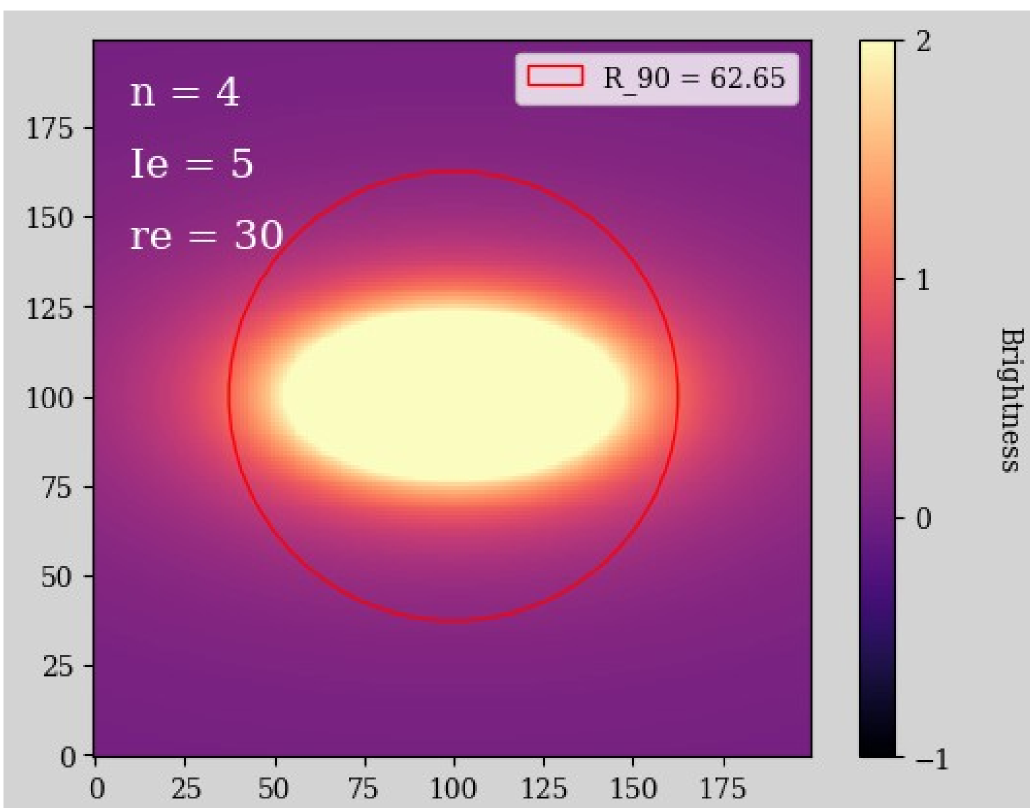


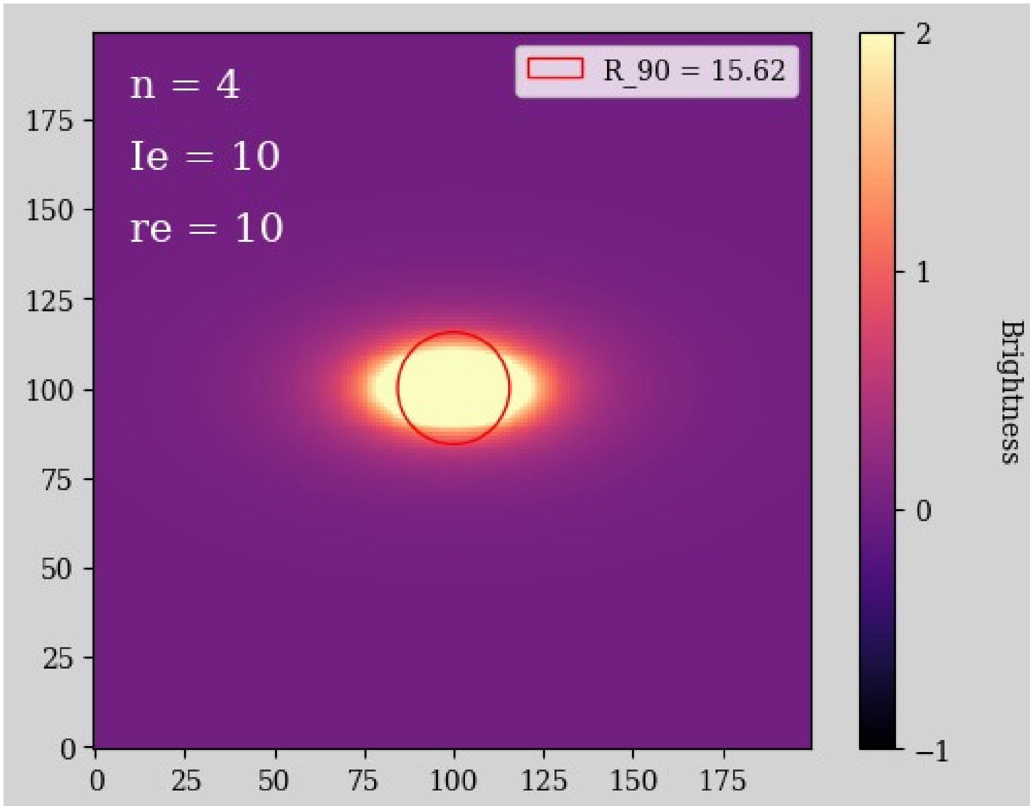
n = 4
Ie = 1
re = 30
R_90 = 62.65
Brightness



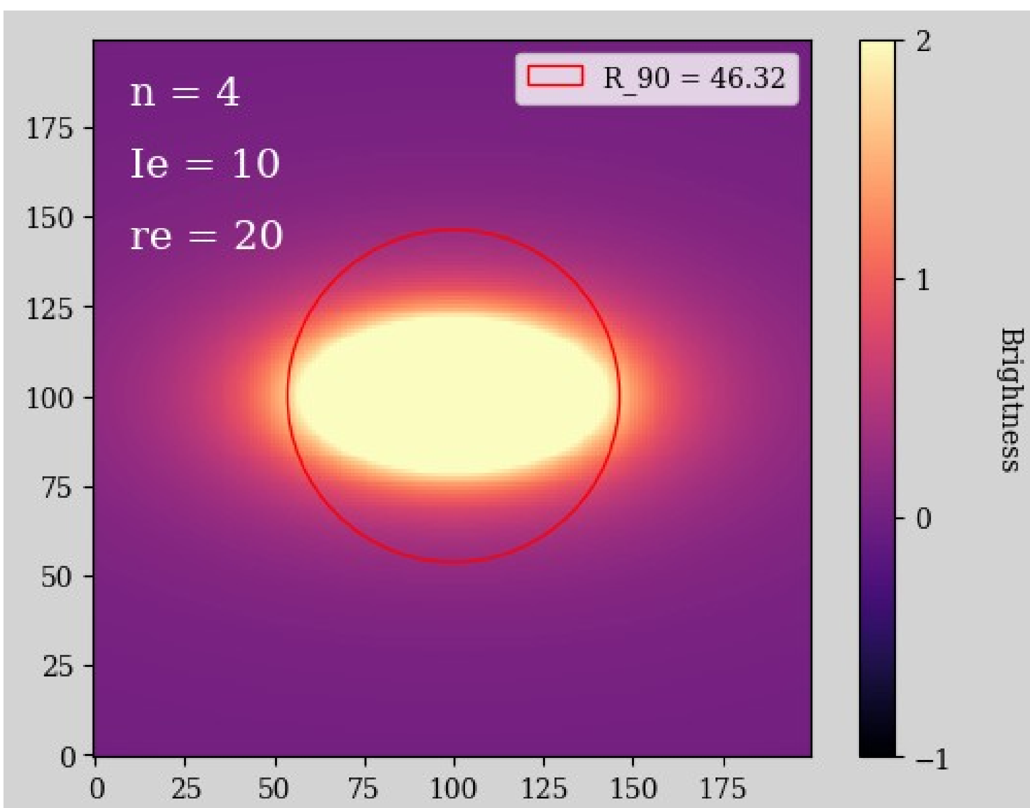
n = 4
Ie = 5
re = 10
R_90 = 15.62
Brightness


R_90 = 46.32
n = 4
Ie = 5
re = 20
Brightness

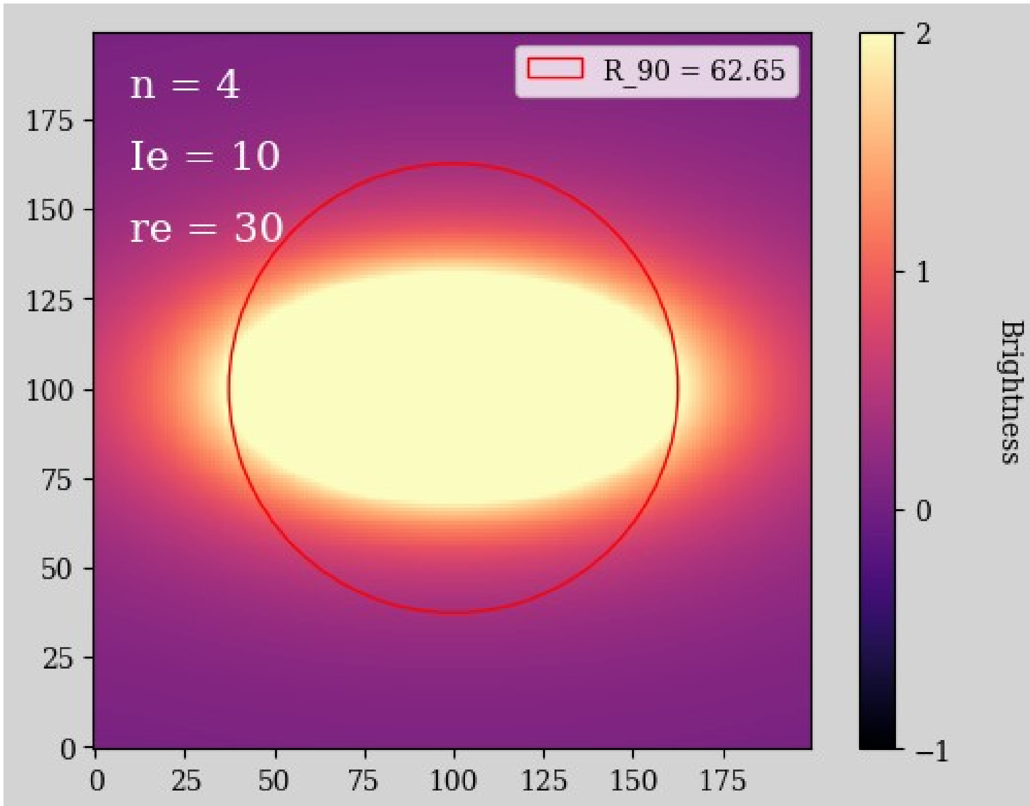

R_90 = 62.65
n = 4
Ie = 5
re = 30
Brightness



R_90 = 15.62
n = 4
Ie = 10
re = 10
Brightness



R_90 = 46.32
n = 4
Ie = 10
re = 20
Brightness



n = 4
Ie = 10
re = 30
R_90 = 62.65
Brightness
0
25
50
75
100
125
150
175
2
1
0
−1