

# Simulación Interactiva de Lentes Gravitacionales

Modelo SIS + Shear Externo y Trazado de Rayos Inverso

Documentación del Código

7 de enero de 2026

## Resumen

Este documento describe la implementación computacional de una simulación de lentes gravitacionales fuertes utilizando Python. El código emplea la técnica de trazado de rayos inverso (*Inverse Ray-Shooting*) para modelar la distorsión de una galaxia fuente extendida bajo la influencia de una Esfera Isotérmica Singular (SIS) y perturbaciones de cizallamiento externo (*Shear*). Se detalla el modelo físico, las ecuaciones matemáticas empleadas y la estructura de la interfaz interactiva desarrollada.

## 1. Introducción

El programa desarrollado es una herramienta interactiva que permite visualizar en tiempo real los efectos de las lentes gravitacionales fuertes. Utiliza bibliotecas científicas estándar (`numpy`, `matplotlib`) para calcular y renderizar la imagen distorsionada de una fuente de fondo al variar los parámetros del potencial gravitatorio y la morfología de la fuente.

## 2. Modelo Físico y Matemático

### 2.1. Ecuación de la Lente

La simulación se basa en la ecuación de la lente gravitacional, que mapea coordenadas del plano de la imagen ( $\vec{\theta}$ ) al plano de la fuente ( $\vec{\beta}$ ):

$$\vec{\beta} = \vec{\theta} - \vec{\alpha}(\vec{\theta}) \quad (1)$$

donde  $\vec{\alpha}(\vec{\theta})$  es el ángulo de deflexión total en la posición  $\vec{\theta}$ . El código implementa el mapeo inverso: para cada píxel en la malla de la imagen, se calcula la posición correspondiente en el plano fuente para determinar su brillo superficial.

### 2.2. Potencial de la Lente (Deflexiones)

El potencial gravitatorio total se modela como la suma de dos componentes: una lente principal y una perturbación externa.

#### 2.2.1. Esfera Isotérmica Singular (SIS)

La lente principal se modela como una distribución de masa SIS. El ángulo de deflexión tiene una magnitud constante igual al radio de Einstein  $\theta_E$  y apunta radialmente hacia el centro de la lente ( $\theta_{xc}, \theta_{yc}$ ):

$$\vec{\alpha}_{SIS}(\vec{\theta}) = \theta_E \frac{\vec{\theta} - \vec{\theta}_c}{|\vec{\theta} - \vec{\theta}_c|} \quad (2)$$

En el código, se evita la singularidad numérica en el centro añadiendo un término  $\epsilon$  pequeño al denominador.

### 2.2.2. Cizallamiento Externo (Shear)

Para simular la influencia de masas externas, se añade un término de cizallamiento  $\gamma$  con un ángulo de posición  $\phi_\gamma$ . Las componentes del cizallamiento se calculan como:

$$\gamma_1 = \gamma \cos(2\phi_\gamma) \quad (3)$$

$$\gamma_2 = \gamma \sin(2\phi_\gamma) \quad (4)$$

El ángulo de deflexión debido al shear es:

$$\vec{\alpha}_{shear}(\vec{\theta}) = \begin{pmatrix} \gamma_1 & \gamma_2 \\ \gamma_2 & -\gamma_1 \end{pmatrix} \cdot \vec{\theta} \quad (5)$$

## 2.3. Perfil de Brillo de la Fuente

La galaxia fuente se modela utilizando un **Perfil de Sersic**, que permite describir una amplia variedad de morfologías galácticas. La intensidad  $I(R)$  viene dada por:

$$I(R) = I_0 \exp \left( -b_n \left[ \left( \frac{R}{R_e} \right)^{1/n} - 1 \right] \right) \quad (6)$$

donde:

- $R_e$  es el radio efectivo.
- $n$  es el índice de Sersic (e.g.,  $n = 1$  para discos exponenciales,  $n = 4$  para perfiles de De Vaucouleurs).
- $b_n \approx 2n - 1/3 + \frac{0,009876}{n}$  es una constante de normalización.

El código implementa este perfil permitiendo **elipticidad** ( $q = b/a$ ) y **rotación** ( $\phi_{src}$ ) mediante una transformación de coordenadas en el plano fuente.

## 3. Implementación Computacional

### 3.1. Generación de Malla y Trazado de Rayos

El script define un campo de visión (FOV) y crea una malla bidimensional (`meshgrid`) de  $N \times N$  píxeles. El algoritmo principal (`render`) sigue estos pasos:

1. Calcula las componentes del shear ( $\gamma_1, \gamma_2$ ) basadas en el ángulo  $\phi_\gamma$ .
2. Calcula la deflexión  $\vec{\alpha}_{SIS}$  y  $\vec{\alpha}_{shear}$  para cada punto de la malla.
3. Aplica la ecuación de la lente para obtener las coordenadas distorsionadas en el plano fuente ( $\beta_x, \beta_y$ ).
4. Evalúa el perfil de Sersic en las coordenadas ( $\beta_x, \beta_y$ ).
5. Añade ruido gaussiano  $\mathcal{N}(0, \sigma_{noise})$  para simular condiciones observacionales reales.

### 3.2. Curvas Críticas

Para el modelo SIS, la curva crítica tangencial es un círculo de radio  $\theta_E$ . El código calcula y dibuja esta curva paramétricamente:

$$x(\varphi) = \theta_E \cos(\varphi), \quad y(\varphi) = \theta_E \sin(\varphi) \quad (7)$$

Esto permite visualizar la región donde la magnificación teórica diverge.

## 4. Interfaz Gráfica Interactiva

La herramienta incorpora una interfaz de usuario (UI) construida con `matplotlib.widgets` que permite la modificación dinámica de los parámetros.

### 4.1. Controles Disponibles

Se han implementado deslizadores (*Sliders*) para controlar las siguientes variables físicas:

- **Lente:** Radio de Einstein ( $\theta_E$ ), Magnitud del Shear ( $\gamma$ ), Ángulo del Shear ( $PA_\gamma$ ).
- **Fuente (Geometría):** Posición ( $\beta_x, \beta_y$ ), Radio Efectivo ( $R_e$ ).
- **Fuente (Morfología):** Índice de Sersic ( $n$ ), Relación Axial ( $q$ ), Ángulo de Posición ( $PA_{src}$ ).
- **Instrumental:** Nivel de Ruido ( $\sigma$ ).

Además, se incluye un botón de reinicio (*Reset*) y una casilla de verificación (*CheckButton*) para alternar la visibilidad de la curva crítica sobre la imagen.

## 5. Conclusión

Este código proporciona una plataforma robusta y didáctica para el estudio de lentes gravitacionales. La implementación modular de los perfiles de deflexión y brillo permite futuras expansiones, como la inclusión de perfiles de masa más complejos (e.g., NFW, PEMD) o múltiples fuentes.