

## Verano de la ciencia 2022 \ Universidad de Guanajuato \ Proyecto: ¡Fotografía de un agujero negro!

Docentes a cargo:

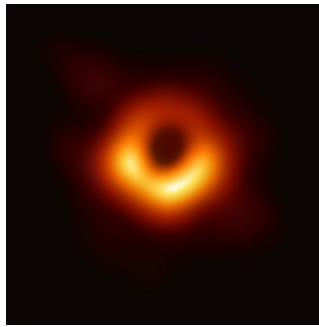
- Dr. Gustavo Niz Quevedo
- Dr. Gustavo Gutiérrez Cano

Equipo:

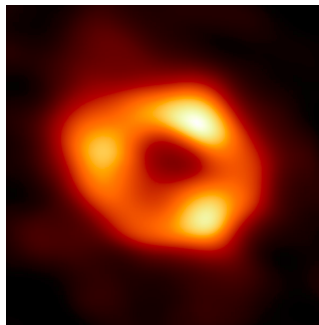
- Fátima Pamela López Salcedo
- Martha Nava Hernández
- Gustavo Andres Concha Valdez

Cuadernillo disponible en el repositorio abierto <https://github.com/DCI-UG/veranoug2022-pamemargus> (<https://github.com/DCI-UG/veranoug2022-pamemargus>)

#Introducción La atención de las personas volvió a centrarse en los agujeros negros en abril de 2019, cuando, por medio del Event Horizon Telescope (EHT), se publicó la primera imagen de la sombra correspondiente a uno de estos objetos en el centro de la galaxia vecina M87.



Recientemente, en mayo del 2022 se reveló la sombra del agujero negro en el centro de nuestra propia galaxia, Sagitario A\*.



La obtención de estas conocidas capturas es fruto en una verdadera proeza científica y tecnológica. No solo fortalece la confianza en el quehacer investigativo, sino que abre las puertas a la consolidación de una cultura científica en la población.

Al hablar de un agujero negro a la mayoría de las personas se les viene a la mente una esfera completamente negra, capaz de atraer todo a su alrededor y engullirlo sin dejar rastro: una cárcel gravitacional absoluta. La idea, que el día de hoy se respalda en evidencia visual, puede pulirse haciendo una revisión de lo que las matemáticas predicen que es susceptible de observarse: la sombra del agujero negro.

Para una mayor rigurosidad en la descripción de estos fascinantes objetos es imperativo recurrir a las herramientas matemáticas que permiten su caracterización. En el presente informe se busca describir teóricamente la observación de un agujero negro, como también plantear un programa capaz de dibujar su sombra en función de las cantidades que lo determinan.

#Marco teórico

## ## ¿Qué es y qué caracteriza a un agujero negro?

Se trata de una región finita del espacio-tiempo en donde se concentra una gran cantidad de masa, la cual genera un campo gravitatorio tan descomunadamente grande que no existe nada capaz de escapar de él, ni siquiera la luz. El concepto de agujero negro es resultado de una de las primeras soluciones a las ecuaciones de campo de la teoría de la relatividad general de A. Einstein, hallada por el físico alemán Karl Schwarzschild.

Por otro lado, los agujeros negros pueden ser clasificados gracias al teorema de No-Pelo. Este teorema establece que un agujero negro queda completamente caracterizado por únicamente tres parámetros: masa, momento angular y carga eléctrica/magnética. Así, aquel que solo posee masa es conocido como agujero negro de Schwarzschild; es el modelo más sencillo. Un agujero negro de Kerr posee masa y momento angular. Si tiene masa, momento angular y carga eléctrica, se trata de un agujero negro de Kerr-Newman. Aunque en este trabajo no se detallará el caso, existe también el agujero negro de Reissner-Nordström, con masa y carga eléctrica. En el trabajo se desarrollará un modelo adicional, conocido como Bardeen-rotante, que es producto de una masa rotante y carga magnética.

Es importante mencionar ciertas partes que componen un agujero negro y que serán relevantes para el trabajo. Posee un disco de acreción, que se trata de un conjunto de gas y polvo a altísimas temperaturas que orbita al objeto. Existe también la esfera de luz, que es aquella región a partir de la cual la luz ya no es capaz de escapar de la atracción gravitacional, marcando así la llamada sombra del agujero negro. La siguiente capa se trata del horizonte de eventos, a partir del cual no es posible escapar, ni siquiera acelerando. Esta región determina un punto de no retorno.

## ## Análisis matemático

### ### Métrica de Schwarzschild

La métrica de Schwarzschild es la primera solución exacta hallada para las ecuaciones de campo de Einstein. Se trata de una solución estática y con simetría esférica que describe el campo gravitatorio alrededor de una concentración de masa dados una carga eléctrica y momento angular nulos.

Si se trabaja con unidades tales que  $c = 1$ , la solución de Schwarzschild para una masa  $M$  se resume en el elemento de línea

$$ds^2 = \left(1 - \frac{r_S}{r}\right) dt^2 - \left(1 - \frac{r_S}{r}\right)^{-1} dr^2 - r^2 (d\theta^2 + \sin^2\theta d\phi^2),$$

en donde el radio de Schwarzschild está dado por  $r_S = 2GM$ , siendo  $G$  la constante de Newton. Puesto que el tamaño de un agujero negro depende del contenido que ha absorbido, cuanto mayor es la masa del agujero negro, mayor es el radio de Schwarzschild. Es precisamente este radio el que determina el horizonte de eventos.

### Singularidad Existen dos puntos de interés en la métrica de Schwarzschild, que aparentan acarrear problemas en las definiciones. Uno de ellos es  $r = r_S$ , para el cual

$$\left(1 - \frac{r_S}{r}\right) \rightarrow 0, \quad \left(1 - \frac{r_S}{r}\right)^{-1} \rightarrow \infty.$$

Aunque tal comportamiento parece implicar consecuencias físicas, mediante un cambio de coordenadas adecuado es posible demostrar que este valor no implica una singularidad. Sin embargo, sí es posible hacer un análisis de este punto: una vez que se cruza dicho valor, en la región  $r < r_S$ , la parte temporal es negativa, mientras que la parte radial, positiva. Esto se describe con cierta frecuencia como un intercambio entre los roles que juegan tiempo y espacio.

Por otro lado, el punto  $r = 0$ , para el cual

$$\left(1 - \frac{r_S}{r}\right) \rightarrow \infty, \quad \left(1 - \frac{r_S}{r}\right)^{-1} \rightarrow 0,$$

sí representa una singularidad física. Lo que sucede en dicho punto suele describirse como una curvatura hasta el infinito de tiempo y espacio, aunque se trata de especulación: realmente, no está del todo claro qué ocurre en la singularidad.

### La sombra de un agujero negro con una métrica más general El campo gravitacional de un agujero negro es capaz de desviar la luz que pasa a su alrededor. Dependiendo de las características del objeto, esta luz "doblada" producirá una sombra particular, la cual se capta desde la Tierra, gracias al EHT. Es necesario modelar al observador en un espacio asintóticamente plano, en tanto se encuentra en una región lo suficientemente lejana del agujero negro. Para esto, resulta conveniente definir las llamadas coordenadas celestes,  $\alpha$  y  $\beta$ .

Veamos cómo se traduce esto en lenguaje matemático. Se recurre a la métrica general

$$ds^2 = \frac{a^2 \sin^2 \theta}{\rho^2} \left( dt^2 - \frac{r^2 + a^2}{a} d\phi \right)^2 + \frac{\rho^2}{Q} dr^2 + \rho^2 d\theta^2 - \frac{Q}{\rho} (dt - a \sin^2 \theta d\phi)^2,$$

en donde  $a = J/M$  es un término asociado al momento angular del agujero negro rotante y  $\rho$  se expresa como

$$\rho = r^2 + a^2 \cos^2 \theta,$$

con  $M$  la masa del agujero negro.

La definición de  $Q$  determinará los distintos tipos de agujero negro:

- Kerr: caracterizado por poseer masa y momento angular

$$Q = r^2 - 2Mr + a^2$$

- Kerr-Newman: posee masa, momento angular y carga eléctrica

$$Q = r^2 - 2Mr + a^2 + q^2$$

- Bardeen-rotante: posee masa, momento angular y carga magnética

$$Q = r^2 + a^2 - \frac{2Mr^4}{(r^2 + g^2)^{3/2}}$$

Desde luego, si el agujero negro no tiene rotación, al  $a = 0$  se recupera la solución estática de Schwarzschild.

###Ecuaciones geodésicas Para el caso general, es necesario tener en cuenta que las componentes inversas de la métrica se forman según

$$\begin{aligned} g^{tt} &= -\frac{1}{\rho^2 Q} [(r^2 + a^2)^2 - a^2 Q \sin^2 \theta] \\ g^{rr} &= \frac{Q}{\rho^2} \\ g^{\theta\theta} &= \frac{1}{\rho^2} \\ g^{\phi\phi} &= \frac{Q - a^2 \sin^2 \theta}{\rho^2 Q \sin^2 \theta} \\ g^{t\phi} &= \frac{a}{\rho^2 Q} [Q - (r^2 + a^2)] \end{aligned}$$

Luego, con ayuda de la ecuación de Hamilton-Jacobi se puede obtener las ecuaciones geodésicas:

$$\begin{aligned} \rho^2 \dot{t} &= \frac{1}{Q} \{ [(r^2 + a^2)^2 - a^2 Q \sin^2 \theta] E + a [Q - (r^2 + a^2)] L_z \}, \\ \rho^2 \dot{\phi} &= \frac{1}{Q} \left\{ \frac{Q - a^2 \sin^2 \theta}{\sin^2 \theta} L_z - a [Q - (r^2 + a^2)] E \right\}, \\ \rho^2 \dot{r} &= \sqrt{\mathcal{R}(r)}, \\ \rho^2 \dot{\theta} &= \sqrt{\Theta(\theta)}, \end{aligned}$$

en donde  $L_z$  es la componente  $z$  del momento angular y las funciones  $\mathcal{R}(r)$  y  $\Theta(\theta)$  se definen como

$$\begin{aligned} \mathcal{R}(r) &= [(r^2 + a^2)E - aL_z]^2 - Q [m^2 r^2 + \mathcal{K} + (L_z - aE)^2] \\ \Theta(\theta) &= \mathcal{K} \cos^2 \theta [(E^2 - m^2)a^2 - L_z^2 \csc^2 \theta] \end{aligned}$$

###Geodésicas nulas: trayectoria de la luz Ya que el interés radica en la imagen generada por la observación del agujero negro en cuestión, el análisis se debe centrar en las características que tiene la propagación de la luz. Como es sabido, la luz viaja siguiendo geodésicas nulas. Las ecuaciones de movimiento para un haz luminoso pueden determinarse haciendo uso nuevamente de las ecuaciones de Hamilton-Jacobi, considerando que  $m = 0$  sería lo correspondiente a un fotón luminoso.

Haciendo  $m = 0$ ,  $\xi = L_z/E$ ,  $\eta = \mathcal{K}/E^2$  y cambiando el parámetro de la curva por  $\lambda_* = E\lambda$ , se tiene que

$$\begin{aligned} \rho^2 \dot{t} &= \frac{1}{Q} [(r^2 + a^2)^2 - a^2 Q \sin^2 \theta + a (Q - r^2 - a^2) \xi], \\ \rho^2 \dot{\phi} &= \frac{1}{Q} \left[ \frac{Q - a^2 \sin^2 \theta}{\sin^2 \theta} \xi - a (Q - r^2 - a^2) \right], \\ \rho^2 \dot{r} &= \sqrt{\mathcal{R}(r)}, \\ \rho^2 \dot{\theta} &= \sqrt{\Theta(\theta)}, \end{aligned}$$

en donde

$$\begin{aligned} \mathcal{R}(r) &= (r^2 + a^2 - a\xi)^2 - Q [\eta + (\xi - a)^2] \\ \Theta(\theta) &= \eta + \cos^2 \theta (a^2 - \xi^2 \csc^2 \theta) \end{aligned}$$

Las órbitas que son de interés son las de equilibrio inestable, es decir, las que delimitan un valor de la coordenada radial tal que un fotón describe varias vueltas alrededor del agujero negro, para luego ser capturado o expulsado hacia el infinito. Ya que la expresión  $\mathcal{R} = 0$  determina los puntos de retorno en la trayectoria de los fotones, para hallar tal órbita basta con

buscar las soluciones de

$$\mathcal{R} = \frac{d\mathcal{R}}{dr} = 0$$

Este sistema da como resultado los valores de interés para  $\xi$  y  $\eta$ , dados a continuación:

$$\xi = \frac{(r^2 + a^2)Q' - 4Qr}{aQ'}$$

$$\eta = \frac{r^2 [16Q(a^2 - Q) + 8QQ'r - Q'^2 r^2]}{a^2 Q'^2}$$

Como se mencionaba, la forma más práctica de representar la sombra del agujero negro viene dada por el uso de las coordenadas celestes  $\alpha$  y  $\beta$ . Considerando a un observador ubicado en un espacio asintóticamente plano, en  $(r_0, \theta_0, 0)$ , estas coordenadas se definen como

$$\alpha = \lim_{r_0 \rightarrow \infty} \left( -r_0^2 \sin \theta_0 \frac{d\phi}{dr} \right)$$

$$\beta = \lim_{r_0 \rightarrow \infty} r_0^2 \frac{d\theta}{dr}$$

Si se utiliza las relaciones anteriormente escritas, las coordenadas celestes quedan dadas por

$$\alpha = -\frac{\xi}{\sin \theta_0}$$

$$\beta = \pm \sqrt{\eta + a^2 \cos^2 \theta_0 - \xi^2 \cot^2 \theta_0}$$

Los fotones que son capaces de llegar al observador son aquellos para los cuales  $\beta$  queda bien definido. En ese sentido, los valores permitidos para la coordenada radial quedan determinados por el intervalo entre las raíces de la ecuación  $\beta^2 = 0$ .

En el presente trabajo se busca modelar un observador ubicado en  $\theta_0 = \pi/2$ . Con esto establecido, la ecuación que determinará el intervalo de valores de  $r$  es

$$\beta^2 = \eta = 0.$$

Por lo tanto, tomando en cuenta la expresión que define  $\eta$ , la ecuación de interés resulta ser

$$16Q(a^2 - Q) + 8QQ'r - Q'^2 r^2 = 0.$$

El programa que se planteará en este informe busca resolver numéricamente la ecuación mostrada. Una vez determinados los valores que puede asumir  $r$ , se procede a dibujar la sombra del agujero negro en un gráfico de  $\alpha$  vs  $\beta$ .

###Consideraciones adicionales Es necesario garantizar que los valores que asuma la dimensión radial queden fuera del horizonte de eventos, de lo contrario se estaría entrando en contradicción. A continuación el horizonte de eventos para distintos valores de  $a$  para los casos de Kerr ( $M = 1$ ) y Kerr-Newman ( $M = 1, q = 0.5$ ):

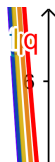
Kerr		Kerr-Newman	
$M + \sqrt{M^2 - a^2}$		$M + \sqrt{M^2 - a^2 - q^2}$	
a	$r_H$	a	$r_H$
0.1	1.994987437106619955	0.15	1.852936105461599
0.25	1.968245836551854221	0.50	1.7071067811865475

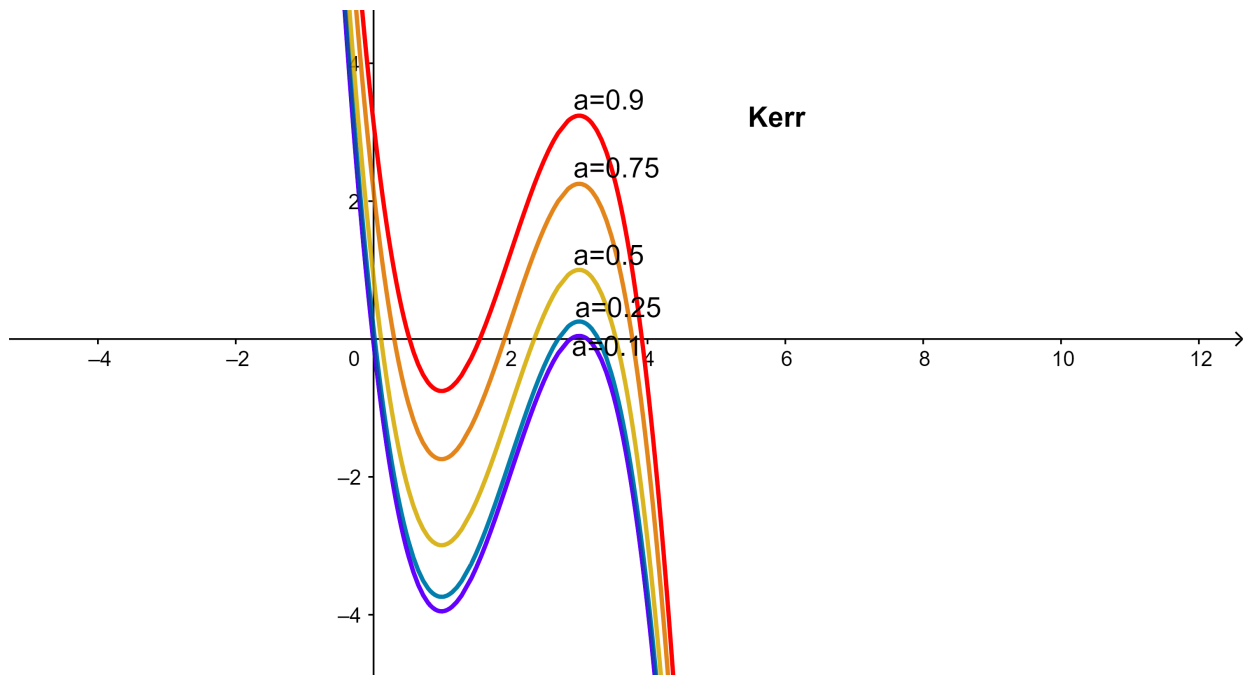
#Código para obtener la sombra del agujero negro

En esta primera parte se importan las librerías necesarias para la ejecución del programa:

```
In [ ]: 1 import numpy as np
        2 from matplotlib import pyplot as plt
```

##Kerr Se hará variaciones en el momento angular. Ya que se desea tener las raíces, se grafica  $16Q(a^2 - Q) + 8QQ'r - Q'^2 r^2$  para encontrar los valores límite de  $r$ .





Se comienza definiendo el valor de la masa  $M$ , el valor de  $a$ , asociado a la rotación del cuerpo, y el ángulo  $\theta$  con que se observa, que en adelante será siempre ecuatorial, como se había indicado. Los valores para la masa y el ángulo se mantendrán a lo largo del documento.

```
In [ ]: 1 M=1
        2 theta=np.pi/2
```

Se dividirá el código en varias partes, de manera que se pueda variar el parámetro  $a$ . Además, recurrentemente se utiliza la función `HallarRaices(A,B,nmax,tolerancia)`, la cual se trata de un método numérico para hallar raíces reales de funciones. Se usa el conocido método de la secante para tal fin. La función denominada `anul(r)` es, precisamente,

$$anul(r) = 16Q(a^2 - Q) + 8QQ'r - Q'^2r^2,$$

que tendrá diferentes formas dependiendo del tipo de agujero negro (que determina  $Q$ ).

In [ ]:

```
1 a1=0.1
2
3 def anul1(r):
4     Q=r**2-2*M*r+a1**2
5     Qp=2*r-2*M
6     return 16*Q*(a1**2-Q)+8*r*Q*Qp-r**2*Qp**2
7
8 def HallarRaices1(A,B,nmax,tolerancia):
9     fA=anul1(A)
10    fB=anul1(B)
11    if abs(fA)>abs(fB):
12        aux=B
13        auxF=fB
14        B=A
15        A=aux
16        fB=fA
17        fA=auxF
18    for n in range(2,nmax+1):
19        if abs(fA)>abs(fB):
20            aux=B
21            auxF=fB
22            B=A
23            A=aux
24            fB=fA
25            fA=auxF
26        d=(B-A)/(fB-fA)
27        B=A
28        fB=fA
29        d=d*fA
30        A=A-d
31        fA=anul1(A)
32        if abs(d)<tolerancia:
33            break
34    return A
35
36 rInic1=HallarRaices1(2.8,2.9,1000,0.0000001)
37 rFin1=HallarRaices1(3.1,3.2,1000,0.0000001)
38
39 def alfa1(r):
40     Q=r**2-2*M*r+a1**2
41     Qp=2*r-2*M
42     xi=((a1**2+r**2)*Qp-4*r*Q)/(a1*Qp)
43     return -xi/np.sin(theta)
44
45 def beta1(r):
46     Q=r**2-2*M*r+a1**2
47     Qp=2*r-2*M
48     xi=((a1**2+r**2)*Qp-4*r*Q)/(a1*Qp)
49     eta=(r**2/(a1**2*Qp**2))*(anul1(r))
50     return np.sqrt(eta+a1**2*(np.cos(theta))**2-xi**2/(np.tan(theta))**2)
```

In [ ]:

```
1 a2=0.25
2
3 def anul2(r):
4     Q=r**2-2*M*r+a2**2
5     Qp=2*r-2*M
6     return 16*Q*(a2**2-Q)+8*r*Q*Qp-r**2*Qp**2
7
8 def HallarRaices2(A,B,nmax,tolerancia):
9     fA=anul2(A)
10    fB=anul2(B)
11    if abs(fA)>abs(fB):
12        aux=B
13        auxF=fB
14        B=A
15        A=aux
16        fB=fA
17        fA=auxF
18    for n in range(2,nmax+1):
19        if abs(fA)>abs(fB):
20            aux=B
21            auxF=fB
22            B=A
23            A=aux
24            fB=fA
25            fA=auxF
26        d=(B-A)/(fB-fA)
27        B=A
28        fB=fA
29        d=d*fA
30        A=A-d
31        fA=anul2(A)
32        if abs(d)<tolerancia:
33            break
34    return A
35
36 rInic2=HallarRaices2(2.6,2.8,1000,0.0000001)
37 rFin2=HallarRaices2(3.2,3.4,1000,0.0000001)
38
39 def alfa2(r):
40     Q=r**2-2*M*r+a2**2
41     Qp=2*r-2*M
42     xi=((a2**2+r**2)*Qp-4*r*Q)/(a2*Qp)
43     return -xi/np.sin(theta)
44
45 def beta2(r):
46     Q=r**2-2*M*r+a2**2
47     Qp=2*r-2*M
48     xi=((a2**2+r**2)*Qp-4*r*Q)/(a2*Qp)
49     eta=(r**2/(a2**2*Qp**2))*(anul2(r))
50     return np.sqrt(eta+a2**2*(np.cos(theta))**2-xi**2/(np.tan(theta))**2)
```

In [ ]:

```
1 a3=0.5
2
3 def anul3(r):
4     Q=r**2-2*M*r+a3**2
5     Qp=2*r-2*M
6     return 16*Q*(a3**2-Q)+8*r*Q*Qp-r**2*Qp**2
7
8 def HallarRaices3(A,B,nmax,tolerancia):
9     fA=anul3(A)
10    fB=anul3(B)
11    if abs(fA)>abs(fB):
12        aux=B
13        auxF=fB
14        B=A
15        A=aux
16        fB=fA
17        fA=auxF
18    for n in range(2,nmax+1):
19        if abs(fA)>abs(fB):
20            aux=B
21            auxF=fB
22            B=A
23            A=aux
24            fB=fA
25            fA=auxF
26        d=(B-A)/(fB-fA)
27        B=A
28        fB=fA
29        d=d*fA
30        A=A-d
31        fA=anul3(A)
32        if abs(d)<tolerancia:
33            break
34    return A
35
36 rInic3=HallarRaices3(2.3,2.4,1000,0.0000001)
37 rFin3=HallarRaices3(3.4,3.6,1000,0.0000001)
38
39 def alfa3(r):
40     Q=r**2-2*M*r+a3**2
41     Qp=2*r-2*M
42     xi=((a3**2+r**2)*Qp-4*r*Q)/(a3*Qp)
43     return -xi/np.sin(theta)
44
45 def beta3(r):
46     Q=r**2-2*M*r+a3**2
47     Qp=2*r-2*M
48     xi=((a3**2+r**2)*Qp-4*r*Q)/(a3*Qp)
49     eta=(r**2/(a3**2*Qp**2))*(anul3(r))
50     return np.sqrt(eta+a3**2*(np.cos(theta))**2-xi**2/(np.tan(theta))**2)
```



In [ ]:

```
1 a4=0.75
2
3 def anul4(r):
4     Q=r**2-2*M*r+a4**2
5     Qp=2*r-2*M
6     return 16*Q*(a4**2-Q)+8*r*Q*Qp-r**2*Qp**2
7
8 def HallarRaices4(A,B,nmax,tolerancia):
9     fA=anul4(A)
10    fB=anul4(B)
11    if abs(fA)>abs(fB):
12        aux=B
13        auxF=fB
14        B=A
15        A=aux
16        fB=fA
17        fA=auxF
18    for n in range(2,nmax+1):
19        if abs(fA)>abs(fB):
20            aux=B
21            auxF=fB
22            B=A
23            A=aux
24            fB=fA
25            fA=auxF
26        d=(B-A)/(fB-fA)
27        B=A
28        fB=fA
29        d=d*fA
30        A=A-d
31        fA=anul4(A)
32        if abs(d)<tolerancia:
33            break
34    return A
35
36 rInic4=HallarRaices4(1.8,2.0,1000,0.0000001)
37 rFin4=HallarRaices4(3.6,4.0,1000,0.0000001)
38
39 def alfa4(r):
40     Q=r**2-2*M*r+a4**2
41     Qp=2*r-2*M
42     xi=((a4**2+r**2)*Qp-4*r*Q)/(a4*Qp)
43     return -xi/np.sin(theta)
44
45 def beta4(r):
46     Q=r**2-2*M*r+a4**2
47     Qp=2*r-2*M
48     xi=((a4**2+r**2)*Qp-4*r*Q)/(a4*Qp)
49     eta=(r**2/(a4**2*Qp**2))*(anul4(r))
50     return np.sqrt(eta+a4**2*(np.cos(theta))**2-xi**2/(np.tan(theta))**2)
```

In [ ]:

```
1 a5=0.9
2
3 def anul5(r):
4     Q=r**2-2*M*r+a5**2
5     Qp=2*r-2*M
6     return 16*Q*(a5**2-Q)+8*r*Q*Qp-r**2*Qp**2
7
8 def HallarRaices5(A,B,nmax,tolerancia):
9     fA=anul5(A)
10    fB=anul5(B)
11    if abs(fA)>abs(fB):
12        aux=B
13        auxF=fB
14        B=A
15        A=aux
16        fB=fA
17        fA=auxF
18    for n in range(2,nmax+1):
19        if abs(fA)>abs(fB):
20            aux=B
21            auxF=fB
22            B=A
23            A=aux
24            fB=fA
25            fA=auxF
26        d=(B-A)/(fB-fA)
27        B=A
28        fB=fA
29        d=d*fA
30        A=A-d
31        fA=anul5(A)
32        if abs(d)<tolerancia:
33            break
34    return A
35
36 rInic5=HallarRaices5(1.4,2.0,1000,0.0000001)
37 rFin5=HallarRaices5(3.5,4.1,1000,0.0000001)
38
39 def alfa5(r):
40     Q=r**2-2*M*r+a5**2
41     Qp=2*r-2*M
42     xi=((a5**2+r**2)*Qp-4*r*Q)/(a5*Qp)
43     return -xi/np.sin(theta)
44
45 def beta5(r):
46     Q=r**2-2*M*r+a5**2
47     Qp=2*r-2*M
48     xi=((a5**2+r**2)*Qp-4*r*Q)/(a5*Qp)
49     eta=(r**2/(a5**2*Qp**2))*(anul5(r))
50     return np.sqrt(eta+a5**2*(np.cos(theta))**2-xi**2/(np.tan(theta))**2)
```

Ya con estas cantidades, es posible configurar el gráfico a realizarse:

```

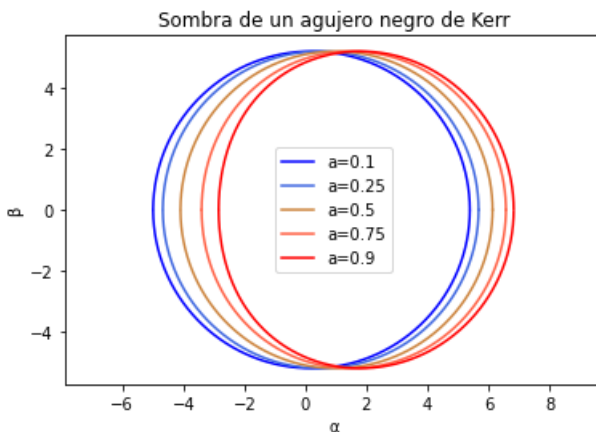
In [ ]: 1 r1=np.linspace(rInic1,rFin1,100000)
2 plt.plot(alfa1(r1),beta1(r1),'b', label="a=0.1")
3 plt.plot(alfa1(r1),-beta1(r1),'b')
4 #purple, olivedrab
5
6
7 r2=np.linspace(rInic2,rFin2,100000)
8 plt.plot(alfa2(r2),beta2(r2),'royalblue', label="a=0.25")
9 plt.plot(alfa2(r2),-beta2(r2),'royalblue')
10 #gold, limegreen
11
12
13 r3=np.linspace(rInic3,rFin3,100000)
14 plt.plot(alfa3(r3),beta3(r3),'peru', label="a=0.5")
15 plt.plot(alfa3(r3),-beta3(r3),'peru')
16 #teal, darkolivegreen
17
18 r4=np.linspace(rInic4,rFin4,100000)
19
20 plt.plot(alfa4(r4),beta4(r4),'tomato', label="a=0.75")
21 plt.plot(alfa4(r4),-beta4(r4),'tomato')
22 #navy, lightseagreen
23
24 r5=np.linspace(rInic5,rFin5,100000)
25 plt.plot(alfa5(r5),beta5(r5),'r', label="a=0.9")
26 plt.plot(alfa5(r5),-beta5(r5),'r')
27
28
29 plt.axis('equal')
30 plt.title('Sombra de un agujero negro de Kerr')
31 plt.xlabel("α")
32 plt.ylabel('β')
33
34 plt.legend()
35
36 plt.show()

```

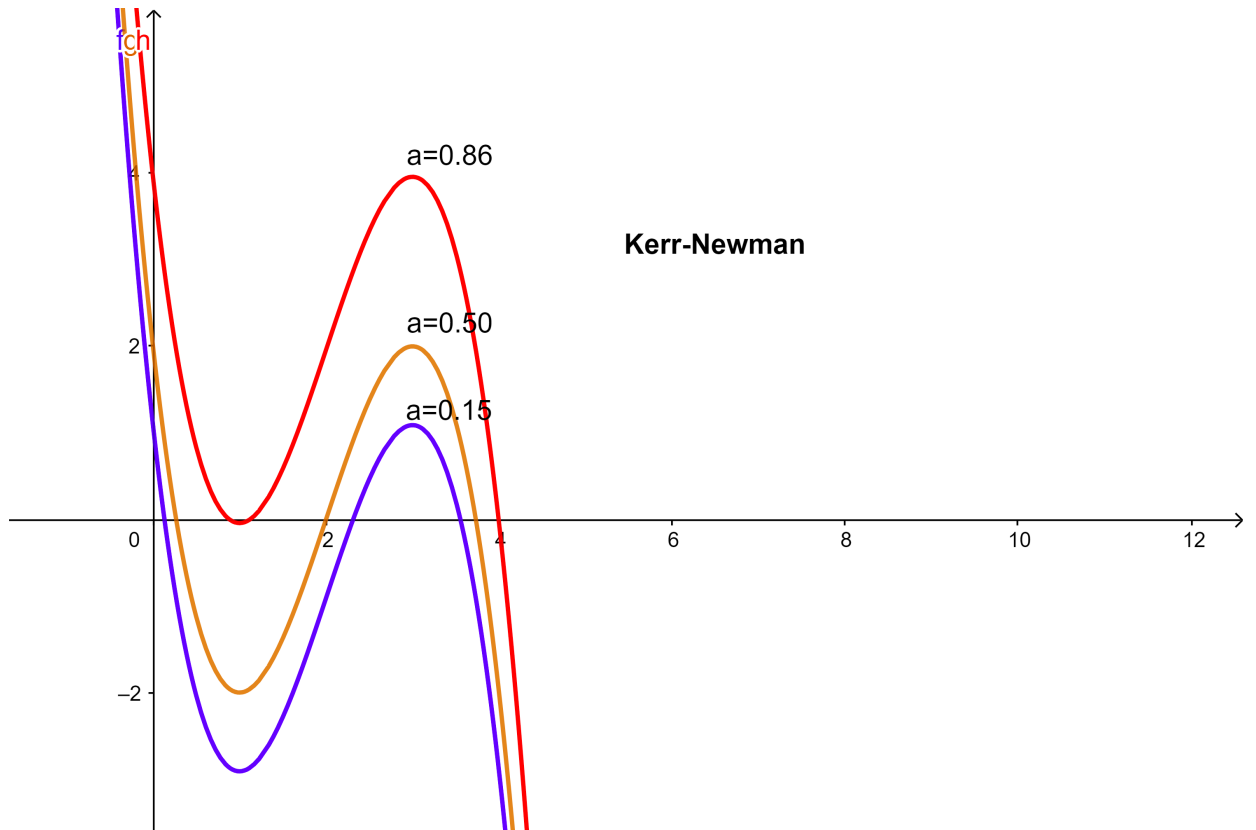
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:50: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:50: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:50: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:50: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:50: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:50: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:50: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:50: RuntimeWarning: invalid va
lue encountered in sqrt

```



##Kerr-Newman Gráfica de  $16Q(a^2 - Q) + 8QQ'r - Q'^2r^2$  utilizada para encontrar los valores límites de r.



Es importante resaltar que para Kerr-Newman se introduce un valor adicional: la carga eléctrica,  $q$ . Para el presente análisis, se toma  $q = 0.5$ .

In [ ]: 1 q=0.5

El proceso a seguir es análogo al anterior:

In [ ]:

```
1 a11=0.15
2
3 def anul11(r):
4     Q=r**2-2*M*r+a11**2+q**2
5     Qp=2*r-2*M
6     return 16*Q*(a11**2-Q)+8*r*Q*Qp-r**2*Qp**2
7
8 def HallarRaices11(A,B,nmax,tolerancia):
9     fA=anul11(A)
10    fB=anul11(B)
11    if abs(fA)>abs(fB):
12        aux=B
13        auxF=fB
14        B=A
15        A=aux
16        fB=fA
17        fA=auxF
18    for n in range(2,nmax+1):
19        if abs(fA)>abs(fB):
20            aux=B
21            auxF=fB
22            B=A
23            A=aux
24            fB=fA
25            fA=auxF
26        d=(B-A)/(fB-fA)
27        B=A
28        fB=fA
29        d=d*fA
30        A=A-d
31        fA=anul11(A)
32        if abs(d)<tolerancia:
33            break
34    return A
35
36 rInic11=HallarRaices11(2,2.6,1000,0.0000001)
37 rFin11=HallarRaices11(3.4,3.8,1000,0.0000001)
38
39 def alfall1(r):
40     Q=r**2-2*M*r+a11**2+q**2
41     Qp=2*r-2*M
42     xi=((a11**2+r**2)*Qp-4*r*Q)/(a11*Qp)
43     return -xi/np.sin(theta)
44
45 def betall1(r):
46     Q=r**2-2*M*r+a11**2+q**2
47     Qp=2*r-2*M
48     xi=((a11**2+r**2)*Qp-4*r*Q)/(a11*Qp)
49     eta=(r**2/(a11**2*Qp**2))*(anul11(r))
50     return np.sqrt(eta+a11**2*(np.cos(theta))**2-xi**2/(np.tan(theta))**2)
```

In [ ]:

```
1 a22=0.50
2
3 def anul22(r):
4     Q=r**2-2*M*r+a22**2+q**2
5     Qp=2*r-2*M
6     return 16*Q*(a22**2-Q)+8*r*Q*Qp-r**2*Qp**2
7
8 def HallarRaices22(A,B,nmax,tolerancia):
9     fA=anul22(A)
10    fB=anul22(B)
11    if abs(fA)>abs(fB):
12        aux=B
13        auxF=fB
14        B=A
15        A=aux
16        fB=fA
17        fA=auxF
18    for n in range(2,nmax+1):
19        if abs(fA)>abs(fB):
20            aux=B
21            auxF=fB
22            B=A
23            A=aux
24            fB=fA
25            fA=auxF
26        d=(B-A)/(fB-fA)
27        B=A
28        fB=fA
29        d=d*fA
30        A=A-d
31        fA=anul22(A)
32        if abs(d)<tolerancia:
33            break
34    return A
35
36 rInic22=HallarRaices22(1.8,2.2,1000,0.0000001)
37 rFin22=HallarRaices22(3.4,4.0,1000,0.0000001)
38
39 def alfa22(r):
40     Q=r**2-2*M*r+a22**2+q**2
41     Qp=2*r-2*M
42     xi=((a22**2+r**2)*Qp-4*r*Q)/(a22*Qp)
43     return -xi/np.sin(theta)
44
45 def beta22(r):
46     Q=r**2-2*M*r+a22**2+q**2
47     Qp=2*r-2*M
48     xi=((a22**2+r**2)*Qp-4*r*Q)/(a22*Qp)
49     eta=(r**2/(a22**2*Qp**2))*(anul22(r))
50     return np.sqrt(eta+a22**2*(np.cos(theta))**2-xi**2/(np.tan(theta))**2)
```

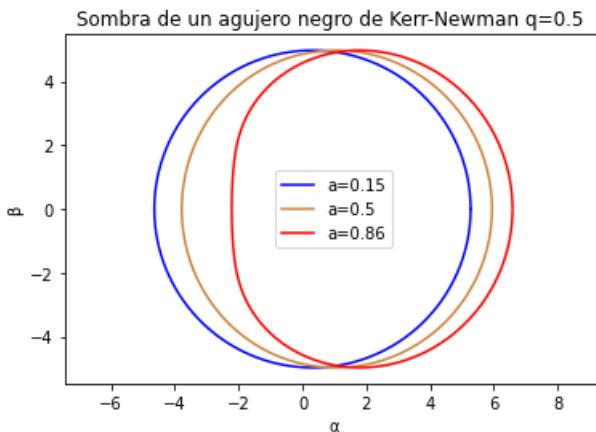
In [ ]:

```
1 a33=0.86
2
3 def anul33(r):
4     Q=r**2-2*M*r+a33**2+q**2
5     Qp=2*r-2*M
6     return 16*Q*(a33**2-Q)+8*r*Q*Qp-r**2*Qp**2
7
8 def HallarRaices33(A,B,nmax,tolerancia):
9     fA=anul33(A)
10    fB=anul33(B)
11    if abs(fA)>abs(fB):
12        aux=B
13        auxF=fB
14        B=A
15        A=aux
16        fB=fA
17        fA=auxF
18    for n in range(2,nmax+1):
19        if abs(fA)>abs(fB):
20            aux=B
21            auxF=fB
22            B=A
23            A=aux
24            fB=fA
25            fA=auxF
26        d=(B-A)/(fB-fA)
27        B=A
28        fB=fA
29        d=d*fA
30        A=A-d
31        fA=anul33(A)
32        if abs(d)<tolerancia:
33            break
34    return A
35
36 rInic33=HallarRaices33(1.0,1.2,1000,0.0000001)
37 rFin33=HallarRaices33(3.8,4.2,1000,0.0000001)
38
39 def alfa33(r):
40     Q=r**2-2*M*r+a33**2+q**2
41     Qp=2*r-2*M
42     xi=((a33**2+r**2)*Qp-4*r*Q)/(a33*Qp)
43     return -xi/np.sin(theta)
44
45 def beta33(r):
46     Q=r**2-2*M*r+a33**2+q**2
47     Qp=2*r-2*M
48     xi=((a33**2+r**2)*Qp-4*r*Q)/(a33*Qp)
49     eta=(r**2/(a33**2*Qp**2))*(anul33(r))
50     return np.sqrt(eta+a33**2*(np.cos(theta))**2-xi**2/(np.tan(theta))**2)
```

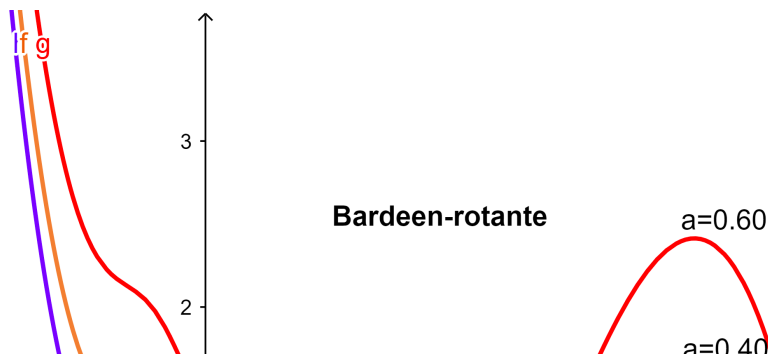
In [ ]:

```
1 r11=np.linspace(rInic11,rFin11,100000)
2 plt.plot(alfa11(r11),beta11(r11),'b', label="a=0.15")
3 plt.plot(alfa11(r11),-beta11(r11),'b')
4 #purple, olivedrab
5
6
7 r22=np.linspace(rInic22,rFin22,100000)
8 plt.plot(alfa22(r22),beta22(r22),'peru', label="a=0.5")
9 plt.plot(alfa22(r22),-beta22(r22),'peru')
10 #gold, limegreen
11
12
13 r33=np.linspace(rInic33,rFin33,100000)
14 plt.plot(alfa33(r33),beta33(r33),'r', label="a=0.86")
15 plt.plot(alfa33(r33),-beta33(r33),'r')
16
17
18 plt.axis('equal')
19 plt.title('Sombra de un agujero negro de Kerr-Newman q=0.5 ')
20 plt.xlabel("α")
21 plt.ylabel("β")
22 plt.legend()
23
24 plt.show()
```

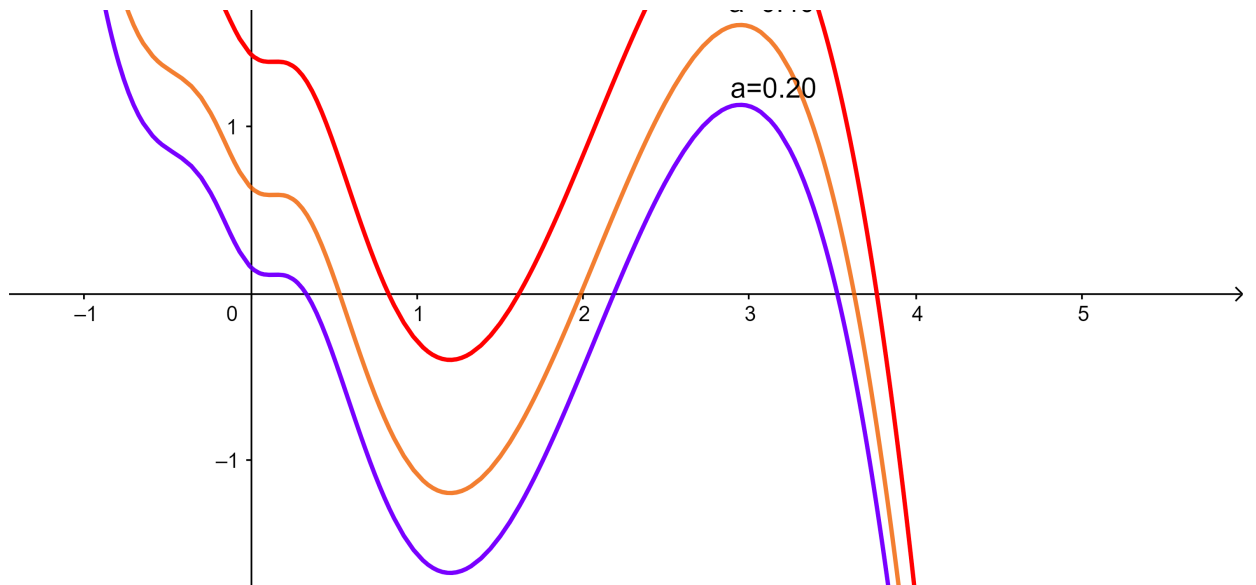
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:50: RuntimeWarning: invalid value encountered in sqrt  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:50: RuntimeWarning: invalid value encountered in sqrt  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:50: RuntimeWarning: invalid value encountered in sqrt  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:50: RuntimeWarning: invalid value encountered in sqrt  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:50: RuntimeWarning: invalid value encountered in sqrt  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:50: RuntimeWarning: invalid value encountered in sqrt  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:50: RuntimeWarning: invalid value encountered in sqrt



##Bardeen-rotante De manera análoga a las anteriores, se grafica para facilitar la obtención de las raíces.







En Bardeen-rotante se añade una carga magnética a la configuración, dada por la cantidad  $g$ . En adelante, se considera  $g = 0.5$ .

In [ ]: 1  $g=0.5$

Similarmente, se procede a configurar cada caso particular.

In [ ]:

```
1 a111=0.20
2
3 def anul111(r):
4     Q=r**2+a111**2-2*M*r**4/(r**2+g**2)**(3/2)
5     Qp=2*r-8*M*r**3*(r**2+g**2)**(-3/2)+6*M*r**5*(r**2+g**2)**(-5/2)
6     return 16*Q*(a111**2-Q)+8*r*Q*Qp-r**2*Qp**2
7
8 def HallarRaices111(A,B,nmax,tolerancia):
9     fA=anul111(A)
10    fB=anul111(B)
11    if abs(fA)>abs(fB):
12        aux=B
13        auxF=fB
14        B=A
15        A=aux
16        fB=fA
17        fA=auxF
18    for n in range(2,nmax+1):
19        if abs(fA)>abs(fB):
20            aux=B
21            auxF=fB
22            B=A
23            A=aux
24            fB=fA
25            fA=auxF
26        d=(B-A)/(fB-fA)
27        B=A
28        fB=fA
29        d=d*fA
30        A=A-d
31        fA=anul111(A)
32        if abs(d)<tolerancia:
33            break
34    return A
35
36 rInic111=HallarRaices111(2.0,2.5,1000,0.0000001)
37 rFin111=HallarRaices111(3.2,3.8,1000,0.0000001)
38
39 def alfa111(r):
40     Q=r**2+a111**2-2*M*r**4/(r**2+g**2)**(3/2)
41     Qp=2*r-8*M*r**3*(r**2+g**2)**(-3/2)+6*M*r**5*(r**2+g**2)**(-5/2)
42     xi=((a111**2+r**2)*Qp-4*r*Q)/(a111*Qp)
43     return -xi/np.sin(theta)
44
45 def beta111(r):
46     Q=r**2+a111**2-2*M*r**4/(r**2+g**2)**(3/2)
47     Qp=2*r-8*M*r**3*(r**2+g**2)**(-3/2)+6*M*r**5*(r**2+g**2)**(-5/2)
48     xi=((a111**2+r**2)*Qp-4*r*Q)/(a111*Qp)
49     eta=(r**2/(a111**2*Qp**2))*(anul111(r))
50     return np.sqrt(eta+a111**2*(np.cos(theta))**2-xi**2/(np.tan(theta))**2)
```

In [ ]:

```
1 a222=0.40
2
3 def anul222(r):
4     Q=r**2+a222**2-2*M*r**4/(r**2+g**2)**(3/2)
5     Qp=2*r-8*M*r**3*(r**2+g**2)**(-3/2)+6*M*r**5*(r**2+g**2)**(-5/2)
6     return 16*Q*(a222**2-Q)+8*r*Q*Qp-r**2*Qp**2
7
8 def HallarRaices222(A,B,nmax,tolerancia):
9     fA=anul222(A)
10    fB=anul222(B)
11    if abs(fA)>abs(fB):
12        aux=B
13        auxF=fB
14        B=A
15        A=aux
16        fB=fA
17        fA=auxF
18    for n in range(2,nmax+1):
19        if abs(fA)>abs(fB):
20            aux=B
21            auxF=fB
22            B=A
23            A=aux
24            fB=fA
25            fA=auxF
26        d=(B-A)/(fB-fA)
27        B=A
28        fB=fA
29        d=d*fA
30        A=A-d
31        fA=anul222(A)
32        if abs(d)<tolerancia:
33            break
34    return A
35
36 rInic222=HallarRaices222(1.8,2.2,1000,0.0000001)
37 rFin222=HallarRaices222(3.4,3.8,1000,0.0000001)
38
39 def alfa222(r):
40     Q=r**2+a222**2-2*M*r**4/(r**2+g**2)**(3/2)
41     Qp=2*r-8*M*r**3*(r**2+g**2)**(-3/2)+6*M*r**5*(r**2+g**2)**(-5/2)
42     xi=((a222**2+r**2)*Qp-4*r*Q)/(a222*Qp)
43     return -xi/np.sin(theta)
44
45 def beta222(r):
46     Q=r**2+a222**2-2*M*r**4/(r**2+g**2)**(3/2)
47     Qp=2*r-8*M*r**3*(r**2+g**2)**(-3/2)+6*M*r**5*(r**2+g**2)**(-5/2)
48     xi=((a222**2+r**2)*Qp-4*r*Q)/(a222*Qp)
49     eta=(r**2/(a222**2*Qp**2))*(anul222(r))
50     return np.sqrt(eta+a222**2*(np.cos(theta))**2-xi**2/(np.tan(theta))**2)
```

In [ ]:

```
1 a333=0.60
2
3 def anul333(r):
4     Q=r**2+a333**2-2*M*r**4/(r**2+g**2)**(3/2)
5     Qp=2*r-8*M*r**3*(r**2+g**2)**(-3/2)+6*M*r**5*(r**2+g**2)**(-5/2)
6     return 16*Q*(a333**2-Q)+8*r*Q*Qp-r**2*Qp**2
7
8 def HallarRaices333(A,B,nmax,tolerancia):
9     fA=anul333(A)
10    fB=anul333(B)
11    if abs(fA)>abs(fB):
12        aux=B
13        auxF=fB
14        B=A
15        A=aux
16        fB=fA
17        fA=auxF
18    for n in range(2,nmax+1):
19        if abs(fA)>abs(fB):
20            aux=B
21            auxF=fB
22            B=A
23            A=aux
24            fB=fA
25            fA=auxF
26        d=(B-A)/(fB-fA)
27        B=A
28        fB=fA
29        d=d*fA
30        A=A-d
31        fA=anul333(A)
32        if abs(d)<tolerancia:
33            break
34    return A
35
36 rInic333=HallarRaices333(1.5,1.8,1000,0.0000001)
37 rFin333=HallarRaices333(3.5,4.0,1000,0.0000001)
38
39 def alfa333(r):
40     Q=r**2+a333**2-2*M*r**4/(r**2+g**2)**(3/2)
41     Qp=2*r-8*M*r**3*(r**2+g**2)**(-3/2)+6*M*r**5*(r**2+g**2)**(-5/2)
42     xi=((a333**2+r**2)*Qp-4*r*Q)/(a333*Qp)
43     return -xi/np.sin(theta)
44
45 def beta333(r):
46     Q=r**2+a333**2-2*M*r**4/(r**2+g**2)**(3/2)
47     Qp=2*r-8*M*r**3*(r**2+g**2)**(-3/2)+6*M*r**5*(r**2+g**2)**(-5/2)
48     xi=((a333**2+r**2)*Qp-4*r*Q)/(a333*Qp)
49     eta=(r**2/(a333**2*Qp**2))*(anul333(r))
50     return np.sqrt(eta+a333**2*(np.cos(theta))**2-xi**2/(np.tan(theta))**2)
```

```

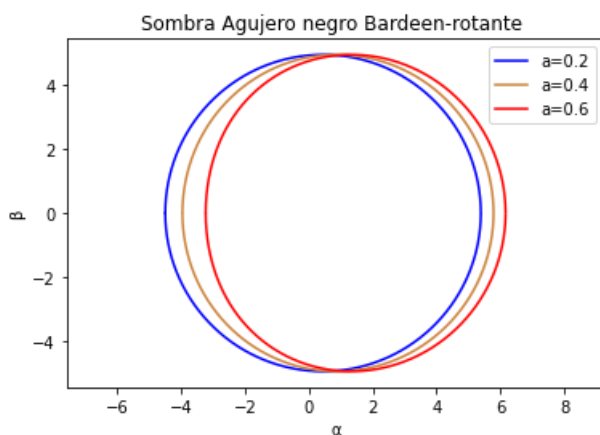
In [ ]: 1 r111=np.linspace(rInic111,rFin111,100000)
2 plt.plot(alfa111(r111),beta111(r111),'b', label="a=0.2")
3 plt.plot(alfa111(r111),-beta111(r111),'b')
4 #purple, olivedrab
5
6
7 r222=np.linspace(rInic222,rFin222,100000)
8 plt.plot(alfa222(r222),beta222(r222),'peru', label="a=0.4")
9 plt.plot(alfa222(r222),-beta222(r222),'peru')
10 #gold, limegreen
11
12
13 r333=np.linspace(rInic333,rFin333,100000)
14 plt.plot(alfa333(r333),beta333(r333),'r', label="a=0.6")
15 plt.plot(alfa333(r333),-beta333(r333),'r')
16 #teal, darkolivegreen
17
18
19 plt.axis('equal')
20
21 plt.title('Sombra de un agujero negro Bardeen-rotante')
22 plt.xlabel("α")
23 plt.ylabel('β')
24
25 plt.legend()
26
27 plt.show()

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:59: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:59: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:57: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:57: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:56: RuntimeWarning: invalid va
lue encountered in sqrt
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:56: RuntimeWarning: invalid va
lue encountered in sqrt

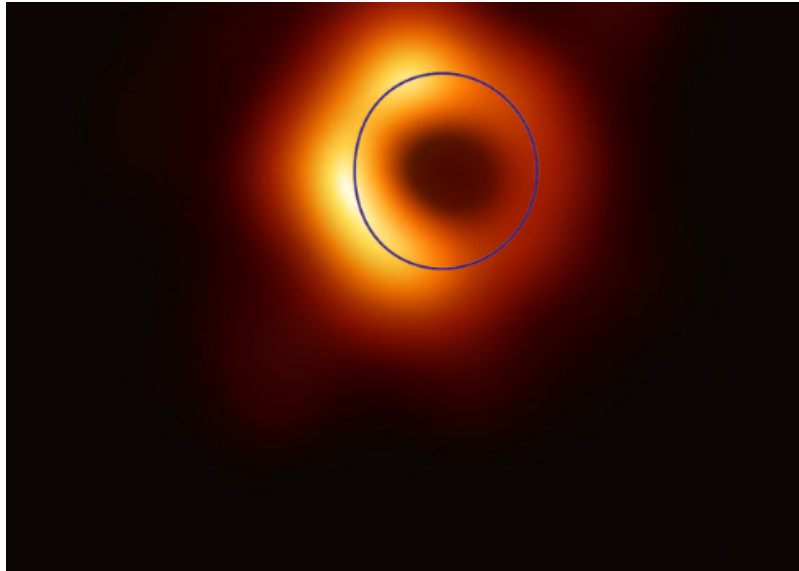
```



#Comparación de la gráfica con la sombra del agujero negro en M87\*

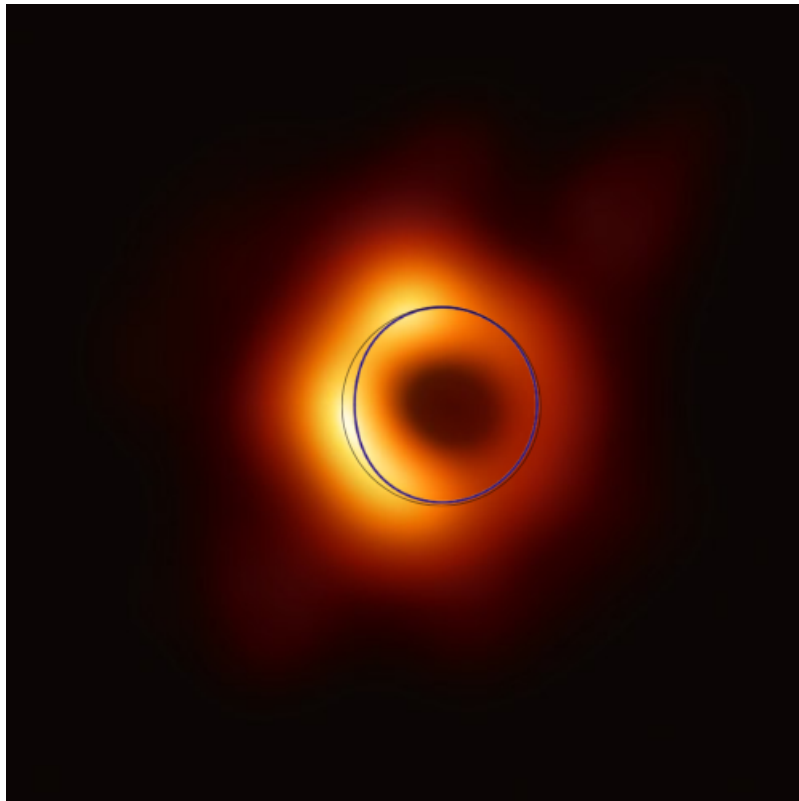
Finalmente, a continuación se visualiza la sombra del agujero negro en M87\* con una de las gráficas obtenidas para Kerr.





Gráfica de la sombra de un agujero negro de Kerr con  $a=0.9$  superpuesta en la sombra del agujero negro M87\*

Como método de comparación en el achatamiento de la sombra se sobrepone también un círculo color negro.



Gráfica de la sombra de un agujero negro de Kerr con  $a=0.9$  superpuesta en la sombra del agujero negro M87\* con un círculo perfecto como comparación

#### #Conclusiones

El programa planteado en el presente informe detalla de manera adecuada la sombra de los agujeros negros de interés, concretamente, los de Kerr, Kerr-Newman y Bardeen-rotante, en función de diferentes momentos angulares. Para esto, se llevó a cabo una minuciosa definición de los conceptos de agujero negro y su correspondiente sombra, que se constituyen como puntos clave en el proceso. Se pudo comprobar, además, que los diferentes parámetros que determinan el tipo de agujero negro influyen directamente en la forma que se capta de su sombra. De manera cualitativa, es posible observar que, al superponer la sombra sobre la fotografía, existe una correspondencia entre lo que sostiene la teoría de agujeros negros y lo que resulta en la práctica.

Como perspectivas para proyectos relacionados, sería un complemento rico en información plantear un análisis de las

zonas brillantes que se distinguen en la fotografía. Explorar el porqué existen zonas anaranjadas más intensas, así como los métodos computacionales usados en las simulaciones previas a la captura de la imagen, son potenciales trabajos que contribuyan a la difusión científica acerca de estos extravagantes elementos de la naturaleza

#### #Bibliografía

- Wild, F. (18 de agosto de 2018). *What is a Black Hole?*. NASA. Recuperado el 10 de julio de 2022 de <https://www.nasa.gov/audience/forstudents/k-4/stories/nasa-knows/what-is-a-black-hole-k4.html> (<https://www.nasa.gov/audience/forstudents/k-4/stories/nasa-knows/what-is-a-black-hole-k4.html>)
- Contreras, E., Ramirez–Velasquez, J. M., Rincón, Á., Panotopoulos, G., y Bargueño, P. (2019). Black hole shadow of a rotating polytropic black hole by the Newman–Janis algorithm without complexification. *The European Physical Journal C*, 79(9), 1-10.
- Amarilla, L. (2013). *Sombras de agujeros negros en teorías alternativas de gravitación* [Tesis de doctorado. Universidad de Buenos Aires]. Repositorio Digital Institucional de la Universidad de Buenos Aires.
- Londoño, J. y Velasquez, E. (2017). *Agujeros Negros de Kerr*. Universidad Nacional de Colombia. [https://www.researchgate.net/publication/328380970\\_Agujeros\\_Negros\\_de\\_Kerr](https://www.researchgate.net/publication/328380970_Agujeros_Negros_de_Kerr) ([https://www.researchgate.net/publication/328380970\\_Agujeros\\_Negros\\_de\\_Kerr](https://www.researchgate.net/publication/328380970_Agujeros_Negros_de_Kerr))