VNIVERSITAT DE VALÈNCIA Facultat de Física

Autora: María Victoria Gómez Bifante
Tutor (1): Genaro Saavedra Tortosa
Tutor (2): Juan Carlos Barreiro Hervás

Trabajo de Fin de Grado en Física

Introducción a la holografía digital en eje

Septiembre 2023

Presento el código fuente que he creado para verificar la solidez de modelo teórico del método de variación de fase de la holografía digital en eje.

Este código lo he desarrollado en base de Python 3.9 haciendo uso de la plataforma informática interactiva Jupyter Notebook. Python es un lenguaje de código abierto que se organiza en módulos de extensión *.py* que contienen funciones, variables u otros objetos predefinidos. A su vez, estos módulos se agrupan en paquetes según su funcionalidad, de modo que para poder desarrollar cualquier código en Python, es preciso importar previamente los módulos con los paquetes que se vayan a emplear. Además, Python es un lenguaje de código abierto que permite personalizar y crear tus propias funciones.

En este caso, para verificar la técnica del método de variación de fase, he importado el módulo NumPy y el módulo Matplotlib de las librerías de Python. Por un lado, NumPy proporciona las funciones necesarias para trabajar con matrices multidimensionales y contiene una gran variedad de rutinas que permiten realizar operaciones rápidas como la transformada discreta de Fourier. Por otro lado, el módulo Matplotlib contiene las herramientas necesarias para poder representar funciones gráficamente. También he extraído el paquete Pyplot de Matplotlib dado que proporciona herramientas rápidas y cómodas para representar funciones de n dimensiones. Además, he creado tres funciones que emplearé posteriormente en la simulación: *Fresnel_Diffraction*, *UnitBox* y *Hat*. La función *Fresnel_Diffraction* propaga una distancia z la distribución de amplitud compleja de una onda, mientras que las funciones *UnitBox* y *Hat* proporcionan las fases de dos objetos distintos puros de fase.

Método de variación de fase para dos objetos distintos puros de fase

```
# Importamos los modulos y las librerias necesarias
import numpy as np # herramientas numéricas y matriciales
import matplotlib.pyplot as plt # funciones para representar figuras

# Definición de constantes, puntos de muestreo y creación de gradillas
lamb = 500*10**(-6); # mm
phi = np.array([0, np.pi/2, np.pi]); # rad
data = np.array([0, np.pi/8, np.pi/4, 3*np.pi/8, np.pi/2]); # rad
N = 551; # número de puntos
```

```
# Datos del caso A
X = np.linspace(-1, 1, num = N-1);
Y = np.linspace(-1, 1, num = N-1);
x, y = np.meshgrid(X, Y); # crea una matriz que varía por columna y otra por fila

# Datos del caso B
A = np.linspace(-5, 5, num = N-1);
B = np.linspace(-5, 5, num = N-1);
a, b = np.meshgrid(A, B);  # crea una matriz que varía por columna y otra por fila
z = 500; #mm
Delta = 0.018; # mm

# Definición de funciones

# Función de transferencia
def Fresnel_Diffraction(U_inicial, dx, dy, z, lamb, delta):
    n1 = dx.ndim;
    n2 = dy.ndim;
    if (n1 == 1 & n2 == 1):
        u = np.arange(-1/(2*Delta) + 1/(N*Delta), 1/(2*Delta), 1/(N*Delta));
        v = np.arange(-1/(2*Delta) + 1/(N*Delta), 1/(2*Delta), 1/(N*Delta));
        U_in = np.fft.fftshift(np.fft.fft(U_inicial, norm = 'forward'));
        H = np.exp(-1j*np.pi*lamb*np.abs(z)*(u**2 + v**2));
        if (z >= 0):
            U_final = np.fft.ifft(np.fft.ifftshift(U_in*H), norm = 'forward');
        elif (z < 0):
             U_final = np.fft.ifft(np.fft.ifftshift(U_in*np.conjugate(H)),␣
↪norm='forward');
    elif (n1 == 2 & n2 == 2):
        u = np.arange(-1/(2*Delta) + 1/(N*Delta), 1/(2*Delta), 1/(N*Delta));
        v = np.arange(-1/(2*Delta) + 1/(N*Delta), 1/(2*Delta), 1/(N*Delta));
        u, v = np.meshgrid(u, v);
        U_in = np.fft.fftshift(np.fft.fft2(U_inicial, norm = 'forward'));
        H = np.exp(-1j*np.pi*lamb*np.abs(z)*(u**2 + v**2));
        if (z > 0):
            U_final = np.fft.ifft2(np.fft.ifftshift(U_in*H), norm = 'forward');
        elif (z < 0):
            U_final = np.fft.ifft2(np.fft.ifftshift(U_in*np.conjugate(H)), norm =␣
↪'forward');
        elif (z == 0):
            U_final = U_in;
    return U_final, u

# Disco de fase constante con radio máximo r_o = 1/2 (u.a)
def UnitBox(coordenate1, coordenate2):
    n1 = coordenate1.ndim;
    n2 = coordenate2.ndim;
```

```
    Radius= np.sqrt(coordenate1**2 + coordenate2**2);
    Disk = np.zeros_like((Radius));
    limit = 1/2;
    if (n1 == 1 & n2 == 1):
        for ii in range(0,N-1):
            if  ((coordenate1[ii] <= limit) & (coordenate1[ii] >= -limit)):
                Disk[ii] = 1.0;
    elif (n1 == 2 & n2 == 2):
        for ii in range(0,N-1):
            for jj in range(0,N-1):
                if  (Radius[ii][jj] < limit):
                    Disk[ii][jj] = 1.0;
    return Disk

#  Variación lineal de fase desde el centro hasta el borde con radio máximo r_o = 1/2 (u.a)
def Hat(radius, coordenate1, coordenate2):
    Hat = radius.copy();
    limit = 0.72;
    Hat[(radius <= limit)] = 1 - (1/limit)*np.abs(radius[(radius <= limit)]);
    Hat[(radius > limit)] = 0;
    return Hat
```

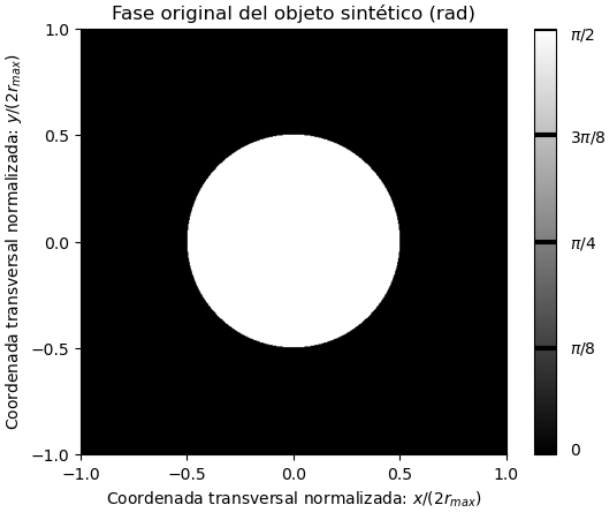
Caso 1. El plano del holograma se encuentra sobre el objeto

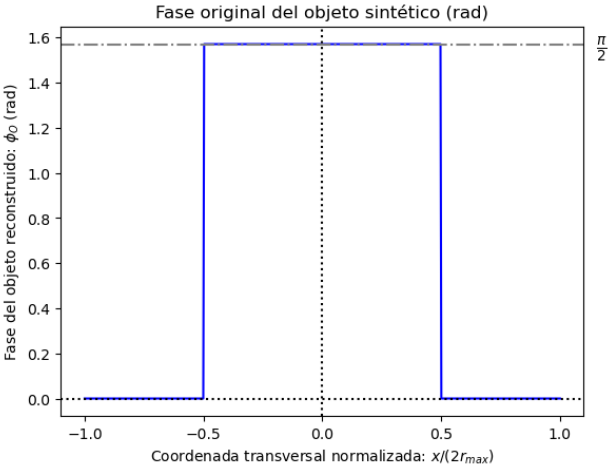
```
# Frente de ondas trasmitido por un objeto sintético en el plano objeto
phi_0 = np.pi/2; # rad
Obj_1d = np.exp(1j*phi_0*UnitBox(X, Y)); # 1D disco de fase constante
Obj_2d =  np.exp(1j*phi_0*UnitBox(x, y)); # 2D disco de fase constante
#Obj_1d = np.exp(1j*phi_0*Hat(np.sqrt(X**2 + Y**2), X, Y)); # 1D variación de fase desde el centro hasta el borde
#Obj_2d = np.exp(1j*phi_0*Hat(np.sqrt(x**2 + y**2), x, y)); # 2D variación de fase desde el centro hasta el borde

# Frente de ondas emergente de la fuente retardada una fase phi
Ref_1d = np.array([np.ones_like(Obj_1d)*np.exp(1j*phi[0]), np.ones_like(Obj_1d)*np.exp(1j*phi[1]), np.ones_like(Obj_1d)*np.exp(1j*phi[2])]); # 1D
Ref_2d = np.array([np.ones_like(Obj_2d)*np.exp(1j*phi[0]), np.ones_like(Obj_2d)*np.exp(1j*phi[1]), np.ones_like(Obj_2d)*np.exp(1j*phi[2])]); # 2D

# Sistema de ecuaciones formado por tres patrones de intensidad (1D)
I1_1d = np.abs(Obj_1d + Ref_1d[0])**2;
I2_1d = np.abs(Obj_1d + Ref_1d[1])**2;
I3_1d = np.abs(Obj_1d + Ref_1d[2])**2;

# Sistema de ecuaciones formado por tres patrones de intensidad (2D)
I1_2d = np.abs(Obj_2d + Ref_2d[0])**2;
I2_2d = np.abs(Obj_2d + Ref_2d[1])**2;
```

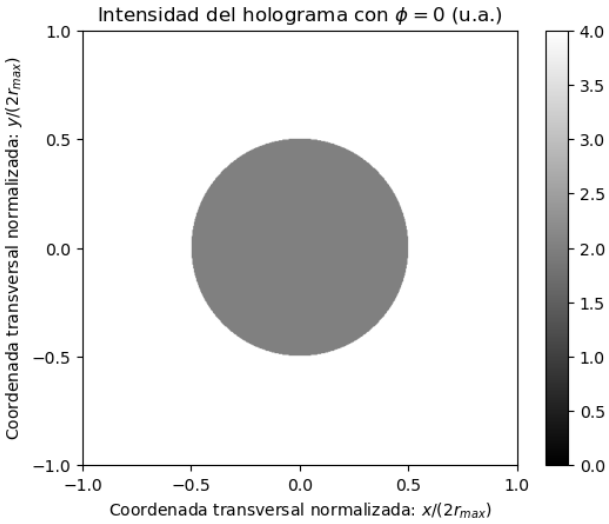
```
I3_2d = np.abs(Obj_2d + Ref_2d[2])**2;

#Resolución de ambos sistemas
Rec_1d = ((1+1j)*I1_1d - 2*I2_1d + (1-1j)*I3_1d)/(4j); # 1D
Rec_2d = ((1+1j)*I1_2d - 2*I2_2d + (1-1j)*I3_2d)/(4j); # 2D

# Representación de la fase original de la onda objeto (2D)
plt.figure();
cf0_z0 = plt.imshow(np.angle(Obj_2d), cmap = 'gray', extent = [-1,1,-1,1], vmin =␣
 ↪0, vmax = np.pi /2);
cbar0_z0 = plt.colorbar(ticks = [])
cbar0_z0.ax.hlines(data, 0, 1, color="black", linewidth=3);
plt.text(1.3, -1, r'$0$', fontsize=10,  color='black');
plt.text(1.3, -0.52, r'$\pi/8}$', fontsize=10,  color='black');
plt.text(1.3, -0.03, r'$\pi/4$', fontsize=10, color='black');
plt.text(1.3, 0.47, r'$3\pi/8$', fontsize=10,  color='black');
plt.text(1.3, 0.95, r'$\pi/2$', fontsize=10, color='black');
plt.title("Fase original del objeto sintético (rad)");
plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-1, 1, num = 5));
plt.yticks(np.linspace(-1, 1, num = 5));

# Representación del perfil de la fase original de la onda objeto (1D)
plt.figure();
plt.plot(X, np.angle(Obj_1d), color ='b');
plt.axhline(y = 0, color = "black", linestyle = ":");
plt.axvline(x = 0, color = "black", linestyle = ":");
plt.axhline(y = np.max(np.angle(Obj_1d)), color =  "gray", linestyle = "-.");
plt.text(np.max(X) + 0.15, np.max(np.angle(Obj_1d)) - 0.04, r'$\frac{\pi}{2}$',␣
 ↪fontsize = 14);
plt.title("Fase original del objeto sintético (rad)");
plt.ylabel("Fase del objeto reconstruido: $\phi_0$ (rad)");
plt.xlabel("Coordenada transversal normalizada: $x /(2r_{max})$");
plt.xticks(np.linspace(-1, 1, num = 5));

# Representación gráfica de la intensidad I1 (2D)
plt.figure();
cf1_z0 = plt.imshow(I1_2d,  cmap = 'gray', extent = [-1,1,-1,1], vmin = 0, vmax =␣
 ↪4);
plt.colorbar(cf1_z0);
plt.title("Intensidad del holograma con $\phi = 0$ (u.a.)");
plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-1, 1, num = 5));
plt.yticks(np.linspace(-1, 1, num = 5));
```

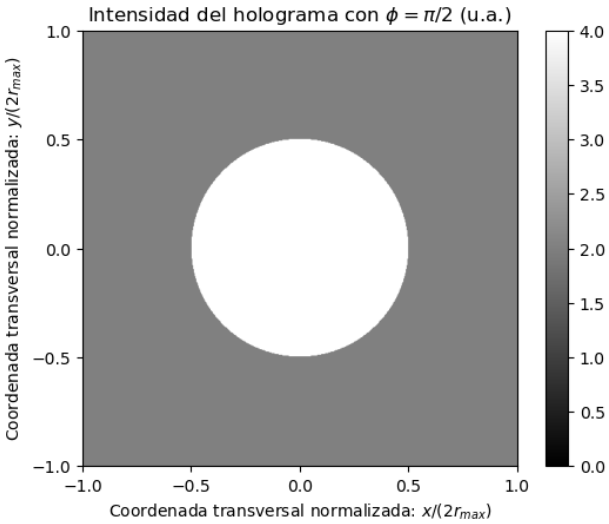
```
# Representación gráfica de la intensidad I2 (2D)
plt.figure();
cf2_z0 = plt.imshow(I2_2d,cmap = 'gray', extent = [-1,1,-1,1], vmin = 0, vmax = 4);
plt.colorbar(cf2_z0);
plt.title("Intensidad del holograma con $\phi = \pi/2$ (u.a.)");
plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-1, 1, num = 5));
plt.yticks(np.linspace(-1, 1, num = 5));

# Representación gráfica de la intensidad I3 (2D)
plt.figure();
cf3_z0 = plt.imshow(I3_2d,  cmap = 'gray', extent = [-1,1,-1,1], vmin = 0, vmax =␣
 ↪4);
plt.colorbar(cf3_z0);
plt.title("Intensidad del holograma con $\phi = \pi$ (u.a.)");
plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-1, 1, num = 5));
plt.yticks(np.linspace(-1, 1, num = 5));

# Representación gráfica de las tres intensidades (1D)
plt.figure();
plt.plot(X, I1_1d, '--', color = 'b', label = '$\phi = 0$');
plt.plot(X, I2_1d, ':', color = 'r', label = '$\phi = \pi /2$', linewidth = 2);
plt.plot(X, I3_1d, '--', color = 'g', label = '$\phi = \pi$');
plt.axhline(y = 0, color = "black", linestyle=":");
plt.axvline(x = 0, color = "black", linestyle=":");
plt.title("Intensidad de los tres hologramas (u.a.)");
plt.legend(bbox_to_anchor =(1.25,0.5), loc='center right')
plt.ylabel("Intensidad del holograma (u.a.)");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-1, 1, num = 5));

# Representación de la fase reconstruida de la onda objeto (2D)
plt.figure();
cf4_z0 = plt.imshow(np.angle(Obj_2d),  cmap = 'gray', extent = [-1,1,-1,1], vmin =␣
 ↪0, vmax = np.pi/2);
cbar4_z0 = plt.colorbar(ticks = [])
cbar4_z0.ax.hlines(data, 0, 1, color="black", linewidth=3);
plt.text(1.3, -1, r'$0$', fontsize=10,  color='black');
plt.text(1.3, -0.52, r'$\pi/8}$', fontsize=10,  color='black');
plt.text(1.3, -0.03, r'$\pi/4$', fontsize=10, color='black');
plt.text(1.3, 0.47, r'$3\pi/8$', fontsize=10,  color='black');
plt.text(1.3, 0.95, r'$\pi/2$', fontsize=10, color='black');
plt.title("Fase reconstruida del objeto (rad)");
plt.ylabel("Coordenada transversal normalizada: $y /(2r_{max})$");
```

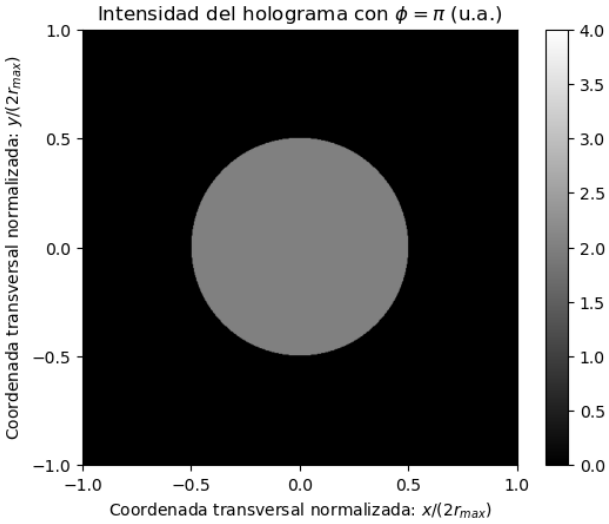
```
plt.xlabel("Coordenada transversal normalizada: $x /(2r_{max})$")
plt.xticks(np.linspace(-1, 1, num = 5));
plt.yticks(np.linspace(-1, 1, num = 5));

# Representación del perfil de la fase reconstruida de la onda objeto (1D)
plt.figure();
plt.plot(X, np.angle(Rec_1d), color ='r');
plt.axhline(y = 0, color = "black", linestyle = ":");
plt.axvline(x = 0, color = "black", linestyle = ":");
plt.axhline(y = np.max(np.angle(Obj_1d)), color =  "gray", linestyle = "-.");
plt.text(np.max(X) + 0.15, np.max(np.angle(Obj_1d)) - 0.04, r'$\frac{\pi}{2}$',␣
 ↪fontsize = 14);
plt.title("Fase reconstruida del objeto sintético (rad)");
plt.ylabel("Fase del objeto reconstruido: $\phi_O$ (rad)");
plt.xlabel("Coordenada transversal normalizada: $x /(2r_{max})$");
plt.xticks(np.linspace(-1, 1, num = 5));

# Comparación entre la fase original y la reconstruida de la distribución de␣
 ↪amplitud emergente del objeto
plt.figure();
plt.plot(X, np.angle(Obj_1d),  linestyle = ":", color = 'b', label = '$\phi_O$␣
 ↪original', linewidth = 4);
plt.plot(X, np.angle(Rec_1d),  linestyle = "--",color = 'r', label = '$\phi_O$␣
 ↪reconstruida', linewidth = 1);
plt.axvline(x = 0, color="black", linestyle=":");
plt.axhline(y = np.max(np.angle(Rec_1d)), color="gray", linestyle="-.");
plt.text(np.max(X) + 0.15, np.max(np.angle(Rec_1d)) - 0.04, r'$\frac{\pi}{2}$',␣
 ↪fontsize=14);
plt.title("Fase original y reconstruida del objeto (rad)");
plt.ylabel("Fase del objeto reconstruido: $\phi_O$ (rad)");
plt.xlabel("Coordenada transversal normalizada: $x /(2r_{max})$");
plt.legend(bbox_to_anchor =(1.4,0.5), loc='center right')
plt.xticks(np.linspace(-1, 1, num = 5));
```

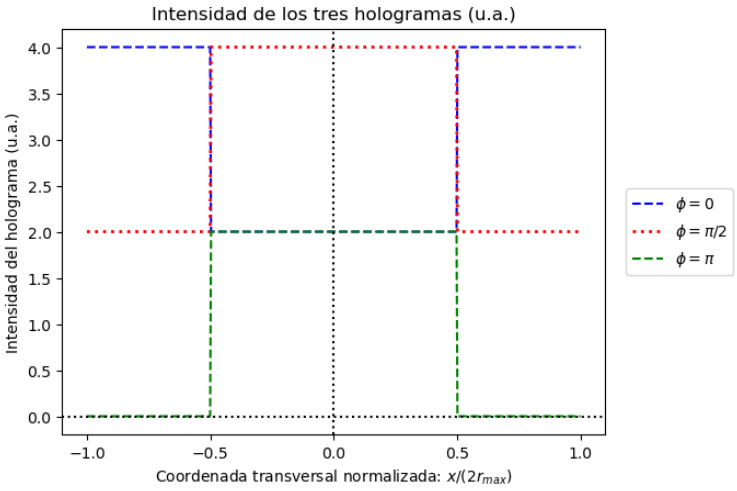

Fase original del objeto sintético (rad)
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
$\pi/2$
$3\pi/8$
$\pi/4$
$\pi/8$
0

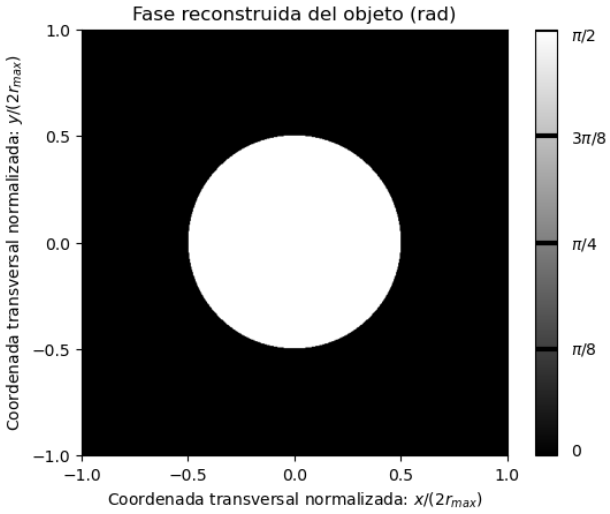

Fase original del objeto sintético (rad)
Fase del objeto reconstruido: ϕ_O (rad)
Coordenada transversal normalizada: $x/(2r_{max})$
$\frac{\pi}{2}$

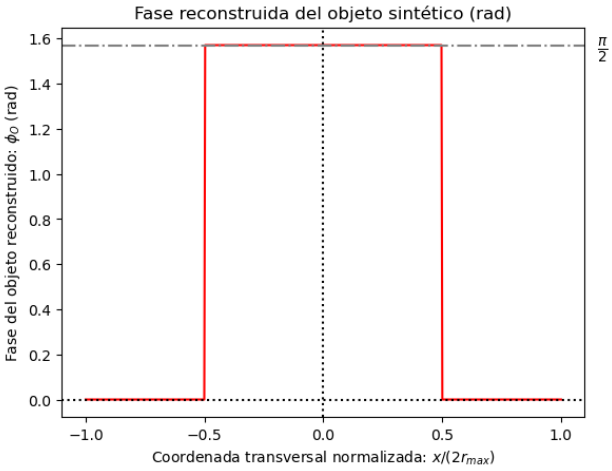

Intensidad del holograma con $\phi = 0$ (u.a.)
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$

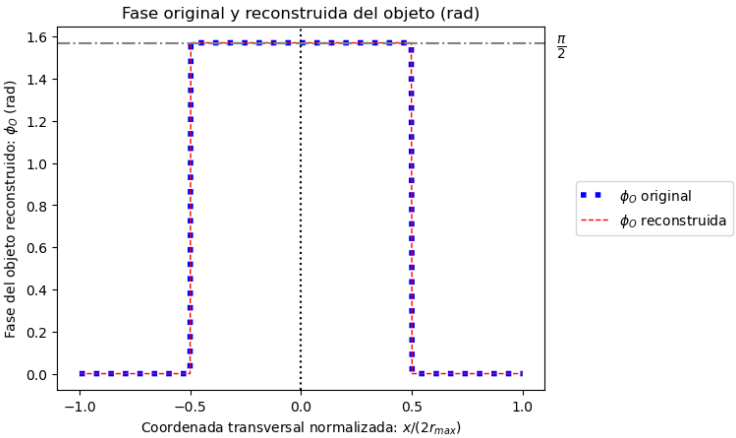

Intensidad del holograma con $\phi = \pi/2$ (u.a.)
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$


Intensidad del holograma con $\phi = \pi$ (u.a.)
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$


Intensidad de los tres hologramas (u.a.)
Intensidad del holograma (u.a.)
Coordenada transversal normalizada: $x/(2r_{max})$
$\phi = 0$
$\phi = \pi/2$
$\phi = \pi$







Caso 2. El plano del holograma se encuentra a z = 500 mm del objeto

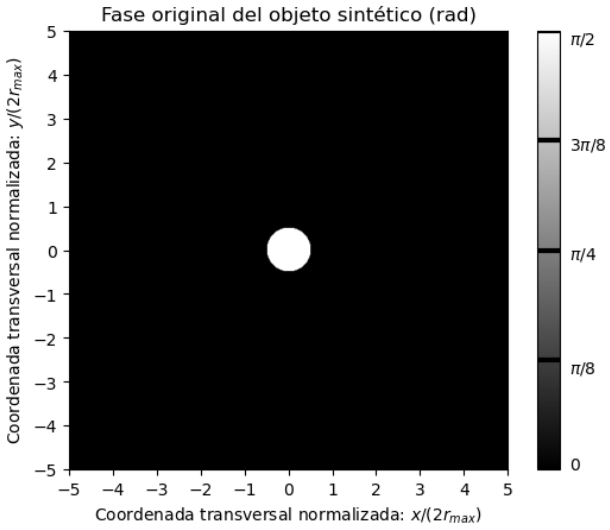
```
# Frente de ondas trasmitido por un objeto sintético en el plano objeto
phi_0 = np.pi/2; # rad
Objeto_1d = np.exp(1j*phi_0*UnitBox(A, B)); # 1D disco de fase constante
```

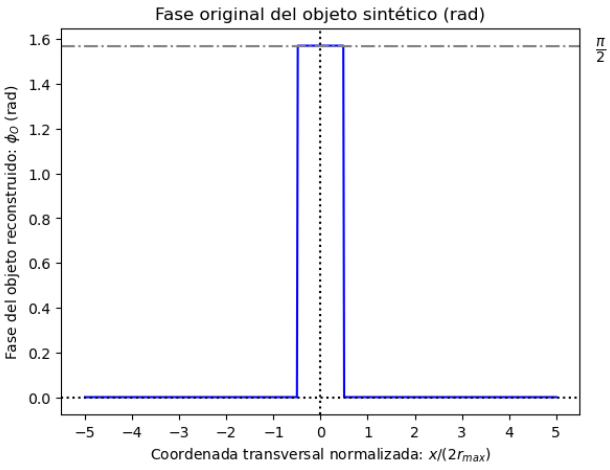
```
Objeto_2d = np.exp(1j*phi_0*UnitBox(a, b)); # 2D disco de fase constante
#Objeto_1d = np.exp(1j*phi_0*Hat(np.sqrt(A**2 + B**2), A, B)); # 1D variación de␣
 ↪fase desde el centro hasta el borde
#Objeto_2d = np.exp(1j*phi_0*Hat(np.sqrt(a**2 + b**2), a, b)); # 2D variación de␣
 ↪fase desde el centro hasta el borde

# Propagación de la onda objeto original a  z = 500 mm
Ob_1d, u_x = Fresnel_Diffraction(Objeto_1d, A, B, z, lamb, Delta);
Ob_2d, uu_x = Fresnel_Diffraction(Objeto_2d, a, b, z, lamb, Delta);

# Frente de ondas emergente de la fuente retardada una fase phi
Re_1d = np.array([np.ones_like(Ob_1d)*np.exp(1j*phi[0]), np.ones_like(Ob_1d)*np.
 ↪exp(1j*phi[1]), np.ones_like(Ob_1d)*np.exp(1j*phi[2])]); # 1D
Re_2d = np.array([np.ones_like(Ob_2d)*np.exp(1j*phi[0]), np.ones_like(Ob_2d)*np.
 ↪exp(1j*phi[1]), np.ones_like(Ob_2d)*np.exp(1j*phi[2])]); # 2D

# Sistema de ecuaciones formado por tres patrones de intensidad (1D)
I1d_1 = np.abs(Ob_1d + Re_1d[0])**2;
I1d_2 = np.abs(Ob_1d + Re_1d[1])**2;
I1d_3 = np.abs(Ob_1d + Re_1d[2])**2;

# Sistema de ecuaciones formado por tres patrones de intensidad (2D)
I2d_1 = np.abs(Ob_2d + Re_2d[0])**2;
I2d_2 = np.abs(Ob_2d + Re_2d[1])**2;
I2d_3 = np.abs(Ob_2d + Re_2d[2])**2;

#Resolución de ambos sistemas
R_1d = ((1+1j)*I1d_1 - 2*I1d_2 + (1-1j)*I1d_3)/(4j); # 1D
R_2d = ((1+1j)*I2d_1 - 2*I2d_2 + (1-1j)*I2d_3)/(4j); # 2D

# Propagación de la onda objeto reconstruida a z = - 500 mm
UO_1d, U_x = Fresnel_Diffraction(R_1d, A, B, -z, lamb, Delta)
UO_2d, u_x = Fresnel_Diffraction(R_2d, a, b, -z, lamb, Delta)

# Representación de la fase original de la onda objeto (2D)
plt.figure();
cf0_z0 = plt.imshow(np.angle(Objeto_2d), cmap = 'gray', extent = [-5,5,-5,5], vmin␣
 ↪= 0, vmax = np.pi /2);
cbar0_z0 = plt.colorbar(ticks = [])
cbar0_z0.ax.hlines(data, 0, 1, color="black", linewidth=3);
plt.text(6.4, -5, r'$0$', fontsize=10, color='black');
plt.text(6.4, -2.8, r'$\pi /8$', fontsize=10, color='black');
plt.text(6.4, -0.2, r'$\pi/4$', fontsize=10, color='black');
plt.text(6.4, 2.3, r'$3 \pi /8$', fontsize=10, color='black');
plt.text(6.4, 4.7, r'$\pi/2$', fontsize=10, color='black');
plt.title("Fase original del objeto sintético (rad)");
plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
```

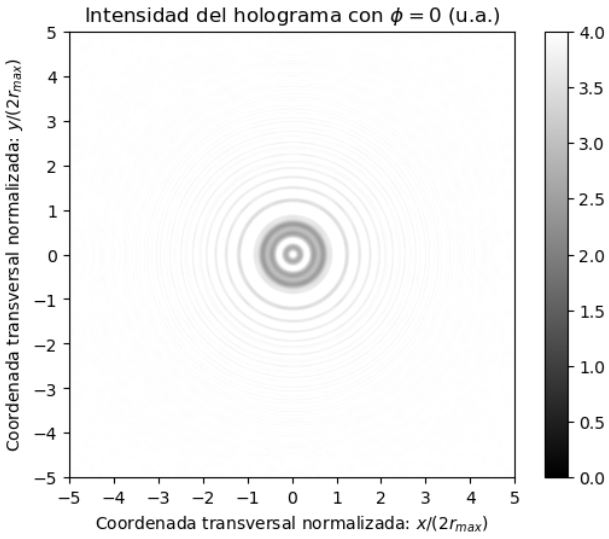
```
plt.xticks(np.linspace(-5, 5, num = 11));
plt.yticks(np.linspace(-5, 5, num = 11));

# Representación del perfil de la fase original de la onda objeto (1D)
plt.figure();
plt.plot(A, np.angle(Objeto_1d), color ='b');
plt.axhline(y = 0, color = "black", linestyle = ":");
plt.axvline(x = 0, color = "black", linestyle = ":");
plt.axhline(y = np.max(np.angle(Obj_1d)), color =  "gray", linestyle = "-.");
plt.text(np.max(A) + np.max(A)/6, np.max(np.angle(Obj_1d)) - 0.04,␣
 ↪r'$\frac{\pi}{2}$', fontsize = 14);
plt.title("Fase original del objeto sintético (rad)");
plt.ylabel("Fase del objeto reconstruido: $\phi_0$ (rad)");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-5, 5, num = 11));

# Representación gráfica de la intensidad I1 (2D)
plt.figure();
cf1_z = plt.imshow(I2d_1, cmap = 'gray', extent = [-5,5,-5,5], vmin = 0, vmax = 4);
plt.colorbar(cf1_z);
plt.title("Intensidad del holograma con $\phi = 0$ (u.a.)");
plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-5, 5, num = 11));
plt.yticks(np.linspace(-5, 5, num = 11));

# Representación gráfica de la intensidad I2 (2D)
plt.figure();
cf2_z = plt.imshow(I2d_2, cmap = 'gray', extent = [-5,5,-5,5], vmin = 0, vmax = 4);
plt.colorbar(cf2_z);
plt.title("Intensidad del holograma con $\phi = \pi/2$ (u.a.)");
plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-5, 5, num = 11));
plt.yticks(np.linspace(-5, 5, num = 11));

# Representación gráfica de la intensidad I3 (2D)
plt.figure();
cf3_z = plt.imshow(I2d_3,  cmap = 'gray', extent = [-5,5,-5,5], vmin = 0, vmax =␣
 ↪4);
plt.colorbar(cf3_z);
plt.title("Intensidad del holograma con $\phi = \pi$ (u.a.)");
plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-5, 5, num = 11));
plt.yticks(np.linspace(-5, 5, num = 11));
```

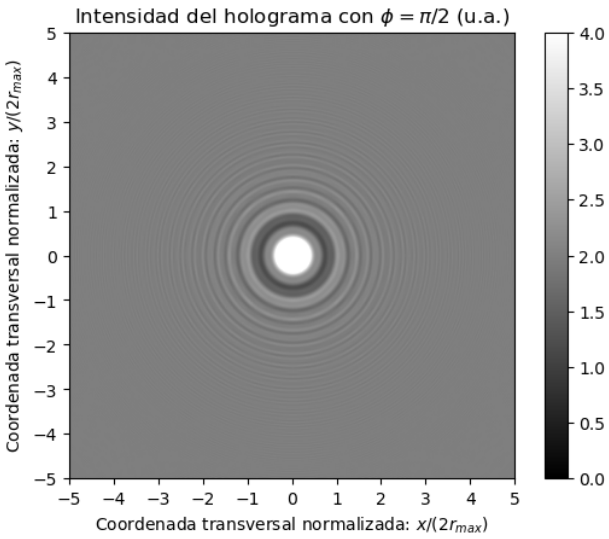
```
# Representación gráfica de las tres intensidades (1D)
plt.figure();
plt.plot(A, I1d_1, '--', color = 'b', label = '$\phi = 0$');
plt.plot(A, I1d_2, ':', color = 'r', label = '$\phi = \pi /2$');
plt.plot(A, I1d_3, '--', color = 'g', label = '$\phi = \pi$');
plt.axhline(y = 0, color = "black", linestyle=":");
plt.axvline(x = 0, color = "black", linestyle=":");
plt.title("Intensidad de los tres hologramas (u.a)");
plt.legend(bbox_to_anchor =(1.25,0.5), loc='center right')
plt.ylabel("Intensidad del holograma (u.a.)");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-5, 5, num = 11));

# Representación de la fase reconstruida de la onda objeto (2D)
plt.figure();
cf4_z = plt.imshow(np.angle(UO_2d),  cmap = 'gray', extent = [-5,5,-5,5]);
cbar4_z = plt.colorbar(ticks = [])
cbar4_z.ax.hlines(data, 0, 1, color="black", linewidth=3);
plt.text(6.4, -5, r'$0$', fontsize=10, color='black');
plt.text(6.4, -2.8, r'$\pi /8$', fontsize=10, color='black');
plt.text(6.4, -0.2, r'$\pi/4$', fontsize=10, color='black');
plt.text(6.4, 2.3, r'$3 \pi /8$', fontsize=10, color='black');
plt.text(6.4, 4.7, r'$\pi/2$', fontsize=10, color='black');
plt.title("Fase reconstruida del objeto (rad)");
plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$")
plt.xticks(np.linspace(-5, 5, num = 11));
plt.yticks(np.linspace(-5, 5, num = 11));

# Representación del perfil de la fase reconstruida de la onda objeto (1D)
plt.figure();
plt.plot(A, np.angle(UO_1d), color ='r');
plt.axhline(y = 0, color = "black", linestyle = ":");
plt.axvline(x = 0, color = "black", linestyle = ":");
plt.axhline(y = np.max(np.angle(Obj_1d)), color =  "gray", linestyle = "-.");
plt.text(np.max(A) + np.max(A)/6, np.max(np.angle(Obj_1d)) - 0.04,␣
 ↪r'$\frac{\pi}{2}$', fontsize = 14);
plt.title("Fase reconstruida del objeto sintético (rad)");
plt.ylabel("Fase del objeto reconstruido: $\phi_O$ (rad)");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-5, 5, num = 11));

# Comparación entre la fase original y la reconstruida de la distribución de␣
 ↪amplitud emergente del objeto
plt.figure();
plt.plot(A, np.angle(Objeto_1d), linestyle=":", color = 'b', label = '$\phi_O$␣
 ↪original', linewidth = 3);
```

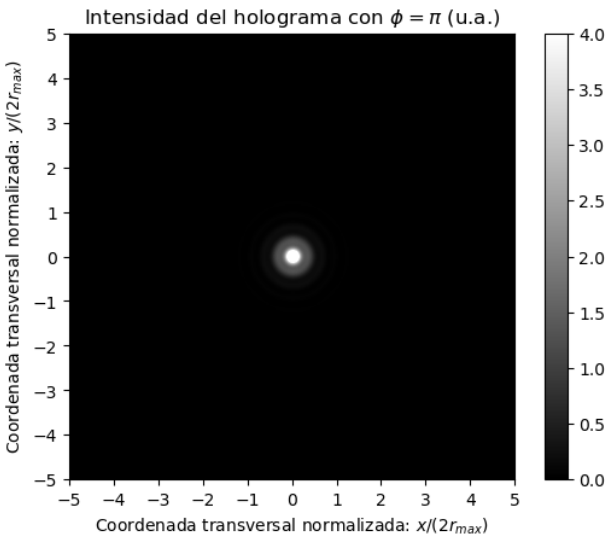
```
plt.plot(A, np.angle(UO_1d), linestyle="--", color = 'r', label = '$\phi_O$␣
 ↪reconstruida', linewidth = 1);
plt.axvline(x = 0, color="black", linestyle=":");
plt.axhline(y = np.max(np.angle(Rec_1d)), color="gray", linestyle = "-.");
plt.text(np.max(A) + np.max(A)/6, np.max(np.angle(Rec_1d)) - 0.05,␣
 ↪r'$\frac{\pi}{2}$', fontsize = 14);
plt.title("Fase original y reconstruida del objeto (rad)");
plt.ylabel("Fase del objeto reconstruido: $\phi_O$ (rad)");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.legend(bbox_to_anchor =(1.4,0.5), loc='center right')
plt.xticks(np.linspace(-5, 5, num = 11));
```

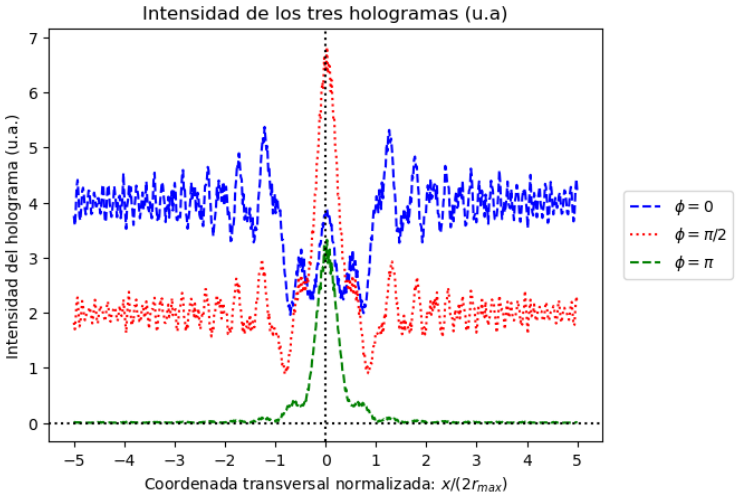


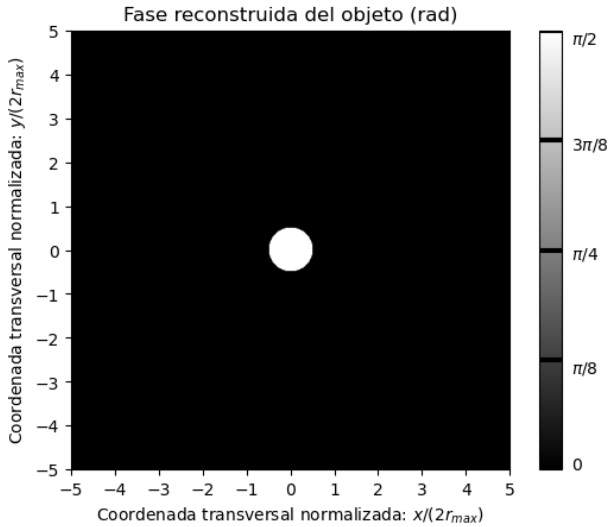


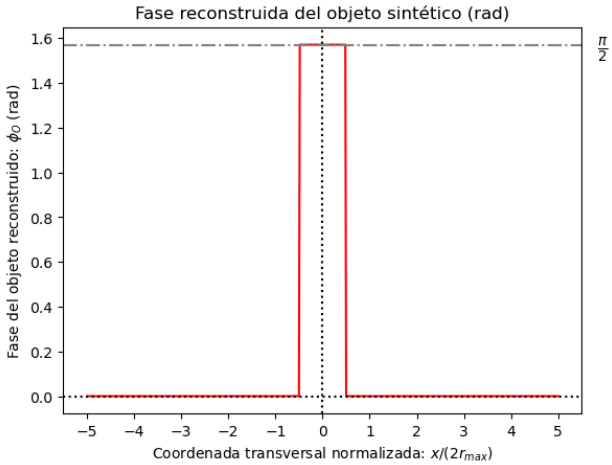

Intensidad del holograma con $\phi = 0$ (u.a.)
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$

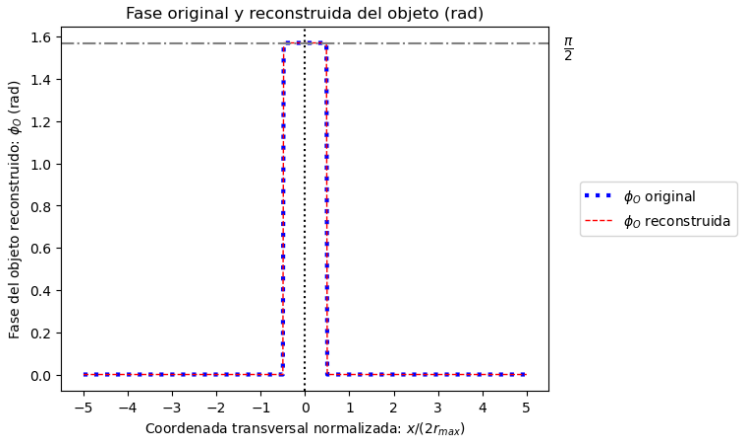

Intensidad del holograma con $\phi = \pi/2$ (u.a.)
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$


Intensidad del holograma con $\phi = \pi$ (u.a.)
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$


Intensidad de los tres hologramas (u.a)
Intensidad del holograma (u.a.)
Coordenada transversal normalizada: $x/(2r_{max})$
$\phi = 0$
$\phi = \pi/2$
$\phi = \pi$


Fase reconstruida del objeto (rad)
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
$\pi/2$
$3\pi/8$
$\pi/4$
$\pi/8$
0

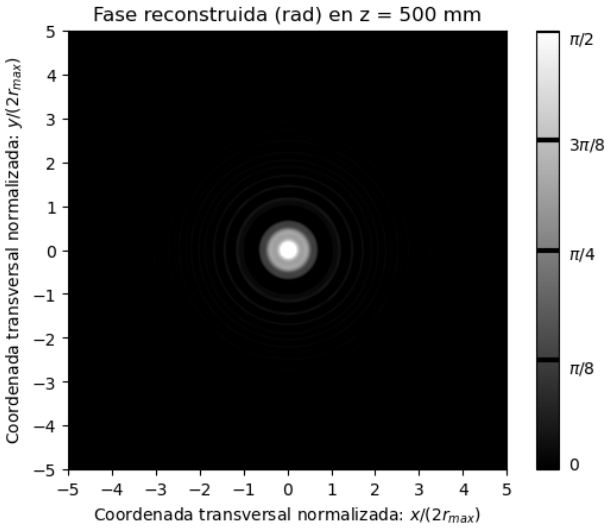

Fase reconstruida del objeto sintético (rad)
Fase del objeto reconstruido: ϕ_O (rad)
Coordenada transversal normalizada: $x/(2r_{max})$
$\frac{\pi}{2}$

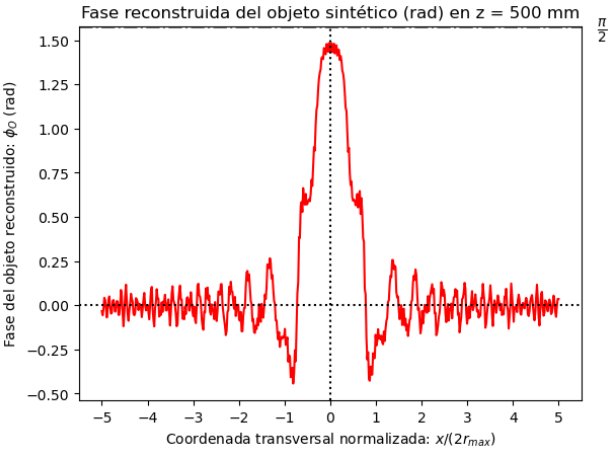


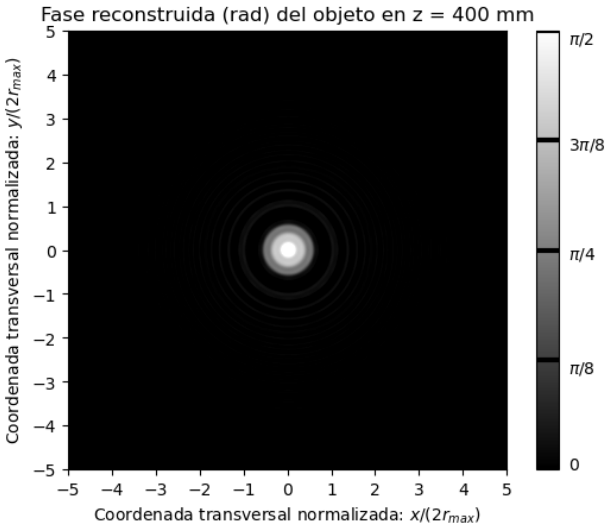
Reenfoque digital del caso 2

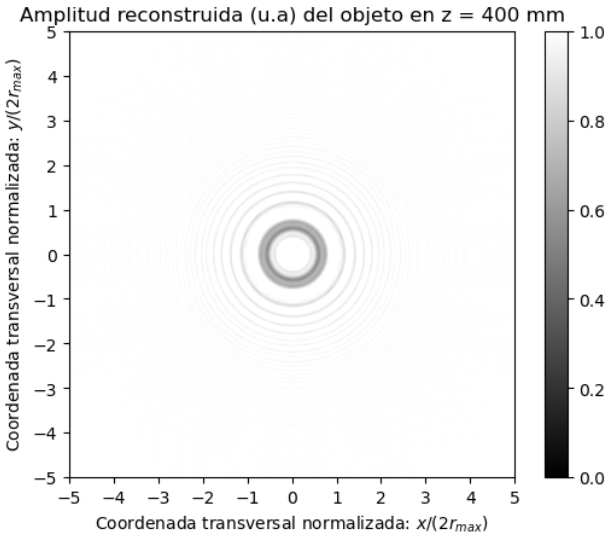
```
z_diffraction = np.array([100,200,300,400,500]);
z_title =([400,300,200,100,0]);
j = 0;
# Representación de la fase reconstruida de la onda objeto en z= 500 mm(2D)
plt.figure();
plt.imshow(np.angle(R_2d),  cmap = 'gray', extent = [-5,5,-5,5], vmin = 0, vmax =␣
 ↪np.pi/2);
cbar_phase = plt.colorbar(ticks = [])
cbar_phase.ax.hlines(data, 0, 1, color="black", linewidth=3);
plt.text(6.4, -5, r'$0$', fontsize=10, color='black');
plt.text(6.4, -2.8, r'$\pi /8$', fontsize=10, color='black');
plt.text(6.4, -0.2, r'$\pi/4$', fontsize=10, color='black');
plt.text(6.4, 2.3, r'$3 \pi /8$', fontsize=10, color='black');
plt.text(6.4, 4.7, r'$\pi/2$', fontsize=10, color='black');
plt.title("Fase reconstruida (rad) en z = 500 mm");
plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$")
plt.xticks(np.linspace(-5, 5, num = 11));
plt.yticks(np.linspace(-5, 5, num = 11));

# Representación del perfil de la fase reconstruida de la onda objeto (1D)
plt.figure();
plt.plot(A, np.angle(R_1d), color ='r');
plt.axhline(y = 0, color = "black", linestyle = ":");
plt.axvline(x = 0, color = "black", linestyle = ":");
plt.axhline(y = np.max(np.angle(Obj_1d)), color =  "gray", linestyle = "-.");
plt.text(np.max(A) + np.max(A)/6, np.max(np.angle(Obj_1d)) - 0.04,␣
 ↪r'$\frac{\pi}{2}$', fontsize = 14);
plt.title("Fase reconstruida del objeto sintético (rad) en z = 500 mm");
plt.ylabel("Fase del objeto reconstruido: $\phi_O$ (rad)");
plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
plt.xticks(np.linspace(-5, 5, num = 11));
```

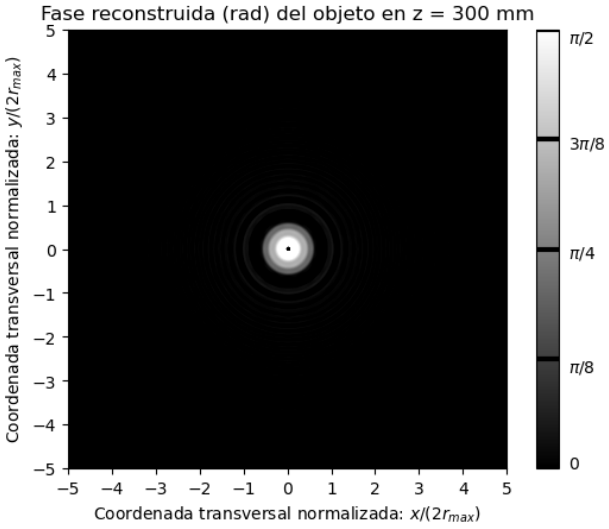
```
for ii in z_diffraction:
    # Reenfoque de la onda objeto reconstruida desde z = 500 hasta z = 0
    UO_1d_, U_x = Fresnel_Diffraction(R_1d, A, B, -ii, lamb, Delta);
    UO_2d_, u_x = Fresnel_Diffraction(R_2d, a, b, -ii, lamb, Delta);

    # Representación de la fase reconstruida de la onda objeto (2D)
    plt.figure();
    cf_phase = plt.imshow(np.angle(UO_2d_),  cmap = 'gray', extent = [-5,5,-5,5],␣
↪vmin = 0, vmax = np.pi/2);
    cbar_phase = plt.colorbar(ticks = [])
    cbar_phase.ax.hlines(data, 0, 1, color="black", linewidth=3);
    plt.text(6.4, -5, r'$0$', fontsize=10, color='black');
    plt.text(6.4, -2.8, r'$\pi /8$', fontsize=10, color='black');
    plt.text(6.4, -0.2, r'$\pi/4$', fontsize=10, color='black');
    plt.text(6.4, 2.3, r'$3 \pi /8$', fontsize=10, color='black');
    plt.text(6.4, 4.7, r'$\pi/2$', fontsize=10, color='black');
    plt.title("Fase reconstruida (rad) del objeto en z = %i mm" %z_title[j]);
    plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
    plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$")
    plt.xticks(np.linspace(-5, 5, num = 11));
    plt.yticks(np.linspace(-5, 5, num = 11));

    # Representación de la amplitud de la onda reconstruida del objeto (2D)
    plt.figure();
    cf_amplitude = plt.imshow(np.sqrt(np.abs(UO_2d_)**2),  cmap = 'gray', extent =␣
↪[-5,5,-5,5], vmin = 0, vmax = 1);
    plt.colorbar(cf_amplitude);
    plt.title("Amplitud reconstruida (u.a) del objeto en z = %i mm" %z_title[j]);
    plt.ylabel("Coordenada transversal normalizada: $y/(2r_{max})$");
    plt.xlabel("Coordenada transversal normalizada: $x/(2r_{max})$");
    plt.xticks(np.linspace(-5, 5, num = 11));
    plt.yticks(np.linspace(-5, 5, num = 11));

    j = j +1;
```

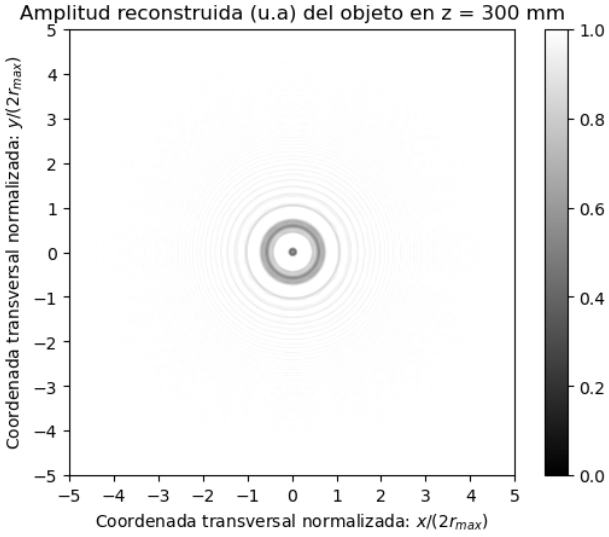

Fase reconstruida (rad) en z = 500 mm
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
$\pi/2$
$3\pi/8$
$\pi/4$
$\pi/8$
0

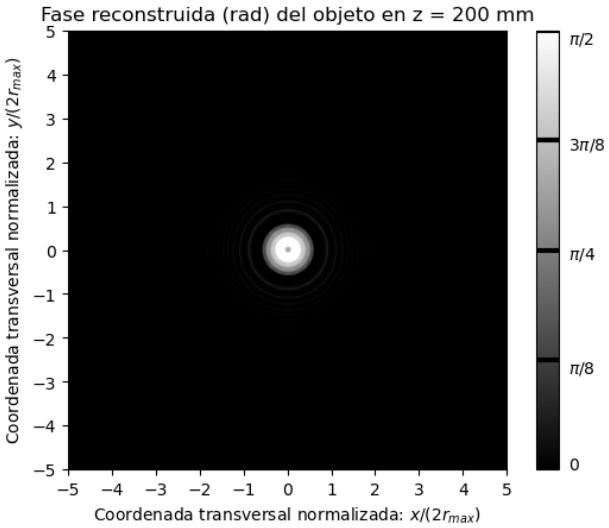

Fase reconstruida del objeto sintético (rad) en z = 500 mm
Fase del objeto reconstruido: ϕ_O (rad)
Coordenada transversal normalizada: $x/(2r_{max})$
$\frac{\pi}{2}$

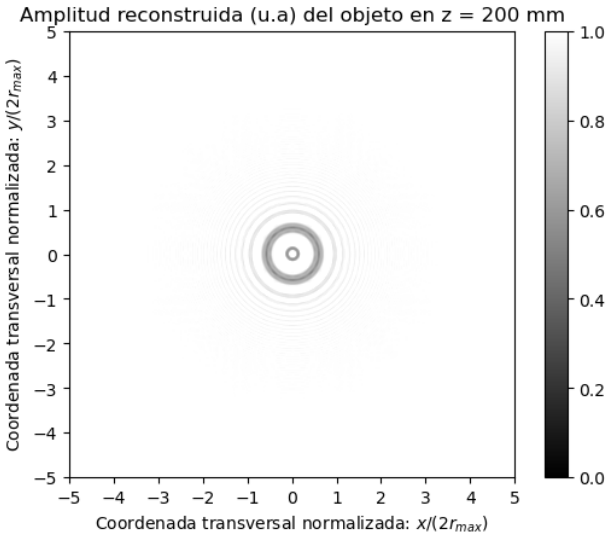

Fase reconstruida (rad) del objeto en z = 400 mm
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
$\pi/2$
$3\pi/8$
$\pi/4$
$\pi/8$
0

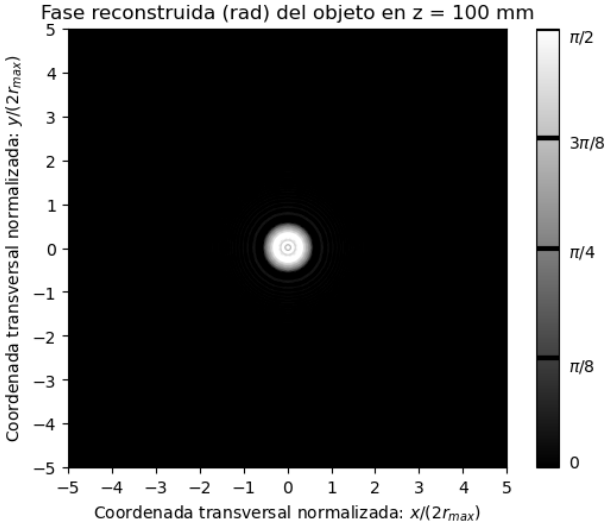

Amplitud reconstruida (u.a) del objeto en z = 400 mm
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
1.0
0.8
0.6
0.4
0.2
0.0

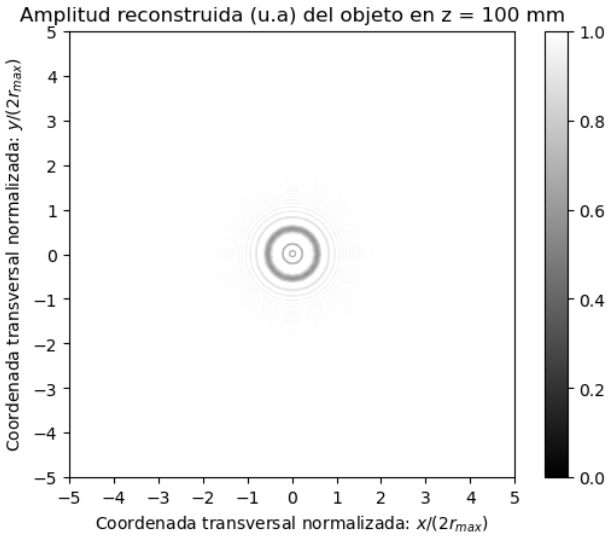

Fase reconstruida (rad) del objeto en z = 300 mm
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
π/2
3π/8
π/4
π/8
0

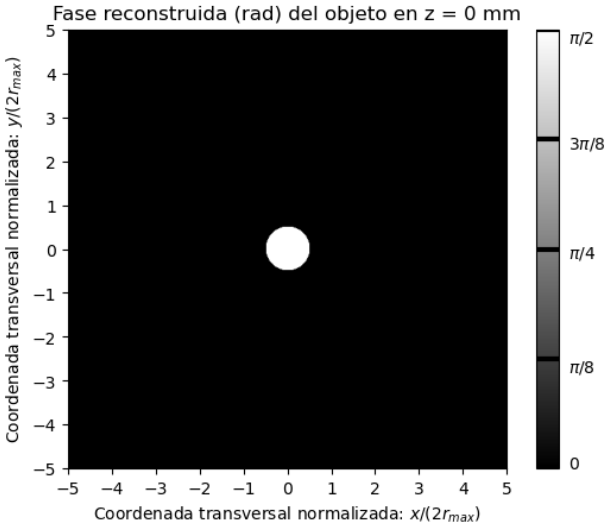

Amplitud reconstruida (u.a) del objeto en z = 300 mm
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
1.0
0.8
0.6
0.4
0.2
0.0

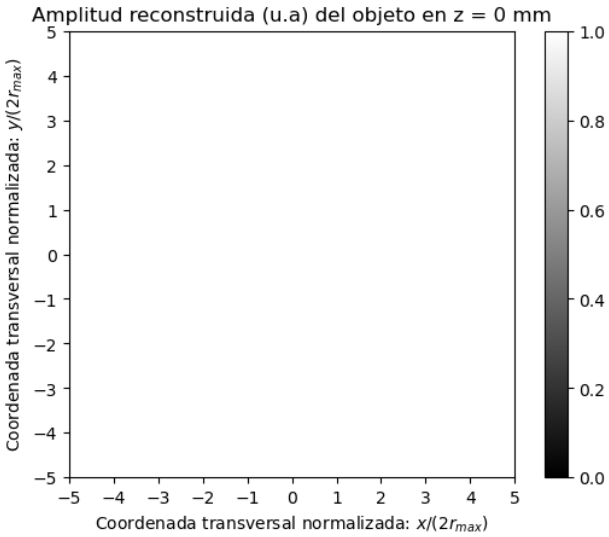

Fase reconstruida (rad) del objeto en z = 200 mm
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
π/2
3π/8
π/4
π/8
0


Amplitud reconstruida (u.a) del objeto en z = 200 mm
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
1.0
0.8
0.6
0.4
0.2
0.0


Fase reconstruida (rad) del objeto en z = 100 mm
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
π/2
3π/8
π/4
π/8
0


Amplitud reconstruida (u.a) del objeto en z = 100 mm
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
1.0
0.8
0.6
0.4
0.2
0.0


Fase reconstruida (rad) del objeto en z = 0 mm
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
π/2
3π/8
π/4
π/8
0


Amplitud reconstruida (u.a) del objeto en z = 0 mm
Coordenada transversal normalizada: $y/(2r_{max})$
Coordenada transversal normalizada: $x/(2r_{max})$
1.0
0.8
0.6
0.4
0.2
0.0