

Summer Project 2023



Pritish Karmakar

21MS179

Contents

1	GAUSSIAN and ITS FOURIER TRANSFORM	1
1.1	Standard normal	1
1.2	Fourier transform of Standard normal	1
2	GAUSSIAN BEAMS	4
2.1	Solutions of the paraxial wave equation	4
2.2	Intensity profile	4
3	HERMITE–GAUSSIAN BEAMS	6
3.1	Solutions of the paraxial wave equation	6
3.2	Intensity profile	6
4	LAGUERRE–GAUSSIAN BEAMS	8
4.1	Solutions of the paraxial wave equation	8
4.2	Intensity profile	8
4.3	Phase plot	9

Listings

1	Standard Normal curve	1
2	Fourier transform of Standard normal	2
	intensity_gaussian.py	4
	intensity_var.py	5
	intensity_hg.py	6
	intensity_lg.py	8
	phase_lg.py	9

1 GAUSSIAN and ITS FOURIER TRANSFORM

1.1 Standard normal

Standard normal curve is

$$f(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$$

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 plt.style.use("classic")
4
5 def f(t):
6     return np.exp(-(t)**2/2)/(np.sqrt(2*np.pi))
7
8 xv = np.linspace(-7,7,1000)
9 yv = f(xv)
10
11 plt.plot(xv, yv, lw=1)
12 plt.xlabel("$t$")
13 plt.ylabel("$f(t)$")
14 plt.grid(True)
```

Listing 1: Standard Normal curve

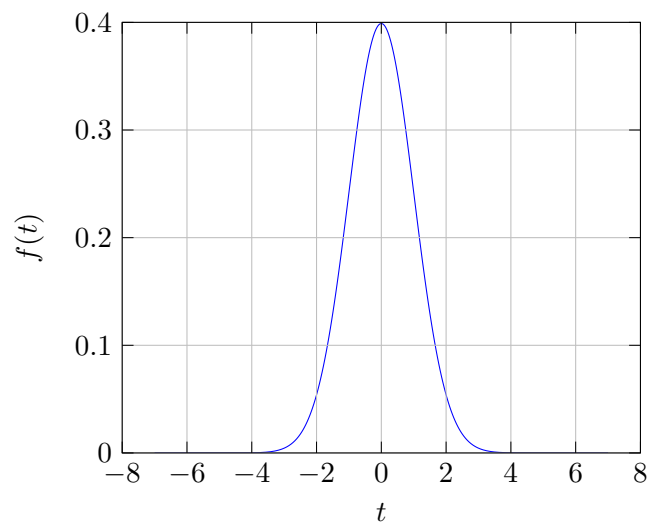


Figure 1: Standard Normal curve

1.2 Fourier transform of Standard normal

If $f(t) = \int_{-\infty}^{\infty} g(\omega) e^{i\omega t} d\omega$, then fourier transform of that is $g(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$. Here, $f(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$

$$\begin{aligned}
g(\omega) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) \cos(\omega t) dt - i \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) \sin(\omega t) dt
\end{aligned}$$

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.integrate import quad
4
5 def f(t):
6     return np.exp(-(t)**2/2)/(np.sqrt(2*np.pi))
7
8 def ft(y):
9     int_re = lambda t: f(t)*np.cos(y*t)
10    int_im = lambda t: f(t)*np.sin(y*t)
11    g_re = quad(int_re,-np.inf,np.inf)[0]/(2*np.pi)
12    g_im = quad(int_im,-np.inf,np.inf)[0]/(2*np.pi)
13    return g_re - 1j*g_im
14 g = np.frompyfunc(ft, 1, 1)
15
16 xv = np.linspace(-7,7,1000)
17 yv = np.abs(g(xv))
18 plt.plot(xv, yv, lw=1)
19 plt.xlabel("$\omega$")
20 plt.ylabel("$abs(g(\omega))$")
21 plt.grid(True)

```

Listing 2: Fourier transform of Standard normal

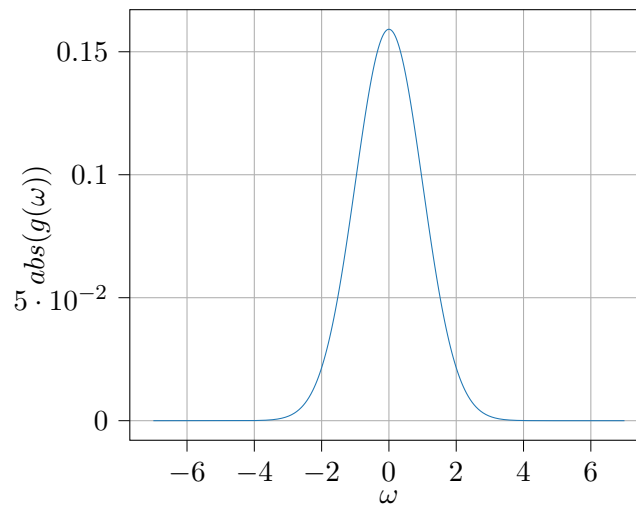


Figure 2: Fourier transform of Standard Normal

Mathematically,

$$\begin{aligned}
g(\omega) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt = \frac{1}{2\pi} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-t^2/2} e^{-i\omega t} dt \\
&= \frac{1}{2\pi} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-t^2/2 - i\omega t} dt = \frac{1}{2\pi} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-(t^2/2 + i\omega t)} dt \\
&= \frac{1}{2\pi} \frac{1}{\sqrt{2\pi}} e^{-\omega^2/2} \int_{-\infty}^{\infty} e^{-(t/\sqrt{2} + i\omega/\sqrt{2})^2} dt \\
&= \frac{1}{2\pi} \frac{1}{\sqrt{\pi}} e^{-\omega^2/2} \int_{-\infty}^{\infty} e^{-\zeta^2} d\zeta \quad [\text{substitute } \zeta = t/\sqrt{2} + i\omega/\sqrt{2}] \\
&= \frac{1}{2\pi} e^{-\omega^2/2} \quad [\text{as } \int_{-\infty}^{\infty} e^{-\zeta^2} d\zeta = \sqrt{\pi}]
\end{aligned}$$

2 GAUSSIAN BEAMS

2.1 Solutions of the paraxial wave equation

$$E(r, z, t) = \psi(r, z)e^{i\omega t}$$
$$\psi(r, z) = A \frac{w_0}{w(z)} \exp \left[\frac{-r^2}{w^2(z)} \right] \exp \left[i \left(kz - \arctan\left(\frac{z}{z_0}\right) + \frac{kr^2}{2R(z)} \right) \right]$$

where,

$$w(z) = w_0 \sqrt{1 + z^2/z_0^2}$$
$$z_0 = \pi w_0^2/\lambda$$
$$R(z) = z + \frac{z_0^2}{z}$$

2.2 Intensity profile

Intensity of Gaussian beam is given by,

$$I(x, y, z) = \frac{c\epsilon}{2} |A|^2 \left[\frac{w_0}{w(z)} \right]^2 \exp(-2(x^2 + y^2)/w^2(z))$$

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # define intensity at z=0, z_0=1
5 R=lambda r: np.exp(-2*r**2)
6 a1=np.linspace(-1.7,1.7,200)
7 xv,yv=np.meshgrid(a1,a1)
8 zv=R(np.sqrt(xv**2+yv**2))
9 plt.contourf(xv,yv,zv,levels=100,cmap='viridis')
10 plt.xlabel("x")
11 plt.ylabel("y")
12 plt.colorbar()
13
14 theta=np.linspace(0,2*np.pi,500)
15 x=np.cos(theta)
16 y=np.sin(theta)
17 plt.plot(x, y, "--", lw=0.8)
```

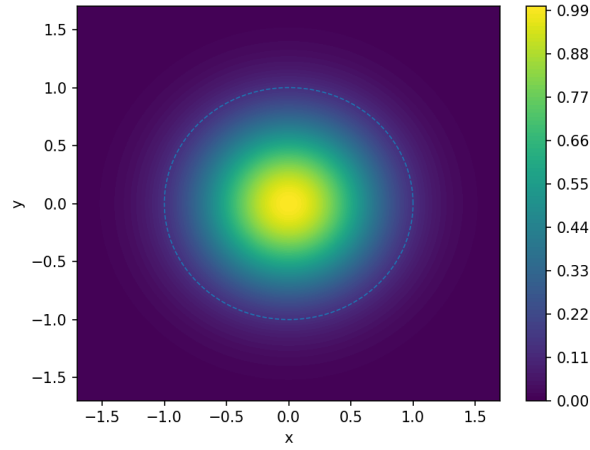


Figure 3: Intensity variation in a cross section ($z = 0, z_0 = 1, w_0 = 1$)

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # define intensity at z_0=1
5 R=lambda x,z: np.exp(-2*x**2/(1+z**2))/(1+z**2)
6 a1=np.linspace(-1.5,1.5,500)
7 a2=np.linspace(-3,3,500)
8 xv,zv=np.meshgrid(a2,a1)
9 I=R(xv,zv)
10 plt.contourf(zv,xv,I,levels=100,cmap='viridis')
11 plt.colorbar()
12 plt.xlabel("z")
13 plt.ylabel("x-y plane")
14
15 w=lambda z1: np.sqrt(1+z1**2)
16 plt.plot(zv,w(zv),"--",lw=0.7)
17 plt.plot(zv,-w(zv),"--",lw=0.7)

```

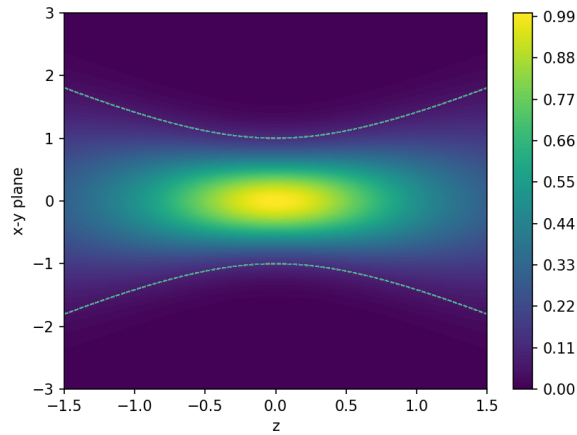


Figure 4: Intensity variation along z ($z_0 = 1, w_0 = 1$)

3 HERMITE-GAUSSIAN BEAMS

3.1 Solutions of the paraxial wave equation

$$E(x, y, z, t) = \psi(x, y, z) e^{i\omega t}$$
$$\psi(x, y, z) = A \frac{w_0}{w(z)} H_m \left(\sqrt{2}x/w(z) \right) H_n \left(\sqrt{2}y/w(z) \right) \exp \left[\frac{-(x^2 + y^2)}{w^2(z)} \right]$$
$$\exp \left[i \left(kz - (m + n + 1) \arctan\left(\frac{z}{z_0}\right) + \frac{k(x^2 + y^2)}{2R(z)} \right) \right]$$

where,

$$w(z) = w_0 \sqrt{1 + z^2/z_0^2}$$
$$z_0 = \pi w_0^2 / \lambda$$
$$R(z) = z + \frac{z_0^2}{z}$$

H_n = n-th order Hermite polynomial

3.2 Intensity profile

Intensity of Hermite-Gaussian beam is given by,

$$I_{m,n}(x, y, z) = \frac{c\epsilon}{2} |A|^2 \left[H_m(\sqrt{2}x/w(z)) \right]^2 \left[H_n(\sqrt{2}y/w(z)) \right]^2 \exp(-2(x^2 + y^2)/w^2(z))$$

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy import special
4
5 H=lambda n: special.hermite(n, monic=False)
6
7 m=int(input("m="))
8 n=int(input("n="))
9
10 # define intensity at z=0, z_0=1
11 intensity=lambda m,n,x,y: (H(m)(np.sqrt(2)*x)*H(n)(np.sqrt(2)*y))**2*np.exp(-2*(
    x**2+y**2))
12
13 l=2
14 a1=np.linspace(-1,1,200)
15 xv,yv=np.meshgrid(a1,a1)
16 zv=intensity(m,n,xv,yv)
17 plt.contourf(xv,yv,zv,levels=100,cmap='viridis')
18 plt.xlabel("x")
19 plt.ylabel("y")
20 #plt.colorbar()
```

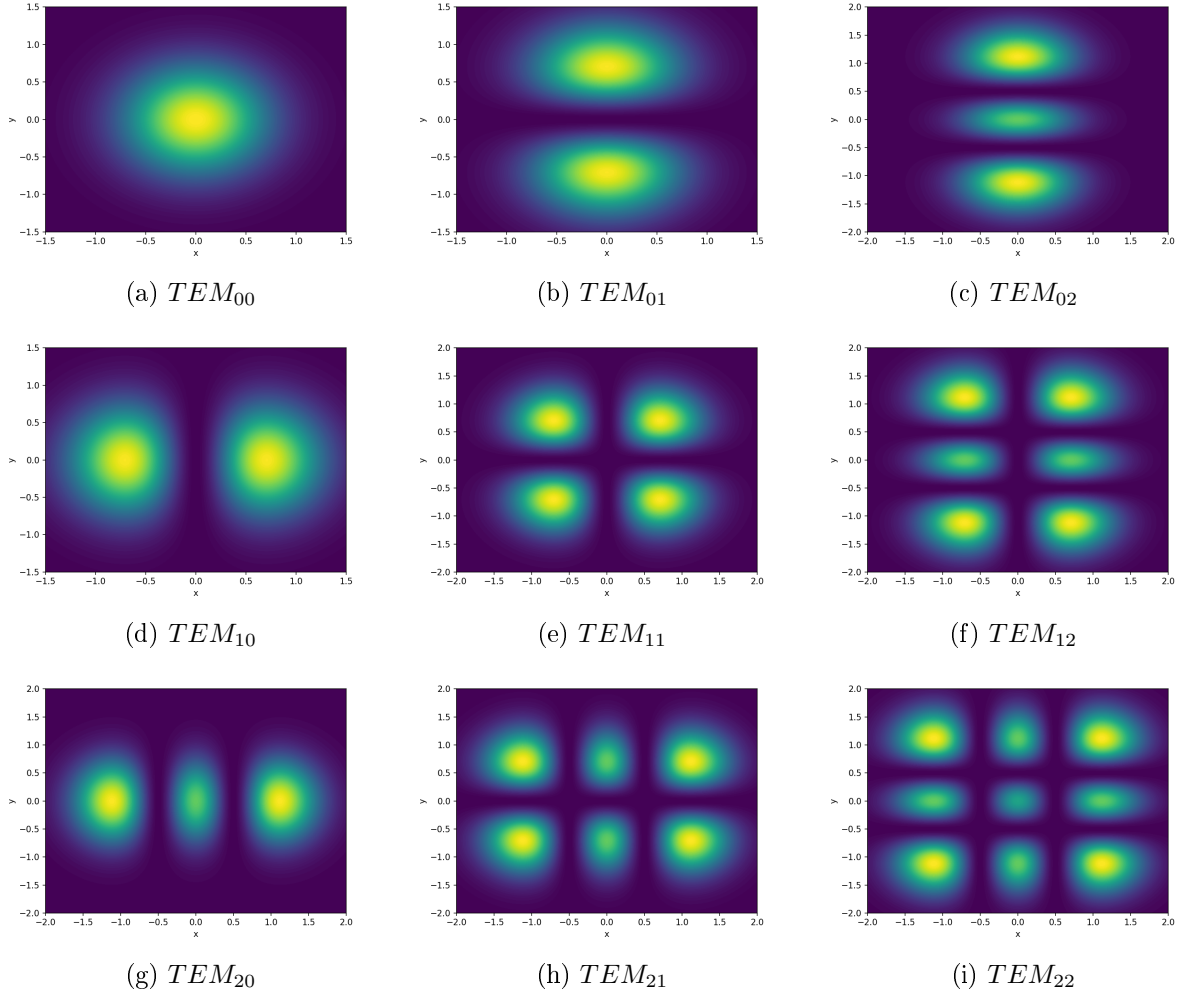


Figure 5: Intensity variation for different TEM in a cross section ($z = 0, z_0 = 1, w_0 = 1$)

4 LAGUERRE-GAUSSIAN BEAMS

4.1 Solutions of the paraxial wave equation

$$E(r, z, t) = \psi_{p,l}(r, z) e^{i\omega t}$$
$$\psi_{p,l}(r, z) = A \frac{w_0}{w(z)} \left[\frac{r\sqrt{2}}{w(z)} \right]^{|l|} L_p^{|l|} \left(\frac{2r^2}{w^2(z)} \right) \exp \left[\frac{-r^2}{w^2(z)} \right]$$
$$\exp \left[i \left(l\phi - (2p + l + 1) \arctan\left(\frac{z}{z_0}\right) + \frac{kr^2}{2R(z)} \right) \right]$$

where,

$$w(z) = w_0 \sqrt{1 + z^2/z_0^2}$$
$$z_0 = \pi w_0^2 / \lambda$$
$$R(z) = z + \frac{z_0^2}{z}$$
$$L_p^{|l|} = \text{Associated Laguerre polynomial}$$

4.2 Intensity profile

Intensity of Hermite-Gaussian beam is given by,

$$I_{p,l}(r, z) = \frac{c\epsilon}{2} |A|^2 \left[\frac{w_0}{w(z)} \right]^2 \left[\frac{r\sqrt{2}}{w(z)} \right]^{2|l|} \left[L_p^{|l|} \left(\frac{2r^2}{w^2(z)} \right) \right]^2 \exp(-2r^2/w^2(z))$$

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy import special
4
5 L= lambda x,p,l: special.assoc_laguerre(x,p,l)
6
7 p=int(input("p="))
8 l=int(input("l="))
9 \
10 # define intensity at z=0, z_0=1
11 Intensity=lambda p,l,x,y: ((np.sqrt(2*(x**2+y**2)))**2*np.abs(1))*(L(2*(x**2+y
    **2),p,np.abs(1)))**2*np.exp(-2*(x**2+y**2))
12
13 a=2
14 a1=np.linspace(-a,a,200)
15 xv,yv=np.meshgrid(a1,a1)
16 zv=Intensity(p,l,xv,yv)
17 plt.contourf(xv,yv,zv,levels=300,cmap='viridis')
18 plt.xlabel("x")
19 plt.ylabel("y")
20 #plt.colorbar()
```

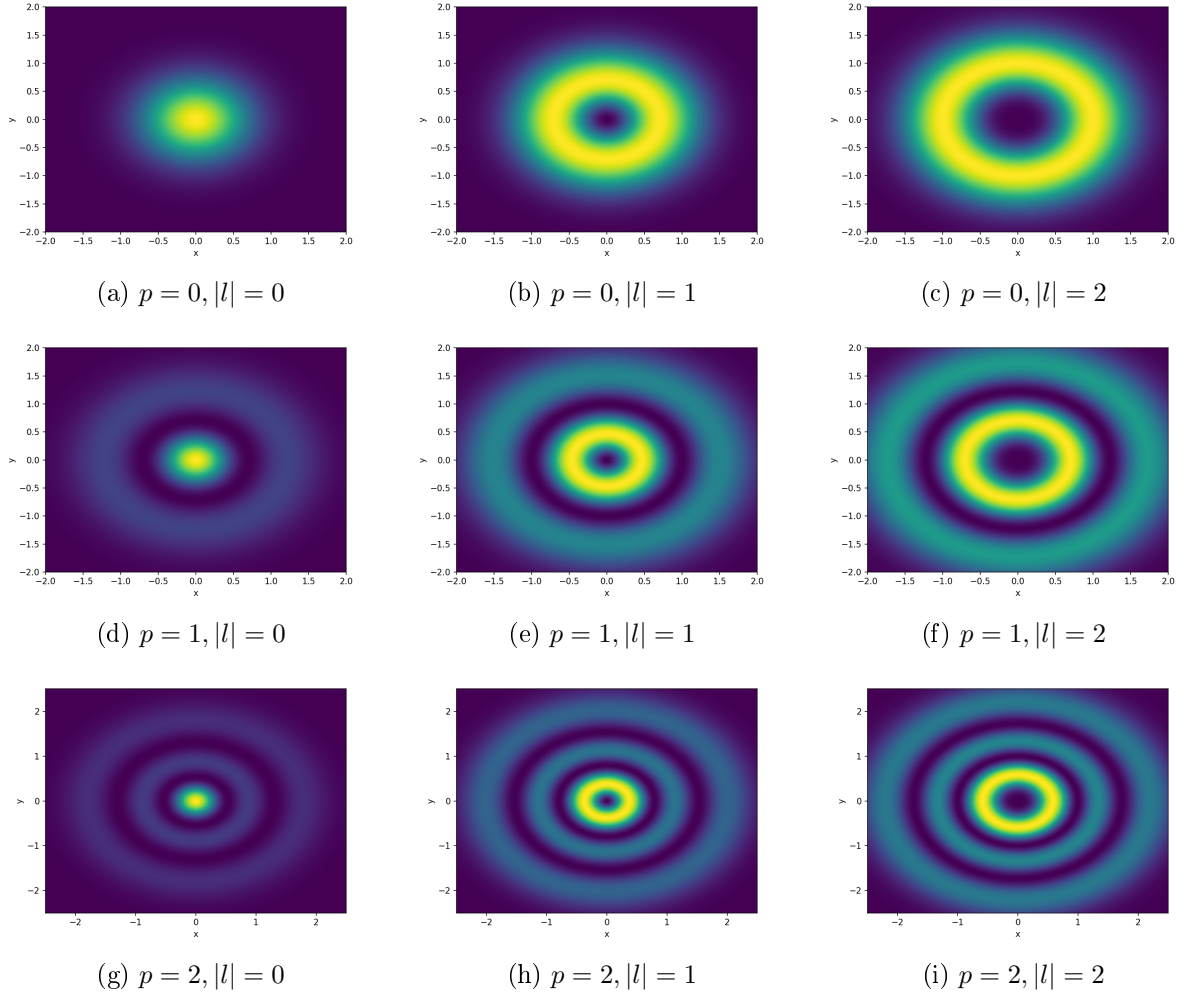


Figure 6: Intensity variation for different modes in a cross section ($z = 0, z_0 = 1, w_0 = 1$)

4.3 Phase plot

Phase difference of Hermite-Gaussian beam at $t = 0$ is given by,

$$\text{Phase} = \arg(\psi_{p,l}(r, z))$$

```

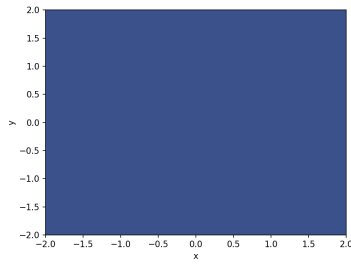
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy import special
4
5 p=int(input("p="))
6 l=int(input("l="))
7
8 # define phi of range 0 to 2*pi
9 def taninv(x, y):
10     return np.angle(x+1j*y)
11 phi_angle = np.vectorize(taninv)
12

```

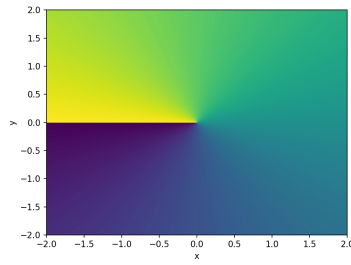
```

13 L= lambda x,p,l: special.assoc_laguerre(x,p,l)
14
15 # define phase function on z=0 plane, z_0=1
16 def R(p,l,r,phi):
17     return np.angle(((np.sqrt(2*(r**2)))**np.abs(l))*(L(2*(r**2),p,np.abs(l)))
18     *np.exp(1j*(l*phi)))
19 phase=np.vectorize(R)
20 a=2
21 a1=np.linspace(-a,a,500)
22 xv,yv=np.meshgrid(a1,a1)
23 zv=phase(p,l,np.sqrt(xv**2+yv**2),phi_angle(xv,yv))
24 plt.contourf(xv,yv,zv,levels=200,cmap='viridis')
25 plt.xlabel("x")
26 plt.ylabel("y")
27 #plt.colorbar()

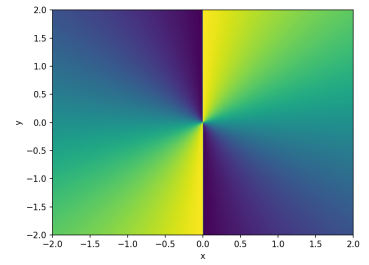
```



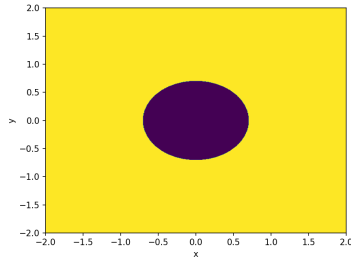
(a) $p = 0, l = 0$



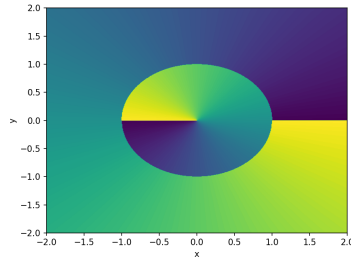
(b) $p = 0, l = 1$



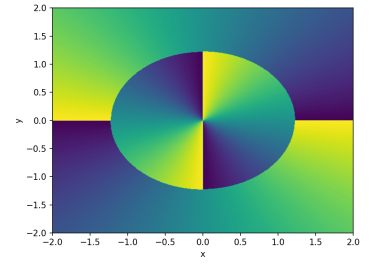
(c) $p = 0, l = 2$



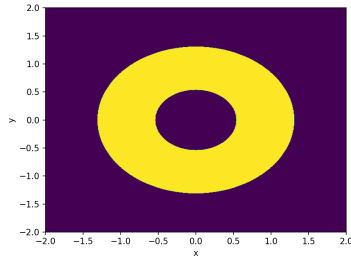
(d) $p = 1, l = 0$



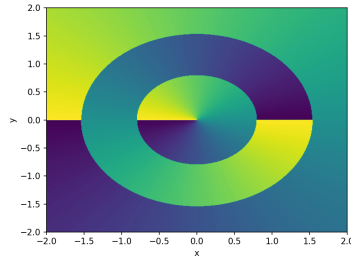
(e) $p = 1, l = 1$



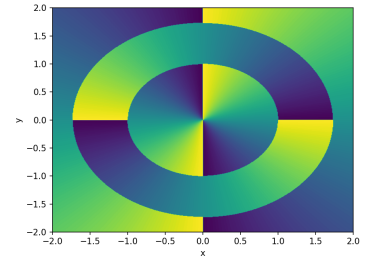
(f) $p = 1, l = 2$



(g) $p = 2, l = 0$



(h) $p = 2, l = 1$



(i) $p = 2, l = 2$

Figure 7: Phase variation for different modes in a cross section ($z = 0, z_0 = 1, w_0 = 1$)