grb

November 22, 2022

1 Radiative Processes: HW4

[45]:
```
#graphing relativistic beaming
```

[46]:
```
import numpy as np
import matplotlib.pyplot as plt
import math
```

[47]:
```
# angular power distributions in the observed K frame
```

[48]:
```
# constant A (given)
#A = (q**2 * a**2)/(4 * np.pi * c**3)
A = 1

# constant phi (given)
phi = 0

# theta array (changing angle)
theta = np.linspace(0, 2*np.pi, 10000)

# gamma values
#gamma = 5, 10, 100
```

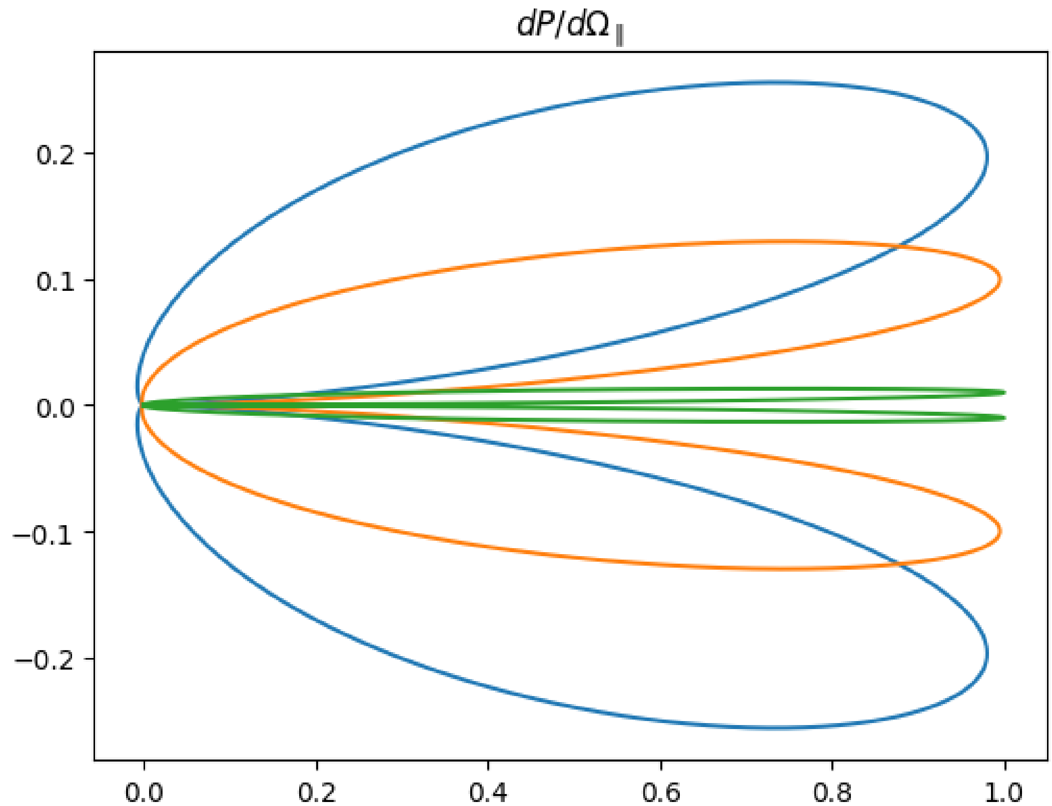
[95]:
```
# define power as a function
#dPdO = ((q**2 a**2)/(4 np.pi c**3))(np.sin(theta))**2
dPdO = A*np.sin(theta)**2

# velocity is dependent on gamma, change beta
#beta = v/c
def beta(gamma):
    return np.sqrt(1 - 1/gamma**2)

def dPpara(gamma, theta):
    dPpara = dPdO/(gamma**2 * (1 - beta(gamma) * np.cos(theta))**2)
    return dPpara

def dPperp(gamma, theta):
```

```python
    dPperp = A * (1/(1 - beta(gamma) * np.cos(theta))**4) * (1 - (((np.
↪sin(theta))**2 * (np.cos(phi))**2)/(gamma**2 * (1 - beta(gamma) * np.
↪cos(theta))**2)))
    return dPperp
```

[96]:
```python
# find power at different gamma values

# define the x and y components of power (parallel)
#define the x and y components of power (perpendicular)
xpara_5 = np.cos(theta) * dPpara(5, theta)
ypara_5 = np.sin(theta) * dPpara(5, theta)

xperp_5 = np.cos(theta) * dPperp(5, theta)
yperp_5 = np.sin(theta) * dPperp(5, theta)

xpara_10 = np.cos(theta) * dPpara(10, theta)
ypara_10 = np.sin(theta) * dPpara(10, theta)

xperp_10 = np.cos(theta) * dPperp(10, theta)
yperp_10 = np.sin(theta) * dPperp(10, theta)

xpara_100 = np.cos(theta) * dPpara(100, theta)
ypara_100 = np.sin(theta) * dPpara(100, theta)

xperp_100 = np.cos(theta) * dPperp(100, theta)
yperp_100 = np.sin(theta) * dPperp(100, theta)
```

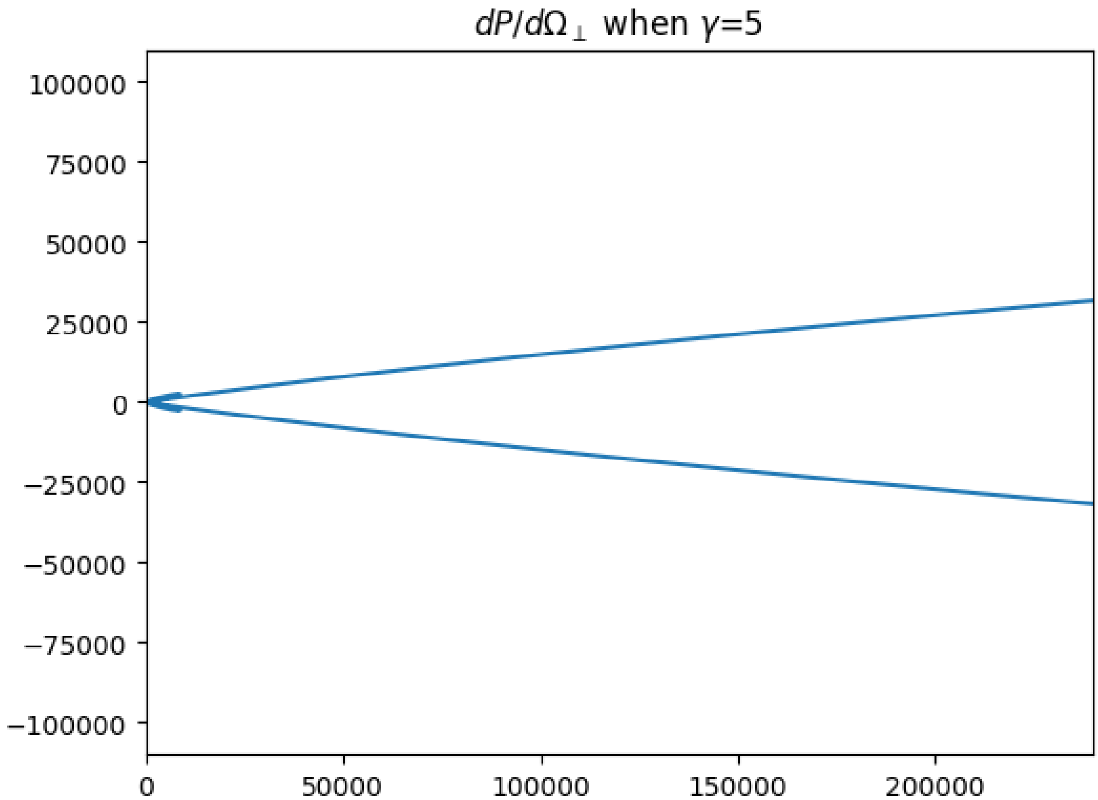
[97]:
```python
# plot power as a function of theta
plt.title('$dP/d\Omega_\parallel$')
plt.plot(xpara_5, ypara_5)
plt.plot(xpara_10, ypara_10)
plt.plot(xpara_100, ypara_100)
```

[97]:
```
[<matplotlib.lines.Line2D at 0x23528093760>]
```



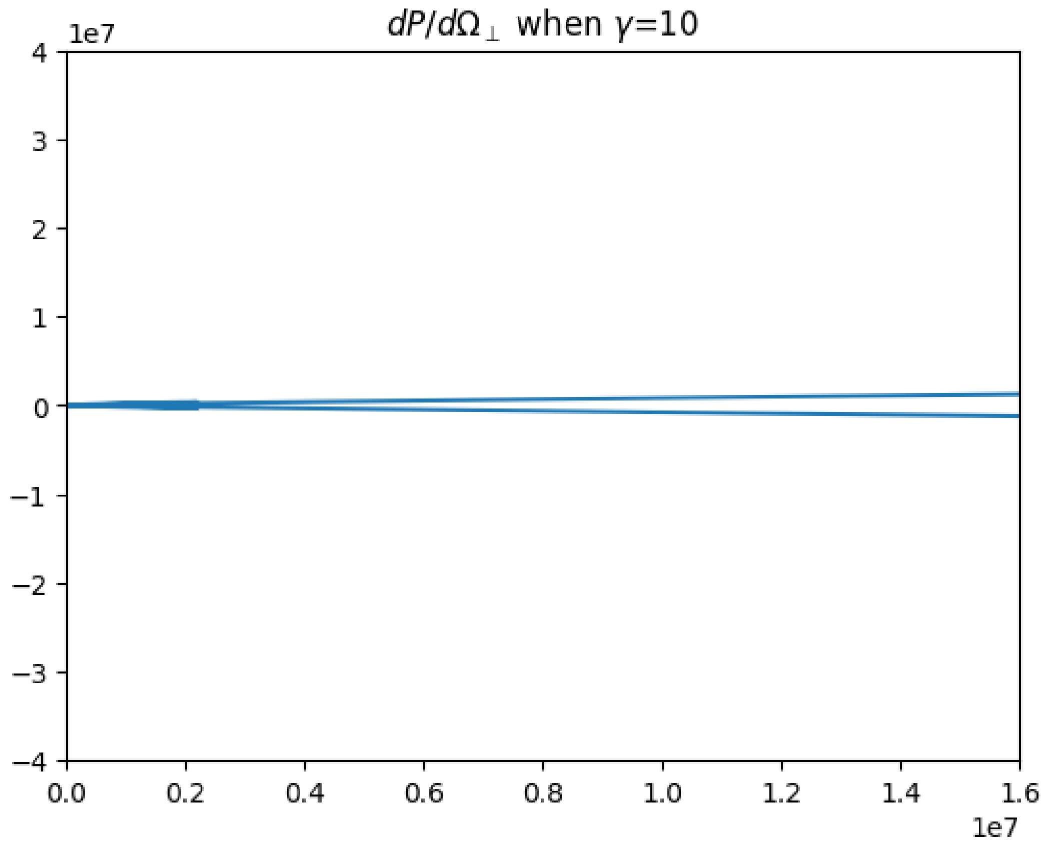
[98]:
```
plt.title('$dP/d\Omega_\perp$ when $\gamma$=5')
plt.plot(xperp_5, yperp_5)
plt.xlim(0, 2.4e5)
plt.ylim(-1.1e5, 1.1e5)
```

[98]: (-110000.0, 110000.0)



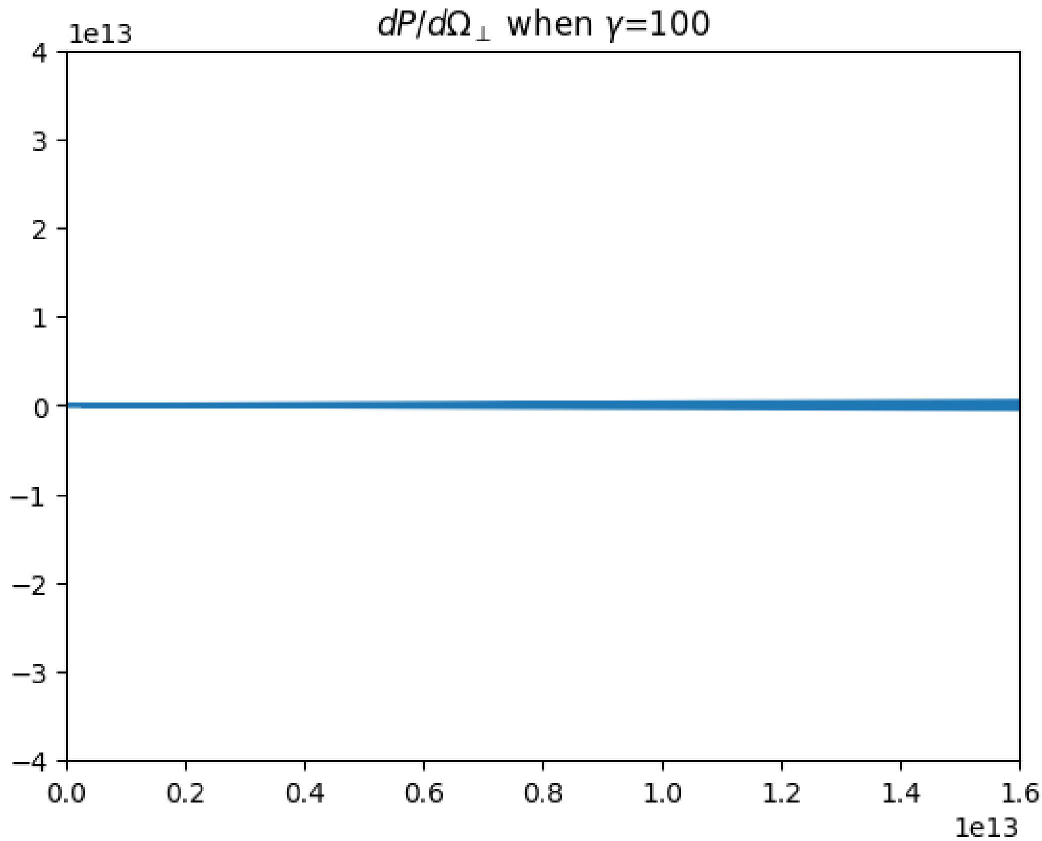
[99]:
```
plt.title('$dP/d\Omega_\perp$ when $\gamma$=10')
plt.plot(xperp_10, yperp_10)
plt.xlim(0, 1.6e7)
plt.ylim(-4e7, 4e7)
```

[99]: (-40000000.0, 40000000.0)



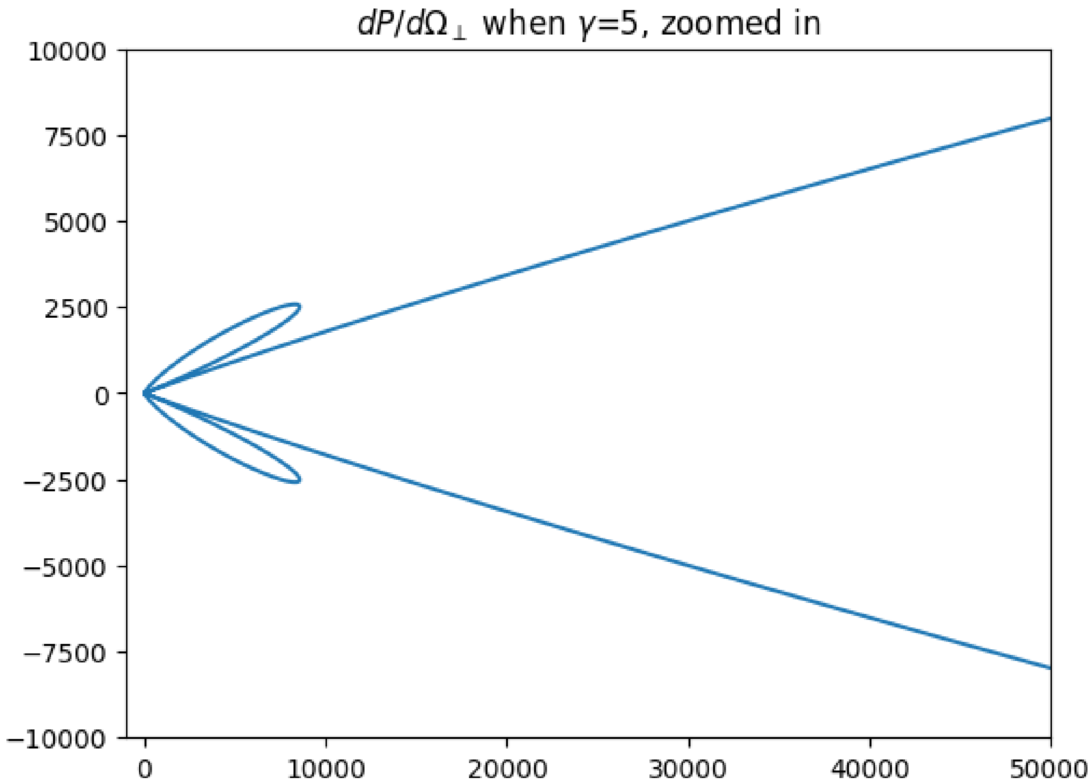
[100]:
```
plt.title('$dP/d\Omega_\perp$ when $\gamma$=100')
plt.plot(xperp_100, yperp_100)
plt.xlim(0, 1.6e13)
plt.ylim(-4e13, 4e13)
```

[100]: (-40000000000000.0, 40000000000000.0)



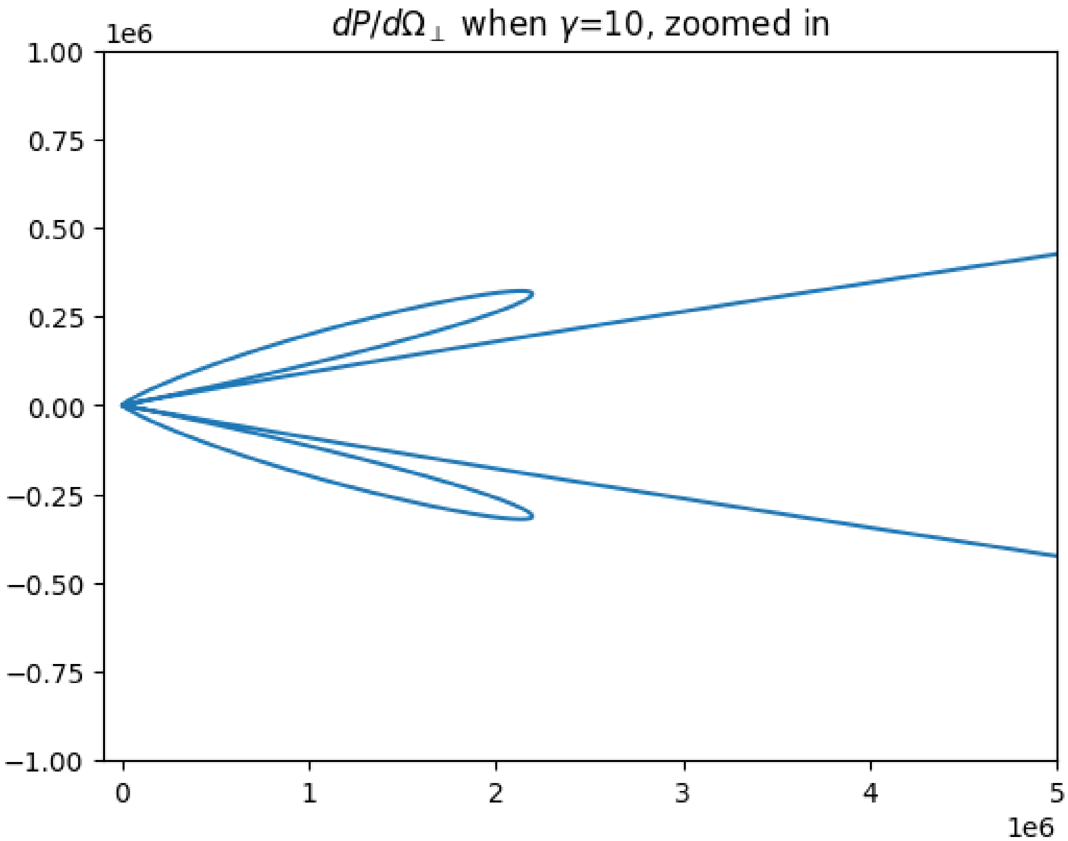
[117]:
```
plt.title('$dP/d\Omega_\perp$ when $\gamma$=5, zoomed in')
plt.plot(xperp_5, yperp_5)
plt.xlim(-.01e5, .5e5)
plt.ylim(-.1e5, .1e5)
```

[117]: (-10000.0, 10000.0)



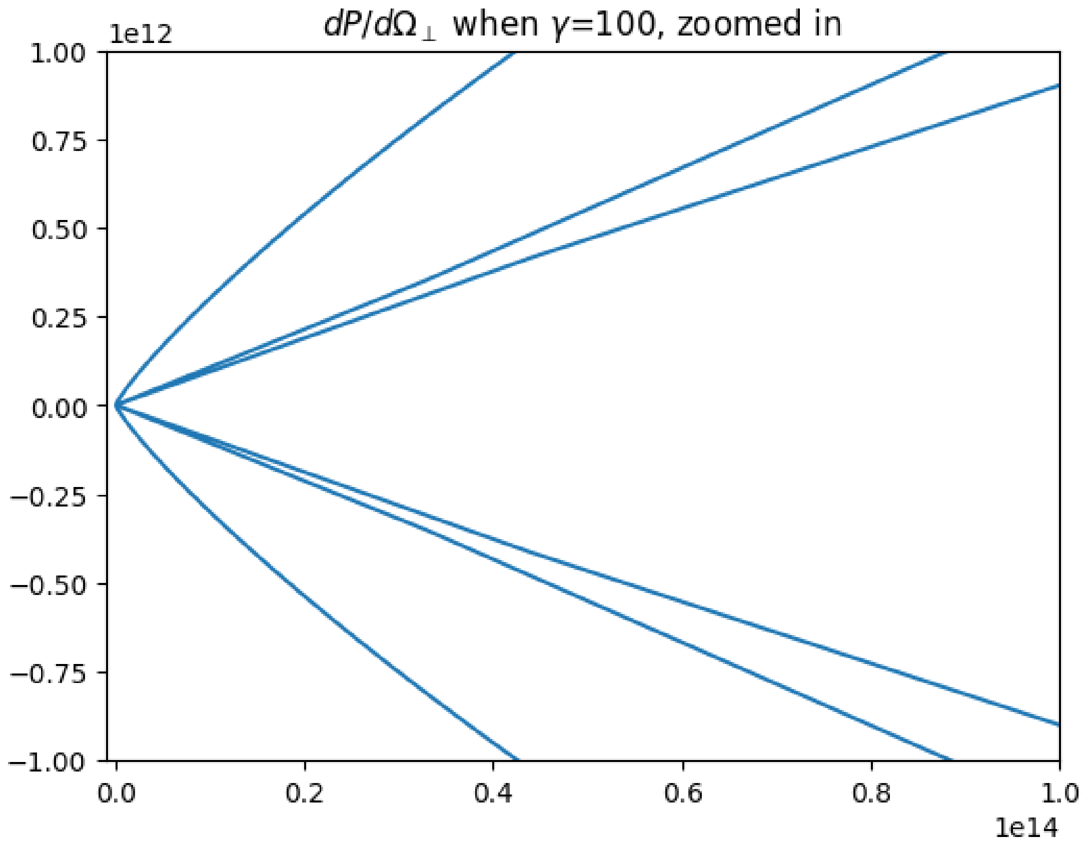
[118]:
```
plt.title('$dP/d\Omega_\perp$ when $\gamma$=10, zoomed in')
plt.plot(xperp_10, yperp_10)
plt.xlim(-.01e7, .5e7)
plt.ylim(-.1e7, .1e7)
```

[118]: (-1000000.0, 1000000.0)



[125]:
```
plt.title('$dP/d\Omega_\perp$ when $\gamma$=100, zoomed in')
plt.plot(xperp_100, yperp_100)
plt.xlim(-.1e13, 10e13)
plt.ylim(-.1e13, .1e13)
```

[125]: (-1000000000000.0, 1000000000000.0)


$dP/d\Omega_\perp$ when γ=100, zoomed in
1e12
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0
1e14

[ ]: