In [226…

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d
from astropy.table import Table
```

In [390…

```python
cd /home/michelle/Downloads/blackhawk/results
```

```
/home/michelle/Downloads/blackhawk/results
```

Formulas and constants

$ R = \frac{2MG}{c^2}$

$T_{\mathrm {H} }={\frac {\hbar c^{3}}{8\pi GMk_{\mathrm {B} }}}$

$F = \sigma T^4$

$ L = AF = A\sigma T^4$

In [5]:

```python
G = 6.6743e-8
c = 3e10
h = 1.0546e-27
kB = 1.380649e-16
sb = 5.670374e-5
```

In [223…

```python
#Radius as function of mass
def R(mass):
    return 2*mass * G/c**2

#Temperature as function of mass for Schwarzchild BH
def Tsc(mass):
    top = h*c**2
    bottom = 8*np.pi*G*mass*kB
    return top/bottom

#Surface area as function of mass
def A(mass):
    return 4 * np.pi * R(mass)**2

#Lumosity as function of mass
def L(mass):
    return sb*A(mass)*Tsc(mass)**4
```

For for Plotting Mass Decay

In [407…
```
def lifetime(path, number = 0, specific = 0):
    """Generates plot of blackhole decay

    Args:
        path (str): path to mass decay file
        number (int): numbers of blackhole generated
        specific (int): which blackhole if there is multiple (0 for all). 1
    """
    plt.title('Lifetime Evolution')
    plt.xlabel('time [s]')
    plt.ylabel('mass [g]')

    if number == 0:
        data = np.loadtxt(path, skiprows = 4)
        plt.plot(data[:, 0], data[:, 1])
    else:
        fin = open(path)
        fin.readline()
        t = fin.readline()
        t = t.split(': ')[1]
        t = int(t)

        if specific == 0:
            start = 1
            end = number + 1
        else:
            start = specific
            end = specific + 1

        for i in np.arange(start, end):
            data = np.loadtxt(path, skiprows = (4*i + (i-1)*t), max_rows = (
            plt.plot(data[:, 0], data[:, 1], label = 'BH {}'.format(i))
            plt.legend()
```

For Plotting Particle Energy Decay Rate

In [267…

```
def particle_decay_tot(path, n, xscale = 'linear', yscale = 'log'):
    """Generates plot of particle decay rate

    Args:
        path (str): path to file
        n (int): numbers energy states
    """

    energy = np.loadtxt(path, skiprows = 1, usecols = (np.arange(1, n+1)), m
    values = np.loadtxt(path, skiprows = 2)

    for i in np.arange(len(values[0])-1):
        plt.plot(values[:, 0], values[:, i+1], label = 'energy = {}'.format(

    plt.title('Particle Decay rate vs Energy')
    plt.legend()
    plt.xlabel('time [s]')
    plt.ylabel('decay rate [GeV^-1 s^-1 cm^-3]')
    plt.yscale(yscale)
    plt.xscale(xscale)
```

In [259…
```
def particle_decay_inst(path, n, dm = True, particles = 'all', xscale = 'lin
    """Generates plot of inital decay rate

    Args:
        path (str): path to file
        n (int): numbers energy states
        dm (bool): is there dark matter decay
        particles (list): particles decay to plot
    """

    data = np.loadtxt(path, skiprows = 2)
    if dm is True:
        names = ['energy', 'photon', 'gluons','higgs', 'W+-', 'Z0',
                 'neutrinos', 'electron', 'muon', 'tau', 'up', 'down',
                 'charm', 'strange', 'top', 'bottom','DM']
    else:
        names = ['energy', 'photon', 'gluons','higgs', 'W+-', 'Z0',
                 'neutrinos', 'electron', 'muon', 'tau', 'up', 'down',
                 'charm', 'strange', 'top', 'bottom']
    rates = Table(data, names = names)

    if particles == 'all':
        particles = names

    for p in particles:
        plt.plot(rates['energy'], rates[p], label = p)

    plt.title('Particle Decay rate vs Energy')
    plt.legend()
    plt.xlabel('time [s]')
    plt.ylabel('decay rate [GeV^-1 s^-1 cm^-3]')
    plt.yscale(yscale)
    plt.xscale(xscale)
```

Lumosity vs Mass for Entire Blackhole

In [378…

```
def Lum_vs_M(path, xscale = 'linear', yscale = 'log', number = 1, specific =

    try:
        data = np.loadtxt(path, skiprows = 4)
        m = data[:, 1]
        plt.plot(m, L(m))

    except:
        fin = open(path)
        fin.readline()
        t = fin.readline()
        t = t.split(': ')[1]
        t = int(t)

        if specific == 0:
            start = 1
            end = number+1
        else:
            start = specific
            end = specific + 1

        for i in np.arange(start, end):
            data = np.loadtxt(path, skiprows = (4*i + (i-1)*t), max_rows = (
            plt.plot(data[:, 1], L(data[:, 1]), label = 'BH {}'.format(i))
            plt.legend()


    plt.title('Mass vs Luminosity')
    plt.xlabel('Mass [g] (inverted axis)')
    plt.ylabel('Lumosity [erg s^-1]')
    plt.yscale(yscale)
    plt.xscale(xscale)
    plt.gca().invert_xaxis()
```

Power Spectra for Particles

This paper got a blackbody like spectra. I found it from the blackhawk manual.

https://journals.aps.org/prd/pdf/10.1103/PhysRevD.13.198

In [400…

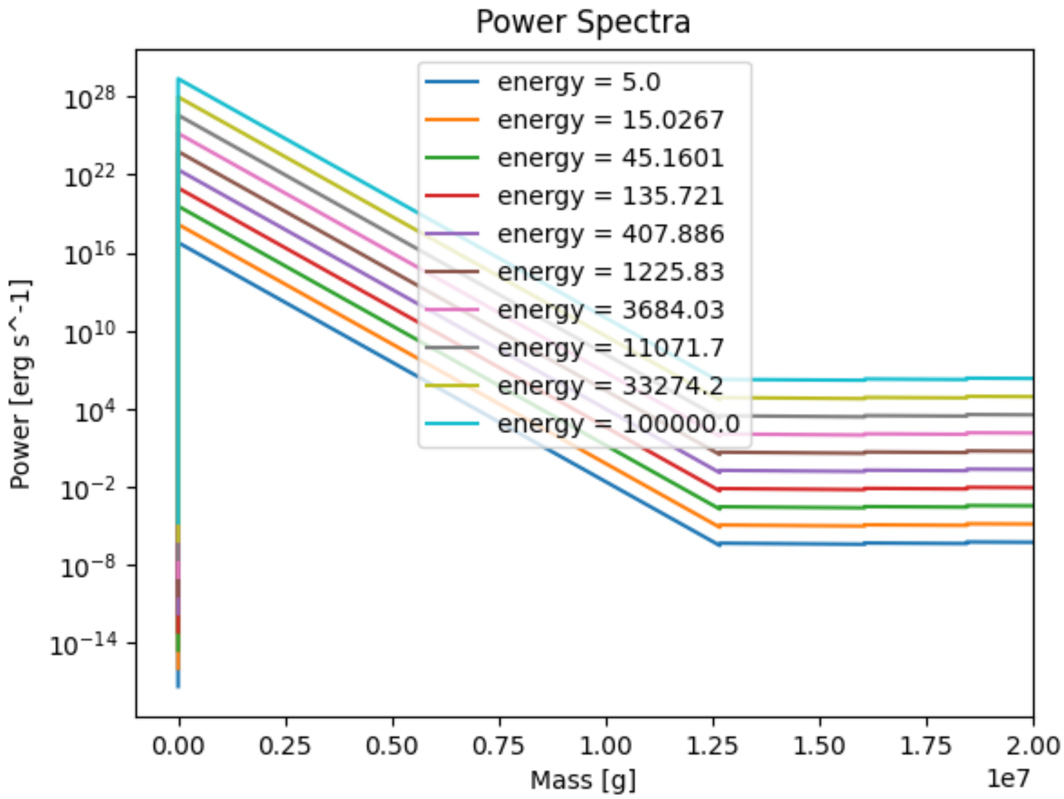
```
def particle_power_spectra(path, n, xscale = 'linear', yscale = 'log'):
    """Generates plot of particle decay rate

    Args:
        path (str): path to file
        n (int): numbers energy states
    """

    folder = path.split('/')[0]

    energy = np.loadtxt(path, skiprows = 1, usecols = (np.arange(1, n+1)), m
    values = np.loadtxt(path, skiprows = 2)
    massdecay = np.loadtxt(folder + '/life_evolutions.txt', skiprows = 4)
    time = values[:, 0]
    interM = interp1d(massdecay[:, 0], massdecay[:, 1])
    M = interM(time)

    for i in np.arange(len(values[0])-1):
        #plt.plot(interM(time), interM(time)* values[:, i+1], label = 'energ
        plt.plot(M, L(M) * values[:, i+1], label = 'energy = {}'.format(ener

    plt.title('Power Spectra')
    plt.legend()
    plt.xlabel('Mass [g]')
    plt.ylabel('Power [erg s^-1]')
    plt.yscale(yscale)
    plt.xscale(xscale)
```

In [401…

```
%matplotlib widget
particle_power_spectra('test/photon_primary_spectrum.txt', 10)
plt.xlim(-.01e8,  0.2e8)
```

Out[401]: (-1000000.0, 20000000.0)

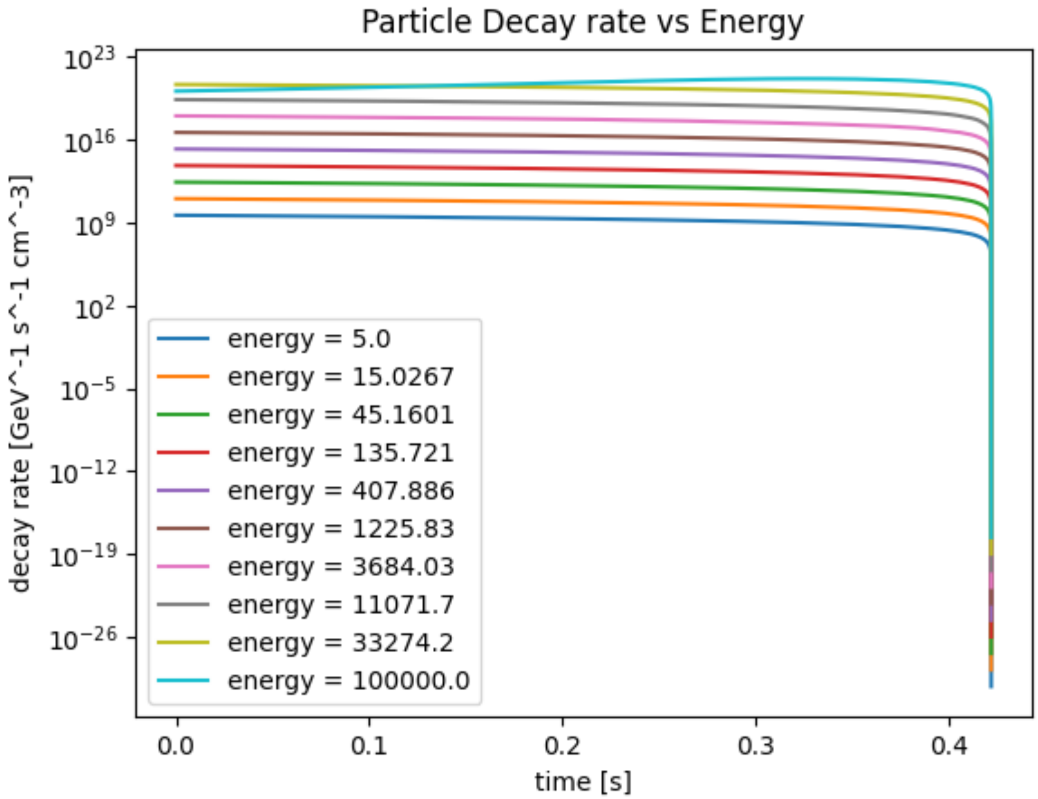
Figure



In [402…

```
%matplotlib widget
particle_decay_tot('test/photon_primary_spectrum.txt', 10)
```

Figure


Particle Decay rate vs Energy
decay rate [GeV^-1 s^-1 cm^-3]
10^23
10^16
10^9
10^2
10^-5
10^-12
10^-19
10^-26
0.0
0.1
0.2
0.3
0.4
time [s]
energy = 5.0
energy = 15.0267
energy = 45.1601
energy = 135.721
energy = 407.886
energy = 1225.83
energy = 3684.03
energy = 11071.7
energy = 33274.2
energy = 100000.0

In [ ]: