

Bailey Piotrowski, PHYS 320 Project

Modelling Quantum Tunneling Rates for the First Step in the Proton-Proton Chain Across Stellar Spectral Type

In my project, I will be calculating fusion reaction rates across stars of different spectral types. Specifically, I will be looking at the first step of the proton-proton chain in nucleosynthesis, with the quantum tunneling of one proton into another. The broad categories of spectral type are: O, B, A, F, G, K, and M. They are further divided into subcategories, being assigned their spectral type letter along with a number between 1 and 9 to specify where within a spectral type a given star falls. For my calculation, I will be looking at the intermediate subcategories of these spectral types: O5, B5, A5, F5, G5, K5, and M5. I used mass and radius values for stars for given spectral types from the very helpful table found at:

<http://www.isthe.com/chongo/tech/astro/HR-temp-mass-table-byhrclass.html> (<http://www.isthe.com/chongo/tech/astro/HR-temp-mass-table-byhrclass.html>).

```
In [2]: %pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

```

In [3]: G = 6.67408e-11 #m^3 * kg^-1 * s^-2
m_sun = 1.989e30 #kg
m_proton = 1.6726e-27 #kg
r_sun = 6.957e8 #m
k_b = 1.3807e-23 #kg * m^2 * s^-2 * K^-1

n_o5 = 60
m_o5 = 14.2
T_o5 = (G*(n_o5*m_sun)*m_proton)/((m_o5*r_sun)*k_b)

n_b5 = 5.9
m_b5 = 3.26
T_b5 = (G*(n_b5*m_sun)*m_proton)/((m_b5*r_sun)*k_b)

n_a5 = 1.9
m_a5 = 1.89
T_a5 = (G*(n_a5*m_sun)*m_proton)/((m_a5*r_sun)*k_b)

n_f5 = 1.4
m_f5 = 1.48
T_f5 = (G*(n_f5*m_sun)*m_proton)/((m_f5*r_sun)*k_b)

n_g5 = 0.9
m_g5 = 1.06
T_g5 = (G*(n_g5*m_sun)*m_proton)/((m_g5*r_sun)*k_b)

n_k5 = 0.7
m_k5 = 1
T_k5 = (G*(n_k5*m_sun)*m_proton)/((m_k5*r_sun)*k_b)

n_m5 = 0.2
m_m5 = 0.38
T_m5 = (G*(n_m5*m_sun)*m_proton)/((m_m5*r_sun)*k_b)

print('The core temperature of the average O5 star is:', T_o5,'K')
print('The core temperature of the average B5 star is:', T_b5,'K')
print('The core temperature of the average A5 star is:', T_a5,'K')
print('The core temperature of the average F5 star is:', T_f5,'K')
print('The core temperature of the average G5 star is:', T_g5,'K')
print('The core temperature of the average K5 star is:', T_k5,'K')
print('The core temperature of the average M5 star is:', T_m5,'K')

The core temperature of the average O5 star is: 97669701.73356695 K
The core temperature of the average B5 star is: 41834190.2415043 K
The core temperature of the average A5 star is: 23237465.19198974 K
The core temperature of the average F5 star is: 21865694.487199448 K
The core temperature of the average G5 star is: 19626081.574763928 K
The core temperature of the average K5 star is: 16180613.92052759 K
The core temperature of the average M5 star is: 12165875.128216237 K

```

Above, I have calculated the core temperatures of these stars using an equation that approximates core temperature:

$T_{core} = (G * M_{star} * m_{proton}) / (R_{star} * k_b)$, where G is the gravitational constant and k_b is the Boltzmann constant.

```

In [4]: k = 8.61733e-8 #keV * K^-1
E = np.linspace(0.001, 20, num=1e4)

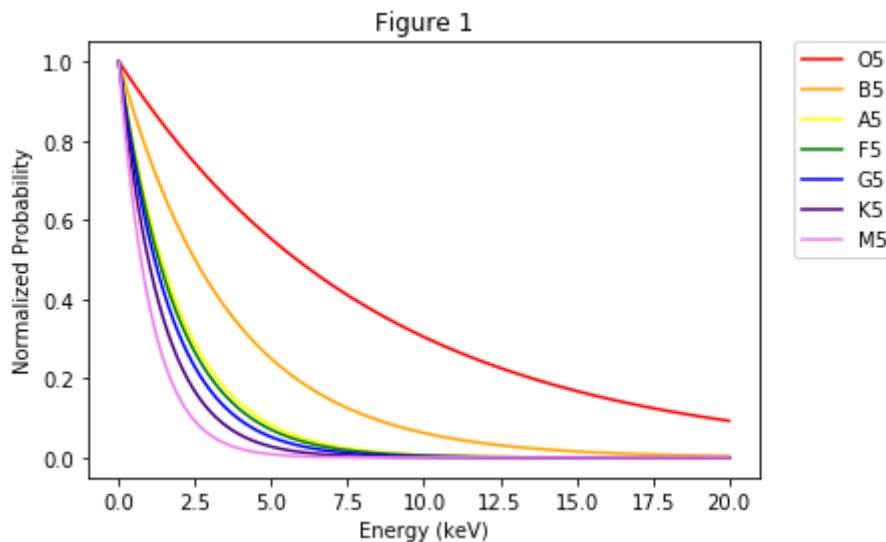
maxwell_o5 = e**(-E/(k*T_o5))
maxwell_b5 = e**(-E/(k*T_b5))
maxwell_a5 = e**(-E/(k*T_a5))
maxwell_f5 = e**(-E/(k*T_f5))
maxwell_g5 = e**(-E/(k*T_g5))
maxwell_k5 = e**(-E/(k*T_k5))
maxwell_m5 = e**(-E/(k*T_m5))

plt.plot(E, maxwell_o5, 'red', label="O5")
plt.plot(E, maxwell_b5, 'orange', label="B5")
plt.plot(E, maxwell_a5, 'yellow', label="A5")
plt.plot(E, maxwell_f5, 'green', label="F5")
plt.plot(E, maxwell_g5, 'blue', label="G5")
plt.plot(E, maxwell_k5, 'indigo', label="K5")
plt.plot(E, maxwell_m5, 'violet', label="M5")
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('Energy (keV)')
plt.ylabel('Normalized Probability')
plt.title('Figure 1')

```

/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:2: DeprecationWarning: object of type <class 'float'> cannot be safely interpreted as an integer.

Out[4]: Text(0.5, 1.0, 'Figure 1')



Above I have produced Maxwellian distributions of protons for each spectral type. From here, I will generate a curve representing the probability that a proton will tunnel into another in a stellar core. This is determined by the equation:

$$Probability = e^{-\sqrt{E_{Gamow}}/E}$$

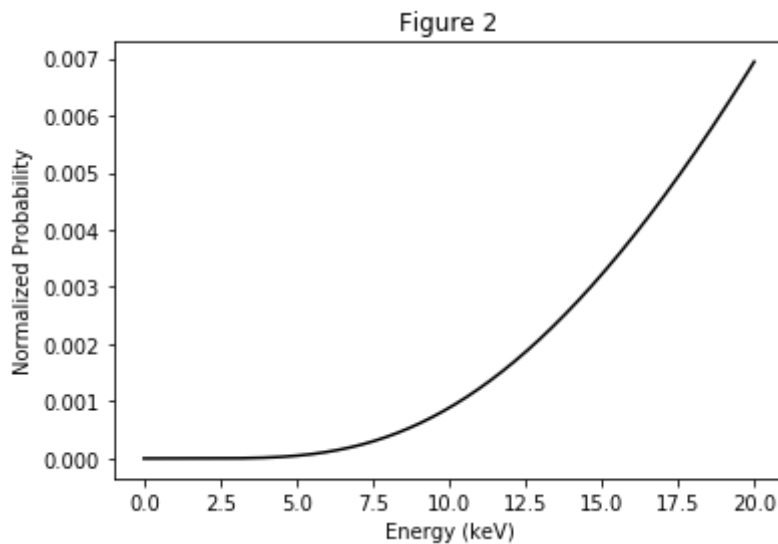
```
In [5]: z_a = 1 #atomic number of a proton
z_b = 1 #atomic number of a proton
c = 3e8 #m * s^-1
alpha = (1/137) #dimensionless fine-structure constant

#Gamow Energy
E_g = ((c*pi*alpha)**2)*(m_proton)*6.242e15

tunnel = e**(-((E_g/E)**.5))

plt.plot(E, tunnel, 'black')
plt.xlabel('Energy (keV)')
plt.ylabel('Normalized Probability')
plt.title('Figure 2')
```

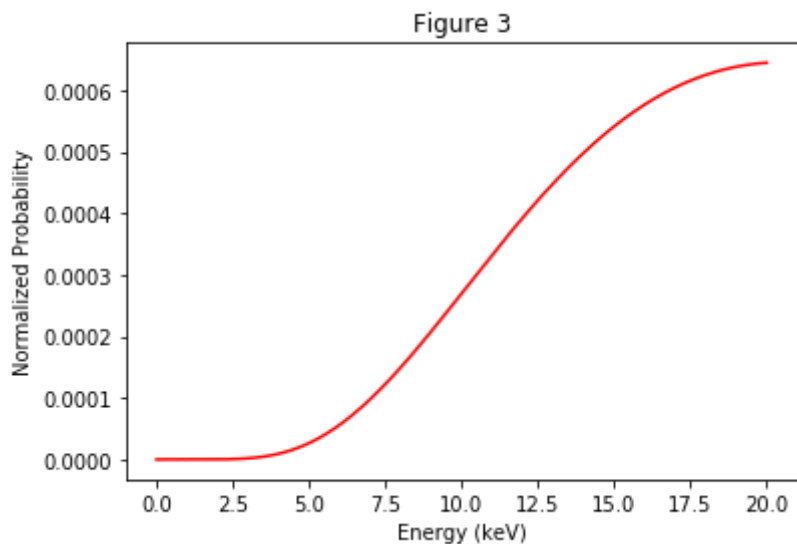
```
Out[5]: Text(0.5, 1.0, 'Figure 2')
```



From here, I will multiply the Maxwellian distribution of particles for each spectral type by the probability distribution of tunneling. When multiplied together, the function produced represents the probability density function of quantum tunneling given a distribution of particles that have various speeds and thus various energies. The peak of this pdf is called the Gamow Peak. The area under the Gamow Peak will be used to determine the reaction rate.

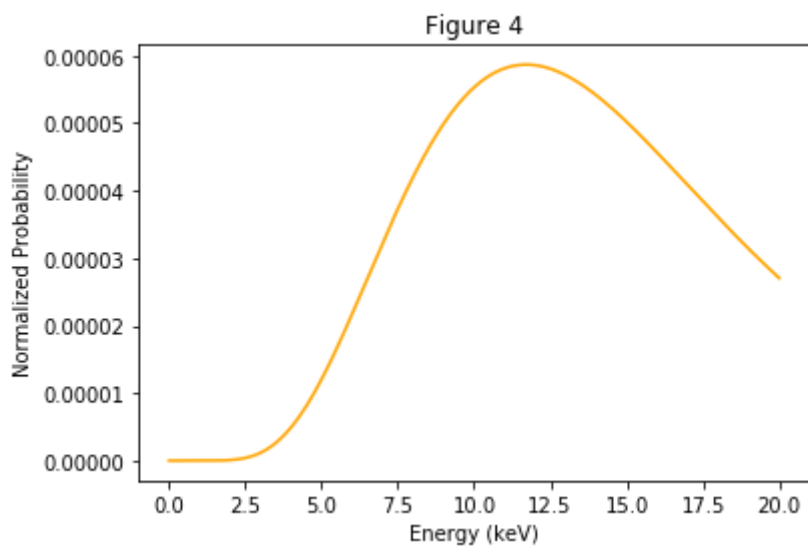
```
In [6]: combined_o5 = maxwell_o5*tunnel
plt.plot(E, combined_o5, 'red')
plt.xlabel('Energy (keV)')
plt.ylabel('Normalized Probability')
plt.title('Figure 3')
```

Out[6]: Text(0.5, 1.0, 'Figure 3')



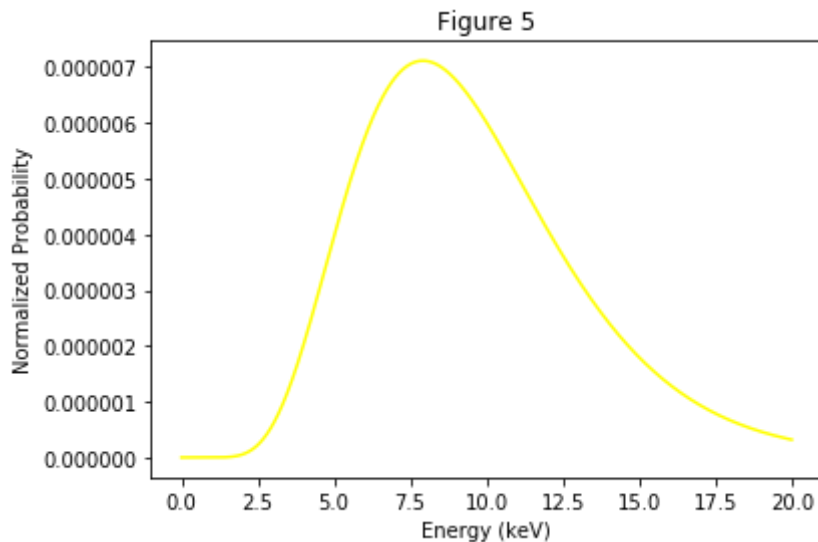
```
In [7]: combined_b5 = maxwell_b5*tunnel
plt.plot(E, combined_b5, 'orange')
plt.xlabel('Energy (keV)')
plt.ylabel('Normalized Probability')
plt.title('Figure 4')
```

Out[7]: Text(0.5, 1.0, 'Figure 4')



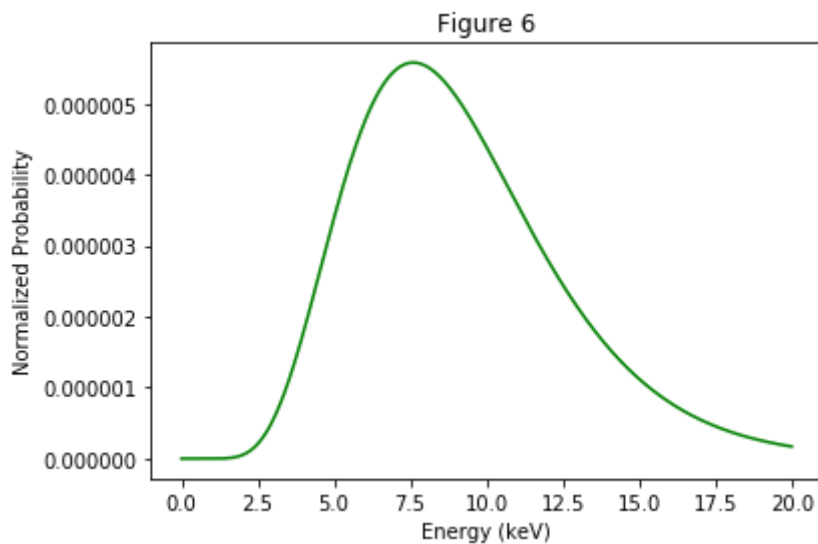
```
In [8]: combined_a5 = maxwell_a5*tunnel
plt.plot(E, combined_a5, 'yellow')
plt.xlabel('Energy (keV)')
plt.ylabel('Normalized Probability')
plt.title('Figure 5')
```

Out[8]: Text(0.5, 1.0, 'Figure 5')



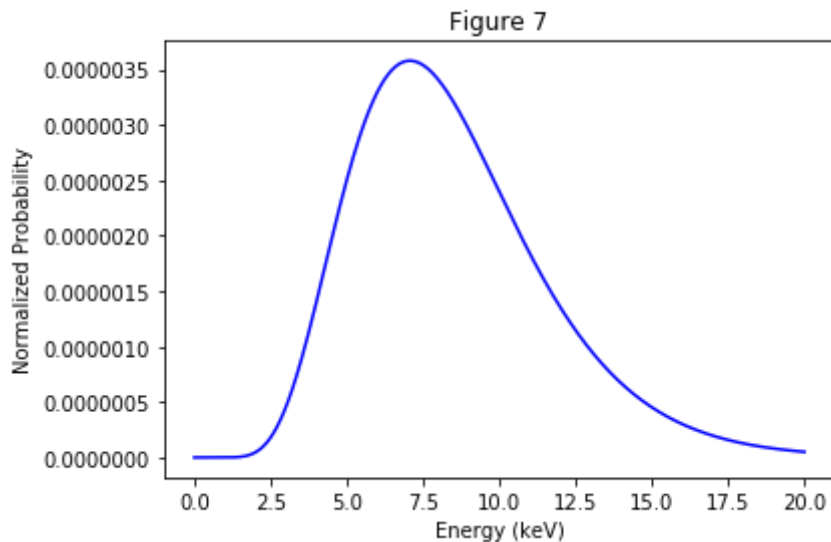
```
In [9]: combined_f5 = maxwell_f5*tunnel
plt.plot(E, combined_f5, 'green')
plt.xlabel('Energy (keV)')
plt.ylabel('Normalized Probability')
plt.title('Figure 6')
```

Out[9]: Text(0.5, 1.0, 'Figure 6')



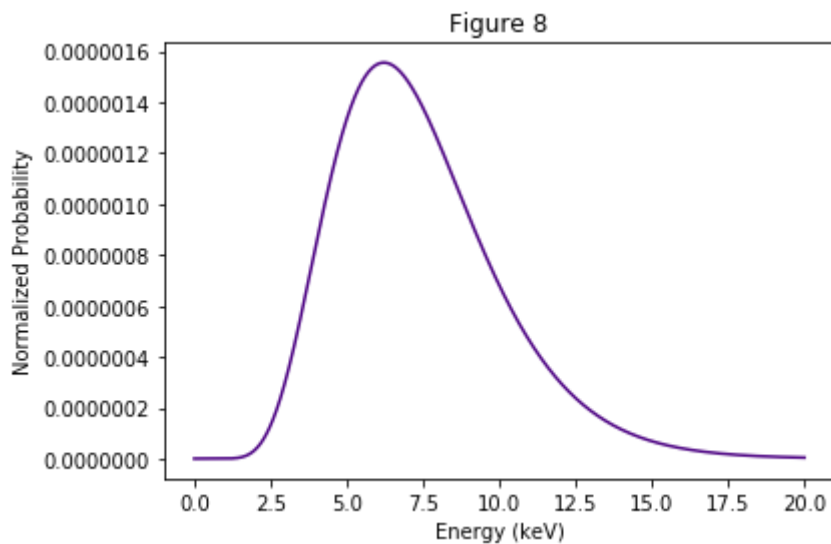
```
In [10]: combined_g5 = maxwell_g5*tunnel
plt.plot(E, combined_g5, 'blue')
plt.xlabel('Energy (keV)')
plt.ylabel('Normalized Probability')
plt.title('Figure 7')
```

Out[10]: Text(0.5, 1.0, 'Figure 7')



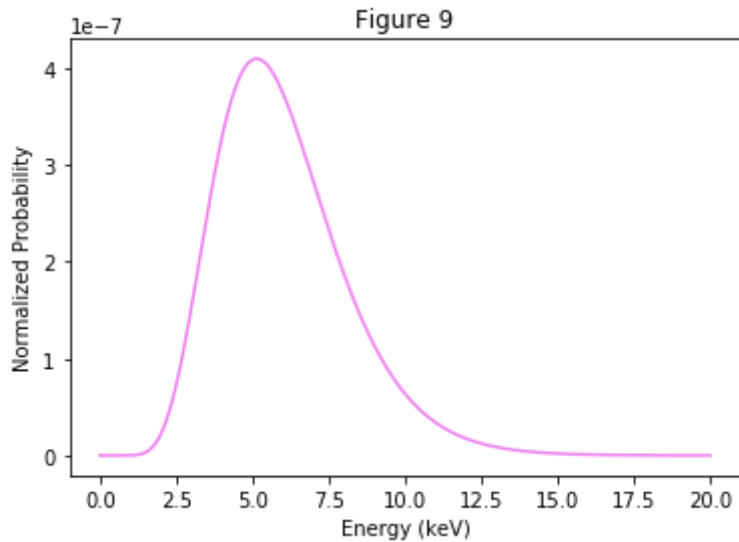
```
In [11]: combined_k5 = maxwell_k5*tunnel
plt.plot(E, combined_k5, 'indigo')
plt.xlabel('Energy (keV)')
plt.ylabel('Normalized Probability')
plt.title('Figure 8')
```

Out[11]: Text(0.5, 1.0, 'Figure 8')



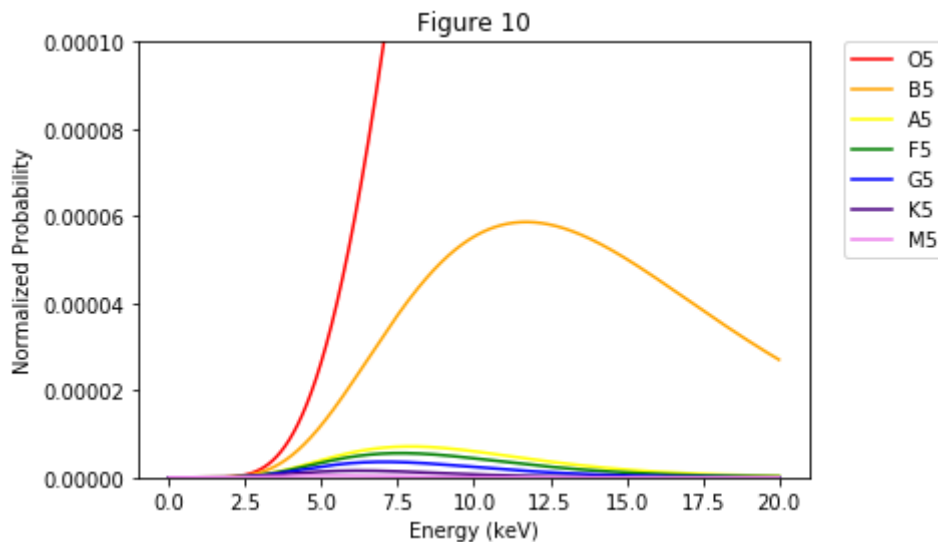
```
In [12]: combined_m5 = maxwell_m5*tunnel
plt.plot(E, combined_m5, 'violet')
plt.xlabel('Energy (keV)')
plt.ylabel('Normalized Probability')
plt.title('Figure 9')
```

Out[12]: Text(0.5, 1.0, 'Figure 9')



```
In [13]: plt.plot(E, combined_o5, 'red', label = "O5")
plt.plot(E, combined_b5, 'orange', label = "B5")
plt.plot(E, combined_a5, 'yellow', label = "A5")
plt.plot(E, combined_f5, 'green', label = "F5")
plt.plot(E, combined_g5, 'blue', label = "G5")
plt.plot(E, combined_k5, 'indigo', label = "K5")
plt.plot(E, combined_m5, 'violet', label = "M5")
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('Energy (keV)')
plt.ylabel('Normalized Probability')
plt.ylim(0, .0001)
plt.title('Figure 10')
```

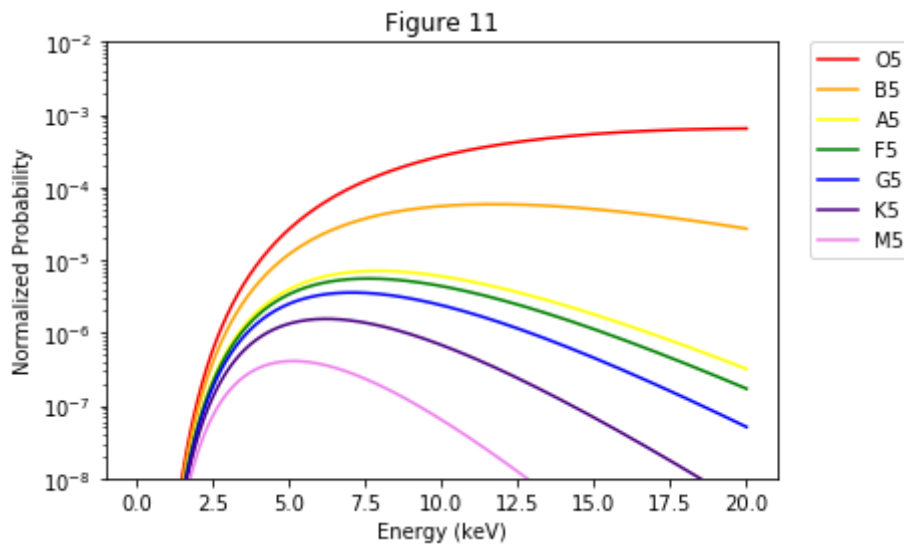
Out[13]: Text(0.5, 1.0, 'Figure 10')



Above is a plot of all the reaction probabilities for each spectral type. It can be seen that the Gamow Peak increases as temperature increases with spectral type, from the coldest stars at M5, to the hottest stars at O5. This indicates that the reaction rates will increase with increasing core temperature, which makes intuitive sense.

```
In [15]: plt.semilogy(E, combined_o5, 'red', label = "O5")
plt.semilogy(E, combined_b5, 'orange', label = "B5")
plt.semilogy(E, combined_a5, 'yellow', label = "A5")
plt.semilogy(E, combined_f5, 'green', label = "F5")
plt.semilogy(E, combined_g5, 'blue', label = "G5")
plt.semilogy(E, combined_k5, 'indigo', label = "K5")
plt.semilogy(E, combined_m5, 'violet', label = "M5")
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('Energy (keV)')
plt.ylabel('Normalized Probability')
plt.ylim(1e-8,1e-2)
plt.title('Figure 11')
```

Out[15]: Text(0.5, 1.0, 'Figure 11')



Since the luminosity axis of the H-R diagram is logarithmic, our dispersion in temperature is logarithmic, so I thought it would be a bit more useful to display the same plot from Figure 11, with a logarithmic y-axis.