▶ Q1

(a) Helmholtz free energy $F(\tau) = -\tau \ln Z$

$$Z = e^{-0} + e^{-\epsilon/\tau} = 1 + e^{-\epsilon/\tau}$$
$$F(\tau) = -\tau \ln\left(1 + e^{-\epsilon/\tau}\right)$$

That was easy.

(b) Helmholtz free energy $F(\tau) = U - \tau\sigma$

In the canonical ensemble, the probability of being in a state is

$$\sigma = -\frac{\partial F}{\partial \tau} = \ln\left(e^{-\epsilon/\tau} + 1\right) + \frac{\epsilon}{\tau} \frac{e^{-\epsilon/\tau}}{e^{-\epsilon/\tau} + 1}$$

This makes finding the energy easy

$$U(\tau) = F(\tau) + \tau\sigma$$
$$= -\tau \ln\left(1 + e^{-\epsilon/\tau}\right) + \tau \ln\left(e^{-\epsilon/\tau} + 1\right) + \epsilon \frac{e^{-\epsilon/\tau}}{e^{-\epsilon/\tau} + 1}$$
$$= \epsilon \frac{e^{-\epsilon/\tau}}{1 + e^{-\epsilon/\tau}}$$

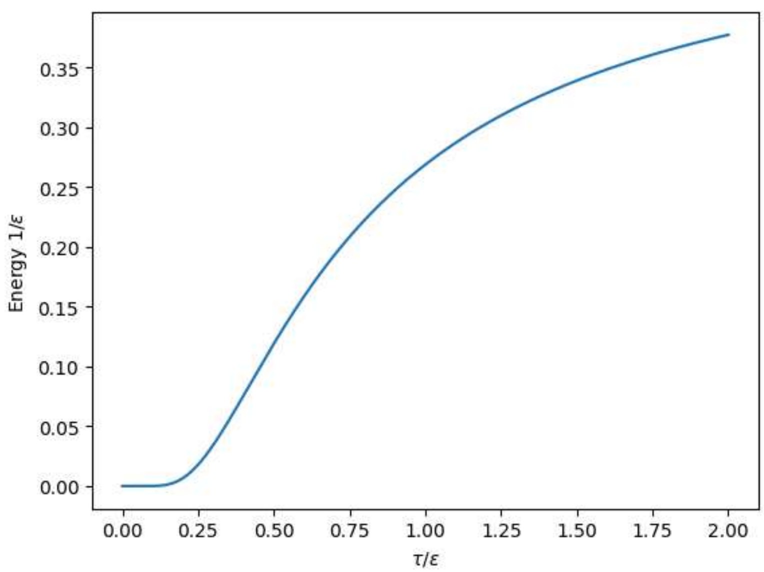
In [133…

```
import matplotlib.pyplot as plt
from numpy import exp, linspace

temp = linspace(0, 2, 1000)
energy = exp(-1/temp)/(1+exp(-1/temp))

plt.plot(temp, energy)
plt.xlabel(r'$\tau/\epsilon$')
plt.ylabel(r'Energy $1/\epsilon$')
```

```
C:\Users\andre\AppData\Local\Temp\ipykernel_27408\2397519148.py:5: RuntimeWarning: divide by zero encountered in divide
  energy = exp(-1/temp)/(1+exp(-1/temp))
```

Out[133…

```
Text(0, 0.5, 'Energy $1/\\epsilon$')
```



▶ Q2

(a)

$$Z = \sum_{s=0}^{\infty} e^{-\epsilon_s/\tau} = \sum_{s=0}^{\infty} e^{-s\hbar\omega/\tau}$$

$$\sum_{a=0}^{\infty} e^{a\,b} = -\frac{1}{e^b - 1} \approx -\frac{1}{2.71828^b - 1} \quad \text{when} \quad e^{\mathrm{Re}(b)} < 1$$

$$Z = \frac{1}{1 - e^{-\hbar\omega/\tau}}$$

(b) Easy! $F = -\tau \ln Z$

$$F = -\tau \ln\left(\frac{1}{1 - e^{-\hbar\omega/\tau}}\right) = \tau \ln\left(1 - e^{\hbar\omega/\tau}\right)$$

(c) I'm actually going to skip this one and come back. I'd rather do the derivative of σ than do this one.

Using $U = F + \tau\sigma$ we have

$$\begin{aligned} U &= \tau \ln\left(1 - e^{\hbar\omega/\tau}\right) - \tau \ln\left(1 - e^{\hbar\omega/\tau}\right) + \frac{\hbar\omega}{e^{\hbar\omega/\tau} - 1} \\ &= \frac{\hbar\omega}{e^{\hbar\omega/\tau} - 1} \end{aligned}$$

See derivation of σ below

(d) Here we'll take the derivative of F

$\sigma = \frac{\partial F}{\partial \tau}$

In [71]:
```python
from sympy import symbols, ln, exp, diff, latex, Derivative

hbar, omega, tau = symbols('hbar omega tau')

print('Helmholtz energy:')
F = tau*ln(1-exp(-hbar*omega/tau))
display(F)

sigma = - diff(F, tau)
print('Entropy:')
sigma
```

Helmholtz energy:

$$\tau \log\left(1 - e^{-\frac{\hbar\omega}{\tau}}\right)$$

Entropy:

Out[71]:

$$\frac{\hbar\omega e^{-\frac{\hbar\omega}{\tau}}}{\tau\left(1 - e^{-\frac{\hbar\omega}{\tau}}\right)} - \log\left(1 - e^{-\frac{\hbar\omega}{\tau}}\right)$$

This works out to be exactly what we're looking for:

$$\frac{\hbar\omega e^{-\frac{\hbar\omega}{\tau}}}{\tau\left(1 - e^{-\frac{\hbar\omega}{\tau}}\right)} \cdot \frac{e^{\hbar\omega/\tau}}{e^{\hbar\omega/\tau}} = \frac{\hbar\omega/\tau}{e^{\hbar\omega/\tau} - 1}$$

$$\begin{aligned} \sigma &= \frac{\hbar\omega e^{-\frac{\hbar\omega}{\tau}}}{\tau\left(1 - e^{-\frac{\hbar\omega}{\tau}}\right)} - \ln\left(1 - e^{-\frac{\hbar\omega}{\tau}}\right) \\ &= \frac{\hbar\omega/\tau}{e^{\hbar\omega/\tau} - 1} - \ln\left(1 - e^{-\frac{\hbar\omega}{\tau}}\right) \end{aligned}$$

(e) HW2P3: $U = \frac{N\hbar\omega}{e^{\hbar\omega/\tau} - 1}$

Yes, it is N times my result. This makes sense since it is N oscillators, where this question asks about a single oscillator.

(f)

HW2P3:

I'll use the result from the answer key

$$\sigma(U,N) = N\ln\left(1+\frac{U}{N\hbar\omega}\right)+\frac{U}{\hbar\omega}\ln\left(1+\frac{N\hbar\omega}{U}\right)$$

Let's insert in $U(\tau)$

$$\sigma(\tau,N) = N\ln\left(1+\frac{\frac{N\hbar\omega}{e^{\hbar\omega/\tau}-1}}{N\hbar\omega}\right)+\frac{\frac{N\hbar\omega}{e^{\hbar\omega/\tau}-1}}{\hbar\omega}\ln\left(1+\frac{N\hbar\omega}{\frac{N\hbar\omega}{e^{\hbar\omega/\tau}-1}}\right)$$

$$= N\ln\left(1+\frac{1}{e^{\hbar\omega/\tau}-1}\right)+\frac{N}{e^{\hbar\omega/\tau}-1}\ln\left(1+e^{\hbar\omega/\tau}-1\right)$$

$$\begin{aligned}
&= N\ln\left(1+\frac{1}{e^{\hbar\omega/\tau}-1}\right)+\frac{N\hbar\omega/\tau}{e^{\hbar\omega/\tau}-1} \\
&= N\ln\left(\frac{e^{\hbar\omega/\tau}}{e^{\hbar\omega/\tau}-1}\right)+\frac{N\hbar\omega/\tau}{e^{\hbar\omega/\tau}-1} \\
&= -N\ln\left(\frac{e^{\hbar\omega/\tau}-1}{e^{\hbar\omega/\tau}}\right)+\frac{N\hbar\omega/\tau}{e^{\hbar\omega/\tau}-1} \\
&= N\left(\frac{\hbar\omega/\tau}{e^{\hbar\omega/\tau}-1}-\ln\left(1-e^{-\frac{\hbar\omega}{\tau}}\right)\right)
\end{aligned}$$

Yippie! Entropy is also just multiplied by N since each oscillator isn't interacting and entropy adds like energy

▶ Q3

(a) Adding energy -> U goes up.

$$F = U - \tau\sigma$$
$$\Delta F = \Delta U - \tau\Delta\sigma$$

The system does no work, so $\Delta F = 0$

$$\Delta U = \tau\Delta\sigma$$
$$U = k_B T(\ln g)$$

Going from g_0 to $2g_0$ states:

$$U_f - U_i = k_B T(\ln 2g_0) - k_B T(\ln(g_0))$$
$$\Delta U = k_B T \ln 2 = 2.87 \times 10^{-21}$$

In [123…
```
from scipy.constants import k
U = k*300*ln(2)
U.evalf()
```

Out[123… $2.87097888507872 \cdot 10^{-21}$

(b) $\Delta U/\tau = \Delta\sigma$

$$S = k_b\sigma = k_b\frac{U}{k_bT} = \frac{U}{T}$$
$$S = 9.57 \times 10^{-24}$$

In [127…
```
S = U/300
S.evalf()
```

Out[127… $9.56992961692908 \cdot 10^{-24}$

▶ Q4

(a) We can find probabilities using Z

$$Z = 1 + e^{-a/\tau} + e^{-3a/\tau}$$
$$P(1) = \frac{e^{-a/\tau}}{1 + e^{-a/\tau} + e^{-3a/\tau}}$$

(b) Using a summation:

$$U = 0 + aP(1) + 3aP(2) = \frac{ae^{-a/\tau} + 3ae^{-3a/\tau}}{1 + e^{-a/\tau} + e^{-3a/\tau}}$$

(c)

	P(0)	P(1)	P(2)
τ = 0	1	0	0
τ = a	0.7	0.26	0.04
τ = 100a	.34	.33	.33

high temp: all energies are equally likely

In [60]:
```
a = symbols('a')

e0 = 1
e1 = exp(-a/tau)
e2 = exp(-3*a/tau)

Z = e0 + e1 + e2
(e0/Z).replace(tau,a*100).evalf()
```

Out[60]: 0.337781308846565

▶ Q5

Return to microcanonical

(a) $\sigma(U) = \ln(g(U))$ easy

$\sigma(U) = \ln\left(e^{\sqrt{NU/U_0}}\right) = \sqrt{NU/U_0}$

(b) $1/\tau = d\sigma/dU$

$$\frac{1}{\tau} = \frac{\partial}{\partial U}\sqrt{\frac{N}{U_0}}\sqrt{U}$$
$$= \sqrt{\frac{N}{U_0}}\frac{\partial}{\partial U}\sqrt{U} = \frac{1}{2}\sqrt{\frac{N}{UU_0}}$$
$$U(\tau) = \frac{N\tau^2}{4U_0}$$

(c)

Note that conventional T and S have different formulas

$$\tau = k_B T$$
$$\sigma = \frac{S}{k_B}$$

τ has units of energy, and N is unitless

$$\tau = 4.14 \times 10^{-21} J$$
$$U(\tau) = 258.19 J$$
$$S(U) = 1.7 J/K$$
$$F = U - \tau\sigma = -258.19 J$$

Interesting how $F = -U$

In [116…

```
from scipy.constants import k

tau = k*300
print(f'tau={tau} J')
N = 6.02e23
U0 = 1e-20

U = N*tau**2/(4*U0)
print(f'U={U:.2f} J')

sigma = (N*U/U0)**(1/2)
S = k*sigma
print(f'Entropy ={S} J/K')

F = U - tau * sigma
print(f'Helmholtz free energy={F:.3f} J')
```

```
tau=4.141947e-21 J
U=258.19 J
Entropy =1.7212910700645034 J/K
Helmholtz free energy=-258.194 J
```

If $\sigma\tau$ has units joules, then σ must be unitless (boltzmann constant J/K)