

# hw4\_ipynb

September 23, 2024

```
[ ]: import numpy as np
from astropy import units as u

# Constants with units
mp = 1.6726e-27 * u.kg # mass of proton
Aul = 2.8843e-15 / u.s # Einstein A coefficient
h = 6.626e-34 * u.J * u.s # Planck's constant
c = 3.00e8 * u.m / u.s # speed of light
k = 1.38e-23 * u.J / u.K # Boltzmann's constant
T = 100 * u.K # Temperature
v = 1000 * u.m / u.s # velocity
lambda_ul = 21.106e-2 * u.m # wavelength (21.106 cm)

# Frequency calculation from wavelength
nu_ul = c / lambda_ul # Frequency in Hz

# Define phi(nu_ul)
def phi(nu_ul, v):
    return lambda_ul / (v * np.sqrt(2 * np.pi))

# Calculate kappa
kappa = (3 / (32 * np.pi)) * (1 / mp) * (h * c**2 / nu_ul) * (Aul / (k * T)) * phi(nu_ul, v)

# Print the result with units
print(f"Opacity (kappa) = {kappa.to(u.cm**2 / u.g):.4e}")
```

Opacity (kappa) = 1.3173e+03 cm<sup>2</sup> / g

```
[ ]: import numpy as np
from astropy import units as u

# Constants with units
mp = 1.6726e-27 * u.kg # mass of proton
N_H = 1e21 / u.cm**2 # hydrogen column density

# Opacity (kappa) calculation (using previous code)
# Assuming kappa is already calculated or using a placeholder for this example
```

```

kappa_placeholder = kappa.to(u.cm**2 / u.g) # Example value for kappa, replace
↳with actual kappa calculation

# Calculate optical depth (tau)
tau = mp * kappa_placeholder * N_H

# Convert tau to a more convenient unit, e.g., dimensionless or with specific
↳context
tau_value = tau.cgs.value

# Print the result
print(f"Optical Depth (tau) = {tau_value:.4e}")

```

Optical Depth (tau) = 2.2033e+00

[ ]: