Homework 1

Question 1: Black-Body Radiation

Build the distribution of curves at different temperatures for the black-body radiation formula:

$$B_\nu(\nu, T) = \frac{2h\nu^3}{c^2} \cdot \frac{1}{e^{\frac{h\nu}{k_B T}} - 1}$$

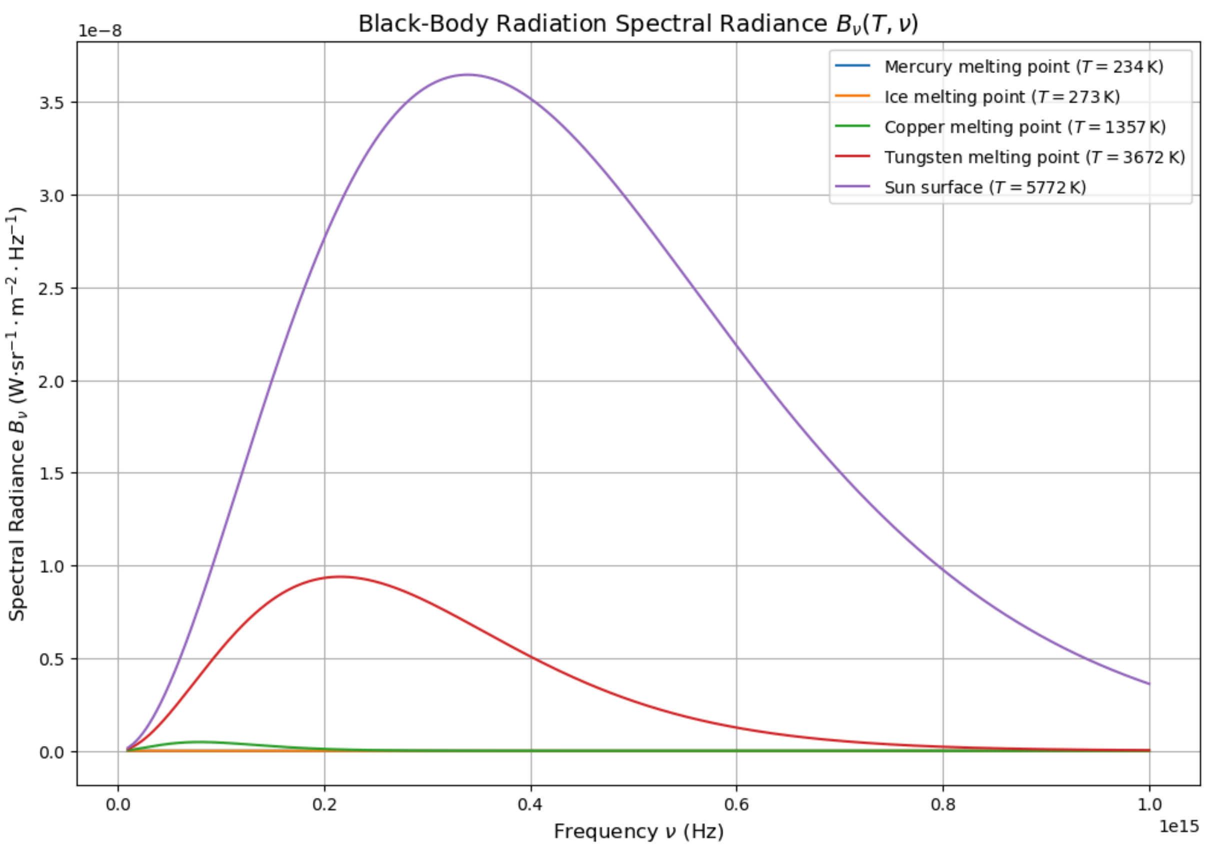
Plot the function $B_\nu(\nu, T)$ for the following temperatures:

- $T = 0\,\text{K (Absolute zero)}$
- $T = 234\,\text{K (Mercury melting point)}$
- $T = 273\,\text{K (Ice melting point)}$
- $T = 1357\,\text{K (Copper)}$
- $T = 3672\,\text{K (Tungsten)}$
- $T = 5772\,\text{K (Sun surface)}$

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt

# Constants
h = 6.62607015e-34  # Planck's constant (J·s)
c = 299792458       # Speed of light (m/s)
k_B = 1.380649e-23  # Boltzmann constant (J/K)

# Black-body radiation formula
def black_body_radiation(v, T):
    """
    Calculate spectral radiance for black-body radiation at a given temperature.

    Parameters:
    v (float or np.array): Frequency in Hz.
    T (float): Temperature in Kelvin.

    Returns:
    float or np.array: Spectral radiance in W·sr^-1·m^-2·Hz^-1.
    """
    return (2 * h * v**3 / c**2) / (np.exp(h * v / (k_B * T)) - 1)

# Frequency range (Hz)
v_vals = np.linspace(1e13, 1e15, 1000)

# Temperatures (K)
temperatures = [
    (0, "Absolute zero"),
```

```
    (234, "Mercury melting point"),
    (273, "Ice melting point"),
    (1357, "Copper melting point"),
    (3672, "Tungsten melting point"),
    (5772, "Sun surface")
]

# Updated Plot with LaTeX
plt.figure(figsize=(12, 8))
for T, label in temperatures:
    if T > 0:  # Avoid calculation for absolute zero (non-physical, formula diverge
        B_vals = black_body_radiation(v_vals, T)
        plt.plot(v_vals, B_vals, label=f"{label} ($T = {T} \, \mathrm{{K}}$)")

# Adding LaTeX labels and title
plt.title(r"Black-Body Radiation Spectral Radiance $B_\nu(T, \nu)$", fontsize=14)
plt.xlabel(r"Frequency $\nu$ (Hz)", fontsize=12)
plt.ylabel(r"Spectral Radiance $B_\nu$ (W$\cdot$sr$^{-1}\cdot$m$^{-2}\cdot$Hz$^{-1}

# Adding legend and grid
plt.legend(fontsize=10)
plt.grid()
plt.show()
```



Question 2: Unit Conversion

Convert the energy difference ΔE from Erg to eV:

$$\Delta E = \frac{\hbar}{\Delta t} = \frac{10^{-27}\,\mathrm{Erg}\cdot\mathrm{s}}{10^{-8}\,\mathrm{s}} = 10^{-19}\,\mathrm{Erg} = ?\,\mathrm{eV}$$

In [2]:
```
# Constants for conversion
# From 2019 revision of the SI (https://en.wikipedia.org/wiki/SI_base_unit)
erg_to_joule = 1e-7  # 1 Erg = 10^-7 Joules
eV_to_joule = 1.602176634e-19  # 1 eV = 1.602176634 × 10^-19 Joules

# Function to convert Erg to eV
def Erg2eV(energy_erg):
    """
    Converts energy from Ergs to eV (electronvolts).

    Parameters:
    energy_erg (float): Energy in Ergs.

    Returns:
    float: Energy in eV.
    """
    conversion_factor = erg_to_joule / eV_to_joule
    return energy_erg * conversion_factor

# Function to convert eV to Erg
def eV2Erg(energy_eV):
    """
    Converts energy from eV (electronvolts) to Ergs.

    Parameters:
    energy_eV (float): Energy in eV.

    Returns:
    float: Energy in Ergs.
    """
    conversion_factor = eV_to_joule / erg_to_joule
    return energy_eV * conversion_factor

# Run the Erg to eV conversion for the given value in the question
energy_erg = 1e-19  # Energy in Ergs
energy_ev = Erg2eV(energy_erg)
```

In [3]:
```
from IPython.display import display, Math

# Prepare the result in LaTeX format for better display in Jupyter Notebook
energy_erg_str = "10^{-19} \\ \\mathrm{Erg}"
energy_ev_str = f"6.241509 \\times 10^{{-8}} \\ \\mathrm{{eV}}"

# Display the result
display(Math(f"\\text{{The energy difference }} \\Delta E = {energy_erg_str} \\text
```

The energy difference $\Delta E = 10^{-19}\ \mathrm{Erg}$ converts to: $\Delta E = 6.241509 \times 10^{-8}\ \mathrm{eV}$

Question 1B: Black-Body Radiation (B)

In physics, Planck's law (also Planck radiation law) describes the spectral density of electromagnetic radiation emitted by a black body in thermal equilibrium at a given temperature T, when there is no net flow of matter or energy between the body and its environment.

Planck Radiation Formula for $B_\lambda(T)$

$$B_\lambda(T) = \frac{2hc^2}{\lambda^5} \cdot \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1}$$

Where:

- $B_\lambda(T)$: Spectral radiance (energy emitted per unit area, per unit wavelength, per unit solid angle).
- λ: Wavelength (m)
- h: Planck's constant (6.626×10^-34 J·s)
- c: Speed of light (3.0×10^8 m/s)
- k_B: Boltzmann constant (1.380649×10^-23 J·K)
- T: Temperature (K)

Rayleigh-Jeans Approximation

At long wavelengths ($\lambda \gg \frac{hc}{k_B T}$), the exponential term in $B_\lambda(T)$ can be approximated as:

$$e^{\frac{hc}{\lambda k_B T}} \approx 1 + \frac{hc}{\lambda k_B T}$$

Thus:

$$B_\lambda(T) \approx \frac{2ck_B T}{\lambda^4}$$

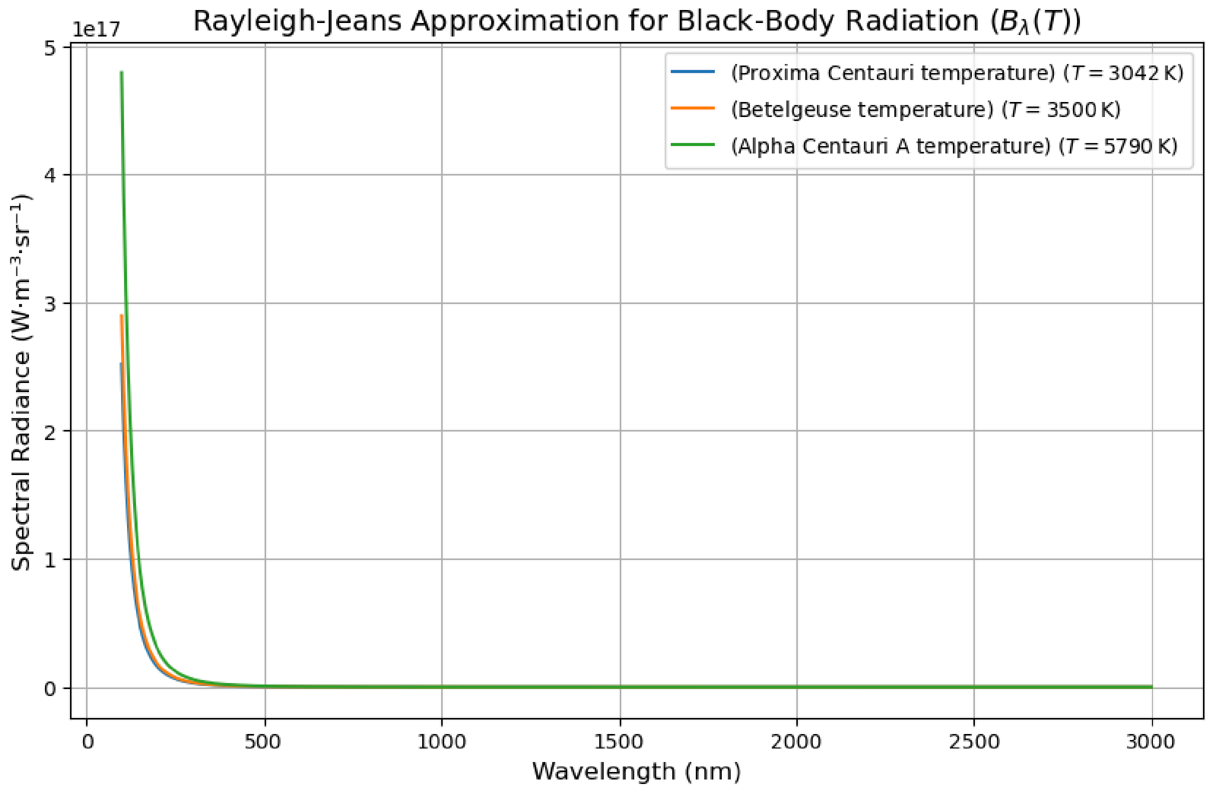
Build the distribution of curves at different temperatures for the Rayleigh-Jeans black-body radiation formula:

$$B_\lambda(T) \approx \frac{2ck_B T}{\lambda^4}$$

Plot the function $B_\lambda(T)$ for the following temperatures:

- $T = 3042\,\mathrm{K}\ (\mathrm{Proxima\ Centauri})$
- $T = 3500\,\mathrm{K}\ (\mathrm{Betelgeuse})$
- $T = 5790\,\mathrm{K}\ (\mathrm{Alpha\ Centauri\ A})$

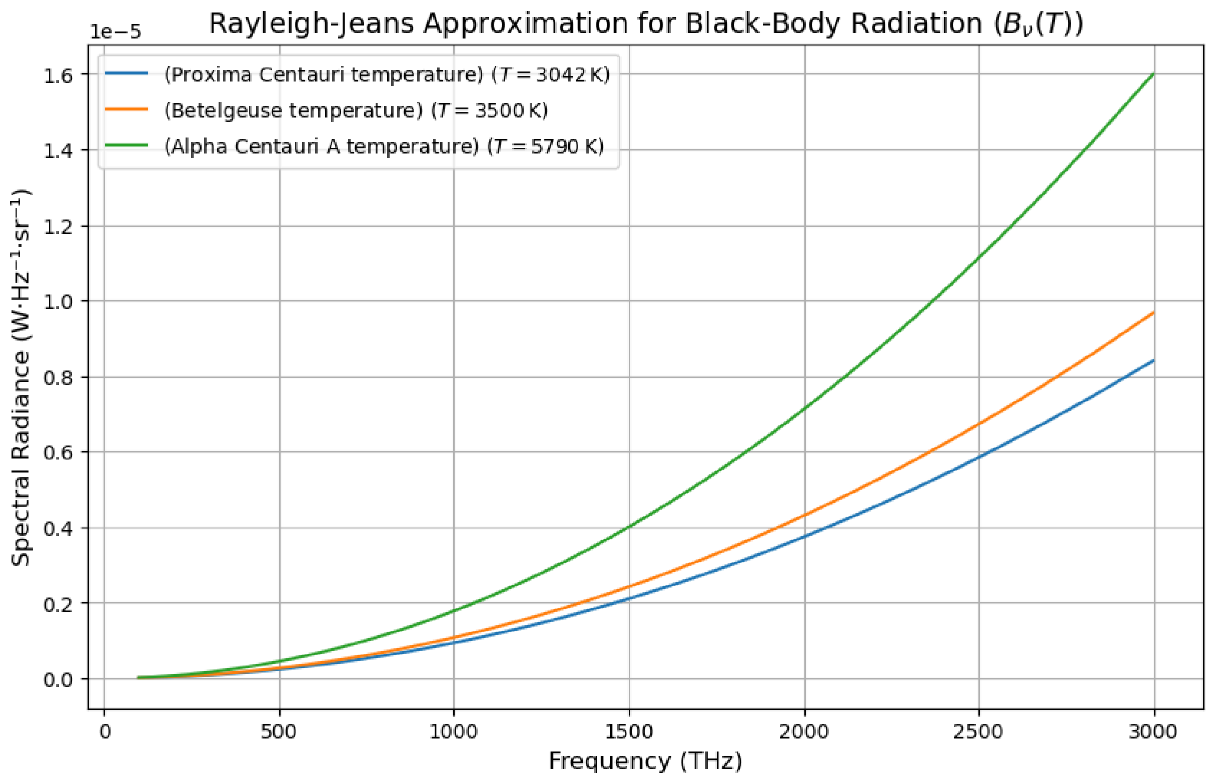
In [4]:
```
# Rayleigh-Jeans approximation for black-body radiation in terms of wavelength
def B_lambda_rayleigh(wavelength, T):
    """
    Calculate spectral radiance B_lambda (Rayleigh-Jeans Law) at a given temperatur

    Parameters:
    wavelength (float or np.array): Wavelength in meters.
    T (float): Temperature in Kelvin.

    Returns:
    float or np.array: Spectral radiance in W·sr⁻¹·m⁻³.
    """
    return (2 * c * k_B * T) / (wavelength**4)

# Wavelength range (meters)
wavelengths = np.linspace(1e-7, 3e-6, 500)  # 100 nm to 3000 nm

# Temperatures (K)
temperatures = [
    (3042, "(Proxima Centauri temperature)"),
    (3500, "(Betelgeuse temperature)"),
    (5790, "(Alpha Centauri A temperature)")
]

# Plot
plt.figure(figsize=(10, 6))
for T, label in temperatures:
    radiance = B_lambda_rayleigh(wavelengths, T)
    plt.plot(wavelengths * 1e9, radiance, label=f"{label} ($T = {T} \, \\mathrm{{K}

# Formatting
plt.title("Rayleigh-Jeans Approximation for Black-Body Radiation ($B_\\lambda(T)$)"
plt.xlabel("Wavelength (nm)", fontsize=12)
plt.ylabel("Spectral Radiance (W·m⁻³·sr⁻¹)", fontsize=12)
plt.legend(fontsize=10)
plt.grid(True)
plt.show()
```



In [5]:

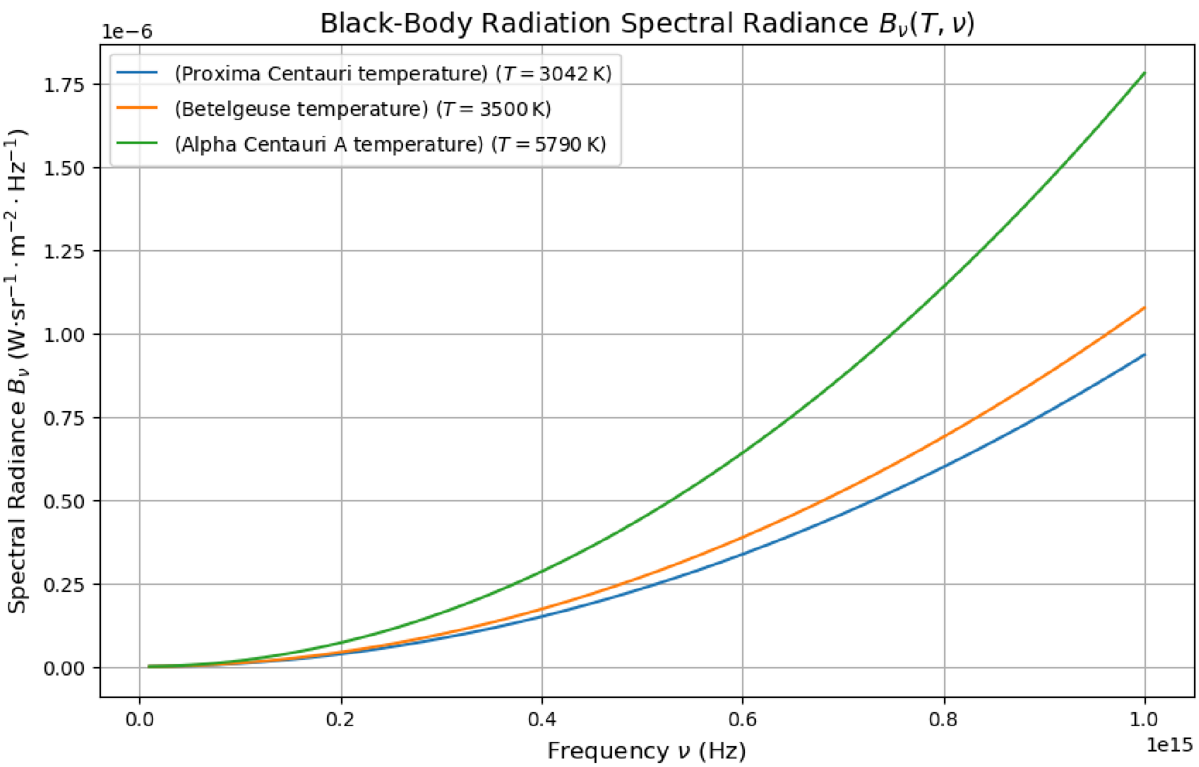
```
# Conversion of B_lambda to B_nu
def B_nu_from_B_lambda(wavelength, B_lambda_values):
    """
    Convert spectral radiance from wavelength to frequency scale.

    Parameters:
    wavelength (float or np.array): Wavelength in meters.
    B_lambda_values (float or np.array): Spectral radiance in W·m⁻³·sr⁻¹.

    Returns:
    tuple: Frequency (Hz) and spectral radiance (W·Hz⁻¹·sr⁻¹).
    """
    frequency = c / wavelength
    B_nu = B_lambda_values * (wavelength**2 / c)
    return frequency, B_nu

# Wavelength range (meters)
wavelengths = np.linspace(1e-7, 3e-6, 500)  # 100 nm to 3000 nm

# Temperatures (K)
temperatures = [
    (3042, "(Proxima Centauri temperature)"),
    (3500, "(Betelgeuse temperature)"),
    (5790, "(Alpha Centauri A temperature)")
]

# Plot in frequency scale
plt.figure(figsize=(10, 6))
for T, label in temperatures:
    # Compute B_lambda
    B_lambda_values = B_lambda_rayleigh(wavelengths, T)
```

```
    # Convert to B_nu
    frequencies, B_nu_values = B_nu_from_B_lambda(wavelengths, B_lambda_values)
    # Plot
    plt.plot(frequencies / 1e12, B_nu_values, label=f"{label} ($T = {T} \, \\mathrm

# Formatting
plt.title("Rayleigh-Jeans Approximation for Black-Body Radiation ($B_\\nu(T)$)", fo
plt.xlabel("Frequency (THz)", fontsize=12)
plt.ylabel("Spectral Radiance (W·Hz⁻¹·sr⁻¹)", fontsize=12)
plt.legend(fontsize=10)
plt.grid(True)
plt.show()
```



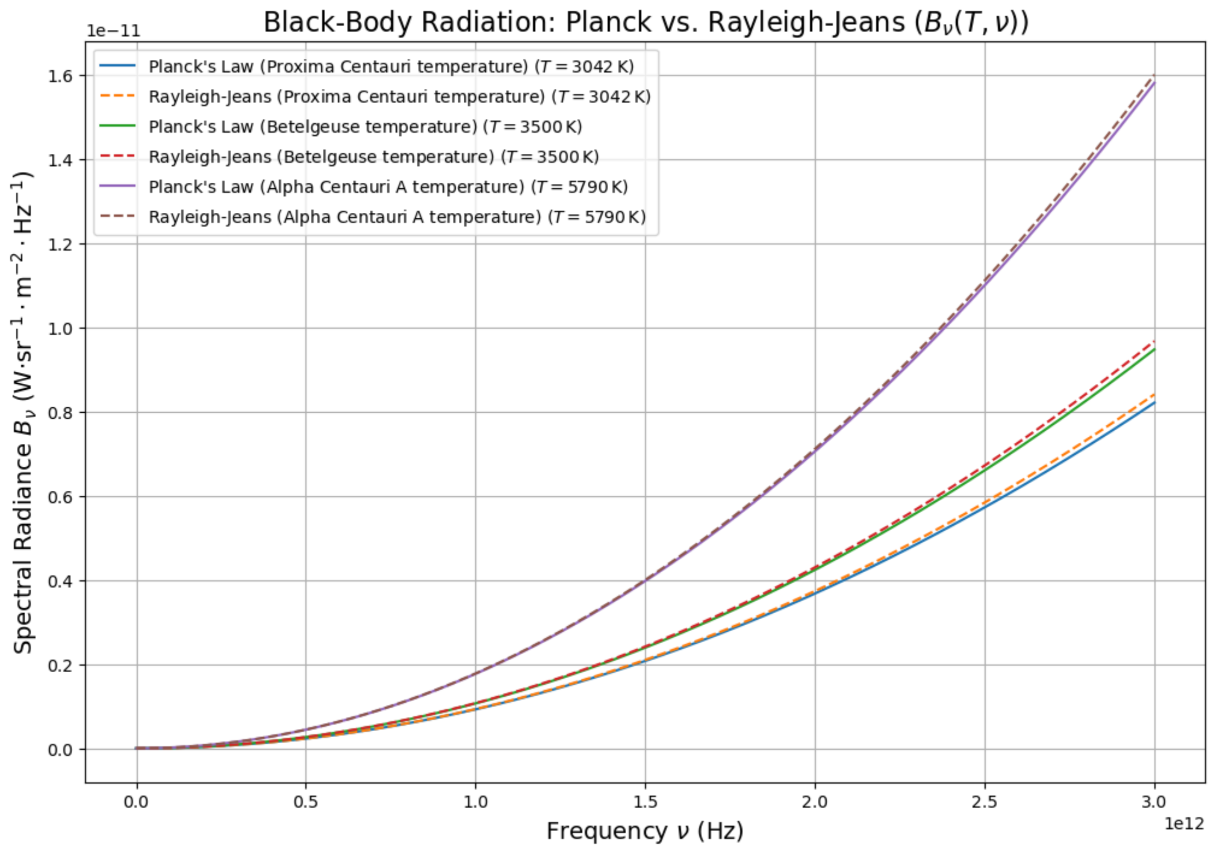
In [6]:
```
# Constants
h = 6.62607015e-34   # Planck's constant (J·s)
c = 299792458        # Speed of light (m/s)
k_B = 1.380649e-23   # Boltzmann constant (J/K)

# Rayleigh-Jeans approximation in terms of frequency
def B_nu_rayleigh(frequency, T):
    """
    Calculate spectral radiance B_nu (Rayleigh-Jeans Law) at a given temperature.

    Parameters:
    frequency (float or np.array): Frequency in Hz.
    T (float): Temperature in Kelvin.

    Returns:
    float or np.array: Spectral radiance in W·sr⁻¹·m⁻²·Hz⁻¹.
    """
    return 2 * k_B * T * (frequency**2) / c**2
```

```
# Frequency range (Hz)
v_vals = np.linspace(1e13, 1e15, 1000)  # 10 THz to 1000 THz

# Temperatures (K)
temperatures = [
    (3042, "(Proxima Centauri temperature)"),
    (3500, "(Betelgeuse temperature)"),
    (5790, "(Alpha Centauri A temperature)")
]

# Plot
plt.figure(figsize=(10, 6))
for T, label in temperatures:
    radiance = B_nu_rayleigh(v_vals, T)
    plt.plot(v_vals, radiance, label=f"{label} ($T = {T} \, \\mathrm{{K}}$)")

# Formatting
plt.title(r"Black-Body Radiation Spectral Radiance $B_\nu(T, \nu)$", fontsize=14)
plt.xlabel(r"Frequency $\nu$ (Hz)", fontsize=12)
plt.ylabel(r"Spectral Radiance $B_\nu$ (W$\cdot$sr$^{-1}\cdot$m$^{-2}\cdot$Hz$^{-1}
plt.legend(fontsize=10)
plt.grid(True)
plt.show()
```



In [7]:
```
import numpy as np
import matplotlib.pyplot as plt

# Constants
h = 6.62607015e-34  # Planck's constant (J·s)
c = 299792458       # Speed of light (m/s)
k_B = 1.380649e-23  # Boltzmann constant (J/K)
```

```
# Rayleigh-Jeans approximation in terms of frequency
def B_nu_rayleigh(frequency, T):
    """
    Calculate spectral radiance B_nu (Rayleigh-Jeans Law) at a given temperature.

    Parameters:
    frequency (float or np.array): Frequency in Hz.
    T (float): Temperature in Kelvin.

    Returns:
    float or np.array: Spectral radiance in W·sr⁻¹·m⁻²·Hz⁻¹.
    """
    return 2 * k_B * T * (frequency**2) / c**2

# Planck's black-body radiation law in terms of frequency
def B_nu_planck(frequency, T):
    """
    Calculate spectral radiance B_nu (Planck's Law) at a given temperature.

    Parameters:
    frequency (float or np.array): Frequency in Hz.
    T (float): Temperature in Kelvin.

    Returns:
    float or np.array: Spectral radiance in W·sr⁻¹·m⁻²·Hz⁻¹.
    """
    return (2 * h * frequency**3 / c**2) / (np.exp(h * frequency / (k_B * T)) - 1)

# Frequency range (Hz)
v_vals = np.linspace(1e8, 3e12, 1000)

# Temperatures (K)
temperatures = [
    (3042, "(Proxima Centauri temperature)"),
    (3500, "(Betelgeuse temperature)"),
    (5790, "(Alpha Centauri A temperature)")
]

# Plot
plt.figure(figsize=(12, 8))
for T, label in temperatures:
    radiance_planck = B_nu_planck(v_vals, T)
    radiance_rayleigh = B_nu_rayleigh(v_vals, T)
    plt.plot(v_vals, radiance_planck, label=f"Planck's Law {label} ($T = {T} \, \\m
    plt.plot(v_vals, radiance_rayleigh, label=f"Rayleigh-Jeans {label} ($T = {T} \,

# Formatting
plt.title(r"Black-Body Radiation: Planck vs. Rayleigh-Jeans ($B_\nu(T, \nu)$)", fon
plt.xlabel(r"Frequency $\nu$ (Hz)", fontsize=14)
plt.ylabel(r"Spectral Radiance $B_\nu$ (W$\cdot$sr$^{-1}\cdot$m$^{-2}\cdot$Hz$^{-1}
plt.legend(fontsize=10)
plt.grid(True)
plt.show()
```



The Rayleigh-Jeans Approximation is only valid at long wavelengths ($\lambda \gg \frac{hc}{k_B T}$)

Assuming 2000K< T < 50000K (star temperature), then: $\lambda \gg \frac{hc}{2000 k_B}$

In [8]:
```
min_lambda = (h*c)/(2000*k_B)
super_lambda = 1e3*min_lambda # Strictly greater by multplying by 1e3=1000
print(f"Super lambda: {super_lambda:.2e} m <-- wavelength must be greater than this
super_frequency = c / super_lambda
print(f"Super frequency: {super_frequency:.2e} Hz <-- Frequency must be lower than
```

```
Super lambda: 7.19e-03 m <-- wavelength must be greater than this
Super frequency: 4.17e+10 Hz <-- Frequency must be lower than this
```

In [ ]: