test

September 10, 2024

[ ]:
```python
import numpy as np
import scipy as sp
```

[ ]:
```python
def f(x):
    return x**3 / (np.exp(x) - 1)
```

[ ]:
```python
value, error = sp.integrate.quad(f,0,50)
```

[ ]:
```python
answer = np.pi**4/15

print(value / answer)
```

```
0.9999999999999989
```

[ ]:
```python
[(1 - np.exp(-t)) for t in [0.1,2/3,1,3,10]]
```

[ ]:
```
[0.09516258196404048,
 0.486582880967408,
 0.6321205588285577,
 0.950212931632136,
 0.9999546000702375]
```

[ ]:
```python
import astropy.constants as const
import astropy.units as u
```

[ ]:
```python
h = const.h
c = const.c
k = const.k_B
```

[ ]:
```python
def T(tau,Teff):
    return Teff * (0.75 * (tau + 2/3))**(0.25)

def S(nu,T):
    return (2 * h * nu**3 / (c**2 * (np.exp((h*nu)/(k*T)) - 1)))
```

[ ]:
```python
S(nus[0],Temp)
```

[ ]: $2.0464793 \times 10^{-8} \ \frac{\mathrm{J}}{\mathrm{m^2}}$

[ ]:
```
nus
```

[ ]: $[1.5523073 \times 10^{14},\ 4.6569219 \times 10^{14},\ 1.3970766 \times 10^{15}]\ \frac{1}{\mathrm{s}}$

[ ]:
```
def integrand(tau,mu,nu,Teff):
    return ((S(nu,T(tau,Teff)) / mu) * np.exp(-tau/mu)).value * 1e8
```

[ ]:
```
integrand(0.1,1,nus[0],Temp)
```

[ ]: 1.4316667747736596

[ ]:
```
Temp = 5700 * u.K #Sun

factor  = np.array([1/3,1,3])
xw = 3.921

def nu(x,Teff):
    return x*k*Teff/h

nus = nu(factor*xw,Temp)
```

[ ]:
```
def I(mu,nu,Teff):
    return sp.integrate.quad(integrand,a = 0,b = 100,args = (mu,nu,Teff))
```

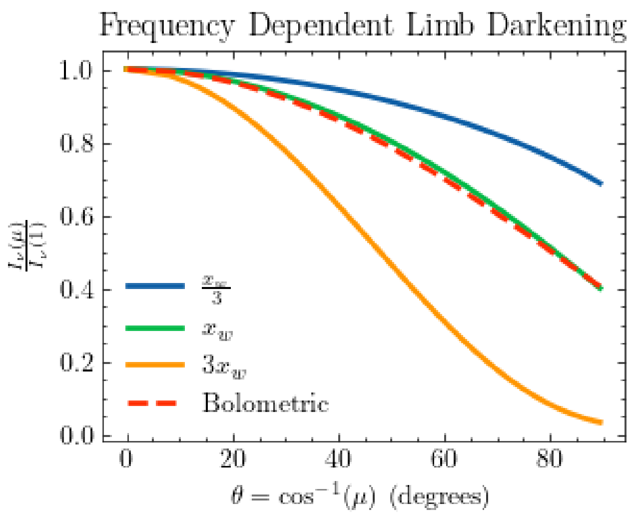
[ ]:
```
mu = np.linspace(0.01,1,100)
theta = np.arccos(mu)
I_mu_nu = np.zeros((len(mu),len(nus)))

for j,n in enumerate(nus):
    for i,m in enumerate(mu):
        I_mu_nu[i,j] = I(m,n,Temp)[0]
    I_nu_1 = I(1,n,Temp)[0]
    I_mu_nu[:,j] /= I_nu_1
```

[ ]:
```
import matplotlib.pyplot as plt
import scienceplots
plt.style.use(['science'])
```

[ ]:
```
b = 0.75 * (const.sigma_sb / np.pi) * Temp**4
a = (2/3) * b
I_bol_mu = (a + b*mu) / (a + b)
```

[ ]:
```
lw = 2
plt.plot(theta*180/np.pi, I_mu_nu[:,0], label = r'$\frac{x_w}{3}$', lw = lw)
plt.plot(theta*180/np.pi, I_mu_nu[:,1], label = r'$x_w$', lw = lw)
plt.plot(theta*180/np.pi, I_mu_nu[:,2], label = r'$3x_w$', lw = lw)
plt.plot(theta*180/np.pi, I_bol_mu, linestyle = 'dashed', label = 'Bolometric',␣
 ↪lw = lw)
```

```python
plt.xlabel(r'$\theta = \rm cos^{-1}(\mu)$ (degrees)')
plt.ylabel(r'$\frac{I_{\nu}(\mu)}{I_{\nu}(1)}$')
plt.title('Frequency Dependent Limb Darkening')
plt.legend()
plt.savefig('limb-darkening.pdf', dpi = 300)
```



[ ]: