Principe - Physics 265 PS2

January 6, 2024

1 Physics 265 Problem Set 2

Rene L. Principe Jr.
PhD Physics
2015-04622

[54]:
```
import numpy as np
import matplotlib.pyplot as plt
```

2 Problem 2.1

Recast the n(λ) expressions of your assigned glass materials (for n_1 and n_2) into its equivalent n(ω) where angular frequency $\omega = 2\pi\nu = \pi c/\lambda$, and c is the speed of light in vacuum.

2.1 $n_1(\lambda)$ of BaSF10

$$n_1^2(\lambda) = 2.6531250 - 8.1388553 \times 10^{-3}\lambda^2 + 2.2995643 \times 10^{-2}\lambda^{-2} + 7.3535957 \times 10^{-4}\lambda^{-4} \quad (1)$$
$$-1.3407390 \times 10^{-5}\lambda^{-6} + 3.6962325 \times 10^{-6}\lambda^{-8} \quad (2)$$

[55]:
```
def BaSF10_y(lambda_):
    term1 = 8.1388553*10**-3 * lambda_**2
    term2 =  2.2995643*10**-2 * lambda_**-2
    term3 =  7.3535957*10**-4 * lambda_**-4
    term4 =  1.3407390*10**-5 * lambda_**-6
    term5 =  3.6962325*10**-6 * lambda_**-8
    return np.sqrt(2.6531250 - term1 + term2 + term3 - term4 + term5)
```

2.2 $n_1(\omega)$ of BaSF10

$$n_1(\omega) = \sqrt{\begin{aligned} &2.6531250 - \frac{8.1388553 \cdot 10^{-3}}{\left(\frac{\omega}{2\pi\cdot 3\cdot 10^8}\right)^2} + 2.2995643 \cdot 10^{-2}\left(\frac{\omega}{2\pi \cdot 3 \cdot 10^8}\right)^2 \\ &+ 7.3535957 \cdot 10^{-4}\left(\frac{\omega}{2\pi \cdot 3 \cdot 10^8}\right)^4 - 1.3407390 \cdot 10^{-5}\left(\frac{\omega}{2\pi \cdot 3 \cdot 10^8}\right)^6 + 3.6962325 \cdot 10^{-6}\left(\frac{\omega}{2\pi \cdot 3 \cdot 10^8}\right)^8 \end{aligned}} \quad (3)$$

[56]:
```
def BaSF10_w(omega):
    term1 =  8.1388553*10**-3 / (omega/(2*np.pi*3e8))**2
```

```
    term2 =  2.2995643*10**-2 * (omega/(2*np.pi*3e8))**2
    term3 =  7.3535957*10**-4 * (omega/(2*np.pi*3e8))**4
    term4 =  1.3407390*10**-5 * (omega/(2*np.pi*3e8))**6
    term5 =  3.6962325*10**-6 * (omega/(2*np.pi*3e8))**8
    return np.sqrt(2.6531250 - term1 + term2 + term3 - term4 + term5)
```

2.3 $n_2(\lambda)$ of BaK1

$$n_2^2(\lambda) = 2.4333007 - 8.4931353\times10^{-3}\lambda^2 + 1.3893512\times10^{-2}\lambda^{-2} + 2.6798268\times10^{-4}\lambda^{-4} \tag{4}$$
$$-6.1946101\times10^{-6}\lambda^{-6} + 6.2209005\times10^{-7}\lambda^{-8} \tag{5}$$

[57]:
```
def BaK1_y(lambda_):
    term1 = 8.4931353*10**-3 * lambda_**2
    term2 =  1.3893512*10**-2 * lambda_**-2
    term3 =  2.6798268*10**-4 * lambda_**-4
    term4 =  6.1946101*10**-6 * lambda_**-6
    term5 =  6.2209005*10**-7 * lambda_**-8
    return np.sqrt(2.4333007 - term1 + term2 + term3 - term4 + term5)
```

2.4 $n_2(\omega)$ of BaK1

$$n_2(\omega) = \sqrt{\begin{aligned}&2.4333007 - \frac{8.4931353\cdot10^{-3}}{\left(\frac{\omega}{2\pi\cdot3\cdot10^8}\right)^2} + 1.3893512\cdot10^{-2}\left(\frac{\omega}{2\pi\cdot3\cdot10^8}\right)^2\\ &+ 2.6798268\cdot10^{-4}\left(\frac{\omega}{2\pi\cdot3\cdot10^8}\right)^4 - 6.1946101\cdot10^{-6}\left(\frac{\omega}{2\pi\cdot3\cdot10^8}\right)^6 + 6.2209005\cdot10^{-7}\left(\frac{\omega}{2\pi\cdot3\cdot10^8}\right)^8\end{aligned}} \tag{6}$$

[58]:
```
def BaK1_w(omega):
    term1 =  8.4931353*10**-3 / (omega/(2*np.pi*3e8))**2
    term2 =  1.3893512*10**-2 * (omega/(2*np.pi*3e8))**2
    term3 =  2.6798268*10**-4 * (omega/(2*np.pi*3e8))**4
    term4 =  6.1946101*10**-6  * (omega/(2*np.pi*3e8))**6
    term5 =  6.2209005*10**-7 * (omega/(2*np.pi*3e8))**8
    return np.sqrt(2.4333007 - term1 + term2 + term3 - term4 + term5)
```

[59]:
```
c = 3e8 #speed of light in vacuum

def Lambda(omega):
    return 2*np.pi*c/omega

def Omega(lambda_):
    return 2*np.pi*c/lambda_
```

[60]:
```
lambda_min, lambda_max = 0.37, 1.011
step = 0.005
lambda_list = np.arange(lambda_min, lambda_max, step)
omega = Omega(lambda_list)
```

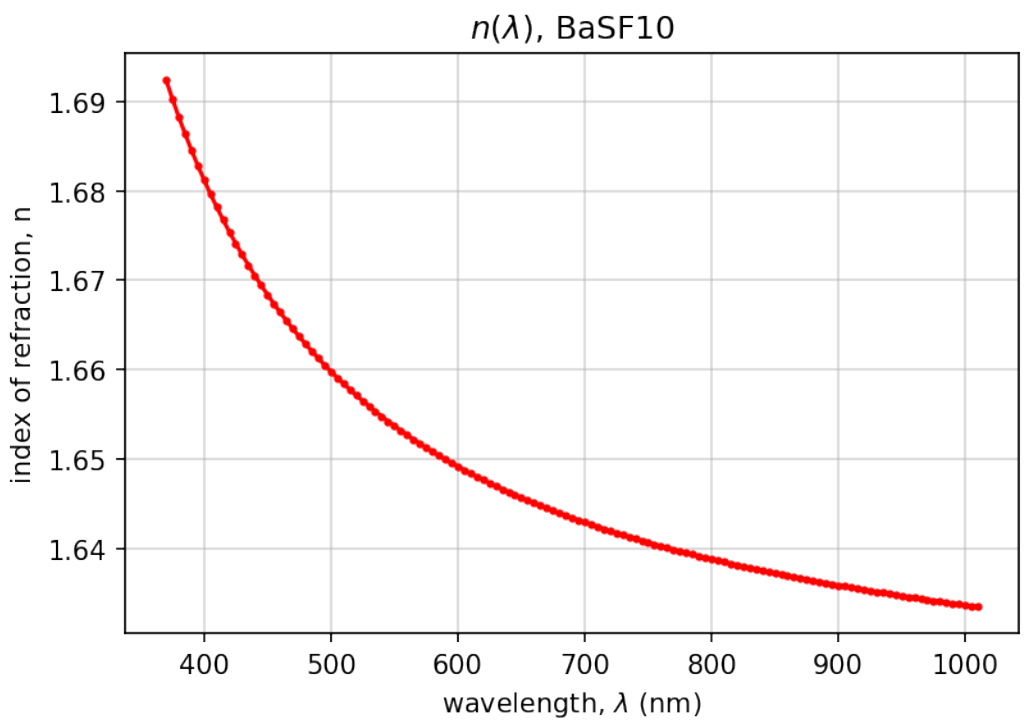
3 Problem 2.2

Plot (at 0.5 nm resolution) the refractive index n( ) for n_1 and n_2 within their corresponding non-dispersive wavelength ranges (in nanometer units). Refer to the attached Dispersion Formula Table 23 (Handbook of Optics Vol 2, Sec 33.67, 2nd Edition, OSA) for details.

[61]:
```
BaSF10_lambda = BaSF10_y(lambda_list)
BaK1_lambda = BaK1_y(lambda_list)

plt.figure(dpi = 150)
plt.grid(alpha = 0.5)
plt.plot(lambda_list*1e3 , BaSF10_lambda , 'ro-', markersize =2,
        markerfacecolor =None)
# plt.xlim (100 , 3800 )
plt.xlabel('wavelength, $\lambda$ (nm)')
plt.ylabel('index of refraction, n')
plt.title('$n(\\lambda )$, BaSF10')
```

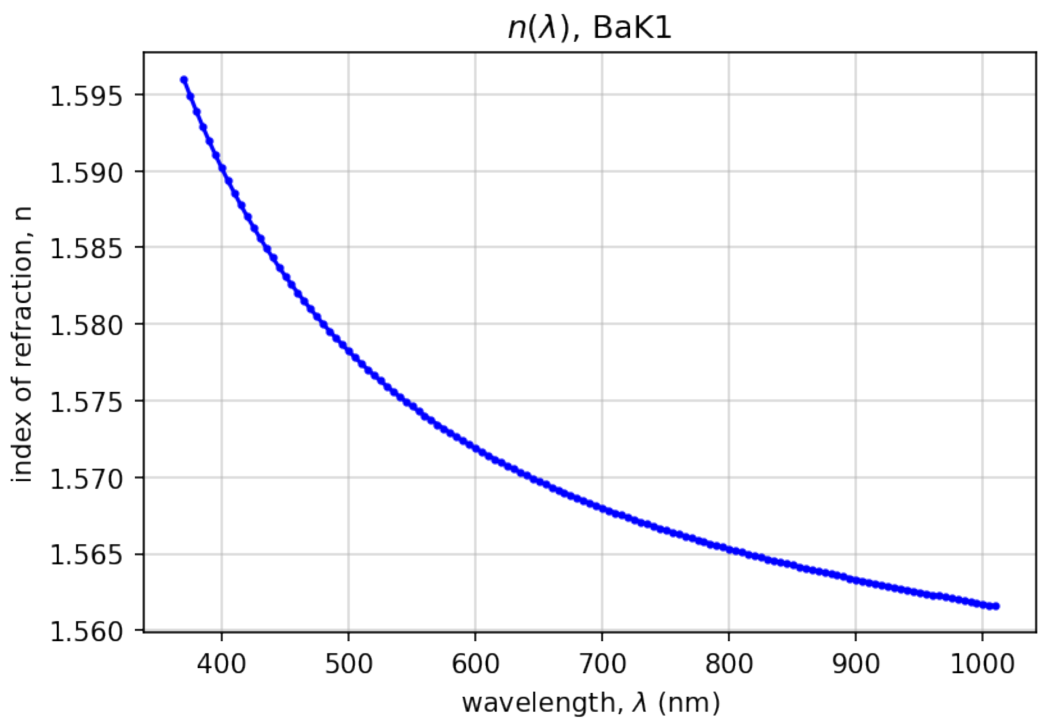
[61]:
```
Text(0.5, 1.0, '$n(\\lambda )$, BaSF10')
```



[62]:
```
plt.figure(dpi = 150)
plt.grid(alpha = 0.5)
plt.plot(lambda_list*1e3 , BaK1_lambda , 'bo-', markersize =2)
plt.xlabel('wavelength, $\lambda$ (nm)')
```

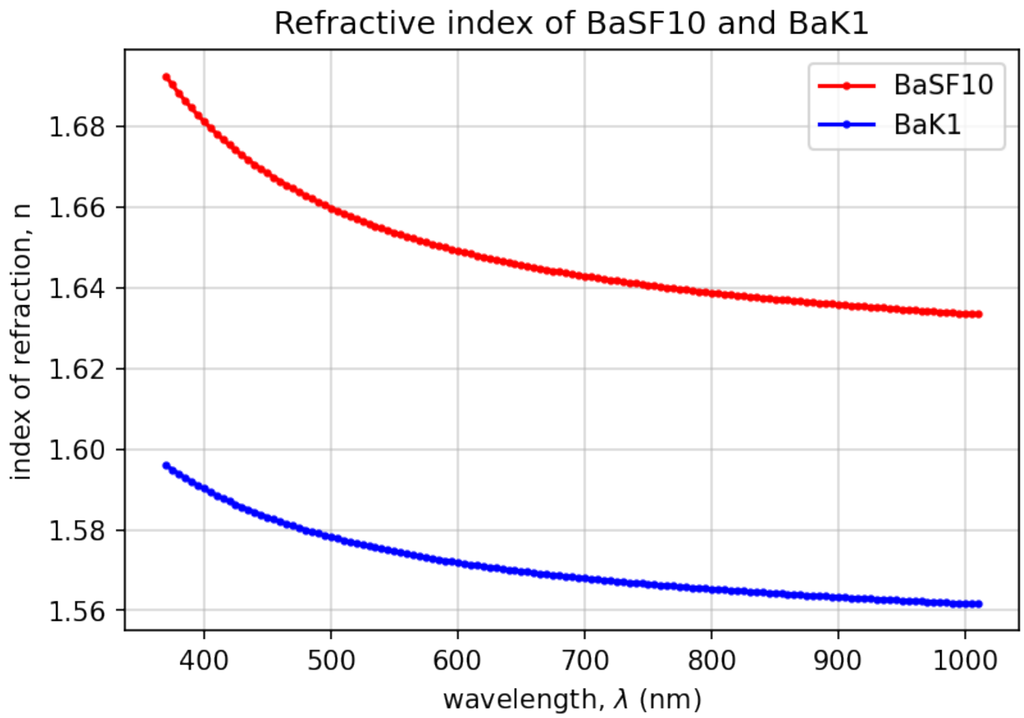
```
plt.ylabel('index of refraction, n')
plt.title('$n(\\lambda )$, BaK1')
```

[62]: Text(0.5, 1.0, '$n(\\lambda)$, BaK1')



[63]:
```
plt.figure(dpi = 150)
plt.grid(alpha = 0.5)
plt.plot(lambda_list*1e3 , BaSF10_lambda , 'ro-', markersize =2, label =␣
 ↪'BaSF10')
plt.plot(lambda_list*1e3 , BaK1_lambda , 'bo-', markersize =2, label = 'BaK1')
plt.xlabel('wavelength, $\lambda$ (nm)')
plt.ylabel('index of refraction, n')
plt.title('Refractive index of BaSF10 and BaK1')
plt.legend()
```

[63]: <matplotlib.legend.Legend at 0x7f953103d520>



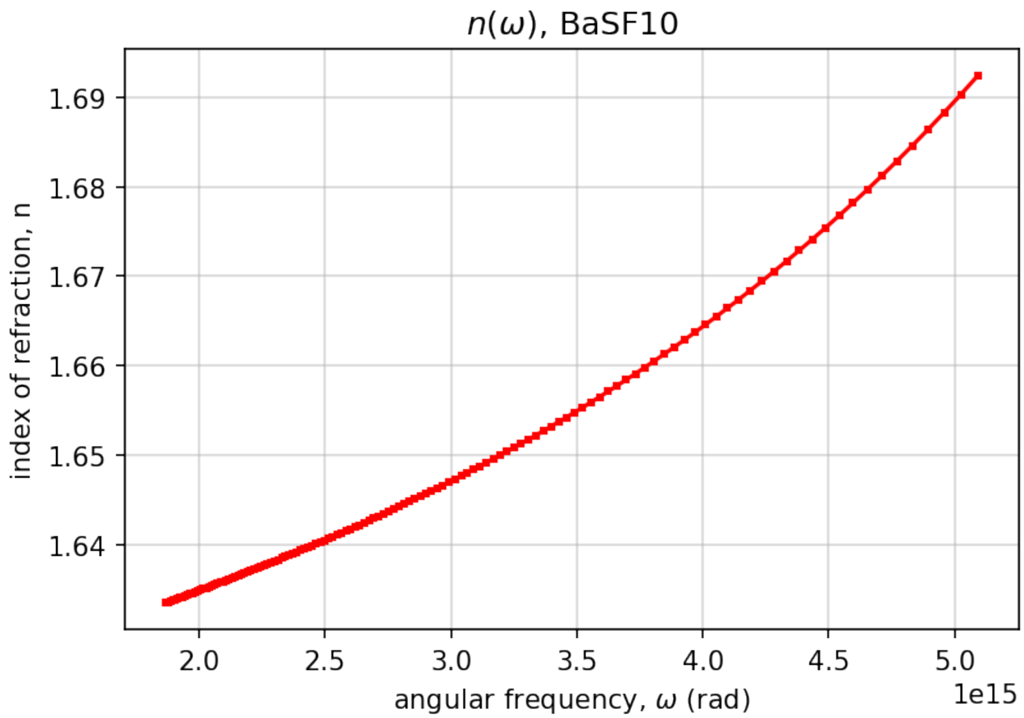
4 Problem 2.3

P lot the equivalent n( ) expressions within their corresponding non-dispersive -range bounded by $_{max}$ and $_{min}$.

[64]:
```
omega = Omega(lambda_list)
```

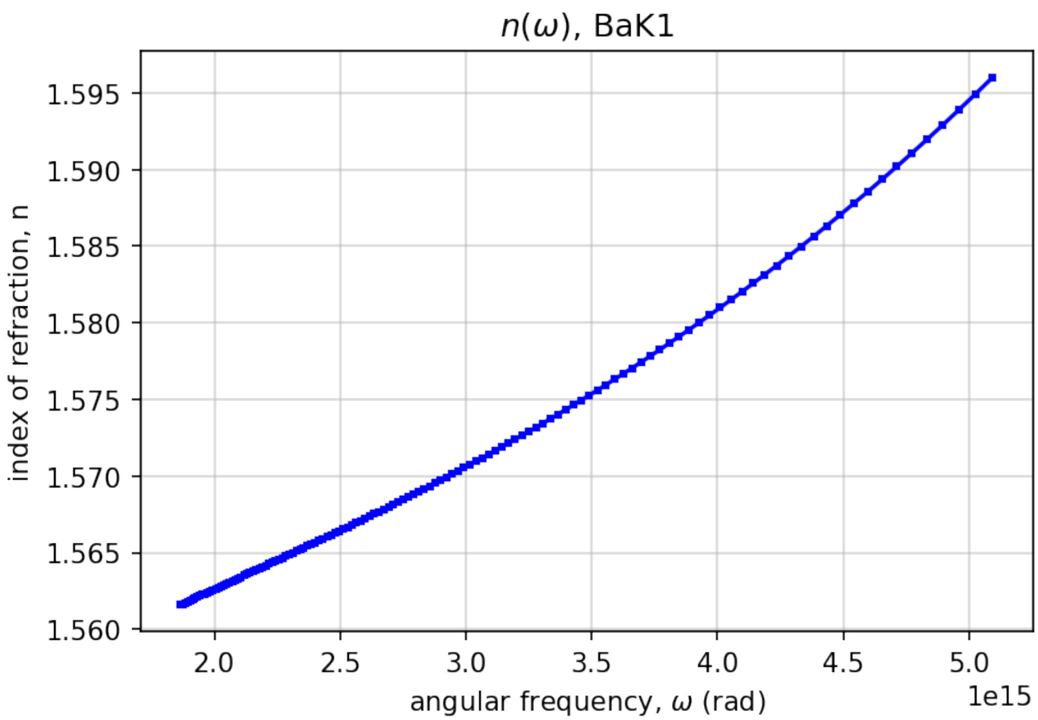
[65]:
```
plt.figure(dpi = 150)
plt.grid(alpha = 0.5)
plt.plot(omega*1e6, BaSF10_w(omega) , 'rs-', markersize =2)
plt.xlabel('angular frequency, $\omega$ (rad)')
plt.ylabel('index of refraction, n')
plt.title('$n(\\omega )$, BaSF10')
```

[65]:
```
Text(0.5, 1.0, '$n(\\omega )$, BaSF10')
```



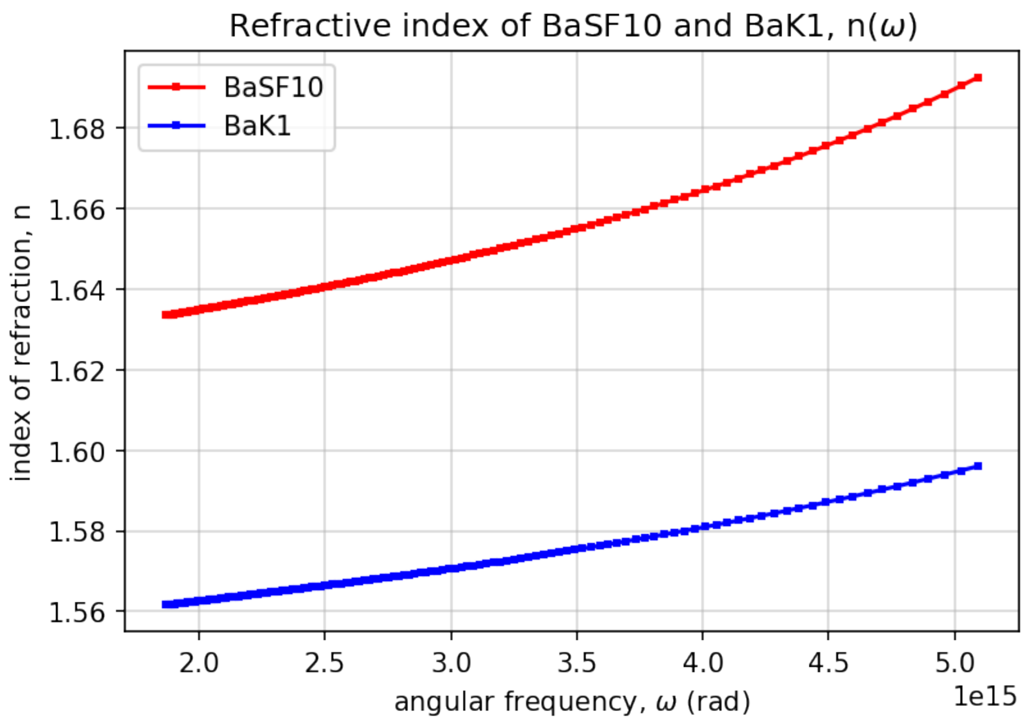
[66]:
```
plt.figure(dpi = 150)
plt.grid(alpha = 0.5)
plt.plot(omega*1e6, BaK1_w(omega) , 'bs-', markersize =2)
plt.xlabel('angular frequency, $\omega$ (rad)')
plt.ylabel('index of refraction, n')
plt.title('$n(\\omega )$, BaK1')
```

[66]:
```
Text(0.5, 1.0, '$n(\\omega )$, BaK1')
```



[67]:
```python
plt.figure(dpi = 150)
plt.grid(alpha = 0.5)
plt.plot(omega*1e6 , BaSF10_w(omega) , 'rs-', markersize =2, label = 'BaSF10')
plt.plot(omega*1e6 , BaK1_w(omega) , 'bs-', markersize =2, label = 'BaK1')
plt.xlabel('angular frequency, $\omega$ (rad)')
plt.ylabel('index of refraction, n')
plt.title('Refractive index of BaSF10 and BaK1, n($\omega$)')
plt.legend()
```

[67]: <matplotlib.legend.Legend at 0x7f95312476a0>



5 Problem 2.4

Plot for both n_1 and n_2 the pertinent n(- $_o$) curves in the presence of a resonant frequency $_o$ = ($_{max}$ + $_{min}$)/2. Please be guided by the discussion in Section 2.3.4 of Born & Wolf particularly Figure 2.3.

[68]:
```
omega = Omega(lambda_list)

omega_min, omega_max = np.max(omega), np.min(omega)
omega_0 = (omega_min + omega_max)/2
omega_res = omega - omega_0
```

[69]:
```
print(' _min = %.6e' % omega_min)
print(' _max = %.6e' % omega_max)
print(' _0 = %.6e' % omega_0)
```

```
 _min = 5.094475e+09
 _max = 1.866293e+09
 _0 = 3.480384e+09
```

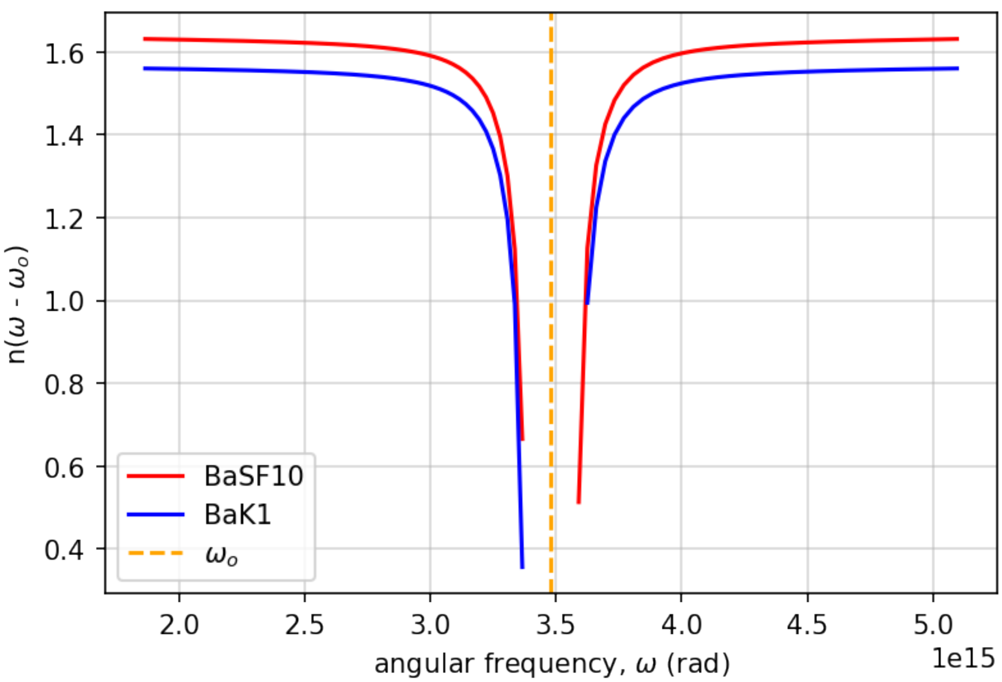
[70]:
```
BaSF10_w_res = BaSF10_w(omega_res)
BaK1_w_res = BaK1_w(omega_res)
```

```
/var/folders/rk/gz59x8xx0bz69pn391mjkglw0000gn/T/ipykernel_2344/1004182490.py:7:
```

```
RuntimeWarning: invalid value encountered in sqrt
  return np.sqrt(2.6531250 - term1 + term2 + term3 - term4 + term5)
```

[71]:
```
plt.figure(dpi = 150)
plt.grid(alpha = 0.5)
plt.plot(omega*1e6, BaSF10_w_res, 'r', label = 'BaSF10')
plt.plot(omega*1e6, BaK1_w_res, 'b', label = 'BaK1')
plt.axvline(omega_0*1e6, ls = '--', color = 'orange', label = '$\omega_o$')
plt.legend()
plt.ylabel('n($\omega$ - $\omega_o$)')
plt.xlabel('angular frequency, $\omega$ (rad)')
```

[71]:
```
Text(0.5, 0, 'angular frequency, $\\omega$ (rad)')
```



[72]:
```
lambda_min, lambda_max = 0.1, 3.011
step = 0.0001
lambda_list = np.arange(lambda_min, lambda_max, step)

omega = Omega(lambda_list)
omega_min, omega_max = np.max(omega), np.min(omega)
omega_0 = (omega_min + omega_max)/2
omega_res = omega - omega_0

BaSF10_w_res = BaSF10_w(omega_res)
BaK1_w_res = BaK1_w(omega_res)
```

```
plt.figure(dpi = 150)
plt.grid(alpha = 0.5)
plt.plot(omega*1e6, BaSF10_w_res, 'r', label = 'BaSF10')
plt.plot(omega*1e6, BaK1_w_res, 'b', label = 'BaK1')
plt.axvline(omega_0*1e6, ls = '--', color = 'orange', label = '$\omega_o$')
plt.legend()
plt.ylabel('n($\omega$ - $\omega_o$)')
plt.xlabel('angular frequency, $\omega$ (rad)')
```

```
/var/folders/rk/gz59x8xx0bz69pn391mjkglw0000gn/T/ipykernel_2344/1004182490.py:7:
RuntimeWarning: invalid value encountered in sqrt
  return np.sqrt(2.6531250 - term1 + term2 + term3 - term4 + term5)
```

[72]:
```
Text(0.5, 0, 'angular frequency, $\\omega$ (rad)')
```