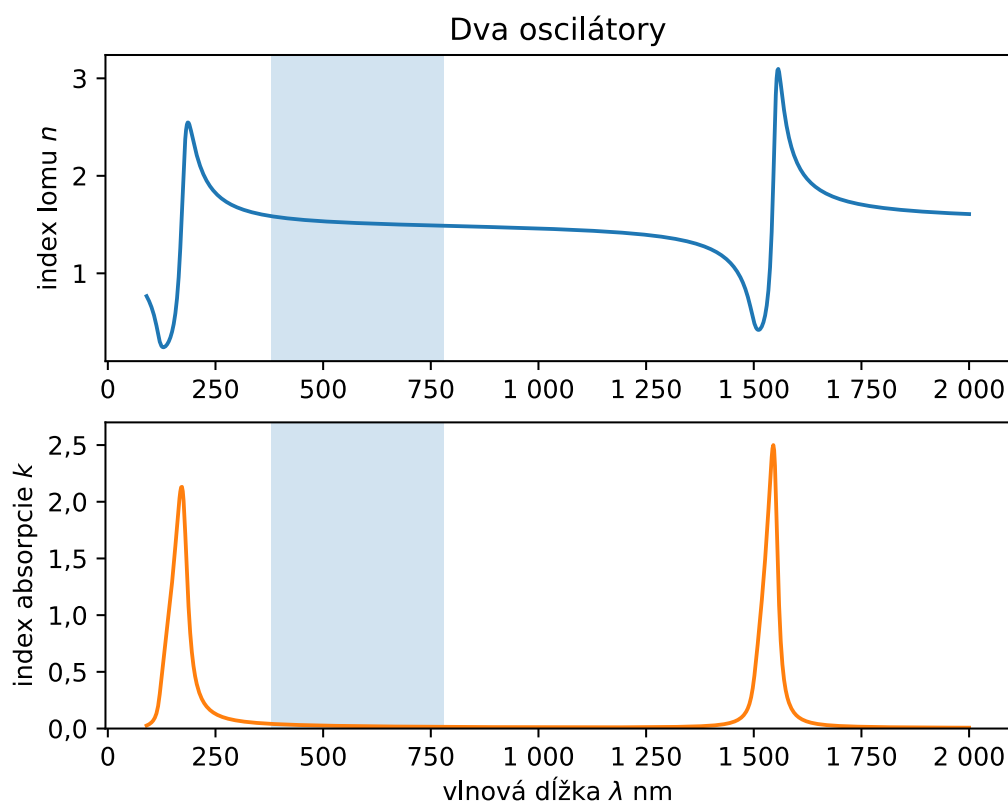


Index lomu optických skiel

Priehľadné optické číre sklá absorbujú v UV oblasti a v ďalekej IČ oblasti. Viditeľné svetlo prechádza sklom takmer bez absorpcie, preto je [Sellmeierov model](#) vhodným priblížením multioscilátorového charakteru takýchto materiálov. Na index lomu vo viditeľnej oblasti majú vplyv zväčša 2 až 4 oscilátory. Farebné sklá absorbujú aj vo viditeľnej oblasti. Tu už Sellmeierov model použiť nemožno, pretože ten neuvažuje absorpciu.

Na nasledujúcom obrázku je index lomu materiálu s absorpčným spektrom, ktoré je charakteristické dvomi absorpčnými čiarami pri vlnových dĺžkach $\lambda_1 = 178 \text{ nm}$ a $\lambda_2 = 1\,550 \text{ nm}$. Interval spektra viditeľného svetla je v grafoch vyznačený modrými obdĺžnikmi. Materiál teda absorbuje v ultrafialovej a infračervenej oblasti, vo viditeľnej takmer neabsorbuje. Index lomu vo viditeľnej oblasti môžeme aproximovať [Cauchyho modelom](#). Infračervená absorpcia však spôsobuje prudší pokles indexu lomu pre väčšie vlnové dĺžky, tu sa skôr uplatní Sellmeierov model.



...

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

# Nastavenie desatinnej čiarky namiesto bodky v grafoch
import locale
locale.setlocale(locale.LC_ALL, "sk_SK")
plt.rcParams["axes.formatter.use_locale"] = True
plt.rcParams["axes.formatter.use_mathtext"] = True
```

```

def Lorenz2(x, x01, F1, G1, x02, F2, G2):
    return 1 + F1*x01**2/(x01**2 - x**2 - 1j*G1*x) + F2*x02**2/(x02**2 - x**2 -
1j*G2*x)

lambdaMin = 90 #nm
lambdaMax = 2000 #nm
numPoints = 1001
wl = np.linspace(lambdaMin, lambdaMax, numPoints)
omega = 1240/wl #eV

#prvý oscilátor
f1 = 0.15
gamma1 = 0.01 #eV
omega1 = 0.8 #eV

#druhý oscilátor
f2 = 1.2
gamma2 = 1.0 #eV
omega2 = 7.0 #eV

n = np.sqrt(Lorenz2(omega, omega1, f1, gamma1, omega2, f2, gamma2))

ax1 = plt.subplot(211)
ax1.plot(wl, n.real, label = "reálna časť")

ax2 = plt.subplot(212, sharex = ax1)
ax2.plot(wl, n.imag, color="tab:orange", label = "imaginárna časť")

plt.xlabel("vlnová dĺžka $\lambda$ nm")
ax1.set_ylabel("index lomu $n$")
ax2.set_ylabel("index absorpcie $k$")
ax1.set_title("Dva oscilátory")

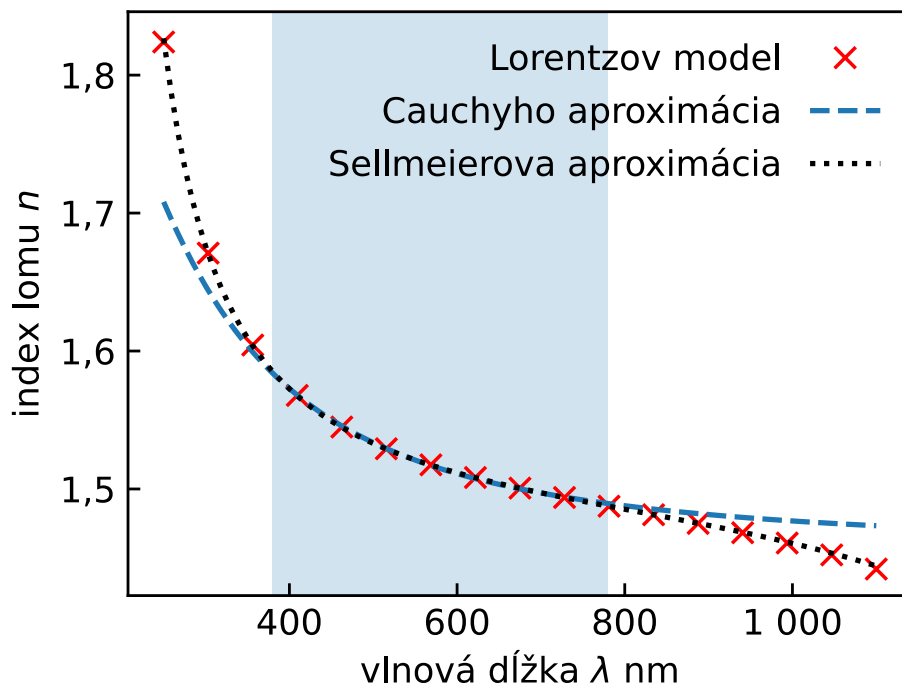
#vyznačenie oblasti viditeľného svetla
limits1 = ax1.axis("tight")
ax1.fill_between([380, 780], limits1[3], limits1[2], alpha = 0.2)
ax2.set_ylim([0, 2.7])
ax2.fill_between([380, 780], 2.7, 0, alpha = 0.2)

plt.show()

```

Pre lepší prehľad zobrazíme index lomu tohto modelu v intervale vlnových dĺžok 250 nm až 1100 nm a porovnáme ho s Cauchyho a Sellmeierovou aproximáciou.

Dvojoscilátorový Lorentzov model



Z grafu je jasné, že vo viditeľnej oblasti aproximujú index lomu rovnako dobre Cauchyho aj Sellmeierov model. Cauchyho model však zlyháva v UV aj v IČ oblasti.

Graf sme získali pomocou nasledujúceho programu:

```
import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

# Nastavenie desatinnej čiarky namiesto bodky v grafoch
import locale
locale.setlocale(locale.LC_ALL, "sk_SK")
plt.rcParams["axes.formatter.use_locale"] = True
plt.rcParams["axes.formatter.use_mathtext"] = True

def Lorentz2(x, x01, F1, G1, x02, F2, G2):
    return 1 + F1*x01**2/(x01**2 - x**2 - 1j*G1*x) + F2*x02**2/(x02**2 - x**2 - 1j*G2*x)

def Cauchy(x, A, B, C):
    return A + B/x**2 + C/x**4

def Sellmeier(x, B1, C1, B2, C2):
    return 1 + B1*x**2/(x**2 - C1) + B2*x**2/(x**2 - C2)

lambdaMin = 250 #nm
lambdaMax = 1100 #nm
numPoints = 101
wl = np.linspace(lambdaMin, lambdaMax, numPoints)

wlFit = np.linspace(0.380, 0.780, 101)
wlLorentz = np.linspace(lambdaMin, lambdaMax, 17)
```

```

omega = 1240/wlLorentz #eV

f1 = 0.15
gamma1 = 0.01 #eV
omega1 = 0.8 #eV

f2 = 1.2
gamma2 = 1.0 #eV
omega2 = 7.0 #eV

epsFit = Lorentz2(1.24/wlFit, omega1, f1, gamma1, omega2, f2, gamma2)
parCa, covCa = curve_fit(Cauchy, wlFit, np.sqrt(epsFit).real, [1.5, 0, 0])
parSe, covSe = curve_fit(Sellmeier, wlFit, epsFit.real,
                        [f1, (1.24/omega1)**2, f2, (1.24/omega2)**2])

n = np.sqrt(Lorentz2(omega, omega1, f1, gamma1, omega2, f2, gamma2))

plt.figure(figsize=(4, 3))
plt.tick_params(axis='both', direction="in")

plt.plot(wlLorentz, n.real, "rx", label = "Lorentzov model")
plt.plot(wl, Cauchy(wl*1e-3, *parCa), "--", label = "Cauchyho aproximácia")
plt.plot(wl, np.sqrt(Sellmeier(wl*1e-3, *parSe)), "k:",
        label = "Sellmeierova aproximácia")

plt.xlabel("vlnová dĺžka  $\lambda$  nm")
plt.ylabel("index lomu  $n$ ")
plt.title("Dvojoscilátorový Lorentzov model")
plt.legend(markerfirst = False, frameon = False)

#vyznačenie oblasti viditeľného svetla
limits = plt.axis("tight")
plt.fill_between([380, 780], limits[3], limits[2], alpha = 0.2)

plt.show()

```