Luca Cordes, 444900
Lisa Evers, 423114

Experimentalphysik Va (WS 2023/2024)

Übung 5

Tutorium: 1 Abgabe: 02.12.2024

Aufgabe 1: Thermische Ausdehnung

Wenn es irgentwo so aussehen sollte, als ob der Rechenweg übersprungen wurde, liegt das daran, dass er in Python gemacht wurde. Der ganze Code steht auf den letzen Seiten.

(a)

Der Gleichgewichtsabstand liegt bei $r_0 = 2^{\frac{1}{6}}\sigma$.

$$V(r) \overset{!}{=} V(r_0) + \frac{D_2}{2}(r-r_0)^2 + \frac{D_3}{3!}(r-r_0)^3 + \mathcal{O}\left((r-r_0)^4\right)$$

$$\implies D_2 = \left.\frac{\partial^2 V}{\partial r^2}\right|_{r_0} = \frac{72\epsilon}{r_0^2} = \frac{36 \cdot 2^{\frac{2}{3}}\epsilon}{\sigma^2}$$

$$\implies D_3 = \left.\frac{\partial^3 V}{\partial r^3}\right|_{r_0} = -\frac{1512\epsilon}{r_0^3} = -\frac{756\sqrt{2}\epsilon}{\sigma^3}$$

(b)

Die Ausweitung der Integrationsgrenzen auf $(-\infty, \infty)$ eigenet sich als Näherung, weil für $|r - r_0| \gg 1$ der Integrand exponentiell unterdrückt wird.

$$\langle x\rangle_T = \frac{\int \mathrm{d}x\, x e^{-\beta U(x)}}{\int \mathrm{d}x\, e^{-\beta U(x)}}$$

$$e^{-\beta U} \approx e^{-\beta\left(\frac{D_2}{2}(x-x_0)^2 + \frac{D_3}{3!}(x-x_0)^3\right)} \approx e^{-\beta\frac{D_2}{2}(x-x_0)^2}\left(1 - \beta\frac{D_3}{3!}(x-x_0)^3\right)$$

$$\int_0^\infty \mathrm{d}x\, e^{-\beta U} \approx \int_{-\infty}^\infty \mathrm{d}x\, e^{-\beta\frac{D_2}{2}(x-x_0)^2}\left(1 - \beta\frac{D_3}{3!}(x-x_0)^3\right) = \frac{2^{\frac{1}{6}}\sqrt{\pi}\sigma}{6\sqrt{\beta\epsilon}}$$

$$\int_0^\infty \mathrm{d}x\, x e^{-\beta U} \approx \int_{-\infty}^\infty \mathrm{d}x\, x e^{-\beta\frac{D_2}{2}(x-x_0)^2}\left(1 - \beta\frac{D_3}{3!}(x-x_0)^3\right) = \frac{2^{\frac{1}{3}}\sqrt{\pi}\sigma^2(48\beta\epsilon + 7)}{288\beta^{\frac{3}{2}}\epsilon^{\frac{3}{2}}}$$

$$\implies \boxed{\langle x\rangle_T \approx 2^{\frac{1}{6}}\sigma\left(1 + \frac{7}{48\beta\epsilon}\right)}$$

(c)

$$
\begin{aligned}
l = l_0(1+\alpha\Delta T) &= N\langle x\rangle_T \\
&= N2^{\frac{1}{6}}\sigma\left(1+\frac{7k_BT}{48\epsilon}\right) \\
\implies \boxed{\alpha = \frac{7k_B}{48\epsilon} \approx 0.00126\frac{1}{\mathrm{K}}}
\end{aligned}
$$

Der Ausdehnungskoeffizient hängt nicht von σ ab, und befindet sich mit $\sim 10^{-3}\frac{1}{\mathrm{K}}$ in der typischen Größenordnung für Flüssigkeiten.

Aufgabe 2: Zweiatomige vs. einatomige lineare Kette

(a)

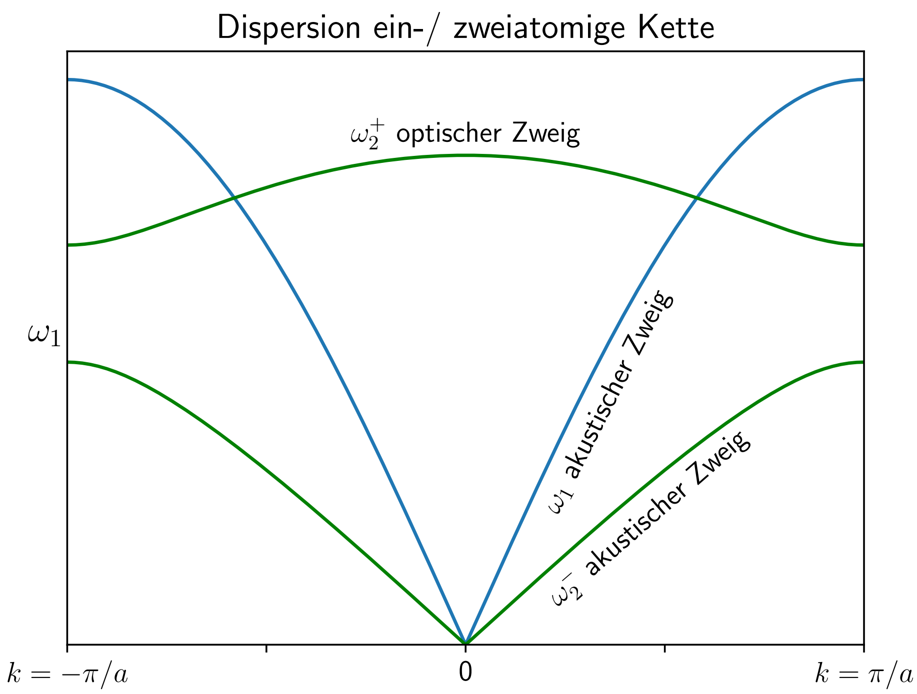


Abbildung 1: Dispersionskurve $\omega_2^{\pm}$ mit $M = 2m$

Beide Dispersionskurven

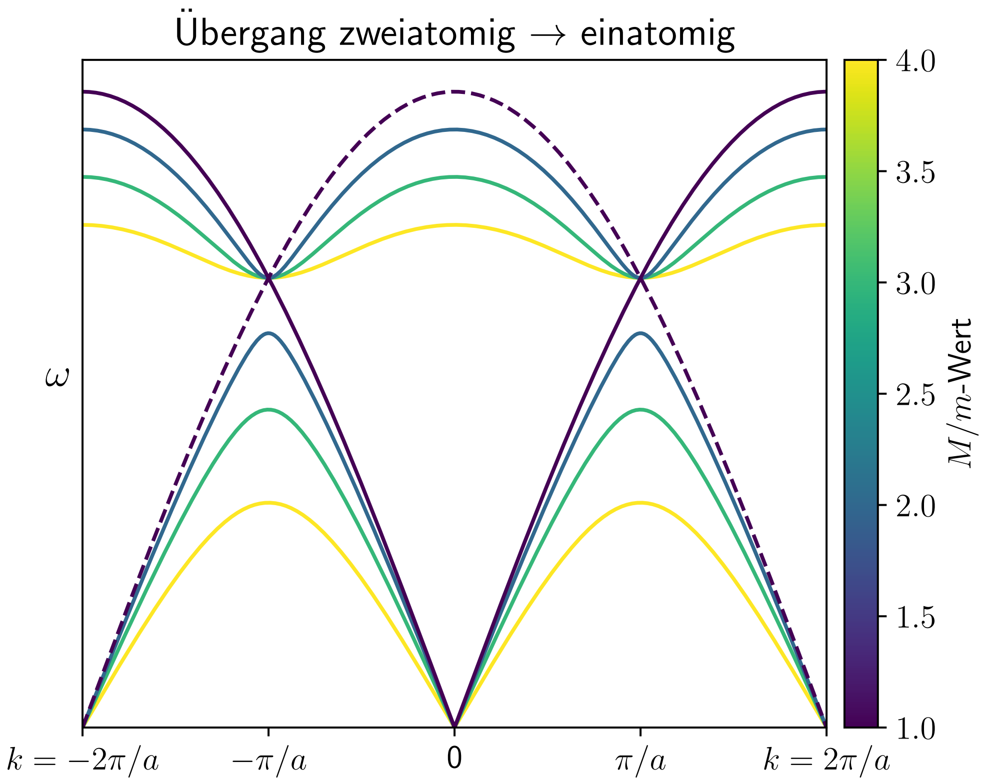
- haben einen (qualitativ identischen) optischen Zweig
- sind im k-Raum periodisch mit der Periode $\frac{2\pi}{a}$.

und unterschieden sich darin, dass

- die Dispersionkurve zu ω_2 zu jedem k eine zweite Lösung hat, welche auf dem optischen Zweig liegt

- es nur bei ω_2 Bandlücken gibt (zwischen den beiden Zweigen).

(b)



Man kann im Plot sehen, wie sich die Dispersionkurve der zweiatomigen Kette, der der einatomigen annähert. Die Frequenzlücke wird immer kleiner, bis sie verschwindet. Im Fall dass $M = m$ ändert sich schlagartig die Größe der Einheitszelle $a \to a/2$, sodass man hier die Achsen neu definieren muss.

Aufgabe 3: Kette mit verschiedenen Federn

(a)

$$\begin{aligned}
\omega_- &= \sqrt{\frac{D_1 + D_2}{m} - \frac{1}{m}\sqrt{D_1^2 + D_2^2 + 2D_1 D_2 \cos(ka)}} \\
v &= \lim_{k\to 0} \frac{\partial \omega_-}{\partial k} \\
&= \lim_{k\to 0} \frac{D_1 D_2 a \sin(ak)}{2m\sqrt{\frac{D_1+D_2}{m} - \frac{1}{m}\sqrt{D_1^2 + D_2^2 + 2D_1 D_2 \cos(ak)}}\sqrt{D_1^2 + D_2^2 + 2D_1 D_2 \cos(ak)}} \\
&= \lim_{k\to 0} \frac{D_1 D_2 a^2 k}{2\sqrt{m}\sqrt{D_1 + D_2 - \left(D_1 + D_2 - \frac{D_1 D_2 a^2 k^2}{2(D_1+D_2)}\right)}(D_1 + D_2)}
\end{aligned}$$

$$= \frac{D_1 D_2 a}{\sqrt{2m}\sqrt{\frac{D_1 D_2}{(D_1+D_2)}}(D_1 + D_2)}$$

$$\boxed{v = \sqrt{\frac{a^2 D_1 D_2}{2m(D_1 + D_2)}}}$$

(b)

Der Term ist 2 mal das harmonisches Mittel der Federkonstanten und stellt eine effektive Federkonstante dar, die die Wechselwirkung zwischen den beiden benachbarten Atomen in der Kette beschreibt.

Aufgabe 4: Phononen in Cu

(a)

$$v = a\sqrt{\frac{D}{m}}, \qquad \omega_{\text{max}} = \omega(k = \pi/a) = 2\sqrt{\frac{D}{m}} = \frac{2v}{a}$$

$$\boxed{\omega_{\text{max}} \approx 23.8\,\text{THz}}$$

(b)

$$\boxed{E_{\text{max}} = \hbar\omega_{\text{max}} \approx 15.7\,\text{meV}}$$

Aufgabe 5: Streuung von Licht an Schallwellen (Brillouin-Streuung)

(a)

$$k_{\text{photon}} = nk_0 = n\frac{2\pi}{\lambda} \approx 13.9 \cdot 10^6 \frac{1}{m}$$
$$\Delta k_{\text{max}} = 2k_{\text{ph}} \approx 27.9 \cdot 10^6 \frac{1}{\text{m}}$$
$$\boxed{\Delta p_{\text{max}} = \hbar\Delta k_{\text{max}} \approx 2.94 \cdot 10^{-27} \frac{\text{kg m}}{\text{s}}}$$

(b)

$$\boxed{\omega_{\text{phonon}} = v\Delta k_{\text{max}} \approx 167\,\text{GHz}}$$

(c)

$$\boxed{\frac{\Delta E}{E} = \frac{\omega_{\text{phonon}}}{\omega_{\text{photon}}} \approx 6.16 \cdot 10^{-5}}$$

jupyter

December 1, 2024

[1]:
```
import scipy.constants as c
from IPython.display import display as print
from sympy import *
init_printing(use_latex="mathjax")
```

[2]:
```
# Nr 1 (a)
sigma, epsilon, r, r_0, beta, k_B, T = symbols("sigma epsilon r r_0 beta k_B␣
↪T",positive=True)
V = 4*epsilon*((sigma/r)**12 - (sigma/r)**6)
D_2 = V.diff(r,2).subs(r,2**(Rational(1,6))*sigma)
D_3 = V.diff(r,3).subs(r,2**(Rational(1,6))*sigma)
print(D_2,D_3)
```

$$\frac{36 \cdot 2^{\frac{2}{3}} \epsilon}{\sigma^2}$$

$$-\frac{756\sqrt{2}\epsilon}{\sigma^3}$$

[3]:
```
# (b)
weight = exp(-beta * D_2/2 * (r-r_0)**2) * (1 - beta * D_3/6 * (r-r_0)**3)
Z = Integral(weight, (r,-oo,oo)).doit()
rZ = simplify(Integral(r*weight, (r,-oo,oo)).doit().subs(r_0,␣
↪2**Rational(1,6)*sigma))
x_ev = simplify(rZ/Z)
print(Z,rZ,x_ev)
```

$$\frac{\sqrt[6]{2}\sqrt{\pi}\sigma}{6\sqrt{\beta}\sqrt{\epsilon}}$$

$$\frac{\sqrt[3]{2}\sqrt{\pi}\sigma^2 \cdot (48\beta\epsilon + 7)}{288\beta^{\frac{3}{2}}\epsilon^{\frac{3}{2}}}$$

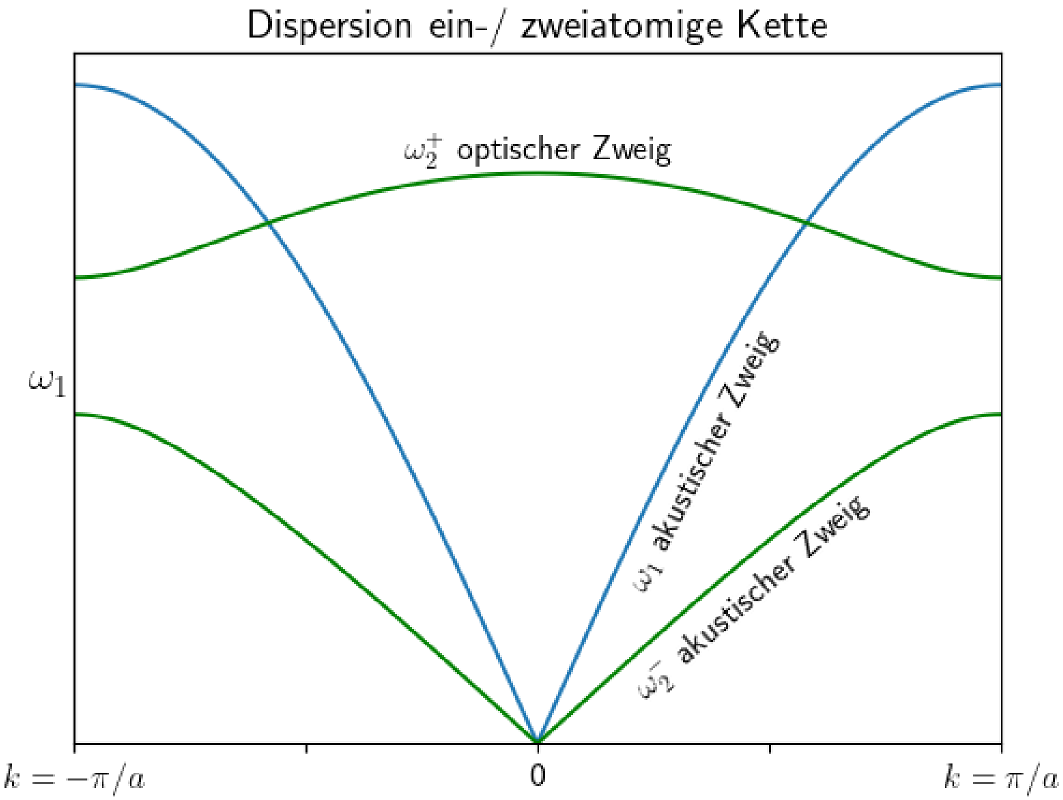
$$\frac{\sqrt[6]{2}\sigma\,(48\beta\epsilon + 7)}{48\beta\epsilon}$$

[4]:
```
# (c)
alpha = 7*c.Boltzmann / (48 * 10e-3 * c.e)
alpha
```

[4]:

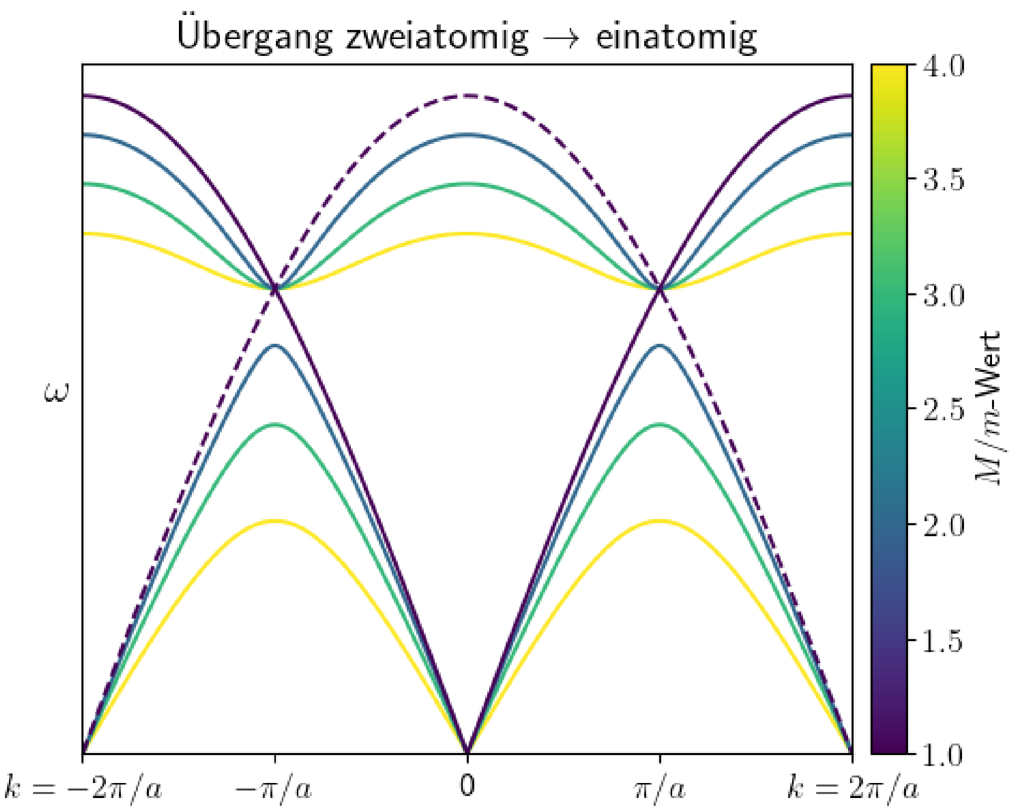
0.00125669443406284

[5]:
```
# Nr 2 (a)
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams.update({
                    "text.usetex": True, "font.size": 13
                    })

k = np.linspace(-np.pi,np.pi,1000)
omega1 = np.sqrt(4 *np.sin(k/2)**2)
omega2optical = np.sqrt(3/2 + np.sqrt((3/2)**2 - 2*np.sin(k/2)**2))
omega2acoustic = np.sqrt(3/2 - np.sqrt((3/2)**2 - 2*np.sin(k/2)**2))

fig,ax = plt.subplots()
ax.plot(k, omega1)
ax.plot(k, omega2acoustic, c="g")
ax.plot(k, omega2optical,c="g")
plt.text(0,1.78, r"$\omega_2^+$ optischer Zweig",ha="center")
plt.text(1.44,0.14, r"$\omega_2^-$ akustischer Zweig",ha="center",rotation=41)
plt.text(1.12,0.47, r"$\omega_1$ akustischer Zweig",ha="center",rotation=63)
ax.set(ylim=(0,2.1),xlim=(-np.pi,np.pi),xticks=[-np.pi,-np.pi/2,0,np.pi/2,np.
 ↪pi],xticklabels=[r"$k=-\pi/a$","","0","",r"$k=\pi/
 ↪a$"],yticks=[],title="Dispersion ein-/ zweiatomige Kette")
ax.set_ylabel(r"$\omega_1$",rotation=0,labelpad=10,size=16)
fig.savefig("dispersion.pdf")
```



[6]:
```python
from matplotlib.cm import ScalarMappable
from matplotlib.colors import LinearSegmentedColormap

M = [4, 2, 1.3, 1]
N = len(M)
cmap = plt.cm.viridis
c = cmap(np.linspace(1, 0, N))
k = np.linspace(-2*np.pi, 2*np.pi, 1000)

omega1 = lambda k,M: np.sqrt(4 *np.sin(k/2)**2)
omega2optical = lambda k,M: np.sqrt((1/M + 1) + np.sqrt(((1/M + 1))**2 - 4/M *
    np.sin(k/2)**2))
omega2acoustic = lambda k,M: np.sqrt((1/M + 1) - np.sqrt(((1/M + 1))**2 - 4/M *
    np.sin(k/2)**2))

fig, ax = plt.subplots()
for i in range(N-1):
    ax.plot(k, omega2acoustic(k,M[i]), c=c[i])
    ax.plot(k, omega2optical(k,M[i]), c=c[i])
ax.plot(k, omega1(k/2, M[-1]), c=c[-1])
```

```python
ax.plot(k, omega1(k/2+np.pi, M[-1]), c=c[-1],linestyle="--")
ax.set(
    ylim=(0, 2.1),
    xlim=(- 2*np.pi, 2*np.pi),
    xticks=[-2*np.pi, -2*np.pi/2, 0, 2*np.pi/2, 2*np.pi],
    xticklabels=[r"$k=-2\pi/a$", r"$-\pi/a$", "0", r"$\pi/a$", r"$k=2\pi/a$"],
    yticks=[],
    title="Übergang zweiatomig $\\to$ einatomig"
)
ax.set_ylabel(r"$\omega$", rotation=0, labelpad=10, size=16)

sm = ScalarMappable(cmap=cmap, norm=plt.Normalize(vmin=min(M), vmax=max(M)))
sm.set_array([])
cbar = fig.colorbar(sm, ax=ax, pad=0.02)
cbar.set_label("$M/m$-Wert")

fig.savefig("transition.pdf")
plt.show()
```



[7]:
```python
# Nr.3 (a)
D_1,D_2,m,M,a,k = symbols("D_1 D_2 m M a k", positive=True)
k = symbols("k")
omega = sqrt((D_1+D_2)/m - 1/m * sqrt(D_1**S(2) + D_2**S(2) +␣
 ↪2*D_1*D_2*cos(k*a)))
f = sqrt(D_1**2 + D_2**2 + 2*D_1*D_2*cos(a*k))
series(f,k,0,3)
```

[7]:
$$\sqrt{D_1^2 + 2D_1D_2 + D_2^2} - \frac{D_1D_2a^2k^2}{2\sqrt{D_1^2 + 2D_1D_2 + D_2^2}} + O\left(k^3\right)$$

[8]:
```python
# Nr.4 (a)
import scipy.constants as c
a = 3.61e-10
v = 4300
omega_max = 2*v/a # THz
print(f"{omega_max*1e-12 = :.3} Hz")

# (b)
E_max = c.hbar*omega_max/c.e# meV
print(f"{E_max*1e3 = :.3} eV")
```

'omega_max*1e-12 = 23.8 Hz'

'E_max*1e3 = 15.7 eV'

[9]:
```python
# Nr.5
lamb = 694e-9
n = 1.54
v = 6000

# (a)
k = n*2*np.pi/lamb
Delta_k = 2*n*2*np.pi/lamb
Delta_p = c.hbar * Delta_k
print(k,Delta_k,Delta_p)

# (b)
omega = v * Delta_k
print(omega)

# (c)
Delta_E_rel = omega / (2*np.pi*c.c/lamb)
print(Delta_E_rel)
```

13942514.9467674

27885029.8935348

$2.94067666599424 \cdot 10^{-27}$

167310179361.209

$6.16426447926185 \cdot 10^{-5}$