

問題

バンド構造 band.csv から、数値微分により $d^2E(k)/dk^2$ を求め、有効質量 m_e^* と k の関係をグラフに描け。

格子定数は $a = 4.0 \text{ \AA}$ とする。

異なる精度の数値微分をし、有効質量の精度の比較をするとbetter。

有効質量

TBバンド

$$E(k) = \varepsilon_1 - 2|h_{12}| \cos(ka) \sim \varepsilon_1 - 2|h_{12}| + |h_{12}|a^2 k^2 + O((ka)^4)$$

自由電子

$$E(k) = E_0 + \frac{|\mathbf{P}|^2}{2m} = E_0 + \frac{\hbar^2}{2m} |\mathbf{k}|^2$$

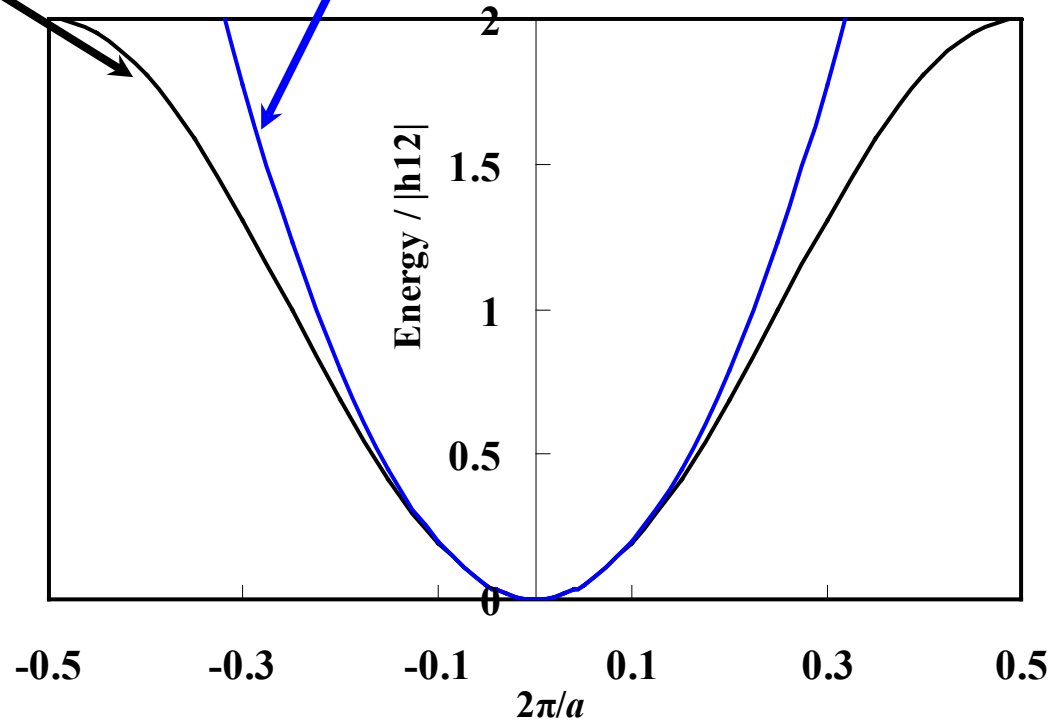
$$\frac{1}{m^*} = \frac{1}{\hbar^2} \frac{\partial^2 E_n(\mathbf{k})}{\partial k^2}$$

$$m^* = \frac{\hbar^2}{2|h_{12}|a^2}$$

大きな混成($|h_{12}|$)により質量 m^* は小さくなる

バンド幅 $W = 4|h_{12}|$

$$m_e^* = 2\hbar^2 / Wa^2$$



有効質量

\mathbf{k} は逆格子の内部座標: 一般に $[-1/2, 1/2]$ の範囲で表示される
単位変換 $k_{\text{real}} = (2\pi/a) k$

$$m^* = \hbar^2 \left(\frac{\partial^2 E_n(\mathbf{k})}{\partial k_{\text{real}}^2} \right)^{-1} = \hbar^2 \left(\frac{2\pi}{a} \right)^2 \left(\frac{\partial^2 E_n(\mathbf{k})}{\partial k^2} \right)^{-1}$$

一般に、有効質量は 電子の静止質量 m_e^0 との比であらわす

$$m^* / m_e^0 = \hbar^2 \left(\frac{\partial^2 E_n(\mathbf{k})}{\partial k_{\text{real}}^2} \right)^{-1} / m_e^0 = \hbar^2 \left(\frac{2\pi}{a} \right)^2 \left(\frac{\partial^2 E_n(\mathbf{k})}{\partial k^2} \right)^{-1} / m_e^0$$

数値計算: 微分

$\frac{df(x)}{dx}$ をコンピュータでどのように計算するか

微分 d を差分 Δ で置き換える

$$\frac{df(x)}{dx} \sim \frac{\Delta f(x)}{\Delta x} = \frac{f(x+h) - f(x)}{(x+h) - x} = \frac{f(x+h) - f(x)}{h}$$

h を小さくすれば精度が上がる \Leftrightarrow 桁落ち誤差

32bit浮動小数点 (~7桁) : 扱う最小数値の 5桁下が限界

64bit浮動小数点 (~16桁) : 扱う最小数値の 14桁下が限界

$$f(x+h) = f(x) + \frac{df(x)}{dx} h + \frac{1}{2} \frac{d^2 f(x)}{dx^2} h^2 + O(h^3)$$

$$\frac{f(x+h) - f(x)}{h} = \frac{df(x)}{dx} + \frac{1}{2} \frac{d^2 f(x)}{dx^2} h + O(h^2)$$

差分誤差

数値微分: 平均を取って精度を上げる

$$\frac{df(x)}{dx} \sim \frac{f(x+h) - f(x)}{h}$$

誤差:
$$\frac{f(x+h) - f(x)}{h} = \frac{df(x)}{dx} + \frac{1}{2} \frac{d^2 f(x)}{dx^2} h + \frac{1}{3!} \frac{d^3 f(x)}{dx^3} h^2 + O(h^4)$$

$$\frac{df(x)}{dx} \sim \left[\frac{f(x+h) - f(x)}{h} + \frac{f(x) - f(x-h)}{h} \right] / 2 = \frac{f(x+h) - f(x-h)}{2h}$$

$$f(x+h) = f(x) + \frac{df(x)}{dx} h + \frac{1}{2} \frac{d^2 f(x)}{dx^2} h^2 + \frac{1}{3!} \frac{d^3 f(x)}{dx^3} h^3 + O(h^4)$$

$$f(x-h) = f(x) - \frac{df(x)}{dx} h + \frac{1}{2} \frac{d^2 f(x)}{dx^2} h^2 - \frac{1}{3!} \frac{d^3 f(x)}{dx^3} h^3 + O(h^4)$$

誤差:
$$\frac{f(x+h) - f(x-h)}{2h} = \frac{df(x)}{dx} + \frac{1}{3!} \frac{d^3 f(x)}{dx^3} h^2 + O(h^4)$$

二階微分

一階微分を前進差分で計算してから二階微分を計算すると・・・

$$\begin{aligned}\frac{d^2x(t)}{dt^2} &= \frac{\frac{dx}{dt}(t + \Delta t) - \frac{dx}{dt}(t)}{\Delta t} \\ &\sim \frac{\frac{x(t + \Delta t) - x(t)}{\Delta t} - \frac{x(t) - x(t - \Delta t)}{\Delta t}}{\Delta t} = \frac{x(t + 2\Delta t) - 2x(t + \Delta t) + x(t)}{\Delta t^2}\end{aligned}$$

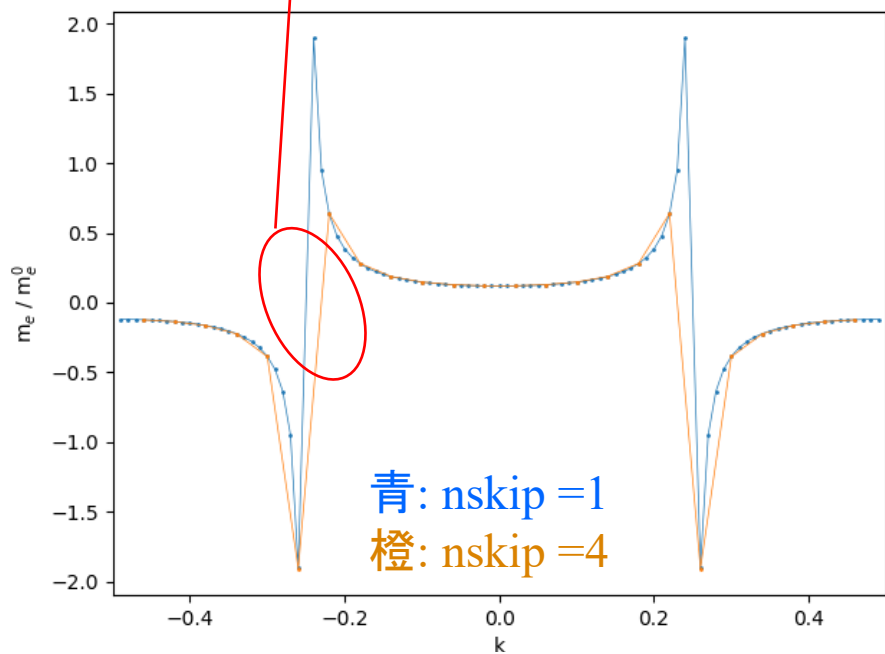
偶数階微分では、結果が $t + \Delta t$ 、 $t - \Delta t$ について対称になる式を取れる

$$\begin{aligned}\frac{d^2x(t)}{dt^2} &\sim \frac{\frac{x(t + \Delta t) - x(t)}{\Delta t} - \frac{x(t) - x(t - \Delta t)}{\Delta t}}{\Delta t} \\ &= \frac{x(t + \Delta t) - 2x(t) + x(t - \Delta t)}{\Delta t^2}\end{aligned}$$

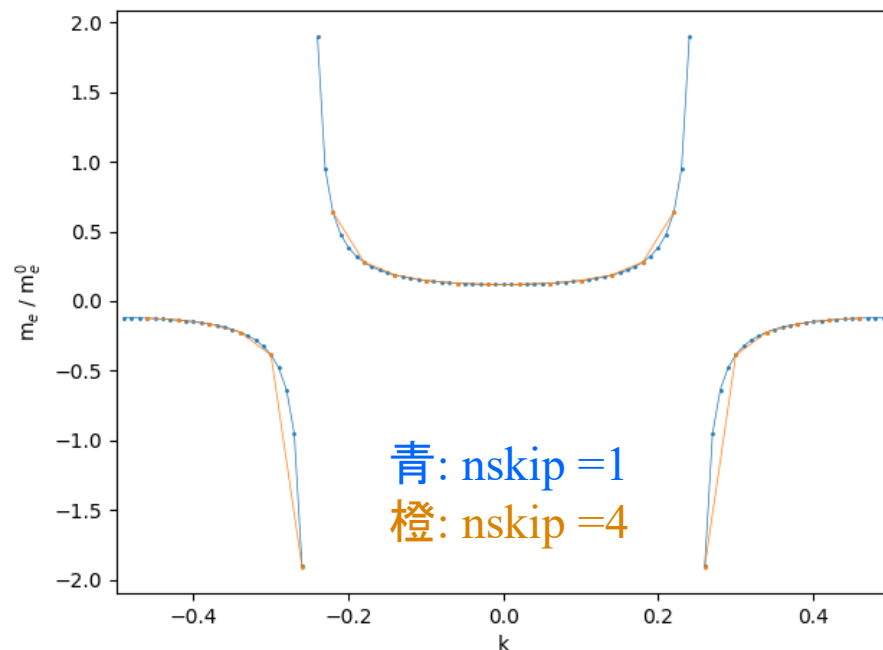
2つの式では、横軸がひとつずれるので注意！

プログラム (抜粋)

描いてはいけない線



データに None (未定義値) を挿入することで描いてはいけない線を消した



プログラム (抜粋)

EffectiveMass.py

```
#=====
# parameters
#=====
#a = 4.0 # A
a = 4.0e-10 #m

infile = 'band.csv'
#有効質量の符号が変わる点をつなぐかどうかのフラグ
cutline = 1

def read_csv(fname):
    x = []
    y = []
    with open(fname) as f:
        fin = csv.reader(f)
        xlabel, ylabel, = next(fin)
        for row in fin:
            try:
                x.append(float(row[0]))
                y.append(float(row[1]))
            except:
                print("Warning: Invalid float data [{}] or [{}]"
                    .format(row[0], row[1]))

    return xlabel, ylabel, x, y

def main():
    xlabel, ylabel, k, E = read_csv(infile)
    #入力データで使う変数を計算
    nk = len(k)
    dk = k[1] - k[0]
```

```
#共通の定数は先に計算
km = hbar * hbar * (pi2 / a)**2.0
#微分の精度を比較するため、h = nskip*dk にする
nskip = 1
xk = []
ymc = []
#符号の変化を検出するため、符号変数を用意
signprev = None
for i in range(nskip, nk - nskip, nskip):
    #2階微分を計算
    d2Edk2c = (E[i+nskip] + E[i-nskip] - 2 * E[i]) * e / pow(nskip *
        dk, 2.0)
    #2回微分はゼロになることがあるので、まずは1/m*を計算
    minv = d2Edk2c / km
    print(i, E[i-1], E[i], E[i+1], minv)
    #1/m*が1/meより非常に小さければ、m*は計算しない
    if abs(minv) <= 1.0e20: # << 1.0/me ~ 1e30
        #符号が反転する場所でグラフの線を切断するときは
        #Noneデータを追加する。
        if cutline:
            xk.append(k[i])
            ymc.append(None)
#反転した符号を記録
    signprev = -signprev
    continue
else:
    m = km / d2Edk2c

if signprev is None:
    signprev = m
elif signprev * m < 0.0:
    if cutline:
```


プログラム (抜粋)

EffectiveMass.py

#共通の定数は先に計算

```
km = hbar * hbar * (pi2 / a)**2.0
```

#微分の精度を比較するため、h = nskip*dk にする

```
nskip = 1
```

```
xk = []
```

```
ymc = []
```

#符号の変化を検出するため、符号変数を用意

```
signprev = None
```

```
for i in range(nskip, nk - nskip, nskip):
```

#2階微分を計算

```
    d2Edk2c = (E[i+nskip] + E[i-nskip] - 2 * E[i]) * e / pow(nskip *  
dk, 2.0)
```

#2回微分はゼロになることがあるので、まずは1/m*を計算

```
    minv = d2Edk2c / km
```

```
    print(i, E[i-1], E[i], E[i+1], minv)
```

#1/m*が1/meより非常に小さければ、m*は計算しない

```
    if abs(minv) <= 1.0e20: # << 1.0/me ~ 1e30
```

#符号が反転する場所でグラフの線を切断するときは

#Noneデータを追加する。

```
        if cutline:
```

```
            xk.append(k[i])
```

```
            ymc.append(None)
```

#反転した符号を記録

```
            signprev = -signprev
```

```
            continue
```

```
        else:
```

```
            m = km / d2Edk2c
```

#符号が反転する場所でグラフの線を切断するときは

#Noneデータを追加する。

```
        if signprev is None:
```

#signprevが初期値 None である場合は符号の最初の値を代入

```
            signprev = m
```

```
        elif signprev * m < 0.0:
```

```
            if cutline:
```

```
                xk.append(k[i])
```

```
                ymc.append(None)
```

#反転した符号を記録

```
                signprev = m
```

```
            xk.append(k[i])
```

```
            ymc.append(m / me)
```

```
    plt.plot(xk, ymc, linewidth = 0.5, marker = 'o', markersize = 1.0,  
label = 'nskip = 1')
```

```
    plt.xlabel(klabel)
```

```
    plt.ylabel("m$_e$ / m$_e^0$")
```

```
    plt.xlim([-0.5, 0.5])
```

```
#    plt.ylim([-0.5, 0.5])
```

```
    plt.tight_layout()
```

```
    plt.pause(0.1)
```

```
    print("Press ENTER to exit>>", end = "")
```

```
    input()
```

```
if __name__ == "__main__":
```

```
    main()
```