Vavilov approximation to Landau

April 12, 2024

[2]:
```python
import numpy as np # NumPy
import matplotlib.pylab as plt # Matplotlib plots
```

1 Calculating the factor the determines if the Vavilov distribution can be approximated by a Landau

Bethe-Bloch mean energy loss:

$$\left\langle -\frac{dE}{dx}\right\rangle \frac{1}{\rho} = Kz^2\frac{1}{\beta^2}\frac{Z}{AM_u}\cdot\left[\frac{1}{2}\ln\left(\frac{2m_ec^2\beta^2W_{max}}{I^2\cdot(1-\beta^2)}\right)-\beta^2\right]$$

$$K = 4\pi N_A r_e^2 m_e c^2/M_u = 0.307075\text{MeV g}^{-1}\text{cm}^2$$

$$r_e = \frac{e^2}{4\pi\varepsilon_0 m_e c^2}$$

$$M_u = 1\frac{g}{mol}$$

from https://nap.nationalacademies.org/read/20066/chapter/10#188

The parameter k of the Vavilov distribution determines if it becomes close to a Landau ($k \to 0$) or a Gaussian ($k >> 1$).

To calculate k ($W_{max} \equiv \epsilon_{max}$):

$$k = 0.30058\frac{m_ec^2}{\beta^2}\frac{Z}{A}\frac{s}{\epsilon_{max}}$$

where $s = \Delta x\cdot\rho$ and

$$\epsilon_{max} = \frac{2m_ec^2\beta^2}{1-\beta^2}\left[1+\frac{2m}{M}\frac{1}{\sqrt{1-\beta^2}}+\left(\frac{m}{M}\right)^2\right]^{-1}$$

But when $\beta \to 1$ then also the maximum energy:

$$\epsilon_{max} \to \frac{Mc^2\beta^2}{1-\beta^2}$$

1.0.1 Calculating β for a proton of energy E

$$E^2 = (pc)^2 + (m_0c^2)^2 = m_0^2\gamma^2c^2 + m_o^2c^4$$

$$\frac{E^2}{m_0^2c^4} = \frac{\beta^2}{1-\beta^2} + 1$$

$$\beta^2 = 1 - \left(\frac{m_0c^2}{E}\right)^2$$

for a proton of energy $E = 120\text{GeV}$, $(m_0 = 938.272\text{MeV}/\text{c}^2)$ $\beta^2 = 0.99993886$

1.0.2 Applying this to a thin silicon layer

[25]:
```
# E = 120 GeV = 120e3 MeV
Energy = 120e3 # MeV
m_proton = 938.272088 # MeV/c^2
m_electron = 0.51099895 # MeV/c^2
# 50 um = 5e-3 cm
delta_x = 5e-3 # cm
### silicon
Z = 14   # atomic number
A = 28.085  # atomic weight
density = 2.329085 # g/cm^3


beta_2 = 1 - (m_proton/(Energy))**2

epsilon_max = (m_proton * beta_2) / (1 - beta_2) # MeV

s = delta_x * density

k = 0.30058 * m_electron / beta_2 * Z / A * s / epsilon_max # g/cm^2
```

[26]:
```
print("k=",k)
```

```
k= 5.8104323137633683e-11
```