exercises_2_solutions

February 2, 2021

[12]:
```
import math as m
from itertools import product
```

0.1 Energy resolution - Sampling term

First we extract relevant quantities from the table

[2]:
```
cm = 1.
mm = 0.1*cm
lead_dEdX = 12.8
sci_dEdX = 2.03
lead_thickness = 5*mm
sci_thickness = 3*mm
```

0.1.1 Sampling fraction

The sampling fraction is defined as the fraction of energy deposited by mips in the active material. In our case:

$$f_s = \frac{d_{sci} \times (dE/dx)_{sci}}{d_{sci} \times (dE/dx)_{sci} + d_{pb} \times (dE/dx)_{pb}}$$

[4]:
```
f_sampling = sci_thickness*sci_dEdX / (sci_thickness*sci_dEdX +
↪lead_thickness*lead_dEdX)
print("Sampling fraction =", f_sampling)
```

```
Sampling fraction = 0.0868882864888001
```

$$f_s = 0.087$$

0.1.2 New sampling fraction required for a 13% sampling term

The sampling term of the energy resolution can be expressed as:

$$\frac{\sigma(E)}{E} = \alpha\sqrt{\frac{d}{f_s}} \times \frac{1}{\sqrt{E}}$$

Where α is a multiplicative coefficient and d is the active material thickness.

We have therefore in the baseline calorimeter:

$$\alpha\sqrt{\frac{d}{f_s}} = 16\ \%$$

While we want to compute $f_{s,new}$, which provides a 13% sampling term:

$$\alpha\sqrt{\frac{d}{f_{s,new}}} = 13\ \%$$

Dividing the two equalities we get:

$$f_{s,new} = \left(\frac{16\ \%}{13\ \%}\right)^2 \times f_s$$

[5]:
```
f_sampling_new = f_sampling * (16./13.)**2
print("New sampling fraction =", f_sampling_new)
```

```
New sampling fraction = 0.13161775941498716
```

$$f_{s,new} = 0.13$$

0.1.3 New lead plates thickness

Using the sampling fraction formula given above we derive:

$$d_{pb,new} = \frac{1 - f_{s,new}}{f_{s,new}} \times \frac{(dE/dx)_{sci}}{(dE/dx)_{pb}} \times d_{sci}$$

[9]:
```
lead_thickness_new = (1-f_sampling_new)/f_sampling_new * sci_thickness *␣
 ↪sci_dEdX/lead_dEdX
print("New lead thickness =", lead_thickness_new, 'cm')
```

```
New lead thickness = 0.3139089965820312 cm
```

$$d_{pb,new} = 3\ \text{mm}$$

0.2 Energy resolution - Comparison of two EM calorimeters

First we define functions to compute each terms as well as the total resolution. For information, the symbol $\oplus$ used in the resolution formulas is the quadratic sum:

$$a \oplus b = \sqrt{a^2 + b^2}$$

[13]:
```
cms = 0
atlas = 1
detector_names = {0:'CMS', 1:'ATLAS'}
GeV = 1.
TeV = 1e3*GeV

def stochastic(E, det):
    if det==cms:
        return 0.03/m.sqrt(E)
    elif det==atlas:
        return 0.1/m.sqrt(E)
    else:
        return 0.

def noise(E, det):
    if det==cms:
        return 0.3/E
    elif det==atlas:
        return 0.3/E
    else:
        return 0.

def constant(det):
    if det==cms:
        return 0.005
    elif det==atlas:
        return 0.007
    else:
        return 0.

def resolution(E, det):
    sigma_square = stochastic(E,det)**2 + noise(E,det)**2 + constant(det)**2
    return m.sqrt(sigma_square)
```

Then we just loop on the different energies and detectors

[16]:
```
energies = [10*GeV, 1*TeV]
detectors = [cms, atlas]
for energy,detector in product(energies, detectors):
    s = stochastic(energy, detector)
    n = noise(energy, detector)
```

```
    c = constant(detector)
    sigma = resolution(energy, detector)
    print(energy, 'GeV in ', detector_names[detector] )
    print('   Stochastic =', s*100., '%')
    print('   Noise =', n*100., '%')
    print('   Constant =', c*100., '%')
    print('   Total =', sigma*100., '%')
```

```
10.0 GeV in  CMS
   Stochastic = 0.9486832980505138 %
   Noise = 3.0 %
   Constant = 0.5 %
   Total = 3.1859064644147983 %
10.0 GeV in  ATLAS
   Stochastic = 3.162277660168379 %
   Noise = 3.0 %
   Constant = 0.7000000000000001 %
   Total = 4.414748010928823 %
1000.0 GeV in  CMS
   Stochastic = 0.09486832980505139 %
   Noise = 0.03 %
   Constant = 0.5 %
   Total = 0.5098038838612354 %
1000.0 GeV in  ATLAS
   Stochastic = 0.31622776601683794 %
   Noise = 0.03 %
   Constant = 0.7000000000000001 %
   Total = 0.7687002016391046 %
```

CMS	10 GeV	1 TeV
Stochastic [%]	0.95	0.095
Noise [%]	3	0.03
Constant [%]	0.5	0.5
$\sigma(E)/E$ [%]	3.2	0.51

ATLAS	10 GeV	1 TeV
Stochastic [%]	3.2	0.32
Noise [%]	3	0.03
Constant [%]	0.7	0.7
$\sigma(E)/E$ [%]	4.4	0.77

A few comments:

- At low energies, noise dominates in CMS while the stochastic and noise terms compete in

ATLAS

- The constant term dominates all other contributions at high energy
- CMS has always a better energy resolution than ATLAS. At low energy it is due to its better stochastic term, which comes from the fact that it is an homogeneous calorimeter while ATLAS ECAL is a sampling calorimeter. But keep in mind that these are test beam results; in real conditions, the resolution is never as good.

[ ]:
```

```