# The Geometry of Scintillator Bars

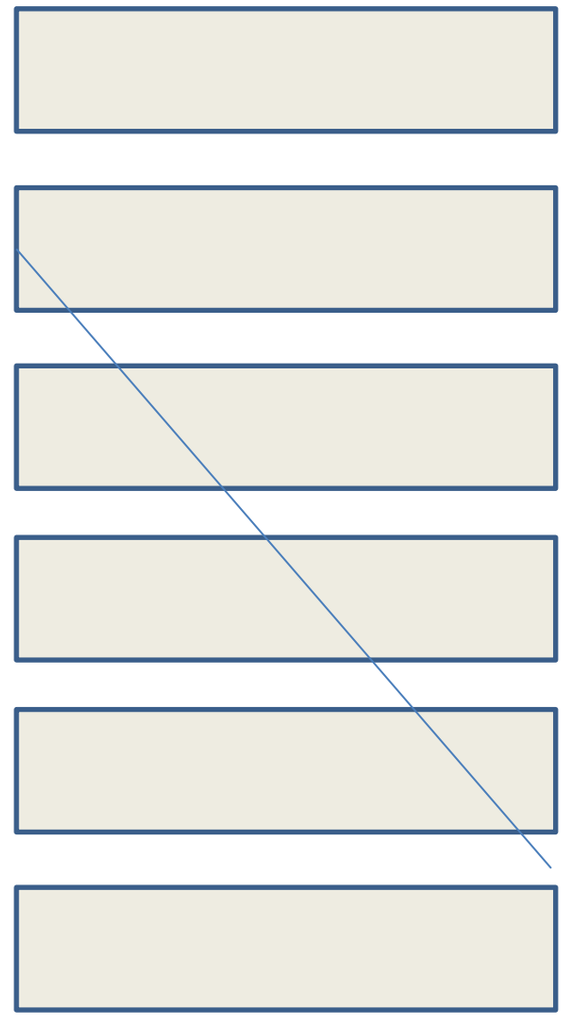
2017/12/01

Kyeong Ro Lee

# Number of Bars a Cosmic Ray Pass Through

- Python code nofbars.py to calculate number of bars a cosmic ray pass through, when the path is given -> successful

- Depends on path (theta, phi, position) and dimension of bars



# nofbars.py

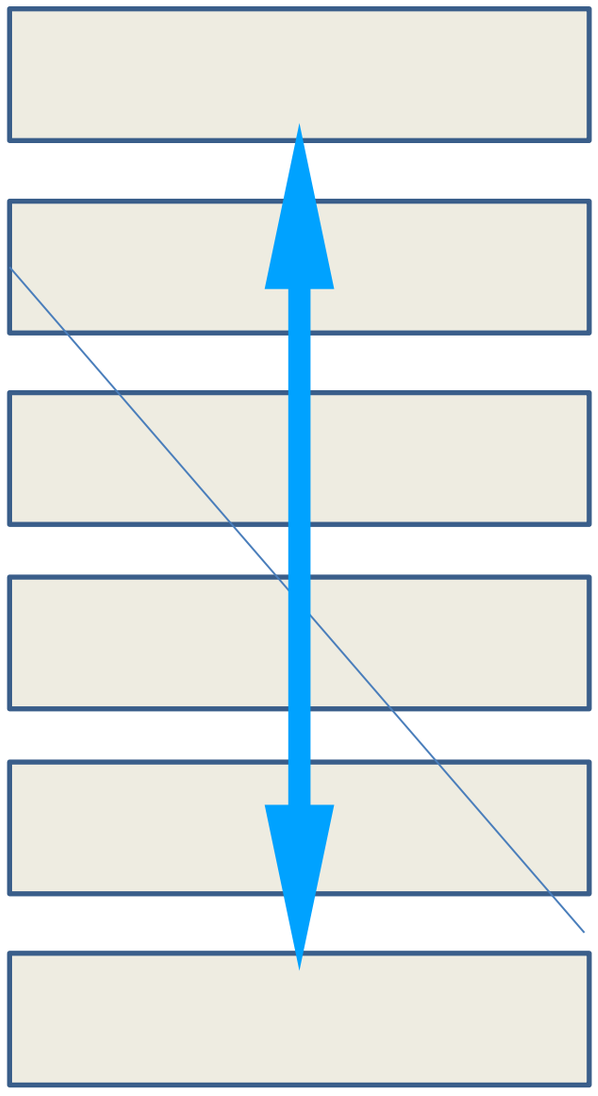
kyeongro@KR-ThinkPad13: ~/CRrate

```
import math

# width means the width of incident plane of cosmic ray
def width(phi, a, b):
    if 0<=phi<=math.acos(b/math.sqrt(a**2+b**2)):
        w=b/math.cos(phi)
    elif math.acos(b/sqrt(a**2+b**2)):
        w=a/math.sin(phi)
    return w

# nofbars is a fuction to calculate the number of bars a cosmic ray passes
# when its trajectory is given
# the number of bars is saved as variable 'm'
def nofbars(theta, phi, x0, a, b, c, h):
    m=0
    for n in range(1,13,1):
        if 0<=math.tan(theta)*(((n-1)*c+(n-1)*h)-x0)<=width(phi, a, b) or 0<=math.tan(theta)*((n*c+(n-1)*h)-x0)<=wid
th(phi, a, b):
            m+=1
        else:
            continue
    return m
~
~
~
~
~
~
~
"nofbars.py" 21L, 671C
```

# Probability Function

- Implemented the probability function of theta and phi, for each case of number (1, 2, 3, ...) -> probability.py -> successful!

- Not an exact solution, but the precision can easily be adjusted to enough standard



# probability.py

kyeongro@KR-ThinkPad13: ~/CRrate

```
15
16 d0=0
17 d1=0
18 d2=0
19 d3=0
20 d4=0
21 d5=0
22 d6=0
23 d7=0
24 d8=0
25 d9=0
26 d10=0
27 d11=0
28 d12=0
29
30 d=0.001 # adjustable increment
31
32 x0=-width(phi, a, b)/math.tan(theta)
33 # error occurs when theta==0
34
35 while x0<=12*c+11*h:
36     x0+=d
37     m=nofbars(theta, phi, x0, a, b, c, h)
38     if m==1:
39         d1+=d
40     elif m==2:
41         d2+=d
42     elif m==3:
43         d3+=d
44     elif m==4:
45         d4+=d
46     elif m==5:
47         d5+=d
48     elif m==6:
49         d6+=d
50     elif m==7:
51         d7+=d
52     elif m==8:
53         d8+=d
54     elif m==9:
                                                    54,5          17%
```

# Remaining Steps

- Implementing theta-dependent (i.e. phi-independent) distribution (probably discrete) to the rate calculation.

- Comparison with data