Homework_2_Problem_2

April 21, 2019

In [9]:
```
import numpy as np
import matplotlib.pyplot as plt

from scipy.stats import linregress
```

In [89]:
```
a = np.pi / 360 * np.array([42.2, 49.2, 72.0, 87.3])
b = np.pi / 360 * np.array([28.8, 41.0, 50.8, 59.6])
c = np.pi / 360 * np.array([42.8, 73.2, 89.0, 115.0])

degrees = [a, b, c]
```

$d_{h,k,l} = \frac{\lambda}{2\sin\theta} = \frac{a}{\sqrt{h^2+k^2+l^2}}$
BCC $h+k+l$ is even
FCC h,k,l ara either odd or even
Diamond...

In [90]:
```
def get_all_ratios(arr):
    arr = np.array(arr)
    perms = sum(list(get_all_permutations(arr.shape[0])), [])
    #print(perms)
    for perm in perms:
        yield (arr[perm[0]] / arr[perm[1]])
```

In [91]:
```
def get_all_permutations(l):
    for i in range(l):
        yield [(i, j + i + 1) for j in range(l - i - 1)]
```

In [92]:
```
ratios = [list(get_all_ratios(np.sin(deg) ** 2)) for deg in degrees]
```

In [93]:
```
ratios
```

Out[93]:
```
[[0.7478673189962223,
  0.37511111214491555,
  0.27200877150053004,
  0.5015744138256834,
  0.36371260595477906,
  0.7251418651533994],
 [0.5042729351944293,
```

```
 0.3361499185992957,
 0.25040845096642766,
 0.6666031332212755,
 0.4965732512887675,
 0.7449308689701771],
[0.3745176427761104,
 0.2709997329815122,
 0.1871690689330239,
 0.7235967068806901,
 0.4997603518638908,
 0.6906614514848718]]
```

```
In [108]: fcc = np.array([3, 4, 8, 11])
          bcc = np.array([2, 4, 6, 8])
          diamond = np.array([3, 8, 11, 16])

In [109]: list(get_all_ratios(fcc))

Out[109]: [0.75, 0.375, 0.2727272727272727, 0.5, 0.36363636363636365, 0.7272727272727273]

In [110]: list(get_all_ratios(bcc))

Out[110]: [0.5, 0.3333333333333333, 0.25, 0.6666666666666666, 0.5, 0.75]

In [111]: list(get_all_ratios(diamond))

Out[111]: [0.375, 0.2727272727272727, 0.1875, 0.7272727272727273, 0.5, 0.6875]
```

Comparing first elements, we conclude that :
$A = FCC$
$B = BCC$
$C = Diamond$

```
In [100]: def lattice_constant(lamb, n, theta):
              return lamb / 2 * np.sqrt(n) / np.sin(theta)

          lamb = 1.5 * 10 ** -10

In [105]: a_a = lattice_constant(lamb, fcc, a)
          a_a

Out[105]: array([3.60847118e-10, 3.60333705e-10, 3.60900573e-10, 3.60371477e-10])

In [104]: np.mean(a_a), np.var(a_a) ** .5

Out[104]: (3.606132181626314e-10, 2.616525346820743e-13)
```

For A:
$a = (3.606 \pm 0.006)A$

```
In [106]: a_b = lattice_constant(lamb, bcc, b)
          a_b
```

```
Out[106]: array([4.26499117e-10, 4.28317643e-10, 4.28297233e-10, 4.26847382e-10])
```

```
In [107]: np.mean(a_b), np.var(a_b) ** .5
```

```
Out[107]: (4.2749034361941377e-10, 8.263510549588644e-13)
```

For B:
$a = (4.27 \pm 0.02)A$

```
In [115]: a_c = lattice_constant(lamb, diamond, c)
          a_c
```

```
Out[115]: array([3.56021035e-10, 3.55791989e-10, 3.54891670e-10, 3.55706714e-10])
```

```
In [116]: np.mean(a_c), np.var(a_c) ** .5
```

```
Out[116]: (3.55602852234001e-10, 4.26385277907221e-13)
```

For C:
$a = (3.56 \pm 0.01)A$

```
In [117]: def reverse_rel(lamb, n, a):
              return 2 * np.arcsin(lamb / 2 * np.sqrt(n) / a)
```

```
In [120]: angles = reverse_rel(lamb, fcc, a_c)
          angles * 180 / np.pi
```

```
Out[120]: array([42.8       , 49.8706111 , 73.41604843, 88.74225978])
```

3) Diffraction will occur at angles:

$42.8, 49.87, 73.42, 88.74$
between $\vec{k}$ and $\vec{k'}$

```
In [ ]:
```