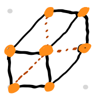
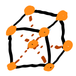
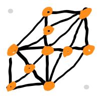
Sergio Montoya Ramirez
Codigo: 202112171

## Punto 1A)

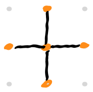
SC) BCC) FCC)

## Punto 1B)

SC)

Dibujemos en corte un atomo



Ademas tiene 2 Atomos vecinos (uno delante y uno atras)

BCC)

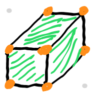
Aparte de los 6 vecinos que ya sabemos que tiene BCC tiene otros 8 vecinos correspondientes a los "centros" de los cubos en donde es una arista

FCC)

Aparte de los 6 que ya conocemos de SC tiene 4 vecinos por eje (Por cada cara) es decir tiene 16 vecinos

## Punto 1C)

SC)



Cada celda convencional tiene $\frac{1}{8}$ por atomo en su esquina y 8 esquinas, Por lo tanto

$8 \cdot \frac{1}{8} = 1$ atomo.

Entonces la celda Primitiva y convencional son Iguales

BCC)

Ademas del atomo que ya sabemos que tiene esta celda tiene 1 atomo extra. teniendo entonces 2 atomos y haciendo que la celda primitiva sea la mitad $\left(\frac{1}{2}\right)$ de la convencional

FCC)

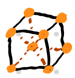
Ademas del atomo de SC hay $\frac{1}{2}$ de atomo por cara (de las cuales hay 6) y por tanto hay 4 atomos y la primitiva es $\frac{1}{4}$ de la convencional

## Punto 1D)

SC)

Siendo un cubo la distancia es el lado (a)

BCC)



tomando distancia euclidia la diagonal mide

$$2d = \sqrt{a^2+a^2+a^2}$$
$$= \sqrt{3a^2}$$
$$= a\sqrt{3}$$

Por lo que al centro seria

$$d = \frac{a\sqrt{3}}{2}$$

FCC)

tomando un lado

Entonces

$$d = \sqrt{\frac{a^2}{4}+\frac{a^2}{4}}$$
$$= \sqrt{\frac{2a^2}{4}}$$
$$= \frac{a\sqrt{2}}{2}$$

## Punto 1e)

$$FE = \frac{N \times V_a}{V_c} \quad \text{con } V_c = a^3$$

SC)

$N = 2$

$V_a$: cada radio ocupa medio lado por tanto $r = \frac{a}{2}$

$$FE = \frac{\frac{4}{3}\pi\frac{a^3}{2^3}}{a^3} = \frac{\pi}{6}$$

BCC)

$N = 2$

$V_a$: la diagonal tiene 4 radios por tanto $r = \frac{a\sqrt{3}}{4}$

$$FE = 2\,\frac{\frac{4}{3}\pi\frac{a^3}{4^3}3^{1.5}}{a^3} = \frac{\sqrt{3}}{8}\pi$$

FCC)

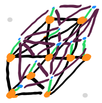
$N = 4$

$V_a$ = de nuevo la mitad del diamdro seria $r = \frac{a\sqrt{2}}{4}$

$$FE = 4\,\frac{\frac{4}{3}\pi\frac{a^3 2^{1.5}}{4^3}}{a^3} = \frac{\sqrt{2}}{6}\pi$$

Punto 2 A)

A)



B)

ticne el doble de Vecinos que en FCC Por lo qur Seria 36

C)

De nuevo se duplica el numero de atomos por lo que seria 8 Y tendria $\frac{1}{8}$ la celda primitiva

D)

tenemos una diagonal que al tener un atomo movido genera una distancia

$$d = \sqrt{\frac{a^2}{4^2} + \frac{a^2}{4^2} + \frac{a^2}{4^2}}$$

$$= \frac{\sqrt{3}\,a}{4}$$

e)

$$r = \frac{\sqrt{3}\,a}{8}$$

$$FE = \frac{8 \cdot \frac{4}{3}\pi\left(\frac{\sqrt{3}\,a}{8}\right)^3}{a^3} = \frac{\sqrt{3}\,\pi}{16}$$

Punto 2 B)

Sabemos que para un a

$$d = \frac{\sqrt{3}\,a}{4}$$

Como podemos saber cuanto es a?

Pues al tener una celda unitaria de Volumen $a^3$,

Este volumen seria

$$a^3 = \frac{\frac{N \times M_a}{N_A}}{\rho} = \frac{N \times M_a}{\rho \times N_a}$$

$\frac{M_a}{N_a}$ : Es la masa atonica dividida en el numero de Avogadro lo que Implica que esta es la masa de una sola molecula

$\rho$ : Es la densidad. Es decir cuanto Volumen ocupa una cantidad de masa

Para simplificar este proceso hice un pequeño codigo

```python
from scipy.constants import Avogadro
from math import cbrt, sqrt

class Element():
    def __init__(self, n: int = 8, m: int | float = 12, density: float = 3.515):
        self.n = n
        self.m = m
        self.d = density

    def side(self) -> float:
        return cbrt((self.n * self.m)/(self.d * Avogadro))

    def distance(self) -> float:
        return (sqrt(3)/4) * self.side()

if __name__=="__main__":
    carbono = Element()
    silicio = Element(m = 28, density=2.329)
    germanio = Element(m = 72.63, density=5.323)
    print(f"Carbono: {carbono.distance()}")
    print(f"Silicio: {silicio.distance()}")
    print(f"Germanio: {germanio.distance()}")
```

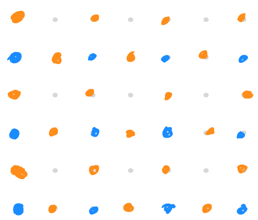
Al Ejecutar Este codigo obtenemos

Carbono: $1.54 \times 10^{-8}$

Silicio: $2.34 \times 10^{-8}$

Germanio: $2.45 \times 10^{-8}$

# Punto 3 A)



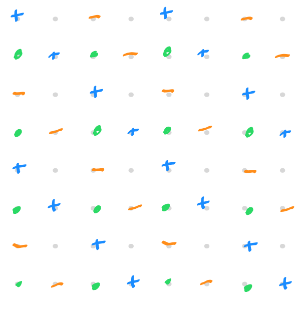
la base

las Flechas naranjas son los Vectores

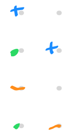
Los puntos encerrados en cuadros Blancos son los puntos de Red de Bravais

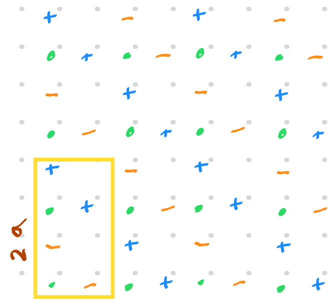
El cuadro verde es la celda unitaria

# Punto 3 B)



Como estamos con un sistema que se repite (Alternando) nuestra base seria





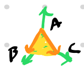

La celda es el cuadro naranja y tiene constante 2a,

Como aclaracion este no es solo un cuadro sino una celda 3D que lo contiene todo

# Punto 4A)

Para hkl el plano cruza en

$A = \frac{a_1}{h}$, $B = \frac{a_2}{k}$, $C = \frac{a_3}{l}$




Construimos

$AB = B - A = \frac{a_2}{k} - \frac{a_1}{h}$

$AC = C - A = \frac{a_3}{l} - \frac{a_1}{h}$

Debemos mostrar que $K \cdot AB = 0$ y $K \cdot AC = 0$

Nota: Recuerde que $b_i \cdot a_j = 2\pi \delta_{ij}$

$K \cdot AB = (hb_1 + kb_2 + lb_3) \cdot \left(\frac{a_2}{k} - \frac{a_1}{h}\right)$

$= \frac{h}{k} b_1 a_2 - \frac{\cancel{h}}{\cancel{h}} b_1 a_1 + \frac{\cancel{k}}{\cancel{k}} b_2 a_2 - \frac{k}{h} b_2 a_1 + \frac{l}{k} b_3 a_2 - \frac{l}{h} b_3 a_1$

$= 0 - 2\pi + 2\pi - 0 + 0 - 0 = 0$

$K \cdot AC = (hb_1 + kb_2 + lb_3) \cdot \left(\frac{a_3}{l} - \frac{a_1}{h}\right)$

$= \frac{h}{l} b_1 a_3 - \frac{\cancel{h}}{\cancel{h}} b_1 a_1 + \frac{k}{l} b_2 a_3 - \frac{k}{h} b_2 a_1 + \frac{\cancel{l}}{\cancel{l}} b_3 a_3 - \frac{l}{h} b_3 a_1$

$= 0 - 2\pi + 0 - 0 + 2\pi - 0 = 0$

# Punto 4 B)

tome un plano paralelo adyacente. La distancia es $d(hkl)$.
El vector $r$ del plano entonces seria $r = d(hkl) \frac{K}{|K|}$

$K \cdot \left(d(hkl) \frac{K}{|K|}\right) = C$

$\frac{d(hkl)\, |K|^{\cancel{2}}}{\cancel{|K|}} = C$

$d(hkl)\,|K| = C$

Ahora para encontrar C tomamos que para que un punto este en el plano cumple

$K \cdot \left(\frac{a_1}{h}\right) = C$

$(hb_1 + kb_2 + lb_3)\left(\frac{a_1}{h}\right) = C$

$\frac{\cancel{h}}{\cancel{h}} b_1 a_1 + \frac{k}{h} b_2 a_1 + \frac{l}{h} b_3 a_1 = C$

$2\pi + 0 + 0 = C = 2\pi$

$d(hkl)\,|K| = 2\pi$

$d(hkl) = \frac{2\pi}{|K|}$

## Punto 4C)

Para una red cubica simple

$a_1 = a\hat{x}, \; a_2 = a\hat{y}, \; a_3 = a\hat{z}$

$b_1 = \frac{2\pi}{a}\hat{x}, \; b_2 = \frac{2\pi}{a}\hat{y}, \; b_3 = \frac{2\pi}{a}\hat{z}$

$$K = h\left(\frac{2\pi}{a}\hat{x}\right) + K\left(\frac{2\pi}{a}\hat{y}\right) + L\left(\frac{2\pi}{a}\hat{z}\right)$$

$$= \frac{2\pi}{a}(h\hat{x} + K\hat{y} + L\hat{z})$$

$$|K| = \left|\frac{2\pi}{a}(h\hat{x} + K\hat{y} + L\hat{z})\right|$$

$$= \sqrt{\left(\frac{2\pi}{a}\right)^2 (h^2 + K^2 + L^2)}$$

$$= \frac{2\pi}{a}\sqrt{h^2 + K^2 + L^2}$$

$$d(hKL) = \frac{2\pi}{|K|}$$

$$= \frac{2\pi}{\frac{2\pi}{a}\sqrt{h^2 + K^2 + L^2}}$$

$$= \frac{\cancel{2\pi} \cdot a}{\cancel{2\pi}\sqrt{h^2 + K^2 + L^2}}$$

$$d(hKL)^2 = \frac{a^2}{h^2 + K^2 + L^2}$$

## Punto 4D)

$$|K|^2 = K \cdot K$$

$$= (hb_1 + Kb_2 + Lb_3)(hb_1 + Kb_2 + Lb_3)$$

$$= h^2(b_1 \cdot b_1) + hK(b_1 \cdot b_2) + hL(b_1 \cdot b_3) + Kh(b_2 \cdot b_1) + K^2(b_2 \cdot b_2) + KL(b_2 \cdot b_3) + Lh(b_3 \cdot b_1) + LK(b_3 \cdot b_2) + L^2(b_3 \cdot b_3)$$

$$= h^2(b_1 \cdot b_1) + K^2(b_2 \cdot b_2) + L^2(b_3 \cdot b_3) + 2hK(b_1 \cdot b_2) + 2hl(b_1 \cdot b_3) + 2Kl(b_2 \cdot b_3)$$

Recordando

$$b_1 = 2\pi \frac{a_2 \times a_3}{a_1 \cdot (a_2 \times a_3)}$$

$$b_2 = 2\pi \frac{a_3 \times a_1}{a_1 \cdot (a_2 \times a_3)}$$

$$b_3 = 2\pi \frac{a_1 \times a_2}{a_1 \cdot (a_2 \times a_3)}$$

Entonces

$$b_x \cdot b_y = \left(2\pi \frac{a_{x+1} \times a_{x-1}}{a_1 \cdot (a_2 \times a_3)}\right)\left(2\pi \frac{a_{y+1} \times a_{y-1}}{a_1 \cdot (a_2 \times a_3)}\right)$$

Nota: Como simplificacion use $x \pm 1$ y $y \pm 1$ esto seria modulo 4 y en caso de ser 0 sube a 1. Es solo una manera de no tener que escribir punto a punto cada caso.

$$= \frac{(2\pi)^2}{(a_1 \cdot (a_2 \times a_3))^2}(a_{x+1} \times a_{x-1})(a_{y+1} \times a_{y-1})$$

$$= \frac{(2\pi)^2}{(a_1 \cdot (a_2 \times a_3))^2}(a_{x+1} \cdot a_{y+1})(a_{x-1} \cdot a_{y-1}) - (a_{x+1} \cdot a_{y-1})(a_{x-1} \cdot a_{y+1})$$

$$= \frac{(2\pi)^2}{(a_1 \cdot (a_2 \times a_3))^2}(C_{x+1} C_{y+1} \cos\theta_{++})(C_{x-1} C_{y-1} \cos\theta_{--}) - (C_{x+1} C_{y-1} \cos\theta_{+-})(C_{x-1} C_{y+1} \cos\theta_{-+})$$

Ahora

$$b_1 \cdot b_1 = \frac{4\pi^2}{(a_1 \cdot (a_2 \times a_3))^2} b^2 c^2 \sin\alpha$$

$$b_2 \cdot b_2 = \frac{4\pi^2}{(a_1 \cdot (a_2 \times a_3))^2} c^2 a^2 \sin\beta$$

$$b_3 \cdot b_3 = \frac{4\pi^2}{(a_1 \cdot (a_2 \times a_3))^2} a^2 b^2 \sin\gamma$$

$$b_1 \cdot b_2 = \frac{4\pi^2}{(a_1 (a_2 \times a_3))^2} abc^2(\cos\alpha\cos\beta - \cos\gamma)$$

$$b_1 \cdot b_3 = \frac{4\pi^2}{(a_1 (a_2 \times a_3))^2} a^2b^2c(\cos\gamma\cos\alpha - \cos\beta)$$

$$b_2 \cdot b_3 = \frac{4\pi^2}{(a_1 (a_2 \times a_3))^2} a^2bc^2(\cos\beta\cos\gamma - \cos\alpha)$$

Reemplazando queda

$$= h^2 \frac{4\pi^2}{(a_1 \cdot (a_2 \times a_3))^2} b^2c^2 \sin\alpha + K^2 \frac{4\pi^2}{(a_1 \cdot (a_2 \times a_3))^2} c^2a^2 \sin\beta + L^2 \frac{4\pi^2}{(a_1 \cdot (a_2 \times a_3))^2} a^2b^2 \sin\gamma +$$

$$2hK \frac{4\pi^2}{(a_1 (a_2 \times a_3))^2} abc^2(\cos\alpha\cos\beta - \cos\gamma) + 2hl \frac{4\pi^2}{(a_1 (a_2 \times a_3))^2} a^2b^2c(\cos\gamma\cos\alpha - \cos\beta) + 2Kl \frac{4\pi^2}{(a_1 (a_2 \times a_3))^2} a^2bc^2(\cos\beta\cos\gamma - \cos\alpha)$$

## Punto 5A)

Iniciamos con

$n\lambda = 2d \sin\theta$

con

$$d = \frac{a}{\sqrt{h^2+k^2+l^2}}$$

$$n\lambda = 2\left(\frac{a}{\sqrt{h^2+k^2+l^2}}\right)\sin\theta$$

$$\sin^2\theta = \underbrace{\left(\frac{n\lambda}{2a}\right)^2}_{\text{Constante para } \theta_i}(h^2+k^2+l^2)$$

$$\frac{\sin^2\theta_1}{\sin^2\theta_2} = \frac{(h_1^2+k_1^2+l_1^2)}{(h_2^2+k_2^2+l_2^2}$$

hice un codigo para calcularlo

```python
from typing import Generator, Tuple
import pandas as pd
import numpy as np
import os

os.system('clear')

def succesion() -> Generator[Tuple[int, int, int], None, None]:
    a = 1
    while True:
        for b in range(a + 1):
            for c in range(b + 1):
                yield (a, b, c)
        a += 1

def index(value):
    def helper(h: int, k: int, l: int):
        return np.pow(h, 2) + np.pow(k, 2) + np.pow(l, 2)
    last = (0, 0, 0)
    for h, k, l in succesion():
        val = helper(h, k, l)
        if val == value:
            return h, k, l
        if val > value:
            return last
        last = (h, k, l)

data = pd.read_csv("./data_5.csv")
data["theta"] = np.radians(data["2theta"] / 2)
data["sin"] = np.sin(data["theta"])
data["sin^2"] = np.pow(data["theta"], 2)
data["s/s"] = (data["sin^2"] / data["sin^2"][0]).round().astype(int)
data.loc[6, "s/s"] = 8
data.loc[9, "s/s"] = 12
data["hkl"] = data["s/s"].apply(index)
print(data)
```

No es el mas lindo pero devuelve

| | 2theta | P | Area[arb] | theta | sin | sin^2 | s/s | hkl |
|---|---|---|---|---|---|---|---|---|
| 0 | 27.3 | 33.50 | 116 | 0.238237 | 0.235990 | 0.056757 | 1 | (1, 0, 0) |
| 1 | 31.7 | 24.00 | 1260 | 0.276635 | 0.273120 | 0.076527 | 1 | (1, 0, 0) |
| 2 | 45.5 | 10.90 | 694 | 0.397062 | 0.386711 | 0.157659 | 3 | (1, 1, 1) |
| 3 | 53.9 | 7.40 | 23 | 0.470366 | 0.453213 | 0.221244 | 4 | (2, 0, 0) |
| 4 | 56.5 | 6.60 | 200 | 0.493056 | 0.473320 | 0.243104 | 4 | (2, 0, 0) |
| 5 | 66.3 | 4.70 | 92 | 0.578577 | 0.546833 | 0.334751 | 6 | (2, 1, 1) |
| 6 | 73.2 | 3.80 | 13 | 0.638791 | 0.596225 | 0.408053 | 8 | (2, 2, 0) |
| 7 | 75.4 | 3.60 | 198 | 0.657989 | 0.611527 | 0.432950 | 8 | (2, 2, 0) |
| 8 | 84.1 | 3.05 | 136 | 0.733911 | 0.669779 | 0.538625 | 9 | (2, 2, 1) |
| 9 | 90.6 | 2.80 | 10 | 0.790634 | 0.710799 | 0.625102 | 12 | (2, 2, 2) |

Punto 5 B)

Usando el Codigo anterior cree otra funcion para conta permutaciones

| | 2theta | P | Area[arb] | theta | sin | sin^2 | s/s | hkl | m |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 27.3 | 33.50 | 116 | 0.238237 | 0.235990 | 0.056757 | 1 | (1, 0, 0) | 6 |
| 1 | 31.7 | 24.00 | 1260 | 0.276635 | 0.273120 | 0.076527 | 1 | (1, 0, 0) | 6 |
| 2 | 45.5 | 10.90 | 694 | 0.397062 | 0.386711 | 0.157659 | 3 | (1, 1, 1) | 6 |
| 3 | 53.9 | 7.40 | 23 | 0.470366 | 0.453213 | 0.221244 | 4 | (2, 0, 0) | 6 |
| 4 | 56.5 | 6.60 | 200 | 0.493056 | 0.473320 | 0.243104 | 4 | (2, 0, 0) | 6 |
| 5 | 66.3 | 4.70 | 92 | 0.578577 | 0.546833 | 0.334751 | 6 | (2, 1, 1) | 24 |
| 6 | 73.2 | 3.80 | 13 | 0.638791 | 0.596225 | 0.408053 | 8 | (2, 2, 0) | 12 |
| 7 | 75.4 | 3.60 | 198 | 0.657989 | 0.611527 | 0.432950 | 8 | (2, 2, 0) | 12 |
| 8 | 84.1 | 3.05 | 136 | 0.733911 | 0.669779 | 0.538625 | 9 | (2, 2, 1) | 24 |
| 9 | 90.6 | 2.80 | 10 | 0.790634 | 0.710799 | 0.625102 | 12 | (2, 2, 2) | 6 |

Este es el resultado y el codigo es

```python
from typing import Generator, Tuple
import pandas as pd
import numpy as np
from math import factorial
from collections import Counter
import os

os.system('clear')

def degeneramiento(tupla):
    total = factorial(len(tupla))
    repeticiones = Counter(tupla)

    for key, rep in repeticiones.items():
        total //= factorial(rep)
        total *= 2*rep if key != 0 else 1

    return total

def succesion() -> Generator[Tuple[int, int, int], None, None]:
    a = 1
    while True:
        for b in range(a + 1):
            for c in range(b + 1):
                yield (a, b, c)
        a += 1

def index(value):
    def helper(h: int, k: int, l: int):
        return np.pow(h, 2) + np.pow(k, 2) + np.pow(l, 2)
    last = (0, 0, 0)
    for h, k, l in succesion():
        val = helper(h, k, l)
        if val == value:
            return h, k, l
        if val > value:
            return last
        last = (h, k, l)

data = pd.read_csv("./data_5.csv")
data["theta"] = np.radians(data["2theta"] / 2)
data["sin"] = np.sin(data["theta"])
data["sin^2"] = np.pow(data["theta"], 2)
data["s/s"] = (data["sin^2"] / data["sin^2"][0]).round().astype(int)
data.loc[6, "s/s"] = 8
data.loc[9, "s/s"] = 12
data["hkl"] = data["s/s"].apply(index)
data["m"] = data["hkl"].apply(degeneramiento)
print(data)
```

Punto 5 D)

Lo mas facil es un codigo

```python
import pandas as pd
import os
import numpy as np
from punto_5_a_b import index


os.system('clear')

data = pd.read_csv("./data_5.csv")
long_onda = 1.542


data["theta"] = np.radians(data["2theta"] / 2)
data["sin"] = np.sin(data["theta"])
data["2sin"] = 2 * data["sin"]
data["d"] = long_onda / data["2sin"]
data["sin^2"] = np.pow(data["theta"], 2)
data["s/s"] = (data["sin^2"] / data["sin^2"][0]).round().astype(int)
data.loc[6, "s/s"] = 8
data.loc[9, "s/s"] = 12
data["hkl"] = data["s/s"].apply(index)
data["sqrt(hkl)"] = data["hkl"].apply(lambda t: np.sqrt(t[0]**2 + t[1]**2 + t[2]**2))
data["a"] = data["d"]*data["sqrt(hkl)"]
print(data)
print(np.mean(data["a"]))
```

```
   2theta      P  Area[arb]     theta       sin      2sin         d     sin^2  s/s        hkl
sqrt(hkl)         a
0    27.3  33.50        116  0.238237  0.235990  0.471980  3.267085  0.056757    1  (1, 0, 0)
1.000000  3.267085
1    31.7  24.00       1260  0.276635  0.273120  0.546240  2.822937  0.076527    1  (1, 0, 0)
1.000000  2.822937
2    45.5  10.90        694  0.397062  0.386711  0.773422  1.993737  0.157659    3  (1, 1, 1)
1.732051  3.453254
3    53.9   7.40         23  0.470366  0.453213  0.906426  1.701188  0.221244    4  (2, 0, 0)
2.000000  3.402375
4    56.5   6.60        200  0.493056  0.473320  0.946639  1.628920  0.243104    4  (2, 0, 0)
2.000000  3.257841
5    66.3   4.70         92  0.578577  0.546833  1.093666  1.409937  0.334751    6  (2, 1, 1)
2.449490  3.453627
6    73.2   3.80         13  0.638791  0.596225  1.192450  1.293136  0.408053    8  (2, 2, 0)
2.828427  3.657542
7    75.4   3.60        198  0.657989  0.611527  1.223054  1.260778  0.432950    8  (2, 2, 0)
2.828427  3.566019
8    84.1   3.05        136  0.733911  0.669779  1.339558  1.151126  0.538625    9  (2, 2, 1)
3.000000  3.453379
9    90.6   2.80         10  0.790634  0.710799  1.421599  1.084694  0.625102   12  (2, 2, 2)
3.464102  3.757491
Promedio: 3.4091548541379892
Divergencia: 0.06669958611705555
```