In [193…
```
import plotly.graph_objects as go
import numpy as np
import matplotlib.pyplot as plt
import tqdm
```

Helper Functions

In [236…
```
def eiktau(ks, tau, minus=False):
    exponetial = np.exp(np.complex(0, 1)*np.dot(ks, tau))
    mod_tau = np.linalg.norm(tau)
    if minus:
        mod_tau = -mod_tau
    return exponetial*mod_tau

def sqrt3by2():
    return np.sqrt(3)/2

def sum_for_ks(ks, points, minus = False):
    sum = 0
    for point in points:
        sum += eiktau(ks, point, minus=minus)

    return np.abs(sum)
```

In [235…
```
eiktau([1,1,1], [2,2,2], minus=True)
```

Out[235… `(-3.3261274407928134+0.9679236786054194j)`

In [233…
```
a=1
b=1
points = [[0,a,0], [a*sqrt3by2(), -a/2, 0], [-a*sqrt3by2(), -a/2, 0],
          [0,0,b], [a*sqrt3by2(), a/2, b], [0, -a, b], [-a*sqrt3by2(), a/2, b]]
sum_for_ks([1,1,1], points, minus=False)
```

Out[233… `4.743291269190796`

Plot

In [241…
```
def plot_band_structure(points, eatomic = 1, gamma = 1):

    kxs = np.linspace(-4, 4, 100)
    kys = np.linspace(-4, 4, 100)
    kzs = np.linspace(-4, 4, 100)

    Esp = np.zeros((100, 100, 100))
    Esn = np.zeros((100, 100, 100))
    # print("Calculating for negative value of gamma.")
    for i in tqdm.tqdm(range(len(kxs)), desc='Calculating...'):
        for l in range(len(kys)):
            for k in range(len(kzs)):
                Esp[i, l, k] = sum_for_ks([kxs[i], kys[l], kzs[k]], points, minus=Fals
    # print("Done!")

    # print("Calculating for positive value of gamma.")
    # for i in tqdm.tqdm(range(len(kxs)), desc='Calculating...'):
    #     for l in range(len(kys)):
    #         for k in range(len(kzs)):
    #             Esn[i, l, k] = sum_for_ks([kxs[i], kys[l], kzs[k]], points, minus=T

    x, y = np.meshgrid(kxs, kys)
    # E1 = eatomic + gamma*Esp[:, :, 0]
    # E1 = eatomic + gamma*Esp[:, :, 0]
    fig =  go.Figure(data = [go.Surface(x=x, y=y, z=Esp[:, :, 0]),
        go.Surface(x=x, y=y, z=-Esp[:, :, 0]),
        # go.Surface(x=x, y=y, z=Esn[:, :, 0]),
        # go.Surface(x=x, y=y, z=-Esn[:, :, 0])
        ])

    fig.update_layout(title='Band Structure',
        scene = dict(
            xaxis_title='kx',
            yaxis_title='ky',
            zaxis_title='E'
        ))
    fig.show()
    return Esp
```

AB Configuration

In [246…
```
a=1
b=1
points = [[0,a,0], [a*sqrt3by2(), -a/2, 0], [-a*sqrt3by2(), -a/2, 0],
          [0,0,b], [a*sqrt3by2(), a/2, b], [0, -a, b], [-a*sqrt3by2(), a/2, b]]

Esp = plot_band_structure(points)
```

```
Calculating...: 100%|██████████| 100/100 [02:58<00:00,  1.78s/it]
```

In [244…
```
a=1
b=1
points = [[0,a,0], [a*sqrt3by2(), -a/2, 0], [-a*sqrt3by2(), -a/2, 0],
          [0,0,b], [a*sqrt3by2(), -a/2, b], [0, a, b], [-a*sqrt3by2(), -a/2, b]]

E = plot_band_structure(points)
```

```
Calculating...: 100%|██████████| 100/100 [03:03<00:00,  1.83s/it]
```

4 Nearest Neighbours

In [245…
```
a=1
b=1
points = [[0,a,0], [a*sqrt3by2(), -a/2, 0], [-a*sqrt3by2(), -a/2, 0],
          [0,0,b],
        #   [a*sqrt3by2(), -a/2, b], [0, a, b], [-a*sqrt3by2(), -a/2, b]
          ]

E = plot_band_structure(points)
```

```
Calculating...: 100%|██████████| 100/100 [01:34<00:00,  1.06it/s]
```

Monolayer

In [201…
```
a=1
b=1
points = [[0,a,0], [a*sqrt3by2(), -a/2, 0], [-a*sqrt3by2(), -a/2, 0],
        #   [0,0,b], [a*sqrt3by2(), -a/2, b], [0, a, b], [-a*sqrt3by2(), -a/2, b]
          ]

E = plot_band_structure(points)
```

```
Calculating...: 100%|██████████| 100/100 [01:15<00:00,  1.32it/s]
```

In [247…
```
a=1
b=1
points = [[-a,0,0], [a/2, a*sqrt3by2(), 0], [a/2, -a*sqrt3by2(), 0],
        #   [0,0,b], [a*sqrt3by2(), -a/2, b], [0, a, b], [-a*sqrt3by2(), -a/2, b]
          ]

E = plot_band_structure(points)
```

```
Calculating...: 100%|██████████| 100/100 [01:12<00:00,  1.38it/s]
```

In [ ]: