DispersionRelation

April 21, 2019

1 Visualization of quadratic and honeycomb lattice dispersion relations in tight-binding model

```
In [23]: import numpy as np
         import matplotlib.pyplot as plt

         import sys
         sys.path.insert(0, "./")
```

```
In [6]: from dispersion import quadratic_lattice_calculator, graphene_dispersion_calculator
        from density_calculator import DensityOfStatesCalculator
```

1.1 Quadratic lattice

Quadratic lattice dispersion relation:

$\epsilon_{\vec{k}} = \epsilon_0 - 2t \cdot (\cos(k_x a_x) + \cos(k_y a_y))$

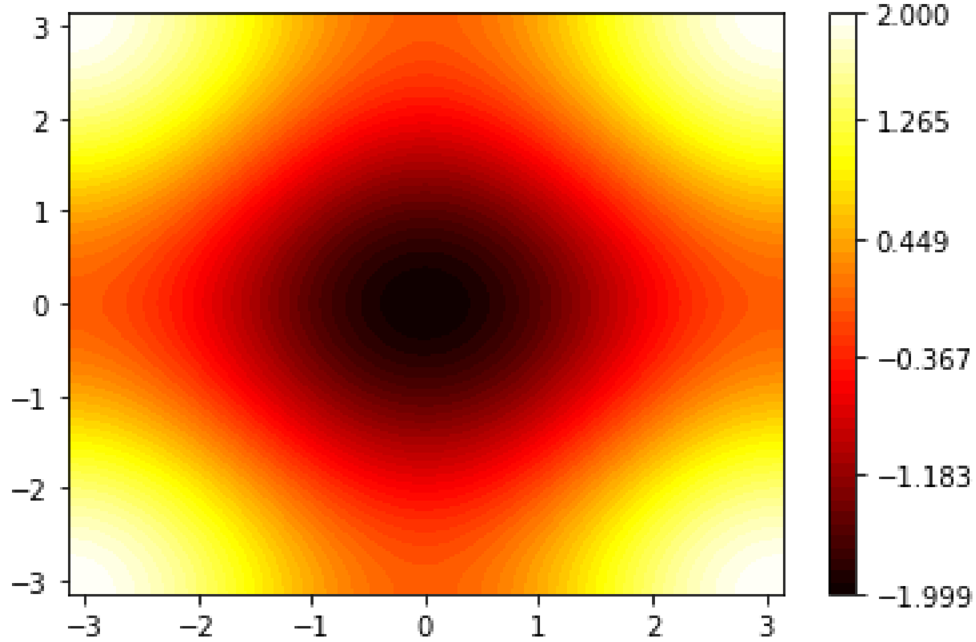
Where ϵ_0 is ground level energy at given site and $-t$ is hopping energy for nearest neighbours.

```
In [31]: disp = quadratic_lattice_calculator()
```

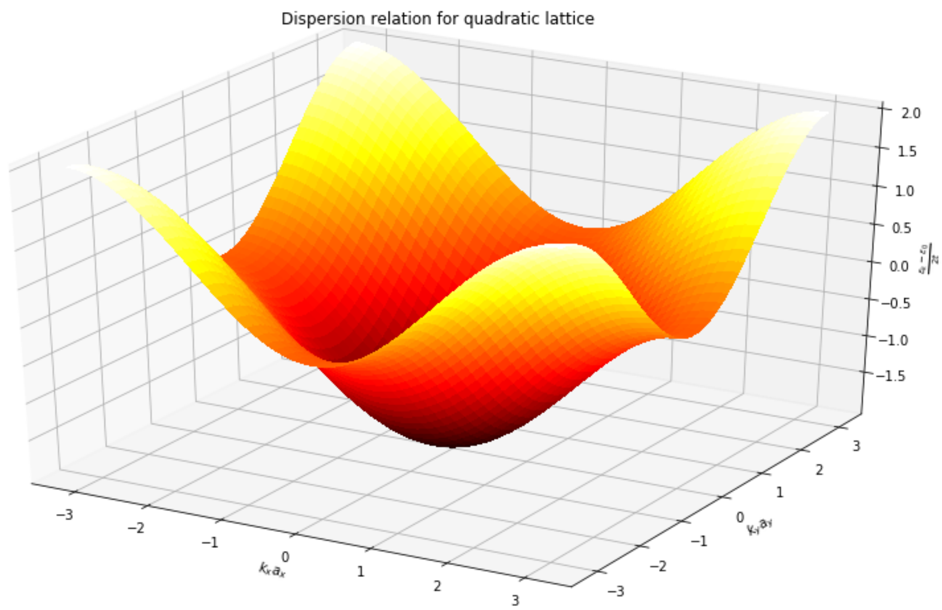
```
In [32]: no_of_grid_points = 100
         grid = np.array([np.linspace(-np.pi, np.pi, no_of_grid_points), np.linspace(-np.pi, np
```

```
In [33]: calc = DensityOfStatesCalculator(disp, grid)
```

```
In [34]: fig, ax = calc.get_2d_contourf_plot()


         #fig
```




```
In [36]: fig, ax = calc.get_3d_plot()

         fig.set_figheight(9)
         fig.set_figwidth(15)

         ax.set_title("Dispersion relation for quadratic lattice")

         ax.set_zlabel(r"$\frac{\epsilon_k - \epsilon_0}{2t}$")
         ax.set_ylabel(r"$k_y a_y$")
         ax.set_xlabel(r"$k_x a_x$")

         #fig

Out[36]: Text(0.5, 0, '$k_x a_x$')
```



1.2 Honeycomb lattice (graphene)

Quadratic lattice dispersion relation:

$\epsilon_{\vec{k}} = \epsilon_0 - 2t \cdot (\cos(\vec{k} \cdot \vec{\delta}_1) + \cos(\vec{k} \cdot \vec{\delta}_2) + \cos(\vec{k} \cdot \vec{\delta}_3))$

Where ϵ_0 is ground level energy at given site and $-t$ is hopping energy for nearest neighbours.

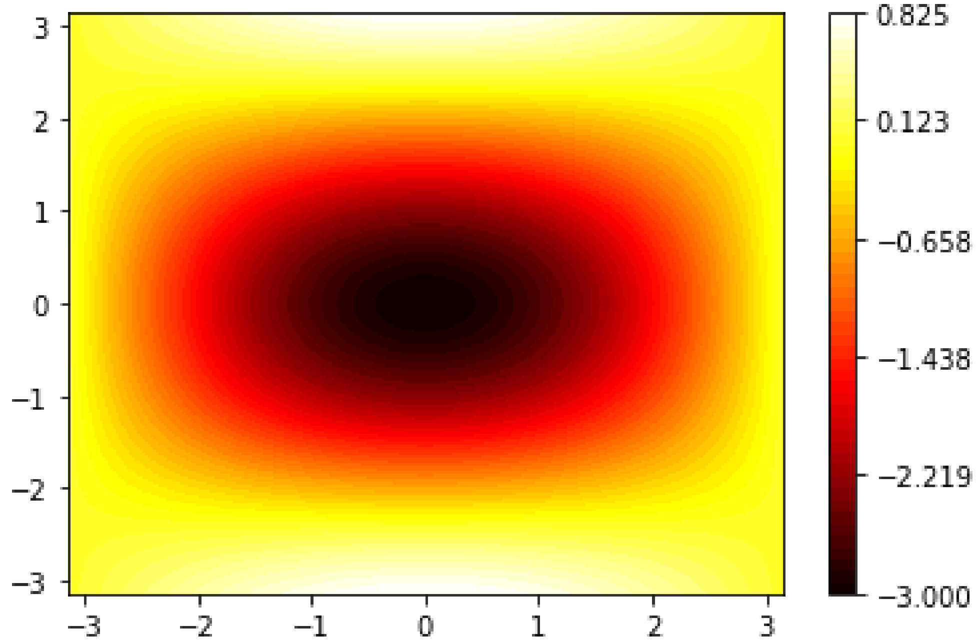
Vectors $\vec{\delta}_1$, $\vec{\delta}_2$, $\vec{\delta}_3$ describe lattice:

$\vec{\delta}_1 = a_0(\frac{1}{2}, \frac{\sqrt{3}}{2})$

$\vec{\delta}_2 = a_0(\frac{1}{2}, -\frac{\sqrt{3}}{2})$

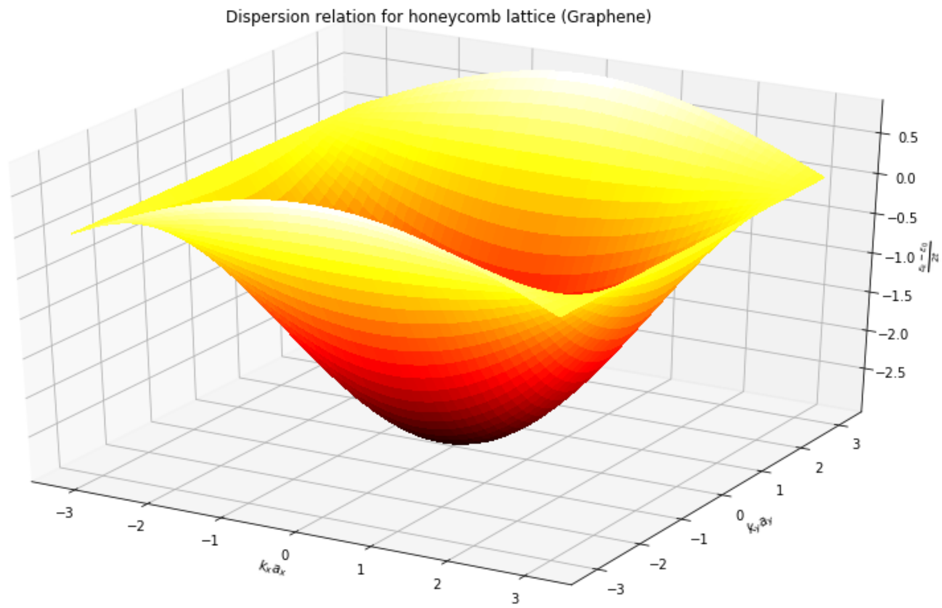
$\vec{\delta}_3 = a_0(-\frac{1}{2}, 0)$

```
In [37]: disp = graphene_dispersion_calculator()

In [38]: no_of_grid_points = 200
         grid = np.array([np.linspace(-np.pi, np.pi, no_of_grid_points), np.linspace(-np.pi, np

In [39]: calc = DensityOfStatesCalculator(disp, grid)

In [40]: fig, ax = calc.get_2d_contourf_plot()
         #fig
```



In [41]:
```
fig, ax = calc.get_3d_plot()

fig.set_figheight(9)
fig.set_figwidth(15)

ax.set_title("Dispersion relation for honeycomb lattice (Graphene)")

ax.set_zlabel(r"$\frac{\epsilon_k - \epsilon_0}{2t}$")
ax.set_ylabel(r"$k_y a_y$")
ax.set_xlabel(r"$k_x a_x$")
```

Out[41]:
```
Text(0.5, 0, '$k_x a_x$')
```



Dispersion relation for honeycomb lattice (Graphene)
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3
−2
−1
0
1
2
3
$k_x a_x$
$k_y a_y$


```
In [ ]:
```