



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Радиоэлектроника и лазерная техника»
Кафедра «Технологии приборостроения»

Индивидуальное задание
на тему дисперсионная зависимость графена по курсу
«Информационное обеспечение разработок в области
наноинженерии»

Выполнил: Веселенко Н.Ю.
РЛ6-61Б

Проверил: доцент кафедры РЛ6, к.т.н.
Ветрова Н.А.

Москва, 2020

Оглавление

МОНОСЛОЙНЫЙ ГРАФЕН.....	3
РЕШЕНИЕ ЗАДАЧ	10
Задача №1, тема «ЧИСЛЕННОЕ РЕШЕНИЕ УРАВНЕНИЯ ШРЕДИНГЕРА».....	10
ВЫВОД ПО ДИСПЕРСИОННОЙ ЗАВИСИМОСТИ ГРАФЕНА	11
О ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON И О ТОМ, КАК	
ЗАПУСТИТЬ ПРОГРАММУ	12
УСТАНОВКА PYTHON НА WINDOWS	13
ПРОСМОТР И РЕДАКТИРОВАНИЕ КОДА НА WINDOWS	18
УСТАНОВКА PYTHON НА MacOS	20
ПРОСМОТР И РЕДАКТИРОВАНИЕ КОДА НА MacOS	22
УСТАНОВКА PYTHON НА LINUX	22
ПРОСМОТР И РЕДАКТИРОВАНИЕ КОДА НА LINUX	23
ИСТОЧНИКИ	24

Монослойный графен

Графен представляет собой однослойную двумерную углеродную структуру, состоящую из правильных шестиугольников со стороной 1,422 Å и атомами углерода в вершинах. Эта структура является составляющей графита, в котором такие графеновые слои располагаются друг над другом.

Каждый атом углерода в графене окружён тремя ближайшими соседями и обладает четырьмя валентными электронами¹, три из которых образуют sp^2 -гибридизированные орбитали, расположенные в одной плоскости под углами 120° и формирующие ковалентные связи с соседними атомами. Четвёртый электрон, представлен ориентированной перпендикулярно этой плоскости $2p_z$ -орбиталью (Рис.1).

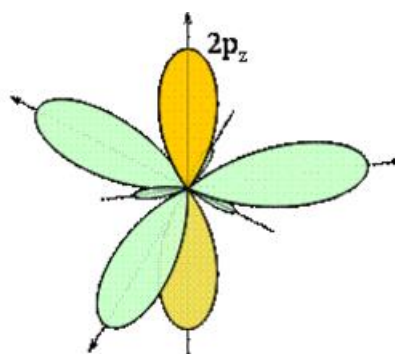


Рис.1 Три sp^2 -гибридизованные орбитали атома углерода, формирующие направленные под углами 120° ковалентные связи с соседними атомами и перпендикулярная им $2p_z$ -орбиталь.

Кристаллическая структура графена представлена на рис.2. Она является двумерной гексагональной решёткой². Постоянная решётки равна 2,464 Å.

¹ Валентные электроны- это электроны, находящиеся на внешней (валентной) оболочке атома.

² Двумерная гексагональная решетка – это такая решетка, у которой трансляционные векторы равны друг другу и угол между ними равен 120° .

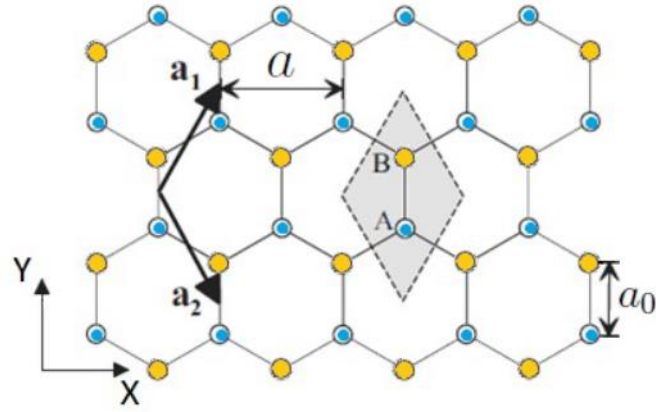


Рис.2 Кристаллическая решетка монослойного графена. Темным цветом выделена элементарная ячейка, состоящая из атомов А и В. a - постоянная решетки, a_0 – расстояние между атомами, a_1 и a_2 – трансляционные вектора³.

Для расчета зонной структуры электронов в графене можно использовать метод сильной связи⁴. Гамильтониан для произвольного импульса k в зоне Бриллюэна будет иметь такой вид:

$$\hat{H} = \begin{pmatrix} 0 & \gamma_0 \sum_{j=1}^3 e^{iku_j} \\ \gamma_0 \sum_{j=1}^3 e^{-iku_j} & 0 \end{pmatrix}, \quad (1.1)$$

где векторы, соединяющие ближайших соседей, имеют вид:

$$u_1 = (-a, 0), u_2 = \left(\frac{1}{2}a, \frac{\sqrt{3}}{2}a\right), u_3 = \left(\frac{1}{2}a, -\frac{\sqrt{3}}{2}a\right) \quad (1.2)$$

Собственные значения гамильтониана принимают значения:

$$E = \sqrt{\gamma_0^2 \sum_{j=1}^3 e^{iku_j} \sum_{j=1}^3 e^{-iku_j}} = \pm \gamma_0 \sqrt{\sum_{j=1}^3 e^{iku_j} \sum_{j=1}^3 e^{iku_j}} \quad (1.3)$$

³ Трансляционный вектор- это определенный вектор, при сдвиге, на который образуется симметрия. В данном случае решетка «пчелиные соты» образована постоянным сдвигом подрешетки А и В на вектор трансляции

⁴ В приближении сильно связанных электронов предполагается, что полный гамильтониан H (оператор полной энергии) системы можно приблизить гамильтонианом изолированного атома, сосредоточенного на каждом узле кристаллической решётки.

$$\begin{aligned}\sum_{j=1}^3 e^{iku_j} &= e^{-ik_x a} e^{0i} + e^{ik_x a/2} e^{i\sqrt{3}k_y a/2} + e^{ik_x a/2} e^{-i\sqrt{3}k_y a/2} \\ &= e^{-ik_x a} + e^{ik_x a/2} \left(e^{i\sqrt{3}k_y a/2} + e^{-i\sqrt{3}k_y a/2} \right)\end{aligned}$$

По формуле Эйлера ($\cos x = \frac{e^{ix} + e^{-ix}}{2}$) получаем: $2\cos \frac{\sqrt{3}k_y a}{2} = e^{i\sqrt{3}k_y a/2} + e^{-i\sqrt{3}k_y a/2}$

$$\sum_{j=1}^3 e^{-iku_j} = e^{-ik_x a} + 2e^{ik_x a/2} \cos \frac{\sqrt{3}k_y a}{2} \quad (1.5)$$

$$\begin{aligned}\sum_{j=1}^3 e^{iku_j} &= e^{ik_x a} e^{0i} + e^{-ik_x a/2} e^{-i\sqrt{3}k_y a/2} + e^{-ik_x a/2} e^{i\sqrt{3}k_y a/2} = e^{ik_x a} + \\ &e^{-ik_x a/2} \left(e^{i\sqrt{3}k_y a/2} + e^{-i\sqrt{3}k_y a/2} \right) = e^{ik_x a} + 2e^{-ik_x a/2} \cos \frac{\sqrt{3}k_y a}{2}\end{aligned} \quad (1.6)$$

Подставляя (1.5) и (1.6) в 1.4 получаем собственные значения энергии:

$$\begin{aligned}E(\vec{k}) &= \pm \gamma_0 \sqrt{\sum_{j=1}^3 e^{iku_j} \sum_{j=1}^3 e^{-iku_j}} \\ &= \pm \gamma_0 \sqrt{\left(e^{-ik_x a} + 2e^{ik_x a/2} \cos \frac{\sqrt{3}k_y a}{2} \right) \left(e^{ik_x a} + 2e^{-ik_x a/2} \cos \frac{\sqrt{3}k_y a}{2} \right)}\end{aligned}$$

Выносим из левой скобки $e^{-ik_x a}$ и из правой скобки $e^{ik_x a}$ и получаем:

$$\begin{aligned}
E(\vec{k}) &= \pm \gamma_0 \sqrt{e^{-ik_x a} \left(1 + 2e^{i3k_x a/2} \cos \frac{\sqrt{3}k_y a}{2}\right) e^{ik_x a} \left(1 + 2e^{-i3k_x a/2} \cos \frac{\sqrt{3}k_y a}{2}\right)} \\
&= \pm \gamma_0 \sqrt{\left(1 + 2e^{i3k_x a/2} \cos \frac{\sqrt{3}k_y a}{2}\right) \left(1 + 2e^{-i3k_x a/2} \cos \frac{\sqrt{3}k_y a}{2}\right)} \\
&= \pm \gamma_0 \sqrt{1 + 2e^{-i3k_x a/2} \cos \frac{\sqrt{3}k_y a}{2} + 4e^{i3k_x a/2} e^{-i3k_x a/2} \cos^2 \frac{\sqrt{3}k_y a}{2} + 2e^{i3k_x a/2} \cos \frac{\sqrt{3}k_y a}{2}} \\
&= \pm \gamma_0 \sqrt{1 + 2\cos \frac{\sqrt{3}k_y a}{2} \left(e^{-i3k_x a/2} + e^{i3k_x a/2} + 2\cos \frac{\sqrt{3}k_y a}{2}\right)}
\end{aligned}$$

Используя уравнение Эйлера получаем: $2\cos \frac{3a}{2}$

$$E(\vec{k}) = \pm \gamma_0 \sqrt{1 + 4\cos \frac{\sqrt{3}k_y a}{2} \left(\cos \frac{3k_x a}{2} + \cos \frac{\sqrt{3}k_y a}{2}\right)} \quad (1.7)$$

Раскрывая скобки (1.7) в итоге получаем закон дисперсии графена:

$$\varepsilon_{\pm}(\vec{k}) = \pm \gamma_0 \sqrt{1 + 4\cos^2 \frac{\sqrt{3}k_y a}{2} + 4\cos \frac{\sqrt{3}k_y a}{2} \cos \frac{3k_x a}{2}} \quad (1.8)$$

где

ε -зонная структура однослойного графена (энергетический спектр электронов в графене), «+»-соответствует электронам (зона проводимости), «-» - дыркам(валентная зона);

\vec{k} -произвольный импульс в зоне Бриллюэна;

γ_0 -энергия перехода к предыдущему или следующему ближайшему соседнему атому (равная 2,7эВ);

$a = 1.42\text{\AA}$, где a -расстояние между атомами(рис.2);

k_x и k_y координаты в импульсном пространстве.

Дисперсионная зависимость графена

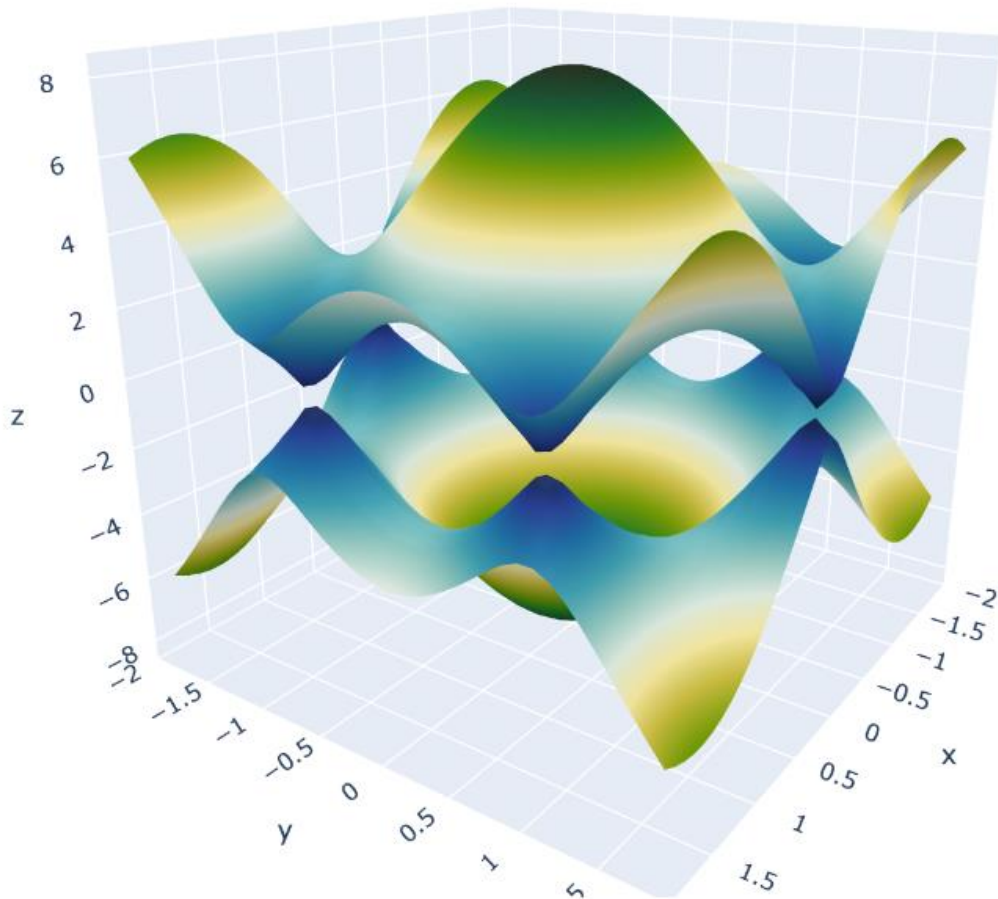


Рис.3 Зонная структура однослойного графена.

Зонная структура однослойного графена, рассчитанная по формуле (1.8), изображена на рисунке 3. В середине первой зоны Бриллюэна⁵ находится Г-точка, а на краях неэквивалентные К и К', соответствующие точкам Дирака или

⁵ БРИЛЛЮЭНА ЗОНЫ - области значений волнового вектора k , при которых энергия электронов изменяется непрерывно, а на границах претерпевает разрыв. Первой зонной называют минимальный по объему многогранник построенный вокруг начала координат в пространстве k , содержащий все возможные различные состояния. Первая Бриллюэна зона — область обратного пространства с центром в начале координат, определяемая следующим образом: если построить плоскости, проходящие через середины векторов, соединяющих начало координат с ближайшими узлами обратной решётки, то образованный ими многогранник и есть 1-я Бриллюэна зона. Каждой кристаллической решётке (прямой решётке) соответствует обратная решётка, в свою очередь определяющая Бриллюэна зону.

точкам электронейтральности. Эти точки представляют большой интерес при изучении электронных свойств графена, т.к. в них происходит пересечение валентной зоны и зоны проводимости, т.е. $\varepsilon_{\pm}(K) = 0$. Одна из точек К соответствует подрешетке А, а другая подрешетке В. Будем рассматривать одну точку, например, К. Чтобы изучить область вокруг К (или К') более подробно, введем $\vec{p} = \vec{k} - K$, тогда уравнение (1.8) примет вид $-\frac{\sqrt{3}a}{2}\gamma_0 e^{-\frac{iK_x}{\sqrt{3}}}(ip_x - p_y) + O(p^2)$, и получается:

$$E(\vec{p}) = \hbar v_F \vec{p}, \quad (1.9)$$

где v_F -скорость Ферми и $\sigma = \{\sigma_x, \sigma_y\}$ – двумерный вектор. Получилось, что носители заряда подчиняются линейному закону дисперсии, как и фотоны, а скорость Ферми в графене $v_F = \frac{\sqrt{3}\gamma_0 a}{2} \approx 10^6$ м/с, что составляет 1/300 скорости света. Таким образом, уравнением Дирака⁶(1.9) для безмассовых фермионов⁷ можно описать носители заряда в графене, и, следовательно, их эффективная масса равна нулю.

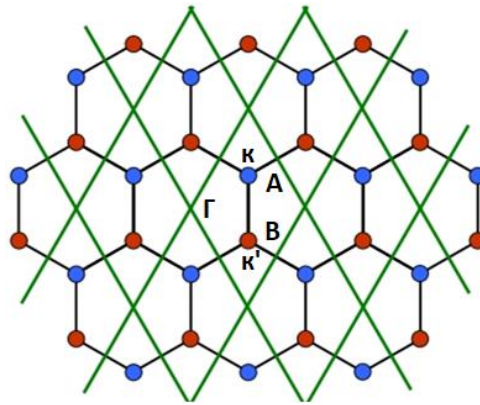


Рис. 4 - Зеленый ромб-элементарная подрешетка. Шестиугольник-обратная решетка, образующая первую зону Бриллюэна. Точки К и К' соответствуют точками Дирака и точкам А и В элементарной подрешетки. Точка Г в центре первой зоны Бриллюэна.

⁶ Уравнение Дирака — релятивистски-инвариантное уравнение движения для биспинорного классического поля электрона, применимое также для описания других точечных фермионов со спином 1/2.

⁷ Фермион — частица или квазичастица с полуцелым значением спина.

Листинг 1. Построение графика дисперсионной зависимости графена

```
import numpy as np # импортирование модуля для общих математических функций, используя
# сокращенное название np
# библиотека для создания графиков
import plotly.graph_objects as go # импортирование модуля создания графиков, используя
# сокращенное название go
from plotly.subplots import make_subplots # импортирование из модуля создания графиков
# функции make_subplots

# Constants
a = 1.42 # задаем расстояние между соседними атомами
d = a * np.sqrt(3) # создание вспомогательной переменной d. np.sqrt(x) - обращение к
# функции квадратного корня из модуля
# numpy. Функция на вход принимает число 'x' и возвращает значение квадратного корня
t = 2.7 # задаем переменную t-энергия перехода к предыдущему или следующему ближайшему
# соседнему атому

"""Получение массива значений x и y
np.arange(x0, x1, dx) - обращение к функции создания массива по двум крайним
значениям с заданным шагом из модуля
numpy, где x0-левый конец отрезка(включенный в массив), x1-правый конец отрезка(не
включенный в массив), dx-шаг.
np.pi - константное значение числа Пи из модуля numpy"""
x = np.arange(-0.6 * np.pi, 0.6 * np.pi, 0.1)
y = np.arange(-0.6 * np.pi, 0.6 * np.pi, 0.1)
"""Получение координатной сетки
np.meshgrid()-функция из модуля numpy для создание матрицы по входным значениями"""
x, y = np.meshgrid(x, y) # Получение координатной сетки
"""Получение функции E(x, y) - энергетический спектр электронов в графене.
np.sqrt(x) - описана выше.
np.cos(x)-функция косинуса из модуля numpy, которая принимает на вход значение x и
возвращает значение косинуса от x"""
E = t * np.sqrt(1 + 4 * pow(np.cos((d * y) / 2), 2) + 4 * np.cos((d * y) / 2) *
np.cos((3 * a * x) / 2))
# Создание прототипа класса для визуализации
fig = make_subplots(rows=1, cols=1, # 1 график
                    specs=[[{'is_3d': True}]], # 3D график
                    subplot_titles=['Дисперсионная зависимость графена'], # Подпись
графика
                    )
"""Построение верхней части графика (+E)
Вызов метода прототипа fig.add_trace(), принимающего на вход график
go.Surface()-вызов функции для построения графика, который на вход принимает
обязательные параметры x, y, z
и все остальные необязательные(опциональные)"""
fig.add_trace(go.Surface(x=x, # Задаем x
                        y=y, # Задаем y
                        z=E, # Задаем z
                        surfacecolor=E + 8, # Задаем градиент цвета
                        colorscale='delta', # Выбираем цветовую гамму
                        showscale=False, # отключение таблицы с зависимостью
градиента цвета от значений(координат)
                        name="+E"))
# Построение нижней части графика (-E)
fig.add_trace(go.Surface(x=x,
                        y=y,
                        z=-E,
                        surfacecolor=E - 8,
                        colorscale='delta',
```

```

        showscale=False,
        name="-E"))
fig.show() # Функция для вывода полученно графика в отдельном окне

```

Решение задач

Задача №1, тема «Численное решение уравнения Шредингера»

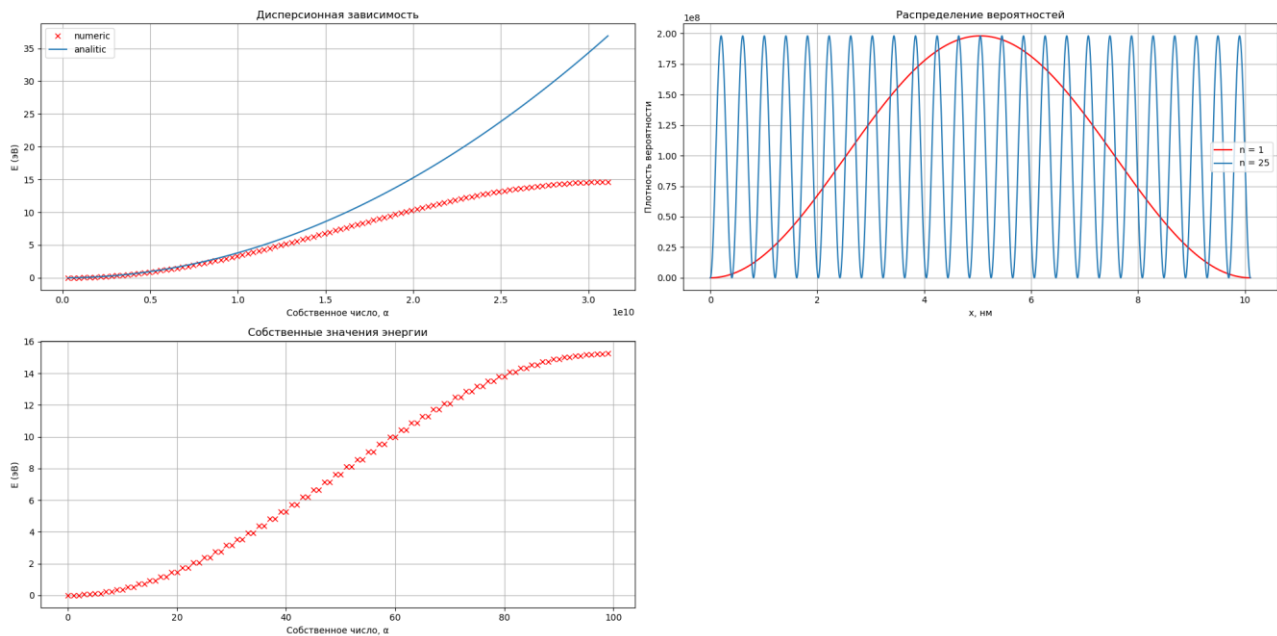


Рис. 5 – Полученные графики.

Листинг 2 – Численное решение уравнения Шредингера

```

# Импорт модулей
import matplotlib.pyplot as plt # модуль для построения графиков
import numpy as np # модуль для общих математических функций, используя сокращенное
название np

def J2eV(E):
    return E * 6.242 * 10 ** 18

m0 = 9.1e-31 # Масса частицы
hbar = 1.0546e-34 # Постоянная Дирака
L = 101e-10 # Длина ямы
Np = 100
# Нахождение Гамильтонианна
Lrr = np.linspace(0, L, Np)
Lr = Lrr[:]
Lan = np.linspace(0, L, 10000)
a = Lr[1] - Lr[0]
t0 = hbar ** 2 / 2 / m0 / a ** 2
H = np.diag(np.ones(len(Lr)) * 2 * t0)
H = H + np.diag(np.diag(H, 1) - t0, 1)
H = H + np.diag(np.diag(H, -1) - t0, -1)
# Получение собственных векторов гамильтониана (матрица Psy)
En, Psy = np.linalg.eigh(H)

```

```

n = 1
plt.figure(figsize=(20, 10)) # Задание размеров области построения графиков
plt.subplot(2, 2, 2) # Расположение графика
# Построение графика плотности вероятности от x при n = 1
plt.plot(Lan * pow(10, 9), 2 / L * np.abs(np.sin(np.pi * n * Lan / L)) ** 2,
color='red', label="n = 1")
n = 25
# Построение графика плотности вероятности от x при n = 25
plt.plot(Lan * pow(10, 9), 2 / L * np.abs(np.sin(np.pi * n * Lan / L)) ** 2, label="n =
25")
plt.xlabel("x, нм") # Подпись оси абцисс
plt.ylabel("Плотность вероятности") # Подпись оси ординат
plt.title("Распределение вероятностей") # Подпись графика
plt.grid() # Включение сетки
plt.legend() # Включение легенды
plt.subplot(2, 2, 1) # Расположение графика
Eanalytic = []
k = [(i + 1) * np.pi / L for i in range(Np)]
# Вычисление энергии численным методом
for i in range(len(En)):
    En[i] = J2eV(En[i])
# Вычисление энергии аналитическим методом
for i in k:
    Eanalytic.append(J2eV(hbar ** 2 * i ** 2 / 2 / m0))
k1 = [i * 2 for i in k]
plt.plot(k, En, "x", label='numeric', color='red') # Построение численного графика
plt.plot(k, Eanalytic, label='analitic') # Построение аналитического графика
plt.xlabel("Собственное число, α") # Подпись оси абцисс
plt.ylabel("E (эВ)") # Подпись оси ординат
plt.title("Дисперсионная зависимость") # Подпись графика
plt.grid() # Включение сетки
plt.legend() # Включение легенды
plt.subplot(2, 2, 3) # Расположение графика
# Вычисление численным методом дисперсионной зависимости
T = np.diag(np.ones(Np) * 7.6265)
t01 = 3.8132
for i in range(len(T)):
    for j in range(len(T[i])):
        if i + 1 == j:
            T[i][j] = -t01
        if i - 1 == j:
            T[i][j] = -t01
T[0][Np - 1] = -t01
T[Np - 1][0] = -t01
D, V = np.linalg.eigh(T)
plt.plot(D, "x", color='red') # Построение графика
plt.title("Собственные значения энергии") # Подпись оси абцисс
plt.xlabel("Собственное число, α") # Подпись оси ординат
plt.ylabel("E (эВ)") # Подпись графика
plt.grid() # Включение сетки
plt.show() # Вывод всех графиков

```

Вывод по дисперсионной зависимости графена

Кристаллическая структура графена состоит из двух эквивалентных подрешеток, что приводит к образованию двух энергетических зон и двух "конических" точек на уровне нулевого заряда носителей K и K' , в которых валентная зона и зона проводимости соприкасаются. В результате носители заряда в графене ведут себя как фотоны, или безмассовые квазичастицы с постоянной "эффективной" скоростью света (скоростью Ферми) $v_F \sim 10^6$ м/с. При этом электроны и дырки являются фермионами, т.е. частицами с полуцелым значением спина, и они заряжены. В настоящее время аналогов для таких безмассовых заряженных фермионов среди известных элементарных частиц нет.

Получив картину дисперсионной зависимости складывается понимание о дисперсии электронов в графене. В отличие от полупроводников, у которых дисперсионная зависимость представляет собой параболу, у графена дисперсионная зависимость представляет собой два соприкасающихся конуса.

О языке программирования Python и о том, как запустить программу

Python — высокоуровневый объектно-ориентированный язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объем полезных функций.

Одной из главных особенностей «питона» является большое количество различных библиотек, с помощью которых можно решать различного рода задачи. Например, с помощью библиотеки pandas можно с легкостью работать

большими данными из различных таблиц, а с помощью библиотеки matplotlib эти самые данные можно визуализировать на графиках и гистограммах.

В данном проекте для построения дисперсионной зависимости графена использовалась библиотека plotly (<https://plotly.com>), с помощью которой можно построить 2D графики, гистограммы, 3D графики и много другое.

Установка Python на Windows

Важно! Команды выделены *курсивом*, некоторые из них можно скопировать и вставить. Перед выполнением каждого пункта инструкции, дочитайте его до конца.

1. Осуществите переход по данной ссылке:
<https://www.python.org/downloads/>
2. И нажмите на кнопку «Download Python...» (рис. 6).

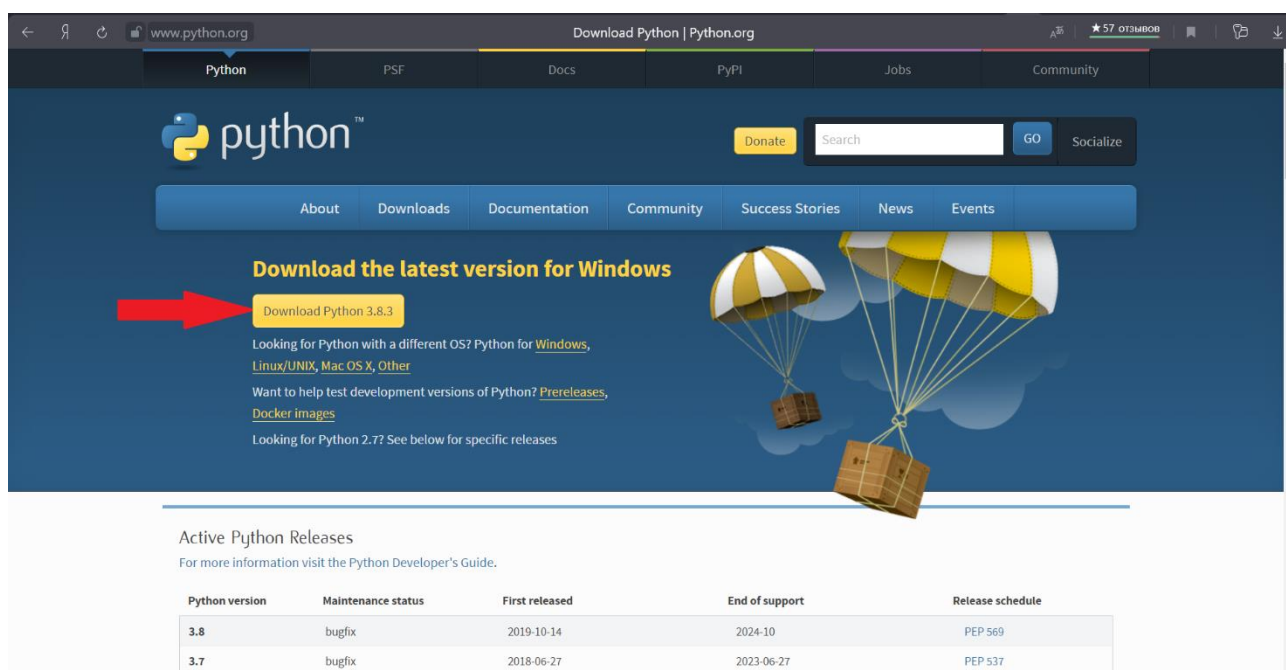


Рис. 6 – Вкладка Downloads сайта python.org

3. После скачивания, запустите скаченный файл.

4. При установке, необходимо поставить галочку напротив «Add Python <версия питона> to PATH» и нажмите кнопку «Install Now» (рис. 7).

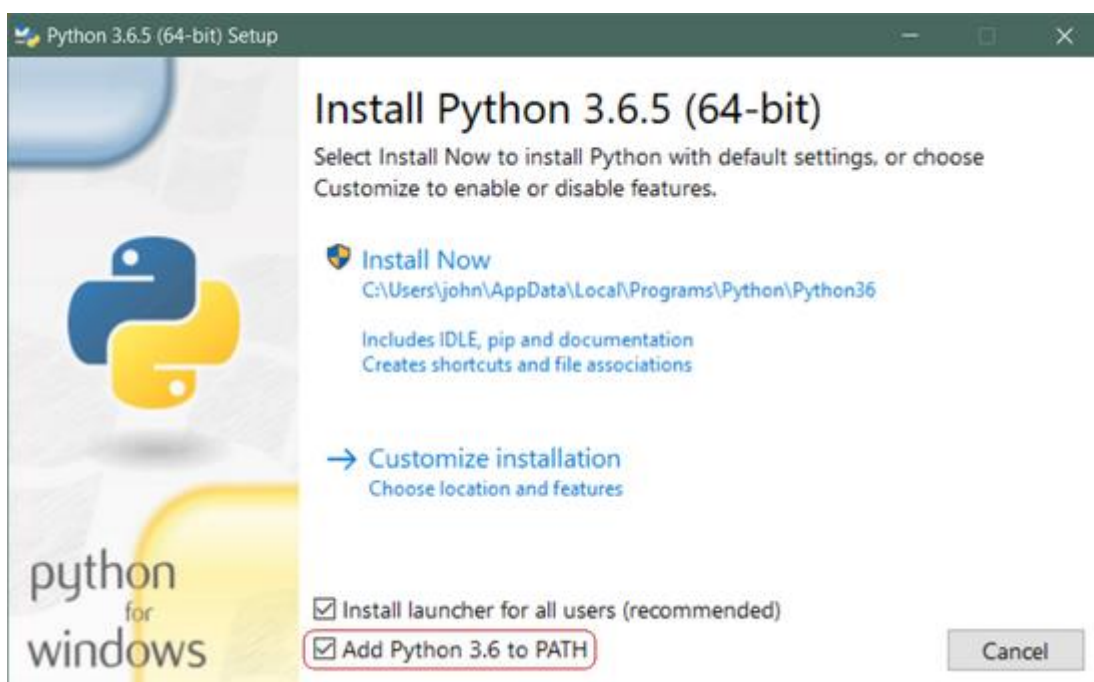


Рис. 7 – Пример окна установки Python на Windows

После завершения установки, можно использовать Python.

5. Откройте терминал/командную строку. Сделать это можно при помощи сочетания клавиш *win+R* (рис. 8). В диалоговом окне введите *cmd*. И нажмите «ОК» или *Enter* (рис. 9).



Рис. 8 – Сочетание клавиш *win+R*

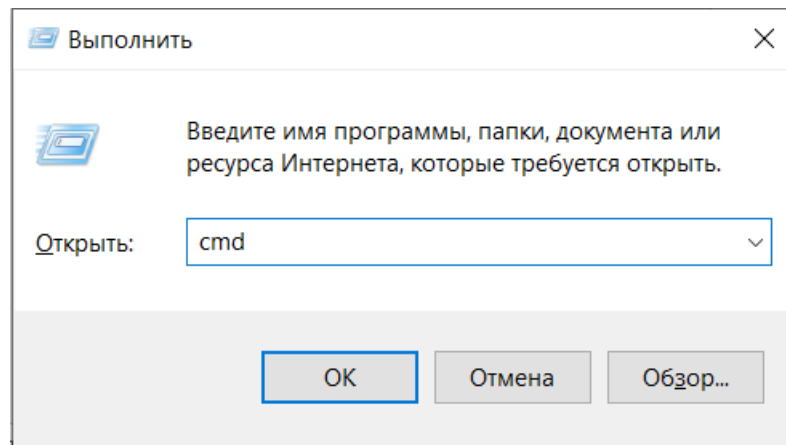


Рис. 9 Выполнение команды cmd

6. В командной строке необходимо ввести следующие команды поочередно (введите сначала одну команду, дождитесь полной загрузки, затем другую).

pip install plotly

pip install numpy

Выполнение этого пункта обязательно, т.к. при помощи этих команд устанавливаются необходимые для запуска программы библиотеки (модули). Обязательно дождитесь полной загрузки модулей. В случае, если загрузка какого-либо из модулей длится очень долго (более 20-ти минут), отмените загрузку с помощью сочетания клавиш *Ctrl+C*, и повторно пропишите установку модуля.

При установке библиотек, может возникнуть предупреждение о том, что пакетный менеджер *pip* устарел (рис. 10), на него можно не обращать внимания.

```
C:\Users\>pip install numpy
Collecting numpy
  Downloading https://files.pythonhosted.org/packages/52/2c/bf86d762ae65550dc8a7ab8381ba610bb69af6db619b3755f2b73052c6b9/numpy-1.18.4-cp38-cp38-win32.whl (10.8MB)
    | 10.8MB 819kB/s
Installing collected packages: numpy
Successfully installed numpy-1.18.4
WARNING: You are using pip version 19.2.3, however version 20.1.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Рис. 10 – Предупреждение *pip*

7. Скачайте программу *graphen_with_plotly.py*.

8. Откройте терминал/командную строку в той директории, где у Вас находится скаченная программа. Для этого, находясь в директории с программой вместо пути пропишите команду *cmd* (рис. 11).

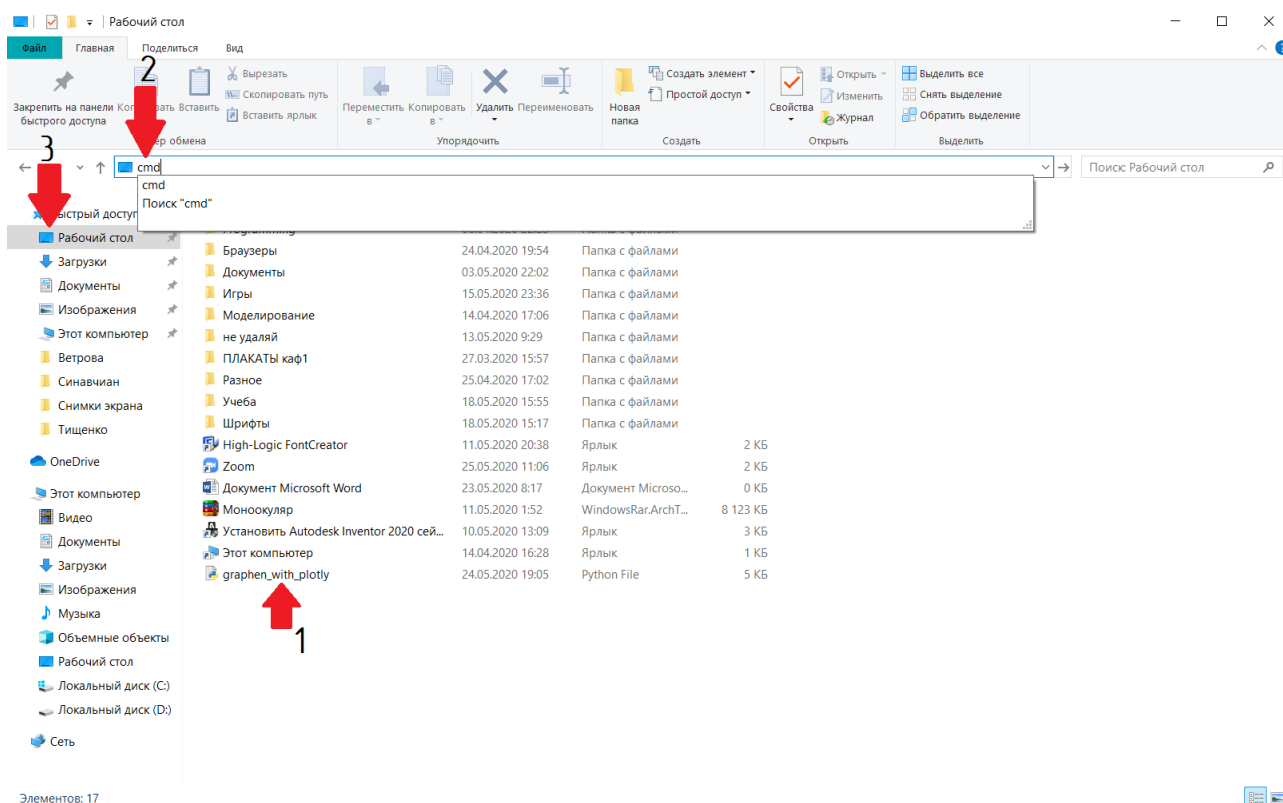


Рис. 11 Снимок окна проводника, для быстрого открытия командной строки в нужной директории. 1 – файл, который будет запускаться, 2 – поля для введения команды *cmd*, 3 – директория, где находится файл 1.

В открывшемся терминале, введите следующую команду:

```
python graphen_with_plotly.py
```

Также вы можете запустить программу, кликнув на нее 2 раза, либо открыв с помощью Python.

Откроется новая вкладка браузера, где появится 3D график дисперсионной зависимости графена, как на рисунке 3. Полученный график можно спокойно вращать, приближать и отдалять.

Важно! Если, открывшаяся в браузере вкладка, не прогрузилась, повторите пункт 8.

9. Удаление. Если Вы не хотите оставлять Python у себя на ПК, Вы можете легко его удалить. В поиске на Вашем ПК введите: “Python”. Выберите установленную ранее Вами версию «питона», и, в открывшемся меню, нажмите кнопку «Удалить» (рис. 12). Далее следует стандартное удаление приложения.

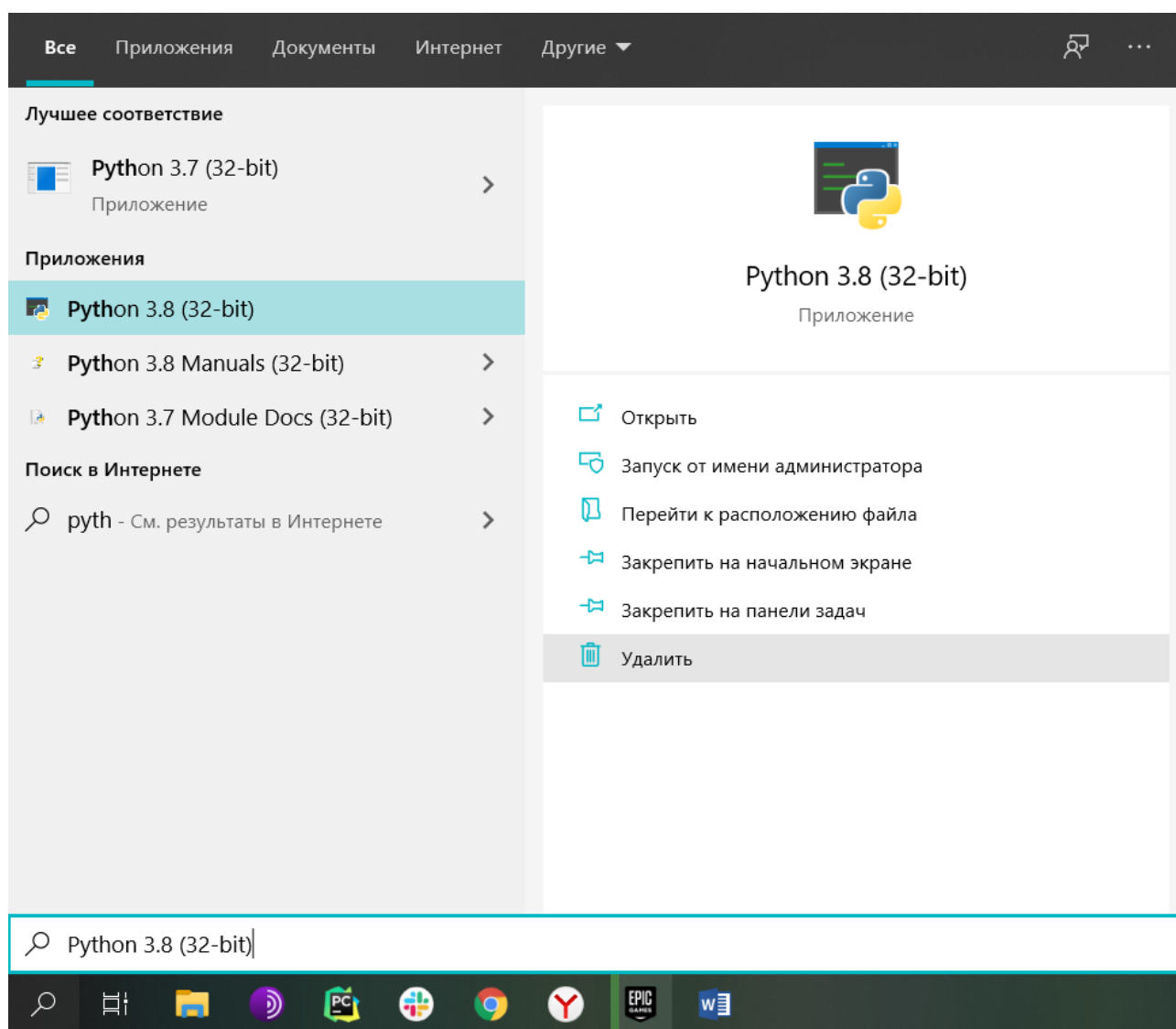


Рис. 12 Удаление Python на Windows 10

Просмотр и редактирование кода на Windows

При установке Python на Windows так же устанавливается стандартное IDLE, позволяющее создавать программы, просматривать их и редактировать.

1. В поиске введите *IDLE*. Выберите его и откройте (рис. 13).

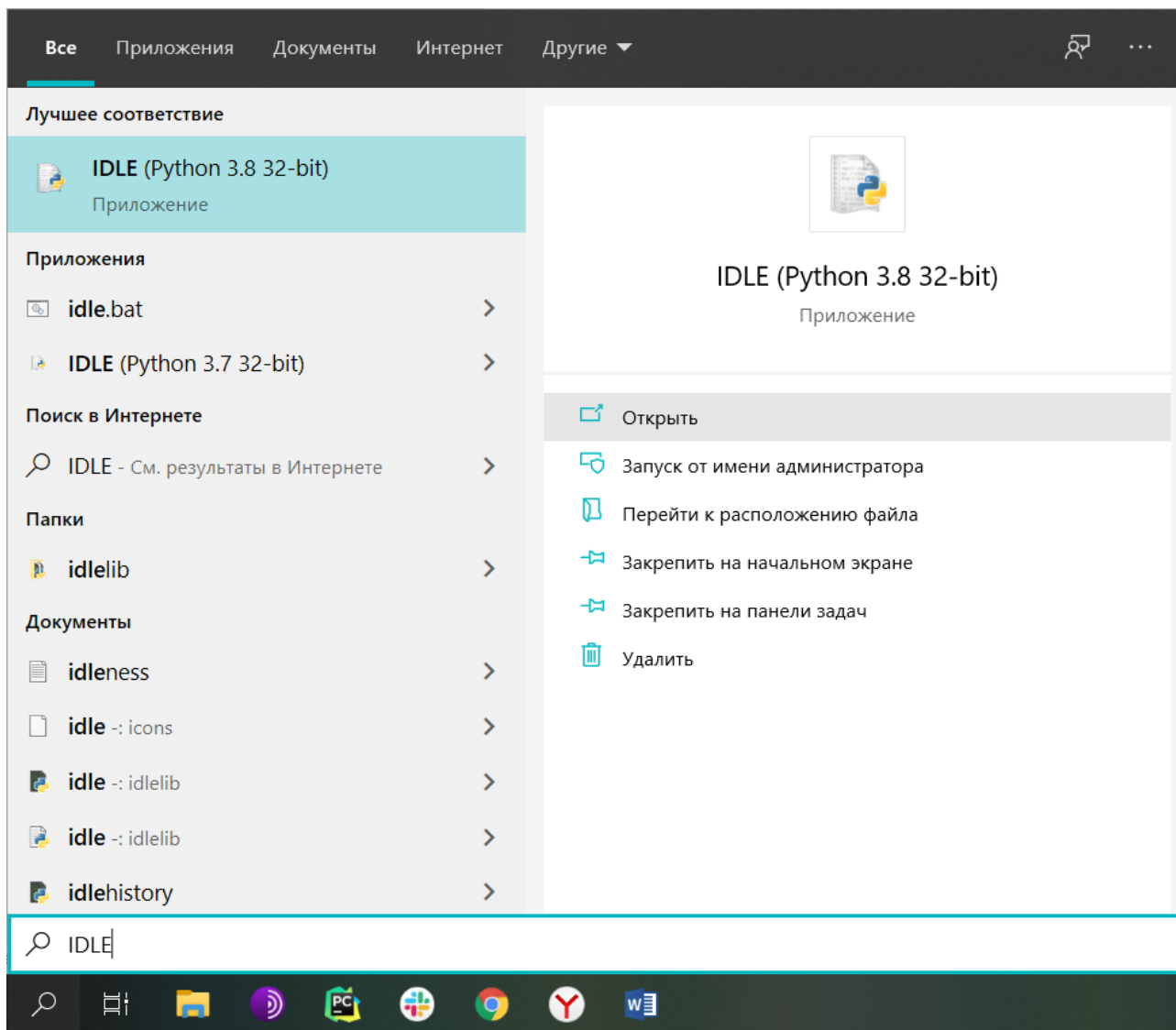


Рис. 13 – Открытие IDLE на Windows

2. В открывшемся окне выберите File –> Open (рис. 14) и выберите файл *graphen_with_plotly.py*. Теперь вы можете с легкостью просмотреть код программы, внести какие-либо изменения, если это требуется и запустить код из IDLE, выбрав Run –> Run_Module или нажав клавишу F5 (рис. 15).

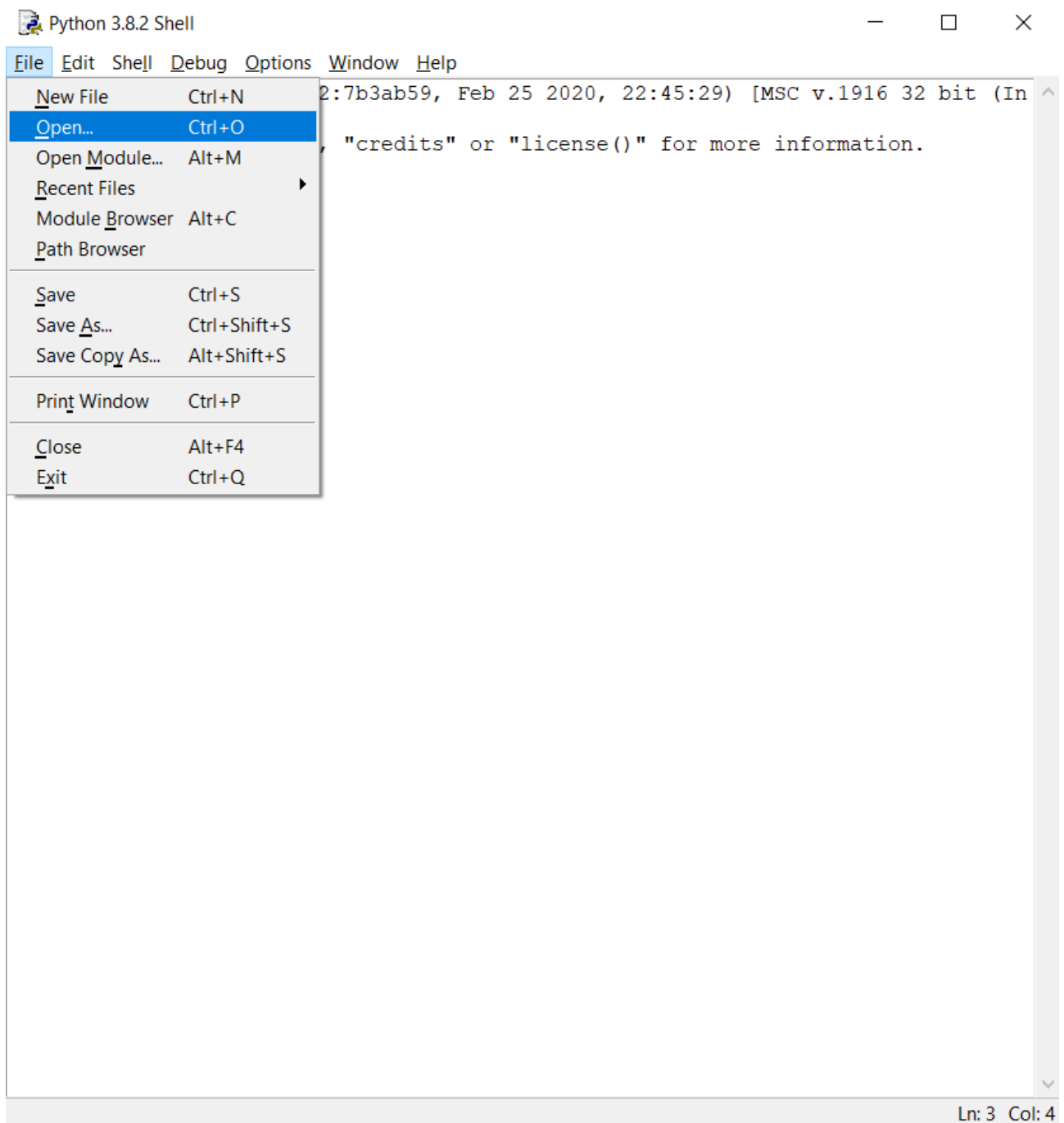


Рис. 14 – Открытие файла с кодом с помощью IDLE

```

"""Получение
np.arange(x0
numpy, где x0
np.pi - конст
"""
x = np.arange(-0.6 * np.pi, 0.6 * np.pi, 0.1)
y = np.arange(-0.6 * np.pi, 0.6 * np.pi, 0.1)
"""Получение координатной сетки
np.meshgrid()-функция из модуля numpy для создание матрицы по входным значениями"""
x, y = np.meshgrid(x, y) # Получение координатной сетки
"""Получение функции E(x, y) - энергетический спектр электронов в графене.
np.sqrt(x) - описана выше.
np.cos(x)-функция косинуса из модуля numpy, которая принимает на вход значение x
E = t * np.sqrt(1 + 4 * pow(np.cos((d * y) / 2), 2) + 4 * np.cos((d * y) / 2) *
# Создание прототипа класса для визуализации
fig = make_subplots(rows=1, cols=1, # 1 график
                    specs=[['is_3d': True]], # 3D график
                    subplot_titles=['Дисперсионная зависимость графена'], # Под
                    )
"""Построение верхней части графика (+E)
Вызов метода прототипа fig.add_trace(), принимающего на вход график
go.Surface()-вызов функции для построения графика, который на вход принимает обя
и все остальные необязательные(опциональные)"""
fig.add_trace(go.Surface(x=x, # Задаем x
                        y=y, # Задаем y
                        z=E, # Задаем z
                        surfacecolor=E + 8, # Задаем градиент цвета
                        colorscale='delta', # Выбираем цветовую гамму
                        showscale=False, # отключение таблицы с зависимостью t
                        name="+E"))
# Построение нижней части графика (-E)
fig.add_trace(go.Surface(x=x,
                        y=y,
                        z=-E,
                        surfacecolor=E - 8,
                        colorscale='delta',
                        showscale=False,
                        name="-E"))
fig.show() # функция для вывода полученно графика в отдельном окне

```

Рис. 15 – Запуск программы из IDLE

Установка Python на MacOS

Важно! Команды выделены *курсивом*, некоторые из них можно скопировать и вставить. Перед выполнением каждого пункта инструкции, дочитайте его до конца.

1. Если у Вас на ПК установлен пакетный менеджер Homebrew, то для установки «питона» Вам подойдет данная инструкция, описанная ниже.

1.1. Чтобы установить Python на MacOS и запустить программу Вам потребуется скачать файлы `installation_on_mac` и `graphen_with_plotly.py`.

1.2. Откройте терминал в директории, где у Вас находятся скаченные файлы и выполните следующую команду: `./installation_on_mac`

Данная команда установит Python и все необходимые библиотеки на Ваш ПК.

В случае возникновения ошибки пропишите следующую команду: `chmod +x installation_on_mac` и повторите предыдущую.

2. Если у Вас на ПК нет пакетного менеджера, и Вы не знаете, что это, то следуйте инструкции написанной ниже.

2.1. Осуществите переход по данной ссылке:

<https://www.python.org/downloads/>

2.2. И нажмите на кнопку «Download Python...» рис. 6.

2.3. Выполните установку, скаченного Python, на Ваш Mac.

2.4. Откройте поиск приложений и найдите приложение «terminal» или «iterm».

2.5. В терминале необходимо ввести следующие команды.

`pip3 install plotly`

`pip3 install numpy`

Выполнение этого пункта обязательно, как и всех остальных, т.к. при помощи этих команд устанавливаются необходимые для запуска программы библиотеки (модули). Обязательно дождитесь полной загрузки модулей. В случае, если загрузка какого-либо из модулей длится очень долго (более 20-ти минут), отмените загрузку с помощью сочетания клавиш `Ctrl+C`, и повторно пропишите установку модуля.

3. После окончания установки запустите код следующей командой: `python3 graphen_with_plotly.py`

Данная команда выполняет скаченный код программы. Откроется новая вкладка браузера, где появится 3D график дисперсионной зависимости графена, как на рисунке 3. Полученный график можно спокойно вращать, приближать и отдалять.

Просмотр и редактирование кода на MacOS

Почти на всех Unix-подобных ОС предустановлен текстовый редактор Vim, каковой и является MacOS.

1. Находясь в директории, откуда производился запуск кода напишите команду *vim graphen_with_plotly.py*.
2. Откроется текстовый редактор Vim. Для редактирования нажмите клавишу *I* (внизу появится фраза --INSERT--, это значит, что можно редактировать код). Для сохранения изменений и выхода из текстового редактора Vim нажмите клавишу *Escape (Esc)*(пропадет фраза --INSERT--), после чего напишите *:wq* (данная команда должна появиться там, где было --INSERT--, w-сохраняет изменения, q-осуществляет выход из Vim).

Установка Python на Linux

Важно! Команды выделены *курсивом*, некоторые из них можно скопировать и вставить. Перед выполнением каждого пункта инструкции, дочитайте его до конца.

1. Чтобы установить Python на Linux и запустить программу Вам потребуется скачать файлы *installation_on_linux* и *graphen_with_plotly.py*.
2. Откройте терминал в директории, где у Вас находятся скаченные файлы и выполните следующую команду: *./installation_on_linux*

Данная команда установит Python и все необходимые библиотеки на Ваш ПК.

В случае возникновения ошибки пропишите следующую команду:
chmod +x installation_on_linux и повторите предыдущую.

3. После окончания установки запустите код следующей командой:

python3 graphen_with_plotly.py

Данная команда выполняет скаченный код программы. Откроется новая вкладка браузера, где появится 3D график дисперсионной зависимости графена. Полученный график можно спокойно вращать, приближать и отдалять.

Просмотр и редактирование кода на Linux

Почти на всех Unix-подобных ОС предустановлен текстовый редактор Vim, каковой и является Linux.

1. Находясь в директории, откуда производился запуск кода напишите команду *vim graphen_with_plotly.py*.
2. Откроется текстовый редактор Vim. Для редактирования нажмите клавишу *I* (внизу появится фраза --INSERT--, это значит, что можно редактировать код). Для сохранения изменений и выхода из текстового редактора Vim нажмите клавишу *Escape* (*Esc*)(пропадет фраза --INSERT--), после чего напишите *:wq* (данная команда должна появиться там, где было --INSERT--, w-сохраняет изменения, q-осуществляет выход из Vim).

Источники

1. ОСОБЕННОСТИ МАГНЕТОСОПРОТИВЛЕНИЯ И ТЕРАГЕРЦОВОЙ ФОТОПРОВОДИМОСТИ В ГРАФЕНЕ //Васильева Галина Юрьевна / Диссертация на соискание ученой степени кандидата физико-математических наук - Санкт-Петербург – 2015 г.
2. Влияние кислорода на взаимодействие графена с металлом // Гращенко Владимир Сергеевич / Бакалаврская работа студента - Санкт-Петербург 2017 г.
3. Уравнение Дирака для графена [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Уравнение_Дирака_для_графена
4. Физика графена [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Физика_графена