

# SciPy. Движение частицы во внешнем электромагнитном поле.

## 1 SciPy

Пакет **SciPy** (<http://www.scipy.org/>) содержит различные инструменты, часто используемые при научном программировании. Модули в пакете соответствуют различным задачам, таким как интерполяция, интегрирование, оптимизация, обработка изображений, спецфункции и другое. Во многом **SciPy** дублирует схожие инструменты из **MATLAB** или библиотек **C++**. **SciPy** основан на **NumPy**, и работает с массивами и матрицами из **NumPy**.

Перед тем, как пытаться самостоятельно реализовать какую-то задачу (например, численное интегрирование каким-либо методом), имеет смысл проверить, не реализована ли эта задача в пакете **SciPy**, так как вряд ли получится реализовать её более эффективно, если вы не являетесь профессионалом в этой области.

### Упражнения по SciPy (4 балла)

1. Ознакомьтесь с возможностями **SciPy**. Все модули перечислены в документации <http://docs.scipy.org/doc/scipy/reference/>. Там же можно найти документацию по всем функциям **SciPy**.  
Возможности **SciPy** очень большие, поэтому для начала предлагается познакомиться лишь с некоторыми из них, а именно с линейной алгеброй, поиском минимумов (оптимизация), нахождением корней уравнений, фитированием и интегрированием. Изучите возможности модулей, а так же разберите и проделайте предложенные ниже примеры (и постройте графики из примеров).
2. (1 балл) Линейная алгебра, модуль `scipy.linalg`.  
Ознакомьтесь с возможностями модуля `scipy.linalg`: <http://docs.scipy.org/doc/scipy/reference/linalg.html#scipy.linalg>.  
Проделайте примеры: <http://scipy-lectures.github.io/intro/scipy.html#linear-algebra-operations-scipy-linalg>
3. (1 балл) Оптимизация и поиск корней уравнений, фитирование, модуль `scipy.optimize`.

Возможности модуля: <http://docs.scipy.org/doc/scipy/reference/optimize.html#scipy.optimize>.

Проделайте примеры: <http://scipy-lectures.github.io/intro/scipy.html#optimization-and-fit-scipy-optimize>

4. (1 балл) Интерполяция, `scipy.interpolate`.

Возможности модуля: <http://docs.scipy.org/doc/scipy/reference/interpolate.html#scipy.interpolate>.

Примеры: <http://scipy-lectures.github.io/intro/scipy.html#interpolation-scipy-interpolate>.

5. (1 балл) Интегрирование (в т.ч. систем ОДУ), `scipy.integrate`.

Документация: <http://docs.scipy.org/doc/scipy/reference/integrate.html#scipy.integrate>.

Примеры: <http://scipy-lectures.github.io/intro/scipy.html#numerical-integration-scipy-integrate>.

## 2 Движение частицы во внешнем электромагнитном поле (4 балла)

Напишите программу, которая рисует траектории движения релятивистской заряженной частицы в произвольном внешнем электромагнитном поле.

1. Уравнения движения частицы во внешнем поле можно записать как

$$\begin{cases} \frac{d\mathbf{p}}{dt} = e \left( \mathbf{E} + \frac{1}{c} [\mathbf{v} \times \mathbf{H}] \right) \\ \frac{d\mathbf{x}}{dt} = \mathbf{v} \end{cases} \quad (1)$$

В релятивистском случае можно выразить  $\mathbf{p} = m\gamma\mathbf{v}$ , где  $\gamma = 1/\sqrt{1-(\mathbf{v}/c)^2} = \sqrt{1+(\mathbf{p}/mc)^2}$  (последнее следует из того, что  $p^0 = mcu^0 = mc\gamma$  и  $(p^0)^2 = m^2c^2 + \mathbf{p}^2$ ).

Таким образом, для решения задачи необходимо проинтегрировать систему ОДУ. Для этого можно использовать модуль `scipy.integrate` из пакета `scipy`, например, функцию `solve_ivp` ([https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve\\_ivp.html#scipy.integrate.solve\\_ivp](https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html#scipy.integrate.solve_ivp))

2. Обезразмерьте систему уравнений (1). Например, для этого можно использовать комптоновские единицы:  $l_C = \hbar/mc$ ,  $t_C = c/l_C$ ,  $p_C = mc$ ,  $E_C = m^2c^3/e\hbar$  (последняя величина называется критическим полем квантовой электродинамики, поле с такой напряжённостью совершает работу порядка  $mc^2$  над электроном при его пе-

ремещении на расстояние  $l_C$ ). Можно обезразмерить и каким-либо другим способом.

3. Напишите программу, которая решает систему уравнений (1) в постоянном скрещенном поле  $\mathbf{E} = (E, 0, 0)$ ,  $\mathbf{H} = (0, E, 0)$  для электрона с произвольными начальными условиями  $\mathbf{p}(t = 0) = \mathbf{p}_0$ ,  $\mathbf{x}(t = 0) = \mathbf{x}_0$ .
4. Постройте траектории движения частиц с помощью `matplotlib.pyplot`. В частности, постройте проекции на плоскости  $(x, y)$ ,  $(y, z)$  и  $(x, z)$ . Также попробуйте построить трехмерный график.
5. Для постоянного скрещенного поля известно точное решение (см. Ландау-Лифшиц, том 2, задача 2 к параграфу 22). Сравните численное решение с точным решением.

### 3 Моделирование движения частицы в произвольном электромагнитном поле с помощью ООП (6 баллов)

Расширьте возможности программы из предыдущего пункта так, чтобы в неё потенциально можно было заложить произвольную модель электромагнитного поля. Для реализации используйте принципы ООП.

1. Напишите программу, которая решает систему уравнений (1). Реализуйте задачу в виде двух классов: `ElectromagneticField` и `Particle`.
2. Класс `ElectromagneticField` должен описывать различные конфигурации электромагнитного поля (подразумевается, что поле может быть простым, вроде постоянного и однородного, а может быть и сложным, например, поле фокусированного лазерного импульса), и должен содержать метод, который возвращает значения электрического и магнитного полей в заданной точке  $(t, \mathbf{x})$ . Придумайте, как класс мог бы содержать в себе различные конфигурации поля. Конфигурация поля и его параметры должны задаваться при инициализации объекта. Для простоты начните с постоянного скрещенного поля:  $\mathbf{E} = (E, 0, 0)$ ,  $\mathbf{H} = (0, E, 0)$ .
3. Класс `ElectromagneticField` должен содержать в себе методы для графического построения распределения компонент электромагнитного поля и интенсивности  $I = (E^2 + H^2)/4\pi$  в заданных координатных плоскостях  $((x, t), (x, y), (y, z)$  и т.д., см. пример на Рис. 1) с использованием `matplotlib`. Напишите универсальный метод для построения распределения поля, вызывая который, пользователь мог бы выбирать координаты, в которых должен строиться

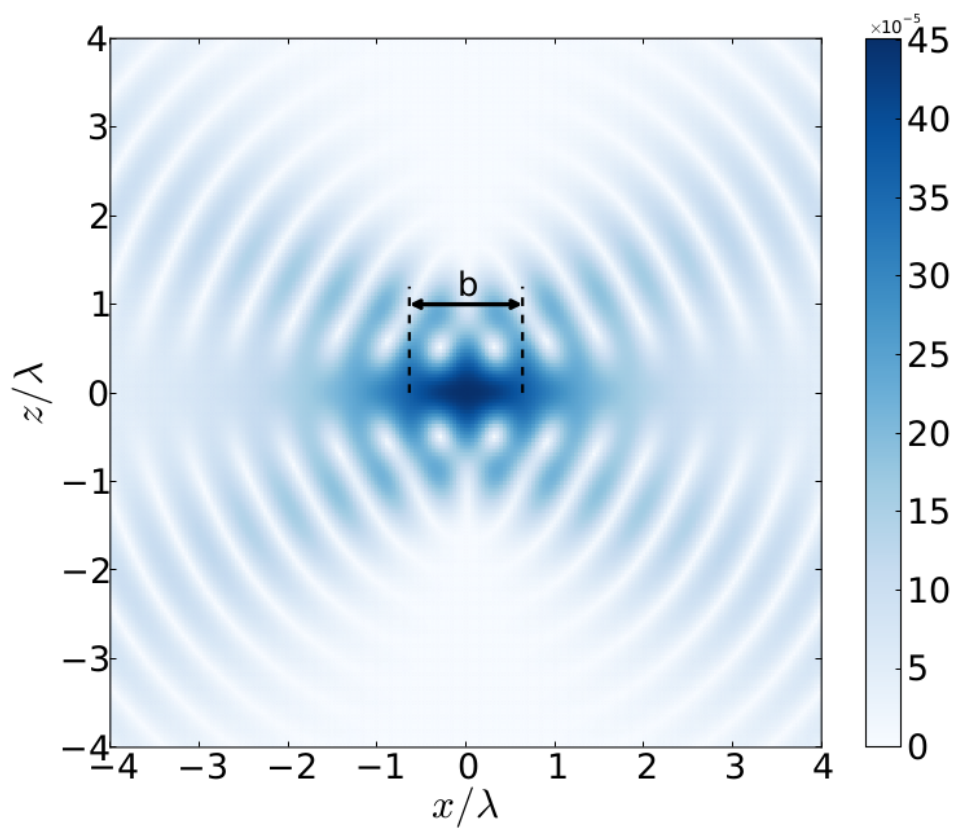


Рис. 1: Пример распределения интенсивности электромагнитного поля фокусированного лазерного импульса.

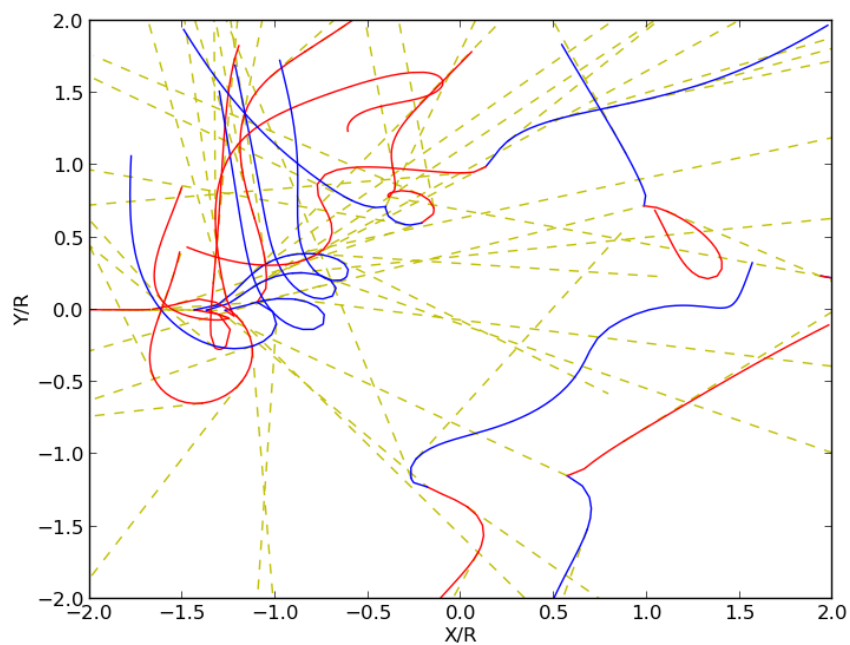
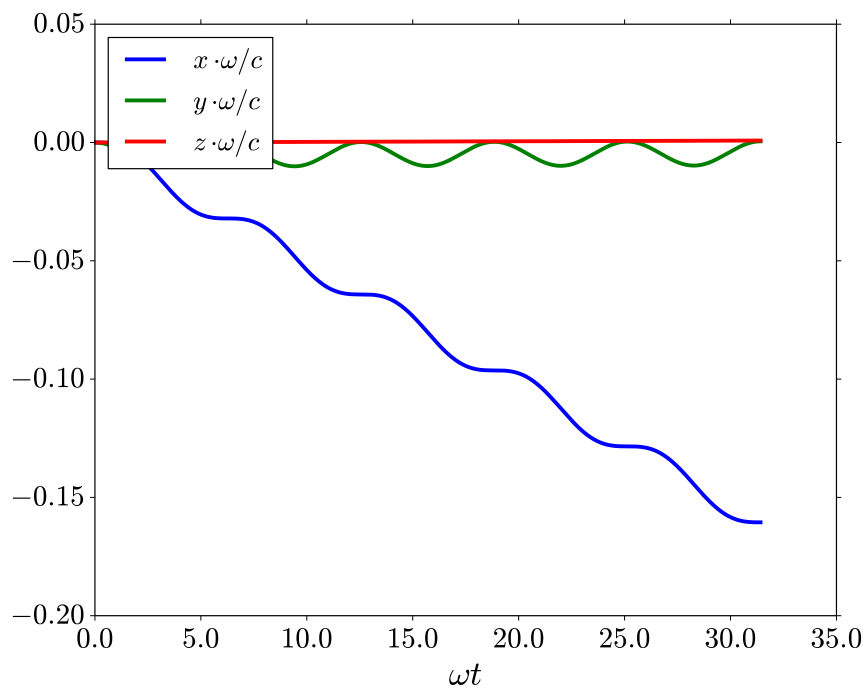


Рис. 2: Пример графиков для траекторий заряженных частиц: сверху — положение одной частицы в зависимости от времени; снизу — траектории частиц в плоскости  $(x, y)$ , движущихся в некотором поле (отличном от примера сверху).

график, а также фиксировать значение остальных координат (например, распределение поля в плоскости  $(x, y)$  при  $z = t = 0$ ). При этом на графике должны выводиться подписи к осям и вся необходимая информация для понимания, что на нём изображено.

4. Класс **Particle** описывает заряженную частицу. При инициализации объекта должны вводиться параметры частицы, её начальное положение, импульс и т.д. Класс должен содержать метод, который решает уравнения движения для данной частицы на заданной сетке по времени, и возвращает траекторию и значения импульса частицы на этой сетке. При решении уравнений метод должен обращаться к объекту из класса **ElectromagneticField** (объект можно передать как при инициализации частицы, так и при вызове метода для решения уравнений движения).
5. Класс **Particle** должен содержать в себе методы для построения графиков:
  - полученной траектории частицы в заданных координатных плоскостях  $((x, t), (x, y), (y, z))$  и т.д., в качестве примера см. Рис. 2),
  - трёхмерного графика траектории частицы,
  - графиков для компонент импульса частицы и её энергии в зависимости от времени.
6. Постройте траектории движения электрона в постоянном скрещенном электромагнитном поле, как в разделе 2.
7. Постройте траектории движения электрона в плоской волне циркулярной и линейной поляризации. Сделайте также возможность плавно включать и выключать поле (например, введите огибающую  $\exp(-t^2/2\tau^2)$ , где  $\tau$  — характерная длительность импульса. Рассмотрите также случай стоячей волны, получаемой при встречном распространении двух импульсов. Как меняется траектория частицы в зависимости от её начального положения?
8. (*\* задание на 3 дополнительных балла*) Придумайте способ, как построить распределение поля и траекторию частицы на одном графике. Возможно для этого вам захочется написать отдельный класс **Plotter**, который полностью возьмёт на себя функцию построения графиков (при этом соответствующие методы классов **ElectromagneticField** и **Particle** можно упростить, передав им объект-построитель **Plotter**).
9. (*\* задание на 2 дополнительных балла*) Постройте траектории движения электрона в поле фокусированного лазерного импульса, описываемого гауссовым пучком.