

# Проект по Физике - Магнетизм, 2022

Васильченко Роман, Киушкин Артем

Октябрь - Декабрь 2022

## 1 Аннотация

В данной статье рассказывается о проекте, который визуализирует линии векторов индукции магнитного поля. Описаны имеющиеся аналоги, используемые теоретические формулы и прикреплены скриншоты рабочего приложения. Также показано, где можно сказать и какие на данный момент имеются проблемы.

## 2 Введение

В качестве проекта мы решили разработать Desktop приложение, в котором пользователь будет иметь возможность добавлять / удалять / перемещать объекты, которые создают вектор силы тока на нас или против нас, что изменяет магнитное поле системы. На области экрана можно увидеть направление магнитных полей. Также можно навести на определенную точку и узнать точное значение модуля индукции магнитных полей, а также и другие данные, которые можно будет использовать для решения некоторых задач.

## 3 Имеющиеся аналоги

### 3.1 PhET

Визуализация магнитного поля в плоскости в составе проекта Университета Колорадо PhET Interactive Simulations. Наличие и направление магнитного поля показывают расставленные вдоль линий поля компасы. Величина магнитного поля измеряется с помощью перетаскивания инструмента внутри программы.

В программе присутствуют визуализации магнитных полей:

- постоянного магнита
- поля Земли
- соленоида с постоянным и переменным током

## 3.2 Falstad vector3dm

Визуализация магнитного поля в трехмерном пространстве. Наличие, направление и величину магнитного поля показывает симуляция частиц, движимых под воздействием поля.

В программе присутствуют всевозможные источники магнитного поля, такие как:

- источник однородного поля
- точечный заряд (движущийся и статичный)
- диполь (движущийся и статичный)
- соленоид (обычный и тороидальный)
- геометрические модели
  - прямая
  - кольцо
  - квадрат
  - сфера

## 3.3 ViziMag

Визуализация магнитного поля в плоскости. Данная программа является по сути конструктором системы источников магнитного поля, поэтому в ней можно изучать поведение магнитного поля в настраиваемой системе источников.

## 3.4 ELCUT

Программа для моделирования электромагнитных, тепловых и упругих полей методом конечных элементов. Платный программный продукт, в котором возможен анализ магнитного поля в плоскости.

# 4 Теория

## 4.1 Случай одного провода

В качестве основной формулы для определения направления векторов мы использовали закон Био-Савара-Лапласа для поля бесконечного прямого провода:

$$\vec{B} = \frac{\mu_0 I}{4\pi R^2} \vec{e}_\phi \quad (1)$$

$R_0$  - расстояние от рассматриваемой точки до проводника,  $I$  - сила тока в нем,  $\mu_0$  - магнитная постоянная,  $\vec{e}_\varphi$  - единичный вектор вдоль окружности, осью симметрии которой является провод:

$$\vec{e}_\varphi = \vec{e}_z \times \vec{e}_r$$

, где  $\vec{e}_z$  - единичный вектор вдоль провода, совпадающий с направлением тока, а  $\vec{e}_r$  - единичный вектор по направлению радиуса-вектора

Рассмотрим в ортогональной системе координат  $O_{xyz}$  создаваемое таким проводом поле в точке  $P(x, y, z)$ , причем  $\vec{e}_z \parallel \vec{z}$ , а ось провода проходит через  $(0,0)$ , тогда:

$$\left. \begin{array}{l} \vec{e}_z = \{0, 0, z\}, \quad z \in \{-1, 1\} \\ \vec{e}_r = \{x_0, y_0, 0\} \end{array} \right| \Rightarrow \vec{e}_\varphi = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 0 & 0 & z \\ x_0 & y_0 & 0 \end{vmatrix} = z(-y_0\vec{i} + x_0\vec{j}) = z\{-y_0, x_0, 0\}$$

Так как  $\vec{e}_r \uparrow \{x, y, 0\}$  (радиусу вектору), то

$$x_0 = |\vec{e}_r| \cos \angle(\vec{e}_r, Ox) = \cos \angle(\{x, y, 0\}, Ox) = \frac{x}{\sqrt{x^2 + y^2}}$$

$$y_0 = |\vec{e}_r| \sin \angle(\vec{e}_r, Ox) = \sin \angle(\{x, y, 0\}, Ox) = \frac{y}{\sqrt{x^2 + y^2}}$$

Итого:

$$\vec{B} = \frac{\mu_0 z I}{4\pi R^2} \left\{ -\frac{y}{\sqrt{x^2 + y^2}}, \frac{x}{\sqrt{x^2 + y^2}}, 0 \right\} \quad (2)$$

В нашей программе мы будем рассматривать поле в плоскости, перпендикулярной направлению тока, поэтому назовем ток втыкающимся в нас, если  $z > 0$ , и втыкающимся в плоскость, если  $z < 0$ . Проекция магнитной индукции на такую плоскость ( $z = Const$ ) по модулю будет равна (2)

## 4.2 Система проводов

До сих пор мы рассматривали систему из одного провода, поэтому перейдем к случаю двух и более компонентов. По принципу суперпозиции магнитная индукция данной системы в какой-то точке является суммой всех ее  $N$  компонентов:

$$\vec{B} = \sum_{i=1}^N B_i$$

Рассмотрим систему из  $N$  параллельных проводов в ортогональной системе координат  $O_{xyz}$  ( $\forall \vec{e}_{z_i} \parallel \vec{z}$ ) на плоскости  $z = Const$ . Тогда магнитную индукцию в точке  $(x_1, y_1, z)$  создаваемого проводом, ось которого проходит через  $(x_2, y_2, z)$ , можно рассмотреть как случай единичного провода (2) =  $\vec{B}(x = x_2 - x_1, y = y_2 - y_1, z = z)$

### 4.3 Поле внутри провода

Дополнительно рассмотрим магнитное поле внутри провода с известным радиусом  $R$ . Плотность тока в поперечном сечении такого провода:

$$J = \frac{I}{S} = \frac{I}{\pi R^2}$$

Окружность радиуса  $r < R$  с центром на оси провода - замкнутый контур, совпадающий с магнитной линией. Тогда по закону полного тока:

$$H = \frac{\sum I}{l_O} = \frac{JS}{2\pi r} = \frac{I\pi r^2}{\pi R^2 2\pi r} = \frac{I}{2\pi R^2} r$$

Тогда магнитная индукция на расстоянии  $r < R$ :

$$\vec{B} = \mu_0 H \vec{e}_\varphi = \frac{\mu_0 I}{2\pi R^2} r \vec{e}_\varphi$$

### 4.4 Фрагмент программного кода

```
#
class Wire(MagneticSource):
    def __init__(self, x, y, radius, electric):
        super().__init__(x, y)
        self.radius = radius # R
        self.electric = electric # z*I

    def vector_at_point(self, dx, dy, epsilon, gamma):
        dist = np.hypot(dx, dy)
        if dist < self.radius:
            H = self.electric / (2 * m.pi * self.radius ** 2) * dist
            B = gamma * H
        else:
            B = (gamma * self.electric) / (2 * m.pi * dist)
        return -B * (dy / dist), B * (dx / dist)

#
def field(self, X, Y):
    u, v = X.shape
    size = np.size(X)
    X.shape = (size)
    Y.shape = (size)
    Vx = np.zeros((u, v))
    Vy = np.zeros((u, v))

    for sources in self.sources:
        tX = [X[i] - sources.x for i in range(np.size(X))]
        tY = [Y[i] - sources.y for i in range(np.size(Y))]
        vectors = [sources.vector_at_point(tX[i], tY[i],
            self.epsilon, self.gamma) for i in range(size)]
        xv, yv = zip(*vectors)
        xv = np.array(xv, dtype=np.float)
        yv = np.array(yv, dtype=np.float)
        xv.shape = (u, v)
        yv.shape = (u, v)
        Vx = Vx + xv
        Vy = Vy + yv

    return Vx, Vy
```

## 5 Реализация и описание работы

В качестве языка программирования был выбран Python 3.10 ввиду его простоты, скорости написания программ, а также наличия удобных и полезных математических библиотек, которые и позволили реализовать линии магнитных полей.

Сам проект является open-source и расположен на сервисе Github, где любой желающий может посмотреть код и изменить его под себя. Последнюю версию проекта можно скачать по данной ссылке.

Рассмотрим само приложение. Как можно увидеть на Рис. 1, то основное место занимает центральный квадрат, где и отображаются линии векторов. Внизу можно заметить кнопки управления.

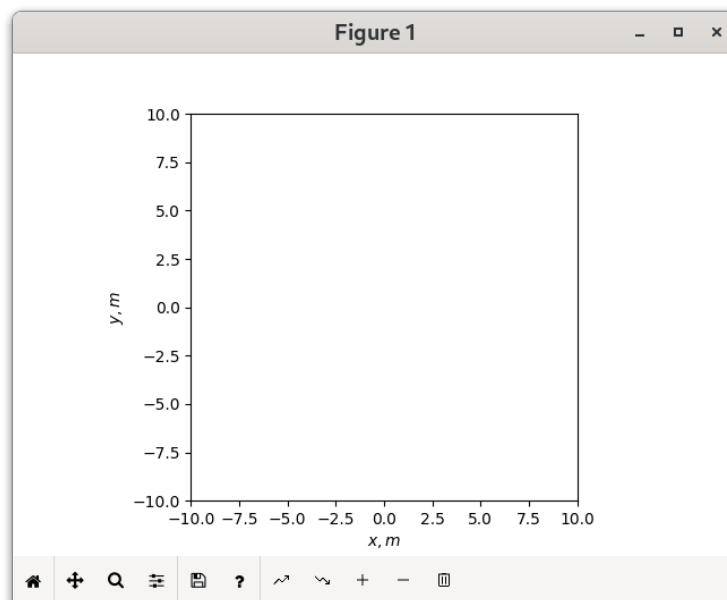


Рис. 1: Основное окно приложения

Если нажать на кнопку  $+$  и разместить 2 заряда, то получится что-то, что напоминает Рис. 2, а именно два объекта у которых, как можно заметить в правом нижнем углу электропроводимость 10См создают вокруг себя линии векторов индукции магнитного поля.

Для исследователей будет полезна функция на Рис. 3. При наведении курсора мыши на область в правом нижнем углу будет отображаться значение индукции магнитного поля, а также его координаты. Размеры поля ограничены по координатам  $(0, 0)$  и  $(10, 10)$  метров. В расчетах это можно учесть. В случае, если понадобится как-то изменить сам холст, то можно нажать на кнопку настройки (4 кнопка слева снизу) и откроется окно Рис. 4.

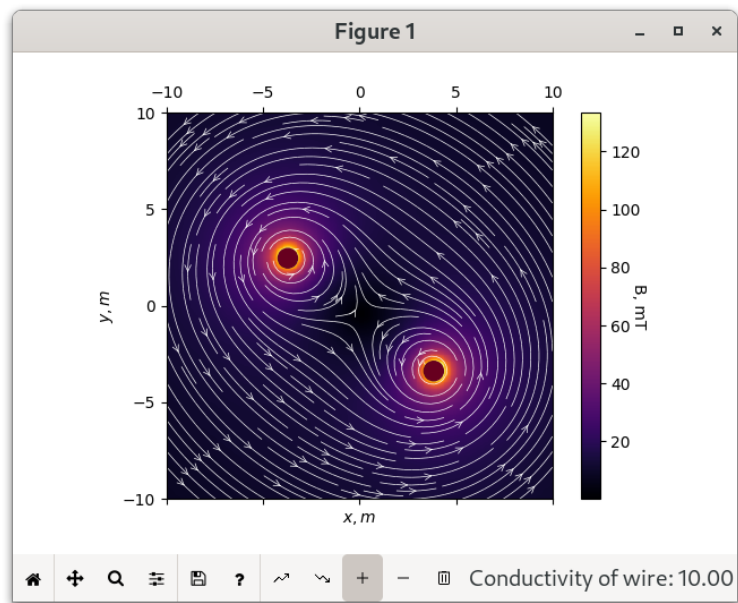


Рис. 2: Пример размещения зарядов

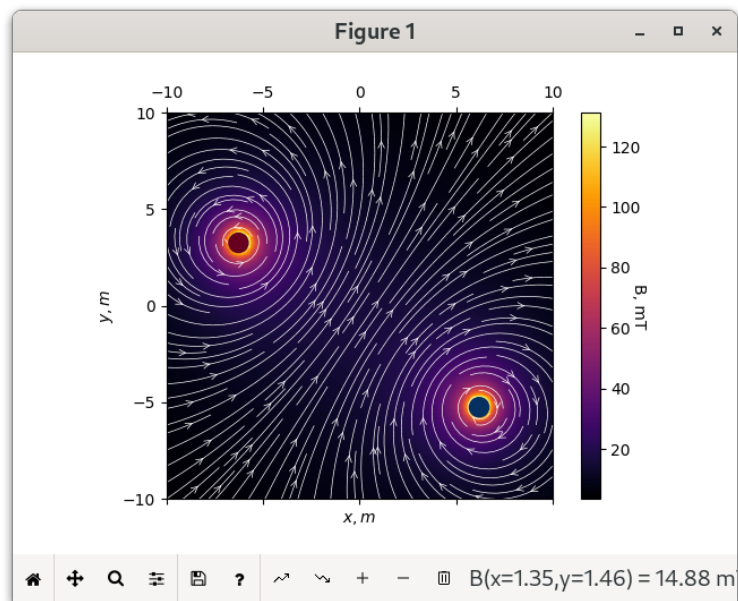


Рис. 3: Вычисление значение индукции магнитного поля

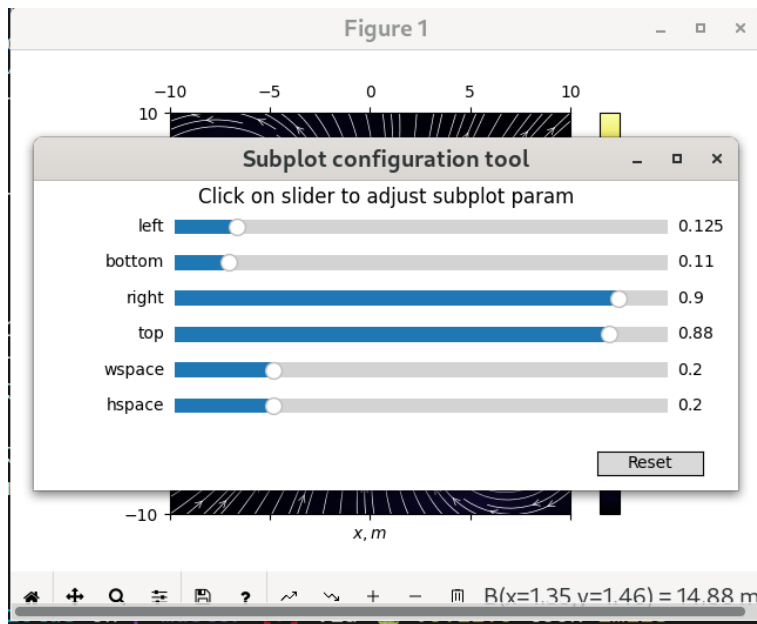


Рис. 4: Настройка холста

Одна из удобнейших функций является Лупа. На Рис. 5 можно ее увидеть и если в этом режиме выделить область, то размер холста уменьшится. Это можно использовать, если нужно брать расстояние не 10 метров, а меньше, что чаще всего и случается в задачах.

Для удобства конечного пользователя было создано отдельное окно, которое открывается при нажатии на кнопку ? и оно еще раз рассказывает о всей функциональности приложения, а также показывает удобные сочетания клавиш, которые позволят намного быстрее пользоваться приложением. Подробнее можно посмотреть на Рис. 6

## 6 Результаты и их обсуждения

Приложение показывает себя достаточно хорошо при различных сценариях использования и не дает "пустых" зон (зоны, где происходит ошибка вычисления и не отображаются вектора индукции магнитного поля). Из-за выбора языка программа начинает зависать при увеличении плотности отображения векторов, что негативно сказывается на производительности.

## 7 Выводы

Посмотрев на результаты можно сделать вывод, что программа достаточно точно показывает линии магнитных полей, а также модуль векторов индукции. Если не брать во внимание то, что она написана на языке Python, который не славится своей производительностью и вызванным из-за этого зависанием при большой плотности векторов, то приложение является удобным инструментом для визуализации и проверки результатов, полученных при расчетах.

## 8 Список литературы, проектов и источников

Chris Malley, Michael Dubson. **INTERACTIVE SIMULATIONS: Magnets and Electromagnets**, University of Colorado Boulder, 2020

Сивухин Д. В. **Общий курс физики.** — Изд. 4-е, стереотипное. М.: Физматлит, 2004.

## 9 Приложение

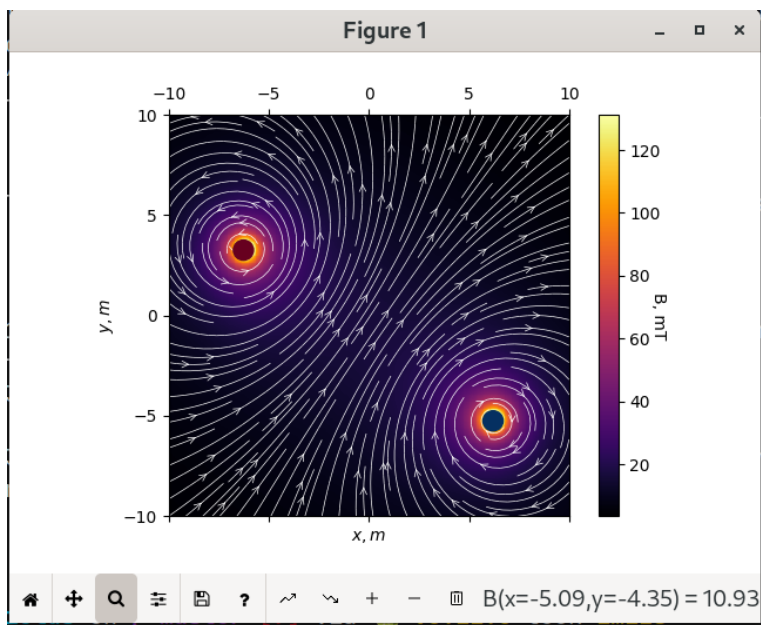


Рис. 5: Лупа: изменить масштаб



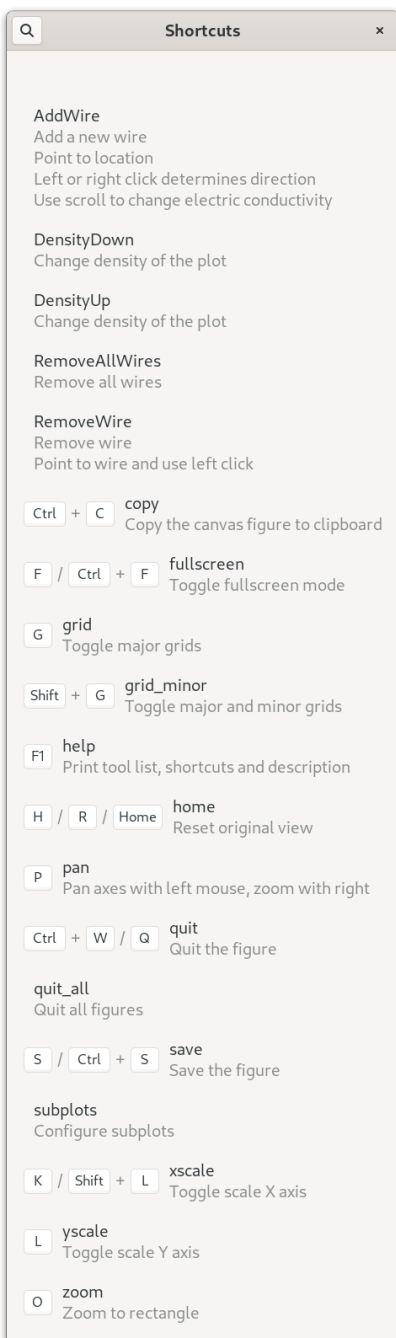


Рис. 6: Информация и сочетания клавиш