

1. Лабораторная работа №1

1.1. Цель лабораторной работы

Лабораторная проверяет умение пользоваться языком Python, библиотеками NumPy и Matplotlib для создания изображений различных математических объектов на плоскости: точек, отрезков, прямых, лучей, кривых, ломанных, многоугольников и т.д.

1.2. Задания

При выполнении всех заданий следует пользоваться объектно-ориентированным интерфейсом библиотеки Matplotlib. Также все изображения должны удовлетворять следующим требованиям:

- Разрешение изображения должно быть не менее 200dpi.
- На всех изображениях должно присутствовать координатная сетка.
- Оси координат должны быть подписаны.
- Формулы должны иметь «математический» вид, например x^2 , а не x^2 или $\sqrt{\sin x}$, а не $\text{sqrt}(\sin(x))$ и т.д.
- Если используются буквы древнегреческого или русского языков, то они также должны отображаться корректно, например μ , а не mu .

1.2.1. Задание №1

В библиотеке Matplotlib есть универсальная функция `plot`, с помощью которой можно отобразить точки, отрезки, лучи, прямые, ломанные, многоугольники, кривые и т.д. Однако, из-за ее универсальности делать это зачастую не удобно, поэтому лучше использовать специализированные функции.

- Задайте диапазон значений по осям Oxy как $[0, n] \times [0, n]$, где n — произвольное целое число.
- Отключите координатную сетку и с помощью функций `vlines` и `hlines` подмодуля `pyplot` нарисуйте квадратные клетки со стороной 1, замостив ими всю координатную плоскость.
- Настройте вид линий, изменив цвет, стиль (пунктир, штрихпунктир, штрих и т.д.), толщину и т.д.
- Изучите справку функций `axvline` и `axhline`. Чем они отличаются от предыдущих и можно ли с помощью них решить предыдущую задачу проще?
- Изучите справку функции `axline`. За что отвечает параметр `slope`? Используйте эту функцию, чтобы нарисовать обе диагонали всех получившихся квадратов.
- Программа должна работать так, чтобы правильно рисовать сетку при любом заданном целом n .

1.2.2. Задание №2

Для рисования векторов можно использовать несколько разных функций, например функцию `arrow` или `annotate`.

- Нарисуйте два произвольных вектора. Найдите ориентированный угол между ними. Используйте функцию `arctan2` из библиотеки NumPy или `atan2` из стандартного модуля `math`. Проверьте корректность вычислений на примерах острого, тупого и прямого углов.
- Примените к векторам такое линейное преобразование, чтобы ориентированный угол поменял свой знак.
- Нарисуйте то же самое переключившись в полярную систему координат.

1.2.3. Задание №3

Написать программу, которая рисует произвольный многоугольник по заданным координатам всех его вершин. Должна быть возможность задавать любое количество вершин в виде массива точек.

- Вычислите ориентированную площадь многоугольника. Проверьте корректность вычислений на примере правильного многоугольника, площадь которого можно вычислить теоретически.
- Как из получившегося массива выбрать координаты x всех вершин? Координаты y ? Как получить координаты (x, y) произвольной i -ой вершины?
- Для рисования многоугольника используйте функцию `Polygon` из подмодуля `Matplotlib.patches`, а для правильного многоугольника — функцию `RegularPolygon`.
- Выделите вершины многоугольника маркерными точками. Как настроить размер маркеров? Цвет? Форму?

1.2.4. Задание №4

Написать программу, которая рисует *правильный* многоугольник с центром в заданной точке, а также вписанную и описанную окружности этого многоугольника см. рис. 1. Количество сторон правильного n -угольника и длину его стороны нужно задать в виде параметра. Вместо длины стороны можно использовать радиус r вписанной или радиус R описанной окружностей.

- Все вершины многоугольника должны быть выделены маркерами и подписаны как P_1, P_2 и т.д. Нумерация точек должна осуществляться автоматически, в зависимости от n .
- Радиусы окружностей должны быть нарисованы в виде отрезков.
- Стороны многоугольника должны быть нарисованы более толстыми линиями, чем остальные элементы изображения.
- Описанная и вписанная окружности должны быть нарисованы разными цветами.
- Масштаб изображения не должен быть искажен (окружности не должны выглядеть как эллипсы).
- Картинка должна корректно перерисовываться при изменении n .

1.2.5. Задание №5

Нарисуйте график циклоиды для двух полных оборотов окружности. Отобразите поле касательных векторов к данному графику, см рис. 2. Уравнение циклоиды:

$$\mathbf{r}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{cases} r(t - \sin t), \\ r(1 - \cos t). \end{cases}$$

Для визуализации поля векторов можно использовать функцию `quiver`

1.2.6. Задание №6

Нарисовать астроиду и ее эволюту. Как выглядит уравнение астроиды, и что такое эволюта вы проходили в курсе дифференциальной геометрии. Вычислите радиус кривизны в произвольной точке астроиды и нарисуйте соприкасающуюся окружность в данной точке, отобразив саму окружность, ее центр и радиус кривизны.

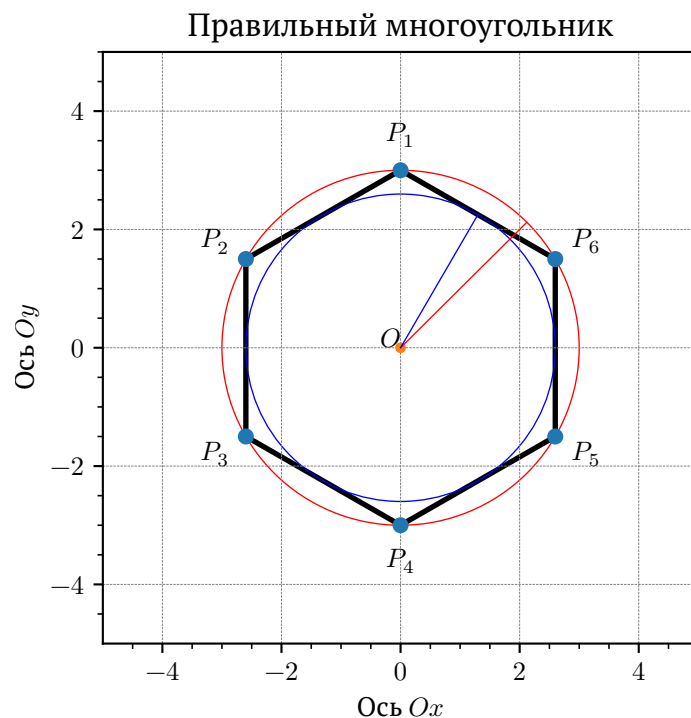


Рис. 1: Правильный шестиугольник. Пример выполнения задания 4

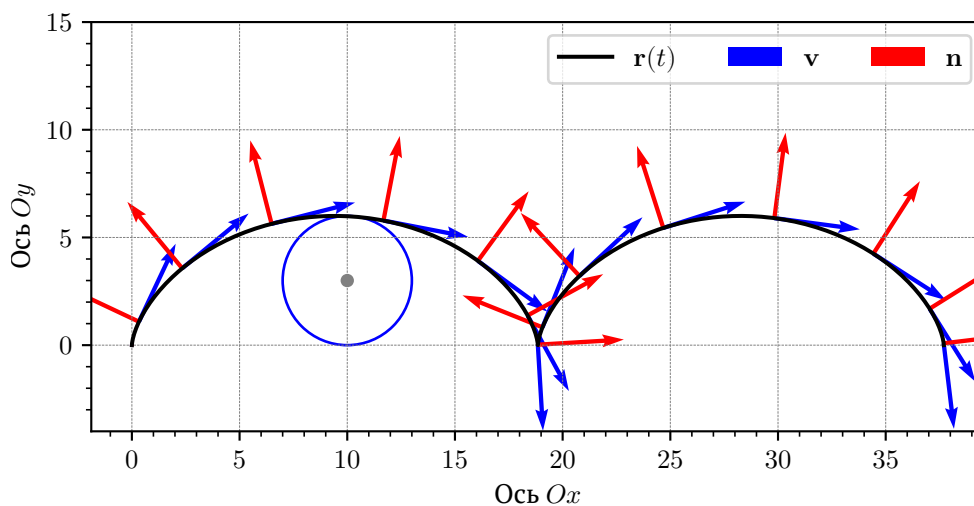


Рис. 2: Циклоида с окружностью и ненормированными направляющими векторами касательных и нормалей в некоторых точках. Пример выполнения задания 5

1.2.7. Задание №7

Частным случаем суперформулы является *суперэллипс* или *кривая Ламе*. В неявном виде в декартовых координатах он задается уравнением:

$$\left| \frac{x}{a} \right|^n + \left| \frac{y}{b} \right|^n = 1, \quad n, a, b > 0.$$

Параметрическое представление:

$$x(t) = \pm a \cos^{\frac{2}{n}} t,$$

$$y(t) = \pm b \sin^{\frac{2}{n}} t,$$

где параметр $t \in [0, \pi/2)$. Постройте суперэллипс, его нормальные векторы $\mathbf{n}(t)$ и кривую $k(t) \cdot \mathbf{n}(t)$, где $k(t)$ — кривизна в точке $(x(t), y(t))$. Пример изображения см. рис. 3.

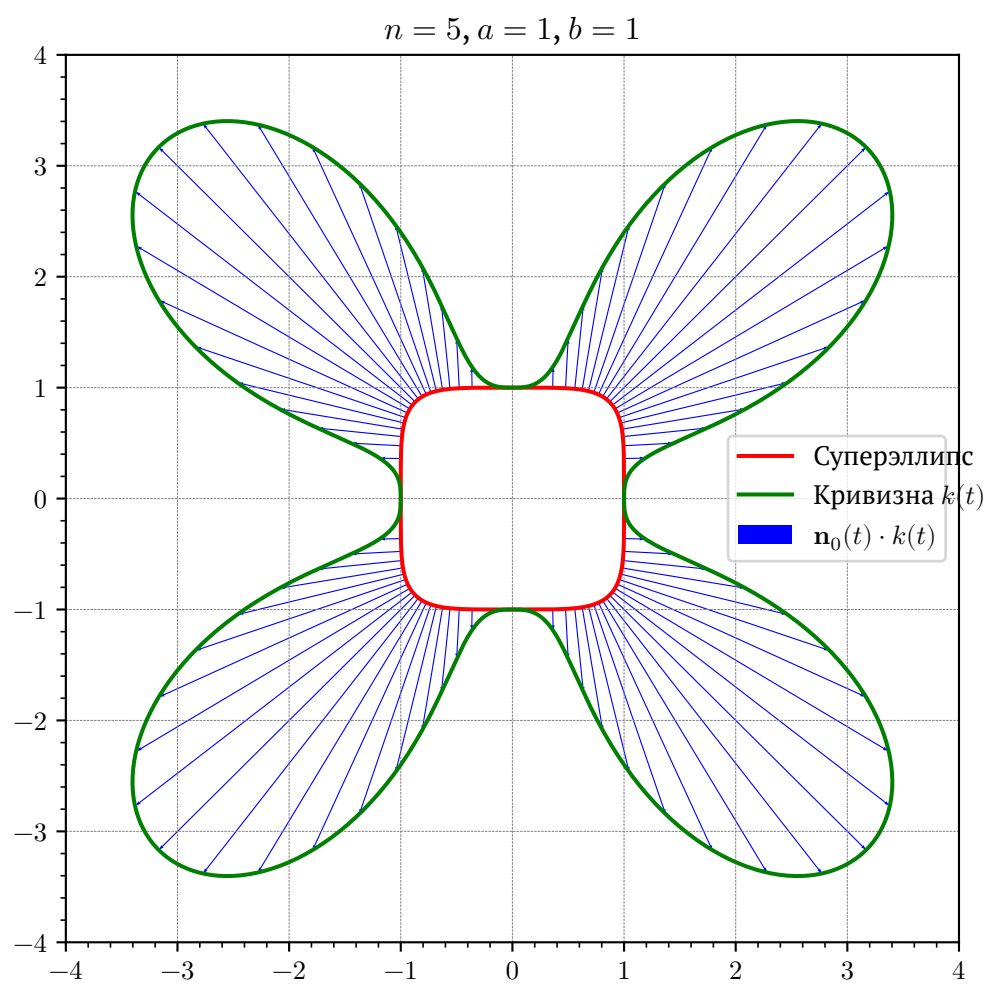


Рис. 3: Суперэллипс с нормальными векторами и кривизной. Пример выполнения задания 7