

Рисование векторов в matplotlib на языке программирования Python

Кизянов Антон Олегович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Создание векторных графиков в matplotlib с помощью языка программирования Python.

Ключевые слова: Python, matplotlib

Drawing vectors in matplotlib in the Python programming language

Kizyanov Anton Olegovich

Sholom-Aleichem Priamursky State University

student

Abstract

Create vector graphics in matplotlib using the Python programming language.

Keywords: Python, matplotlib

Векторные графики используются для визуализации потока в векторных полях. Примеры из науки и природы включают области магнитных и гравитационных сил или движения жидких материалов.

Цель исследования – это написание программы для рисования векторных графиков на языке программирования Python.

Ранее этим вопросом интересовались В.Е. Томин, Д.Ю. Колобов, А.В. Киселев, развивали тему «Обработка данных солнечных телескопов в процессе наблюдений» [1], в которой рассказывается насколько анализ данных астрофизического эксперимента тесно связан с рядом технических задач. А.В. Петрухин, А.С. Стешенко с темой «Компьютерная визуализация биржевых данных о динамике фондового рынка» [2], а подробнее про эффективные построения модулей визуализации данных о динамике фондового рынка. В.В. Волков опубликовал статью «Использование программных средств поддержки планирования эксперимента для оптимизации параметров генетического алгоритма аппроксимации» [3], рассказал как описывается экспериментальная оптимизация скорости сходимости генетического алгоритма аппроксимации, выполненная с использованием специализированных научных программных средств поддержки планирования эксперимента.

Векторное поле можно визуализировать таким образом, где мы назначаем линию и одну или несколько стрелок для каждой точки. Интенсивность может быть представлена длиной линии, а направление -

стрелкой, указывающей в определенном направлении. Обычно интенсивность силы визуализируется с длиной определенной линии тока, но плотность может также использоваться для той же цели. Чтобы визуализировать векторные поля, мы будем использовать функцию `matplotlib.pyplot.streamplot`. Эта функция создает графики из линий.

Наиболее важными аргументами для этой функции являются `x`, `y` равномерно распределенная сетка одномерного массива NumPy и `u`, `v`, соответствующие двумерным массивам NumPy скоростей `x`, `y`. Матрицы `u` и `v` должны иметь такие размеры, чтобы количество строк должно быть одинаковой длины `y`, а число столбцов должно соответствовать длине `x`.

Стрелки используются для указания направления вектора, и мы можем управлять ими, используя два параметра: размер стрелки и стрелочный формат стрелки.

Выполните следующие действия:

- Создайте поле векторов
- Распечатайте промежуточные значения
- Постройте график векторов
- Отобразите фигуру с линиями, визуализирующие наши векторы

Вот сам код:

```
import matplotlib.pyplot as plt
from pprint import pprint as pr
import numpy as np

Y_c, X_c = np.mgrid[0:5:100j, 0:5:100j]

U_c = np.sin(X_c)
V_c = Y_c

print("X")
pr(X_c)

print("Y")
pr(Y_c)

plt.streamplot(X_c, Y_c, U_c, V_c, color='red')
plt.show()
```

Он выведет нам промежуточные данные точек как на рисунке 1.

```

X
array([[ 0.          ,  0.05050505,  0.1010101 , ...,  4.8989899 ,
        4.94949495,  5.          ],
       [ 0.          ,  0.05050505,  0.1010101 , ...,  4.8989899 ,
        4.94949495,  5.          ],
       [ 0.          ,  0.05050505,  0.1010101 , ...,  4.8989899 ,
        4.94949495,  5.          ],
       ...,
       [ 0.          ,  0.05050505,  0.1010101 , ...,  4.8989899 ,
        4.94949495,  5.          ],
       [ 0.          ,  0.05050505,  0.1010101 , ...,  4.8989899 ,
        4.94949495,  5.          ],
       [ 0.          ,  0.05050505,  0.1010101 , ...,  4.8989899 ,
        4.94949495,  5.          ]])

Y
array([[ 0.          ,  0.          ,  0.          , ...,  0.          ,
        0.          ,  0.          ],
       [ 0.05050505,  0.05050505,  0.05050505, ...,  0.05050505,
        0.05050505,  0.05050505],
       [ 0.1010101 ,  0.1010101 ,  0.1010101 , ...,  0.1010101 ,
        0.1010101 ,  0.1010101 ],
       ...,
       [ 4.8989899 ,  4.8989899 ,  4.8989899 , ...,  4.8989899 ,
        4.8989899 ,  4.8989899 ],
       [ 4.94949495,  4.94949495,  4.94949495, ...,  4.94949495,
        4.94949495,  4.94949495],
       [ 5.          ,  5.          ,  5.          , ...,  5.          ,
        5.          ,  5.          ]])
    
```

Рис. 1

График, построенный по точкам, будет выглядеть как на рисунке 2.

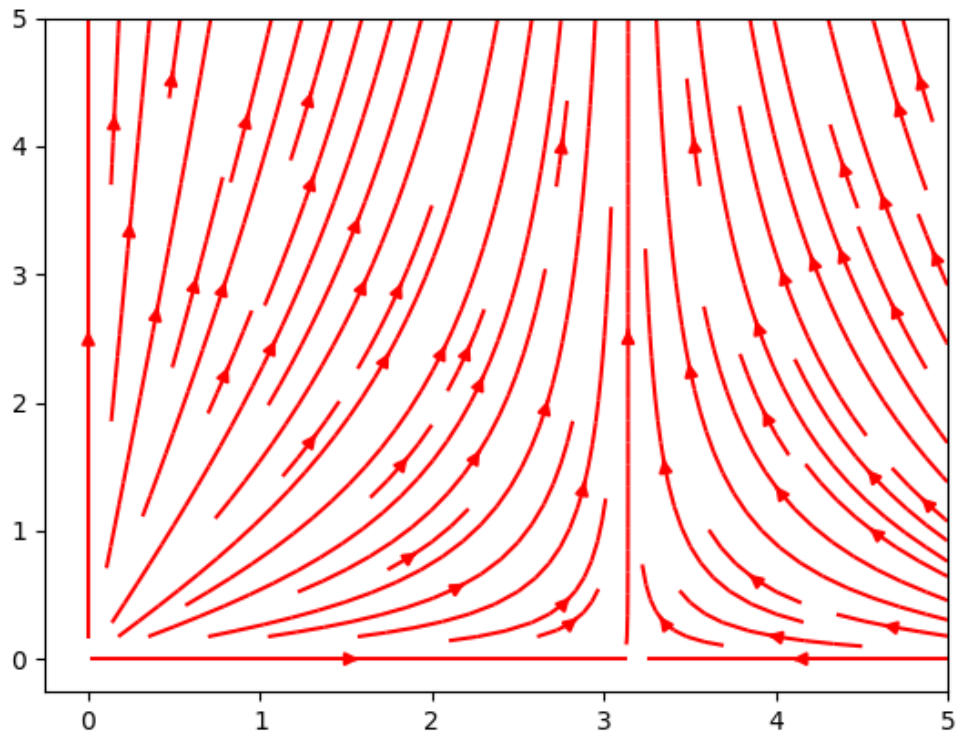


Рис. 2

Мы создаем векторное поле X и Y , индексируя двумерную сетку, используя экземпляр `mgrid NumPy`. Мы определяем диапазон сетки как старт и стоп (-2 и 2 соответственно). Третий индекс представляет собой длину шага. Длина шага - это количество точек, которые должны включать между началом и остановкой.

Вывод

Таким образом, мы можем визуализировать данные магнитных волн и других подобных данных на графике.

Библиографический список

1. Томин В.Е., Колобов Д.Ю., Киселев А.В. Обработка данных солнечных телескопов в процессе наблюдений // Труды Международной Байкальской молодежной научной школы по фундаментальной физике и конференции молодых ученых «Взаимодействие полей и излучения с веществом». 2013. С. 115-117. Ссылка <https://elibrary.ru/item.asp?id=26227260> (Дата обращения: 16.08.2017)
2. Петрухин А.В., Стешенко А.С. Компьютерная визуализация биржевых данных о динамике фондового рынка // Известия Волгоградского государственного технического университета 2015. №6. С. 124-129. Ссылка <https://elibrary.ru/item.asp?id=24334292> (Дата обращения: 16.08.2017)
3. Волков В.В. Использование программных средств поддержки планирования эксперимента для оптимизации параметров генетического алгоритма аппроксимации // Вестник Донского государственного технического университета. 2010. №10. С. 487-491. Ссылка <https://elibrary.ru/item.asp?id=15522965> (Дата обращения: 16.08.2017)