

## Создание анимации в matplotlib на языке программирования Python

*Кизянов Антон Олегович*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

Создание анимированных графиков в matplotlib с помощью языка программирования Python.

**Ключевые слова:** Python, matplotlib

## Creating animations in matplotlib in the Python programming language

*Kizyanov Anton Olegovich*

*Sholom-Aleichem Priamursky State University*

*student*

### Abstract

Create animated graphics in matplotlib using the Python programming language.

**Keywords:** Python, matplotlib

В математике бывает сложно объяснить, как движутся некоторые графики и изменяются. На помощь приходит анимация. Ведь на анимации проще объяснить, как движется график и что произойдет, если мы изменим значения переменных.

Цель исследования – это написание программы для построения анимированных 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.

Фреймворк для анимации добавлен в библиотеку matplotlib с версии 1.1, а его основным классом является matplotlib.animation.Animation. Этот

класс является базовым классом, который имеет подклассы поведения: TimedAnimation, ArtistAnimation и FuncAnimation.

Таблица 1 – Описание классов

<i>Название класса</i>	<i>Описание</i>
Animation (object)	Этот главный класс для создания анимации с использованием matplotlib.
TimedAnimation (Animation)	Подкласс Animation поддерживает анимацию, основанную на времени, и создает новый кадр каждые несколько миллисекунд.
ArtistAnimation (TimedAnimation)	Класс который объединяет холст для анимации и сами графики.
FuncAnimation (TimedAnimation)	Делает анимацию, повторно вызывая функцию, передавая (необязательные) аргументы.

Чтобы иметь возможность сохранять анимацию в видеофайле, должен быть установщик ffmpeg или mencoder.

Вот код, которые создает файлы с анимацией в matplotlib.

```
import numpy as nmp
from matplotlib import pyplot as pmlt
from matplotlib import animation as anm

fg = pmlt.figure()
a_x = pmlt.axes(xlim=(0, 2), ylim=(-2, 2))
lin, = a_x.plot([], [], lw=2)

def initialized():
    lin.set_data([], [])
    return lin,

def animated(a):
    _x = nmp.linspace(0, 2, 1000)
    _y = nmp.sin(4 * nmp.pi * (_x - 0.01 * a)) * nmp.cos(22 * nmp.pi * (_x - 0.01 * a))
    lin.set_data(_x, _y)
    return lin,

anmtor = anm.FuncAnimation(fg, animated, init_func=initialized, frames=200,
interval=20, blit=True)

anmtor.save('animete.mp4', fps=29, extra_args=['-vcodec', 'libx264'],
writer='ffmpeg_file')
pmlt.show()
```

Это создаст файл `basic_animation.mp4` в папке, в которой был запущен этот файл, а также отобразится окно с запущенной анимацией. Видеофайл можно открыть большинством современных видеоплееров, поддерживающих формат MPEG-4. Анимация должна выглядеть так Рис. 1.

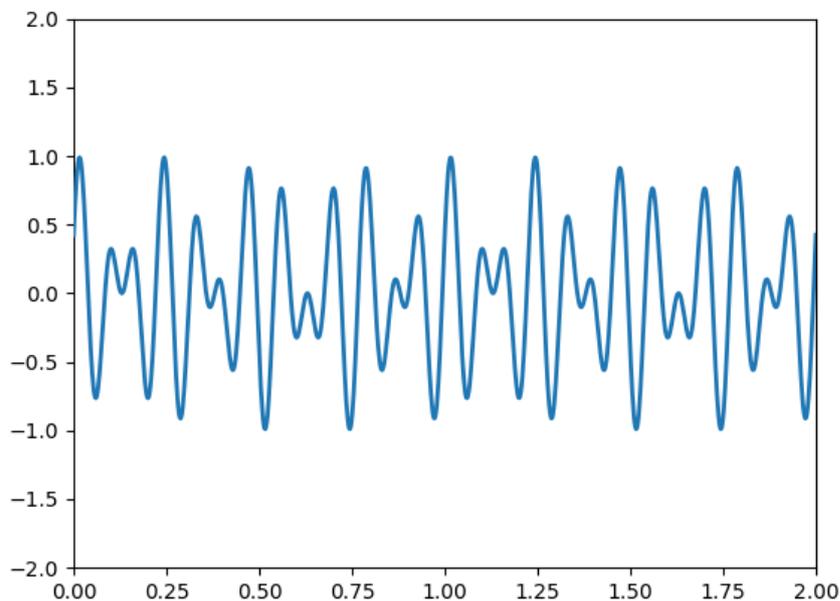


Рис. 1

#### Вывод

Таким образом, мы можем генерировать неограниченное количество файлов с анимацией с разными графиками.

#### 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. URL: <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. Т.10. №4. С. 487-491. URL: <https://elibrary.ru/item.asp?id=15522965> (Дата обращения: 16.08.2017)