

Анимация в Python

Анимация

Анимация — это последовательность изображений, которые быстро сменяют друг друга, в результате чего появляется движение.

Для ее создания в matplotlib предназначен модуль **matplotlib.animation**

Две функции этого модуля:

FuncAnimation и **ArtistAnimation**

Общий вид:

`matplotlib.animation.FuncAnimation(...)`

`FuncAnimation(fig, func, frames=count
[,init_func=None, fargs=None, interval=ms,
repeat=True,...])`

Анимация создается путем многократного вызова функции `func(...)`, которая должна рисовать кадры.

В простейшем случае опция `frames` задает количество кадров, а функция `func(номер_кадра[,fargs])` в качестве первого аргумента принимает номер кадра.

- Опция `interval` задает время в миллисекундах между кадрами.
- Опция `repeat=True` (значение по умолчанию) включает режим повторения анимации.
- В этом случае функция `funcname` будет вызываться в начале каждой «анимационной сессии».

Вначале загружаются необходимые модули и создается графическое окно `fig` с графической областью `ax`.

Затем рисуется «пустая» ломаная.

Она не имеет данных, но для нее сразу устанавливается толщина линии.

Функция `plot` возвращает объект этой линии `line` класса `Line2D`.

Напомним, что если кривых строится несколько, то функция `plot` возвращает кортеж (или список) таких объектов, например, `line1, line2 = plot(x1, y1, x2, y2)`

- Если возвращаемый объект один, то в инструкции `line, = ax.plot(...)` после имени `line` нужно ставить запятую (признак кортежа).
- Используя методы `set_linestyle()`, `set_marker()`, `set_drawstyle()` объекта `line`, можно менять стили оформления кривой и маркеров.

- Функция `redraw(i)` рисует i – й кадр.
- Для рисования кривой используется объект `line`, созданный ранее.
- У него есть метод `line.set_data(x,y)`, который меняет его данные (массивы x и y координат вершин ломаной).

- Функция `animation.FuncAnimation(...)` первым аргументом принимает графическое окно `fig`, в котором создается анимация.
- Второй аргумент является именем функции `redraw`, которая рисует кадры.
- Затем мы указываем количество кадров и интервал времени между ними.

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
fig = plt.figure(facecolor='white')
ax = plt.axes(xlim=(0, 8), ylim=(-1, 1) )
line, = ax.plot([ ], [ ], lw=3) # line = объект кривой
ax.grid(True)
def redraw(i):
    x = np.linspace(0, 8, 200)
    y = np.sin(i * x/10)/(1+x**2)
    line.set_data(x, y)
anim = animation.FuncAnimation(fig, redraw, frames=100, interval=50)
plt.show()
```

Figure 1

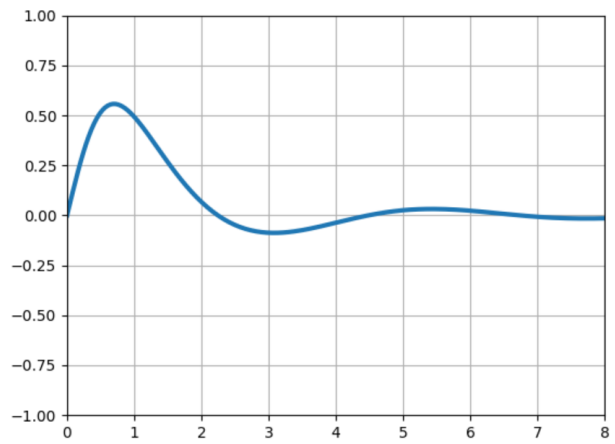
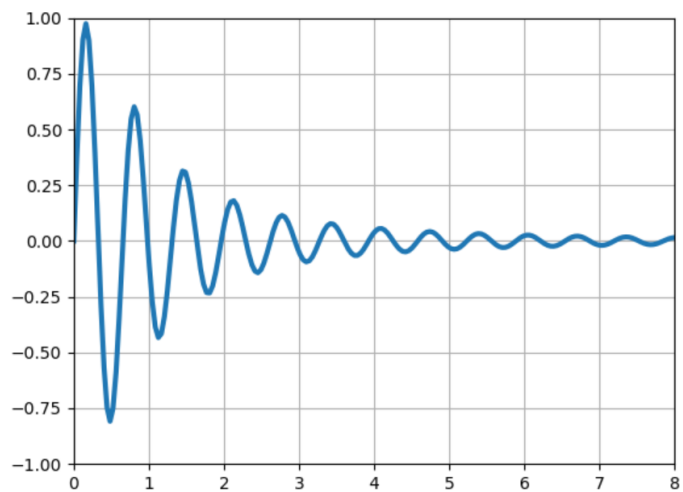
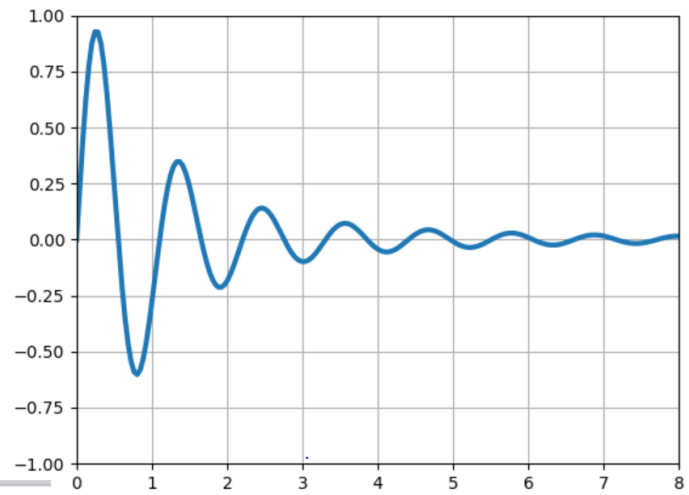


Figure 1

Figure 1



Задаем масштаб осей

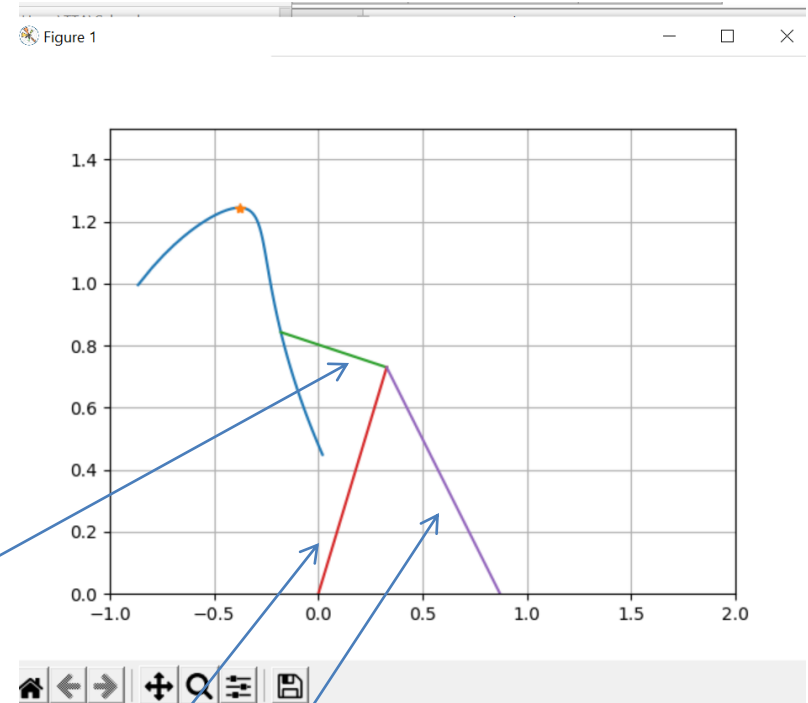
```
ax = plt.axes(xlim=(-1, 2), ylim=(0, 1.5) )
```

Несколько объектов можно создать

```
lin1, = ax.plot([ ], [ ]) 
```

```
lin2, = ax.plot([ ], [ ]) 
```

```
lin3, = ax.plot([ ], [ ]) 
```



- **def** redraw(i):
 x = XC[i]
 y = YC[i]
 x1=XA[i]
 y1=YA[i]
 x2=S1[i]
 lin1.set_data([x,x1], [y,y1])
 lin2.set_data([0, x1], [0, y1])
 lin3.set_data([x2, x1], [0, y1])
 return lin1,lin2,lin3

Формируем кадры анимации

```
anim=animation.FuncAnimation(fig,redraw,  
frames=200,interval=50)
```