

# Лабораторная работа №7. Модель эффективности релкамы

с/б 1032186063 | НФИбд-01-18

---

Доборщук Владимир Владимирович

27 марта 2021

RUDN University, Moscow, Russian Federation

# Цели и задачи

---

Изучить модель эффективности рекламы, а также реализовать её программно.

- изучить теорию о модели эффективности рекламы
- построить модель для 3 различных случаев  $\alpha_1(t)$  и  $\alpha_2(t)$

# **Ход выполнения лабораторной работы**

---

Считаем, что  $\frac{dn}{dt}$  - скорость изменения со временем числа потребителей, узнавших о товаре и готовых его купить,  $t$  - время, прошедшее с начала рекламной кампании,  $n(t)$  - число уже информированных клиентов. Эта величина пропорциональна числу покупателей, еще не знающих о нем, это описывается следующим образом:

$\alpha_1(t)(N - n(t))$ , где  $N$  - общее число потенциальных платежеспособных покупателей,  $\alpha_1(t) \geq 0$  - характеризует интенсивность рекламной кампании (зависит от затрат на рекламу в данный момент времени).

Помимо этого, узнавшие о товаре потребители также распространяют полученную информацию среди потенциальных покупателей, не знающих о нем (в этом случае работает т.н. сарафанное радио). Этот вклад в рекламу описывается величиной  $\alpha_2(t)n(t)(N - n(t))$ , эта величина увеличивается с увеличением потребителей узнавших о товаре. Математическая модель распространения рекламы описывается уравнением:

$$\frac{dn}{dt} = (\alpha_1(t) + \alpha_2(t)n(t))(N - n(t))$$

При  $\alpha_1(t) \gg \alpha_2(t)$  получается модель типа модели Мальтуса, В обратном случае, при  $\alpha_1(t) \ll \alpha_2(t)$  получаем уравнение логистической кривой.



## Инициализация библиотек

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.integrate import odeint
4 from math import sin
5 from scipy.misc import derivative
6
7 from jupyterthemes import jtplot
8 jtplot.style(context='notebook', fscale=1.2,
               gridlines='—')
```

Введём соответствующие нашему варианту начальные данные для построения модели:

```
1  N = 648
2  t0 = 0
3  N0 = 12
4
5  def k(t):
6      return 0.125
7
8  def p(t):
9      return 0.00002
10
11 t = np.arange(0, 30, 0.01)
```

Создадим функцию для нашей СДУ:

```
1 def XD(x, t):  
2     xd = (k(t) + p(t)*x)*(N-x)  
3     return xd
```

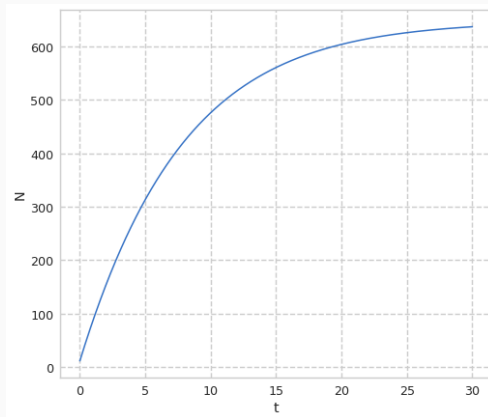
Воспользуемся функцией `odeint` из модуля `scipy.integrate` и решим нашу СДУ.

```
1 x = odeint(XD, NO, t)
```

# Модель 1

```
1 plt.plot(t, x)
2 plt.ylabel('N')
3 plt.xlabel('t')
4 plt.show()
```

# Модель 1

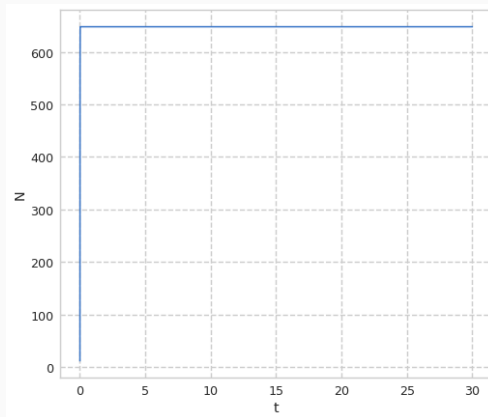


**Рис. 1:** График для модели 1

## Модель 2

```
1 def k(t):  
2     return 0.000095  
3  
4 def p(t):  
5     return 0.92  
6  
7 x = odeint(XD, N0, t)  
8  
9 plt.plot(t, x)  
10 plt.ylabel('N')  
11 plt.xlabel('t')  
12 plt.show()
```

## Модель 2



**Рис. 2:** График для модели 2



## Модель 2

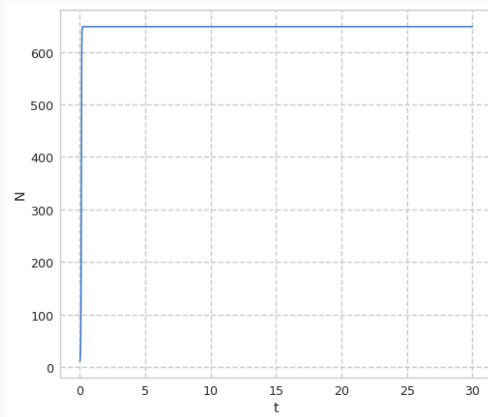
Найдем момент времени, при котором скорость изменения числа потребителей.

```
1 diff = 0
2 ind = -1
3 for i in range(1, len(x[:,0])):
4     if (x[i][0] - x[i-1][0]) > diff:
5         diff = x[i][0] - x[i-1][0]
6         ind = i
```

Максимальная скорость изменения числа потребителей будет при  $t=0.01$

## Модель 3

```
1 def k(t):  
2     return sin(10*t)  
3  
4 def p(t):  
5     return 0.9*t  
6  
7 x = odeint(XD, NO, t)  
8  
9 plt.plot(t, x)  
10 plt.ylabel('N')  
11 plt.xlabel('t')  
12 plt.show()
```



**Рис. 3:** График для модели 3

## Выводы

---

Мы изучили простейшую модель эффективности рекламы, после чего успешно реализовали её с помощью языка Python и дополняющих его модулей.