

## 6 вариант. Копылов В.Я. № В3530904/00022

### Начальные условия

Исходное уравнение:  $W(s) = \frac{k(1+a_1*s)}{1+2*\mu*T*s+T^2*s^2} = \frac{\bar{y}(s)}{\bar{x}(s)} = \frac{P(s)}{Q(s)}$  где:  
 $x(t) = const = 12, k = 2, a_1 = 2, \mu = 0.3, T = 2$

### Введем дополнительную переменную и создадим систему

$$W(s) = W_1(s) * W_2(s) = \frac{1}{Q(s)} * P(s) = \frac{1}{\bar{x}(s)} * \bar{y}(s)$$

Далее выразим  $\bar{x}(s)$  и  $\bar{y}(s)$  в единой системе:

$$\bar{x}(s) = (1 + 2 * \mu * T * s + T^2 * s^2) * \bar{U}(s)$$

$$\bar{y}(s) = (k + k * a_1 * s) * \bar{U}(s)$$

### Переход в вещественную форму

$$\bar{x}(s) = x(t)$$

$$\bar{y}(s) = y(t)$$

$$\bar{U}(s) = U(t)$$

$$S^n \rightarrow \frac{d^n}{dt^n}$$

### Преобразуем систему

$$x(t) = U(t) + 2 * \mu * T * U^{(1)}(t) + T^2 * U^{(2)}(t)$$

$$y(t) = k * U(t) + k * a_1 * U^{(1)}(t)$$

### Выразим из уравнения с $x(t)$ старшую производную

$$U^{(2)}(t) = -\left(\frac{U(t) + 2 * \mu * T * U^{(1)}(t) - x(t)}{T^2}\right)$$

## Приведем систему к канонической форме для применения метода Эйлера

сперва обозначим:

$$z_1(t) = U(t)$$

$$z_2(t) = U^{(1)}(t) = z_1^{(1)}(t)$$

$$z_3(t) = U^{(2)}(t) = z_2^{(1)}(t)$$

Получим систему уравнений:

$$z_1^{(1)}(t) = z_2(t)$$

$$z_2^{(1)}(t) = z_3(t)$$

$$z_3(t) = -\left(\frac{z_1(t) + 2 * \mu * T * z_2(t) - x(t)}{T^2}\right)$$

Выразим уравнение для  $y$ :

$$y(t) = k * z_1(t) + k * a_1 * z_2(t)$$

## Проверка вычисления точки сходимости

исходный предел для проверки имеет вид:

$$\lim_{s \rightarrow 0} y(t) = \lim_{s \rightarrow 0} s * y(s)$$

После множества преобразований получаем:

$$c * \lim_{s \rightarrow 0} W(s)$$

Вычислим предел:

$$c * \lim_{s \rightarrow 0} W(s) = c * \lim_{s \rightarrow 0} \frac{k(1 + a_1 * s)}{1 + 2 * \mu * T * s + T^2 * s^2} = c * \lim_{s \rightarrow 0} \frac{k}{1} = c * k = 12 * 2 = 24$$

## Далее приступим к вычислению точек и построению графика

```
In [165]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [166]: # начальные условия
xt, k, a1, mu, t = 12, 2, 2, 0.3, 2
```

```
In [167]: # функция вычисления точек
def gen_Euler(h: float) -> float:
    """
    h: float -> шаг
    return: значение функции
    """
    z1, z2 = 0,0
    while True:
        yield (k * z1 + k * a1 * z2)
        z1 = z1 + h * z2
        z2 = z2 + h * -((z1 + 2 * mu * T * z2 - xt) / (T*T))
```

```
In [179]: h = 0.005 # шаг с которым функция будет вычислять значения
f = gen_Euler(h)

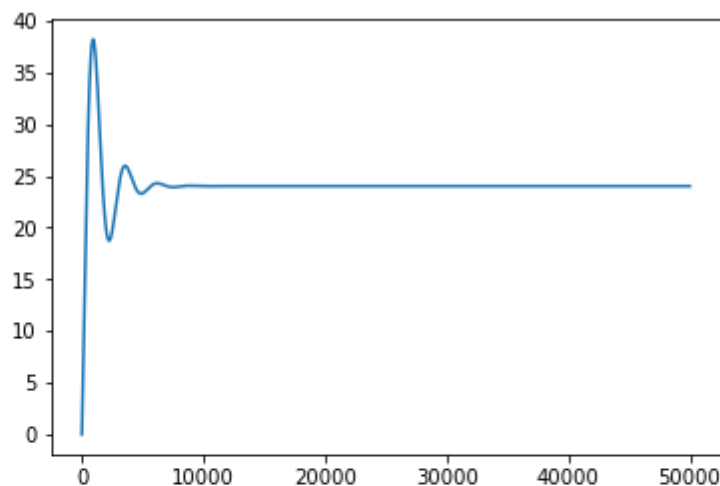
border = 50000 # кол-во точек
dots = []
for _ in range(border):
    dots.append(next(f))

# вывод точек
step = 1000 # шаг для вывода точек
for i, t, dot in zip(range(border), np.arange(0, border, 0.05), dots):
    template = "step: {:<5}| t:{:<5} \t| y:{:<20} | err: {:g}"
    if not i % step and i:
        print(template.format(i, t, dot, abs(dots[i] - dots[i-1])))

# Построение графика
plt.plot(dots)
```

step: 1000	t:50.0	y:37.93984115356972	err: 0.0062944
step: 2000	t:100.0	y:19.772034977406207	err: 0.00903492
step: 3000	t:150.0	y:23.797832254776615	err: 0.00761013
step: 4000	t:200.0	y:25.081219647524765	err: 0.00321468
step: 5000	t:250.0	y:23.302230737860345	err: 0.000512437
step: 6000	t:300.0	y:24.23840586199098	err: 0.000364427
step: 7000	t:350.0	y:23.99172662394127	err: 0.000364583
step: 8000	t:400.0	y:23.9525483743284	err: 0.000169255
step: 9000	t:450.0	y:24.034444559296315	err: 3.50254e-05
step: 10000	t:500.0	y:23.98691159426087	err: 1.36601e-05
step: 11000	t:550.0	y:24.00131513178786	err: 1.71913e-05
step: 12000	t:600.0	y:24.00201359874863	err: 8.76634e-06
step: 13000	t:650.0	y:23.998323485907136	err: 2.19115e-06
step: 14000	t:700.0	y:24.00070302325396	err: 4.48452e-07
step: 15000	t:750.0	y:23.99989066314617	err: 7.96461e-07
step: 16000	t:800.0	y:23.999918428344053	err: 4.4724e-07
step: 17000	t:850.0	y:24.000080410906516	err: 1.29752e-07
step: 18000	t:900.0	y:23.99996293606417	err: 1.05371e-08
step: 19000	t:950.0	y:24.000007542058338	err: 3.61583e-08
step: 20000	t:1000.0	y:24.000003079168454	err: 2.24933e-08
step: 21000	t:1050.0	y:23.99999620355829	err: 7.39459e-09
step: 22000	t:1100.0	y:24.000001921979855	err: 6.70077e-11
step: 23000	t:1150.0	y:23.999999525693173	err: 1.60206e-09
step: 24000	t:1200.0	y:23.999999897493087	err: 1.11559e-09
step: 25000	t:1250.0	y:24.000000176138204	err: 4.09379e-10
step: 26000	t:1300.0	y:23.999999901834755	err: 3.26104e-11
step: 27000	t:1350.0	y:24.00000002818475	err: 6.88374e-11
step: 28000	t:1400.0	y:24.00000000251525	err: 5.4559e-11
step: 29000	t:1450.0	y:23.99999999199015	err: 2.21405e-11
step: 30000	t:1500.0	y:24.0000000049424	err: 3.05533e-12
step: 31000	t:1550.0	y:23.999999998389658	err: 2.83862e-12
step: 32000	t:1600.0	y:24.000000000004814	err: 2.62901e-12
step: 33000	t:1650.0	y:24.000000000355616	err: 1.17595e-12
step: 34000	t:1700.0	y:23.9999999975461	err: 2.23821e-13
step: 35000	t:1750.0	y:24.00000000008932	err: 1.10134e-13
step: 36000	t:1800.0	y:23.9999999999335	err: 1.20792e-13
step: 37000	t:1850.0	y:23.9999999998466	err: 6.03961e-14
step: 38000	t:1900.0	y:24.00000000011976	err: 1.42109e-14
step: 39000	t:1950.0	y:23.9999999999514	err: 3.55271e-15
step: 40000	t:2000.0	y:24.00000000000615	err: 1.06581e-14
step: 41000	t:2050.0	y:24.00000000000735	err: 0
step: 42000	t:2100.0	y:23.9999999999963	err: 3.55271e-15
step: 43000	t:2150.0	y:23.9999999999999	err: 0
step: 44000	t:2200.0	y:24.00000000000007	err: 0
step: 45000	t:2250.0	y:24.00000000000009	err: 0
step: 46000	t:2300.0	y:24.000000000000092	err: 0
step: 47000	t:2350.0	y:24.000000000000092	err: 0
step: 48000	t:2400.0	y:24.000000000000092	err: 0
step: 49000	t:2450.0	y:24.000000000000092	err: 0

Out[179]: [



## Вывод

- При шаге равном  $h = 0.05$  фаза успокоения, с машинным отклонением между точками в  $e-03$  экспоненты, наступает при 1000 шагах. При 5000 и более шагов компьютерное отклонение между точками составляет 0
- При шаге равном  $h = 0.005$  фаза успокоения, с машинным отклонением между точками в  $e-05$  экспоненты, наступает при 1000 шагах. При 43000 и более шагов компьютерное отклонение между точками составляет 0

In [ ]:

# Урезание шагов, оптимизация графика

**Задача:** необходимо найти такой шаг, при котором график бы достигал фазы успокоения на этапе 100-200 шагов

**Решение:** данная задача достигается при шаге 0.15

```
[21]: h = 0.15 # шаг с которым функция будет вычислять значения
      f = gen_Euler(h)

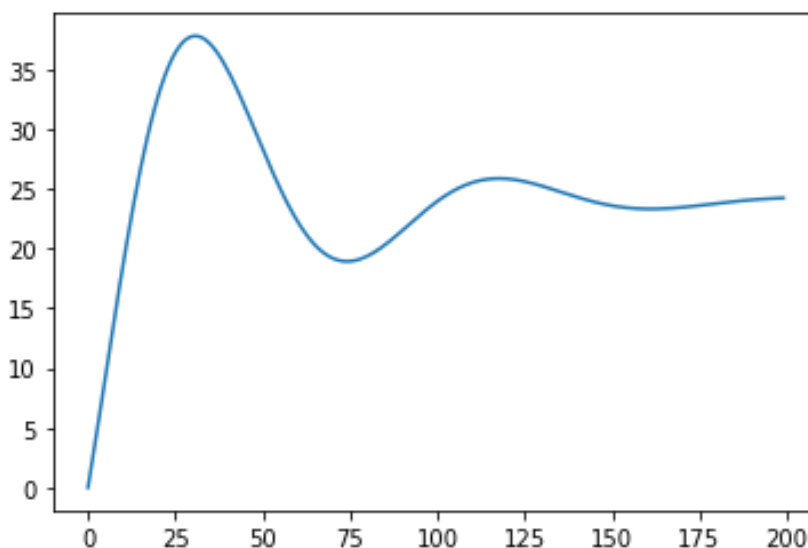
      border = 200 # кол-во точек
      dots = []
      for _ in range(border):
          dots.append(next(f))

      # вывод точек
      step = 25 # шаг для вывода точек
      for i, t, dot in zip(range(border), np.arange(0, border, 0.15), dots):
          template = "step: {:<5}| t:{:<5} \t| y:{:<20} | err: {:g}"
          if not i % step and i:
              print(template.format(i, t, dot, abs(dots[i] - dots[i-1])))

      # Построение графика
      plt.plot(dots)
```

step: 25	t:3.75	y:36.401543580513916	err: 0.549504
step: 50	t:7.5	y:28.317634022926885	err: 0.696474
step: 75	t:11.25	y:18.93791538682245	err: 0.0100276
step: 100	t:15.0	y:23.970483010413616	err: 0.21764
step: 125	t:18.75	y:25.608873237279298	err: 0.0606164
step: 150	t:22.5	y:23.58468444138014	err: 0.052838
step: 175	t:26.25	y:23.600750654559114	err: 0.0331185

```
[21]: [<matplotlib.lines.Line2D at 0x7fe81e9c9dc0>]
```



```
[ ]:
```