



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический  
университет имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

## Лабораторная работа № 4

**Дисциплина**

Моделирование.

**Тема**

Программно-алгоритмическая реализация моделей на основе дифференциальных уравнений в частных производных с краевыми условиями II и III рода.

**Студент**

Сиденко А.Г.

**Группа**

ИУ7-63Б

**Оценка (баллы)**

**Преподаватель**

Градов В.М.

Москва, 2020 г.

**Цель работы:** Получение навыков разработки алгоритмов решения смешанной краевой задачи при реализации моделей, построенных на квазилинейном уравнении параболического типа.

1. Дано:

$$k(T) = a_1(b_1 + c_1 T^{m_1})$$

$$c(T) = a_2 + b_2 T^{m_2} - \frac{c_2}{T^2}$$

$$a_1 = 0.0134$$

$$b_1 = 1$$

$$c_1 = 4.35 \cdot 10^{-4}$$

$$m_1 = 1$$

$$a_2 = 2.049$$

$$b_2 = 0.563 \cdot 10^{-3}$$

$$c_2 = 0.528 \cdot 10^5$$

$$m_2 = 1$$

$$\alpha(x) = \frac{c}{x - d}$$

$$\alpha_0 = 0.05$$

$$\alpha_N = 0.01$$

$$l = 10$$

$$T_0 = 300$$

$$R = 0.5$$

$$F(t) = 50$$

2. Уравнение

$$c(T) \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left( k(T) \frac{\partial T}{\partial x} \right) - \frac{2}{R} \alpha(x) T + \frac{2T_0}{R} \alpha(x) \quad (1)$$

3. Краевые условия

$$\begin{cases} t = 0, & T(x, 0) = T_0 \\ x = 0, & -k(T(0)) \frac{\partial T}{\partial x} = F_0 \\ x = l, & -k(l) \frac{\partial T}{\partial x} = \alpha_N(T(l) - T_0) \end{cases}$$

Функция  $\alpha(x)$  задана:

$$\alpha(x) = \frac{c}{x-d}$$

, где

$$c = -\alpha_0 d = \frac{\alpha_0 \alpha_N l}{\alpha_0 - \alpha_N}, d = \frac{\alpha_N l}{\alpha_N - \alpha_0}$$

#### 4. Разностная схема

$$\begin{aligned} \widehat{A}_n \widehat{y}_{n-1} - \widehat{B}_n \widehat{y}_n + \widehat{D}_n \widehat{y}_{n+1} &= -\widehat{F}_n, 1 \leq n \leq N-1 \\ \widehat{A}_n &= \widehat{\chi}_{n-\frac{1}{2}} \frac{\tau}{h}, \\ \widehat{B}_n &= \widehat{A}_n + \widehat{D}_n + \widehat{c}_n h + p_n h \tau, \\ \widehat{D}_n &= \widehat{\chi}_{n+\frac{1}{2}} \frac{\tau}{h}, \\ \widehat{F}_n &= f_n h \tau + \widehat{c}_n y_n h \end{aligned} \quad (2)$$

#### 5. Краевые условия

Обозначим:

$$\begin{aligned} F &= -k(T) \frac{\partial T}{\partial x} \\ p(x) &= \frac{2}{R} \alpha(x) \\ f(x) &= \frac{2T_0}{R} \alpha(x) \end{aligned} \quad (3)$$

Разностные аналоги краевых условий при  $x=0$ :

$$\begin{aligned} \left( \frac{h}{8} \widehat{c}_{\frac{1}{2}} + \frac{h}{4} \widehat{c}_0 + \widehat{\chi}_{\frac{1}{2}} \frac{\tau}{h} + \frac{\tau h}{8} p_{\frac{1}{2}} + \frac{\tau h}{4} p_0 \right) \widehat{y}_0 + \left( \frac{h}{8} \widehat{c}_{\frac{1}{2}} - \widehat{\chi}_{\frac{1}{2}} \frac{\tau}{h} + \frac{\tau h}{8} p_{\frac{1}{2}} \right) \widehat{y}_1 &= \\ = \frac{h}{8} \widehat{c}_{\frac{1}{2}} (y_0 + y_1) + \frac{h}{4} \widehat{c}_0 y_0 + \widehat{F} \tau + \frac{\tau h}{4} (\widehat{f}_{\frac{1}{2}} + \widehat{c}_0) \end{aligned} \quad (4)$$

Взято из лекции.

Разностные аналоги краевых условий при  $x=l$ , вывод:

Проинтегрируем уравнение 1 с учетом замен 3 на отрезке  $[x_{N-\frac{1}{2}}, x_N]$

$$\int_{x_{N-\frac{1}{2}}}^{x_N} dx = - \int_{t_m}^{t_{m+1}} dt \int_{x_{N-\frac{1}{2}}}^{x_N} \frac{\partial F}{\partial x} dx - \int_{x_{N-\frac{1}{2}}}^{x_N} dx \int_{t_m}^{t_{m+1}} p(x) T dt +$$

$$+ \int_{x_{N-\frac{1}{2}}}^{x_N} dx \int_{t_m}^{t_{m+1}} f(T) dt$$

Вычисляем интегралы методом правых прямоугольников.

$$\int_{x_{N-\frac{1}{2}}}^{x_N} \widehat{c} (\widehat{T} - T) dx = - \int_{t_m}^{t_{m+1}} (F_N - F_{N-\frac{1}{2}}) dt - \int_{x_{N-\frac{1}{2}}}^{x_N} p \widehat{T} \tau dx + \int_{x_{N-\frac{1}{2}}}^{x_N} \widehat{f} \tau dx$$

Вычисляем интегралы методом трапеций, а первый справа – методом правых прямоугольников.

$$\left( \widehat{c}_N (\widehat{y}_N - y_N) + \widehat{c}_{N-\frac{1}{2}} (\widehat{y}_{N-\frac{1}{2}} - y_{N-\frac{1}{2}}) \right) \frac{h}{4} = -\tau (\widehat{F}_N - \widehat{F}_{N-\frac{1}{2}}) -$$

$$- (p_N \widehat{y}_N + p_{N-\frac{1}{2}} \widehat{y}_{N-\frac{1}{2}}) \tau \frac{h}{4} + (\widehat{f}_N + \widehat{f}_{N-\frac{1}{2}}) \tau \frac{h}{4}$$

Зная

$$\widehat{F}_N = \alpha_N (\widehat{y}_N - T_0)$$

$$\widehat{F}_{N-\frac{1}{2}} = \chi_{N-\frac{1}{2}} \frac{\widehat{y}_{N-1} - \widehat{y}_N}{h} \quad (5)$$

$$\widehat{c}_N \widehat{y}_N \frac{h}{4} - \widehat{c}_N y_N \frac{h}{4} + \widehat{c}_{N-\frac{1}{2}} \widehat{y}_N \frac{h}{8} + \widehat{c}_{N-\frac{1}{2}} \widehat{y}_{N-1} \frac{h}{8} - \widehat{c}_{N-\frac{1}{2}} y_N \frac{h}{8} -$$

$$- \widehat{c}_{N-\frac{1}{2}} y_{N-1} \frac{h}{8} = -\tau \alpha_N \widehat{y}_N + \tau \alpha_N T_0 + \tau \chi_{N-\frac{1}{2}} \frac{\widehat{y}_{N-1}}{h} - \tau \chi_{N-\frac{1}{2}} \frac{\widehat{y}_N}{h} -$$

$$- p_N \widehat{y}_N \tau \frac{h}{4} - p_{N-\frac{1}{2}} \widehat{y}_N \tau \frac{h}{8} - p_{N-\frac{1}{2}} \widehat{y}_{N-1} \tau \frac{h}{8} + (\widehat{f}_N + \widehat{f}_{N-\frac{1}{2}}) \tau \frac{h}{4}$$

$$\widehat{y}_{N-1} \left( \widehat{c}_{N-\frac{1}{2}} \frac{h}{8} - \frac{\tau \chi_{N-\frac{1}{2}}}{h} + p_{N-\frac{1}{2}} \tau \frac{h}{8} \right) +$$

$$+ \widehat{y}_N \left( \widehat{c}_N \frac{h}{4} + \widehat{c}_{N-\frac{1}{2}} \frac{h}{8} + \tau \alpha_N + \frac{\tau \chi_{N-\frac{1}{2}}}{h} + p_N \tau \frac{h}{4} + p_{N-\frac{1}{2}} \tau \frac{h}{8} \right) =$$

$$= \widehat{c}_{N-\frac{1}{2}} \left( y_N \frac{h}{8} + y_{N-1} \frac{h}{8} \right) + \widehat{c}_N y_N \frac{h}{4} + \tau \alpha_N T_0 + (\widehat{f}_N + \widehat{f}_{N-\frac{1}{2}}) \tau \frac{h}{4} \quad (6)$$

Соответственно из этих краевых условий (формулы 4, 6) можем найти коэффициенты  $K_0, K_N, M_0, M_N, P_0, P_N$ .

$$\begin{cases} \widehat{K}_0 \widehat{y}_0 + \widehat{M}_0 \widehat{y}_1 = \widehat{P}_0 \\ \widehat{A}_n \widehat{y}_{n-1} - \widehat{B}_n \widehat{y}_n + \widehat{D}_n \widehat{y}_{n+1} = -\widehat{F}_n, 1 \leq n \leq N-1 \\ \widehat{K}_N \widehat{y}_N + \widehat{M}_{N-1} \widehat{y}_{N-1} = \widehat{P}_N \end{cases} \quad (7)$$

Система 7 решается методом простых итераций.

## 6. Метод простых итераций

Обозначим текущую итерацию  $s$ , а предыдущую  $(s-1)$ , тогда итерационный процесс организуется по схеме

$$A_n^{s-1} y_{n+1}^s - B_n^{s-1} y_n^s + D_n^{s-1} y_{n-1}^s = -F_n^{s-1} \quad (8)$$

Решение которой осуществляется методом прогонки. Итерации прекращаются при условии

$$\max \left| \frac{y_n^s - y_n^{s-1}}{y_n^s} \right| \leq \varepsilon, \text{ для всех } n = \overline{0 \dots N}$$

**Приведем листинг.**

Листинг 1: Заданные параметры

```

1 a1 = 0.0134
2 b1 = 1
3 c1 = 4.35e-4
4 m1 = 1
5 a2 = 2.049
6 b2 = 0.563e-3
7 c2 = 0.528e5
8 m2 = 1
9 alpha0 = 0.05
10 alphaN = 0.01
11 l = 10
12 T0 = 300
13 R = 0.5
14 F0 = 50
15 h = 0.01
16 t = 1

```

Листинг 2: Коэффициенты теплопроводности материала стержня и теплоотдачи при обдуве; теплоемкость

```
1 def k(T):
2     return a1 * (b1 + c1 * T ** m1)
3
4 def c(T):
5     return a2 + b2 * T ** m2 - c2 / T ** 2
6
7 def alpha(x):
8     d = (alphaN * l) / (alphaN - alpha0)
9     c = - alpha0 * d
10    return c / (x - d)
```

Листинг 3: Выполненные замены

```
1 def p(x):
2     return 2 * alpha(x) / R
3
4 def f(x):
5     return 2 * alpha(x) * T0 / R
```

Листинг 4: Метод средних для вычисления значения функций

```
1 def func_plus_1_2(x, step, func):
2     return (func(x) + func(x + step)) / 2
3
4 def func_minus_1_2(x, step, func):
5     return (func(x) + func(x - step)) / 2
```

Листинг 5: Функции для нахождения параметров разностной схемы

```
1 def A(T):
2     return func_minus_1_2(T, t, k) * t / h
3
4 def D(T):
5     return func_plus_1_2(T, t, k) * t / h
6
7 def B(x, T):
8     return A(T) + D(T) + c(T) * h + p(x) * h * t
9
10 def F(x, T):
11    return f(x) * h * t + c(T) * T * h
```

Листинг 6: Метод прогонки

```
1 def run_through(prevT):
```

```

2 K0 = h / 8 * func_plus_1_2(prevT[0], t, c) + \
3       h / 4 * c(prevT[0]) + \
4       func_plus_1_2(prevT[0], t, k) * t / h + \
5       t * h / 8 * p(h / 2) + t * h / 4 * p(0)
6
7 M0 = h / 8 * func_plus_1_2(prevT[0], t, c) - \
8       func_plus_1_2(prevT[0], t, k) * t / h + \
9       t * h * p(h / 2) / 8
10
11 P0 = h / 8 * func_plus_1_2(prevT[0], t, c) * (prevT[0] + \
12       prevT[1]) + \
13       h / 4 * c(prevT[0]) * prevT[0] + \
14       F0 * t + t * h / 8 * (3 * f(0) + f(h))
15
16 KN = h / 8 * func_minus_1_2(prevT[-1], t, c) + \
17       h / 4 * c(prevT[-1]) + \
18       func_minus_1_2(prevT[-1], t, k) * t / h + \
19       t * alphaN + \
20       t * h / 8 * p(1 - h / 2) + t * h / 4 * p(1)
21
22 MN = h / 8 * func_minus_1_2(prevT[-1], t, c) - \
23       func_minus_1_2(prevT[-1], t, k) * t / h + \
24       t * h * p(1 - h / 2) / 8
25
26 PN = h / 8 * func_minus_1_2(prevT[-1], t, c) * \
27       (prevT[-1] + prevT[-2]) + \
28       h / 4 * c(prevT[-1]) * prevT[-1] + \
29       t * alphaN * T0 + \
30       t * h / 4 * (f(1) + f(1 - h / 2))
31
32 # Прямой ход
33 eps = [0, -M0 / K0]
34 eta = [0, P0 / K0]
35
36 x = h
37 n = 1
38 while (x + h < 1):
39     eps.append(D(prevT[n]) / (B(x, prevT[n]) - \
40         A(prevT[n]) * eps[n]))
41     eta.append((F(x, prevT[n]) + \
42         A(prevT[n]) * eta[n]) / (B(x, prevT[n]) \
43         - A(prevT[n]) * eps[n]))
44     n += 1

```

```

45         x += h
46
47     # Обратный ход
48     y = [0] * (n + 1)
49     y[n] = (PN - MN * eta[n]) / (KN + MN * eps[n])
50
51     for i in range(n - 1, -1, -1):
52         y[i] = eps[i + 1] * y[i + 1] + eta[i + 1]
53
54     return y

```

#### Листинг 7: Метод простых итераций

```

1  def check_eps(T, newT):
2      for i, j in zip(T, newT):
3          if fabs((i - j) / j) > 1e-3:
4              return True
5      return False
6
7  def check_iter(T, newT):
8      max = fabs((T[0] - newT[0]) / newT[0])
9      for i, j in zip(T, newT):
10         d = fabs(i - j) / j
11         if d > max:
12             max = d
13     return max < 1e-2
14
15  def iter_method():
16     result = []
17     n = int(1 / h)
18     T = [T0] * (n + 1)
19     newT = [0] * n
20
21     result.append(T)
22
23     while (True):
24         buf = T
25         while True:
26             newT = run_through(buf)
27             if check_iter(buf, newT):
28                 break
29             buf = newT
30
31     result.append(newT)

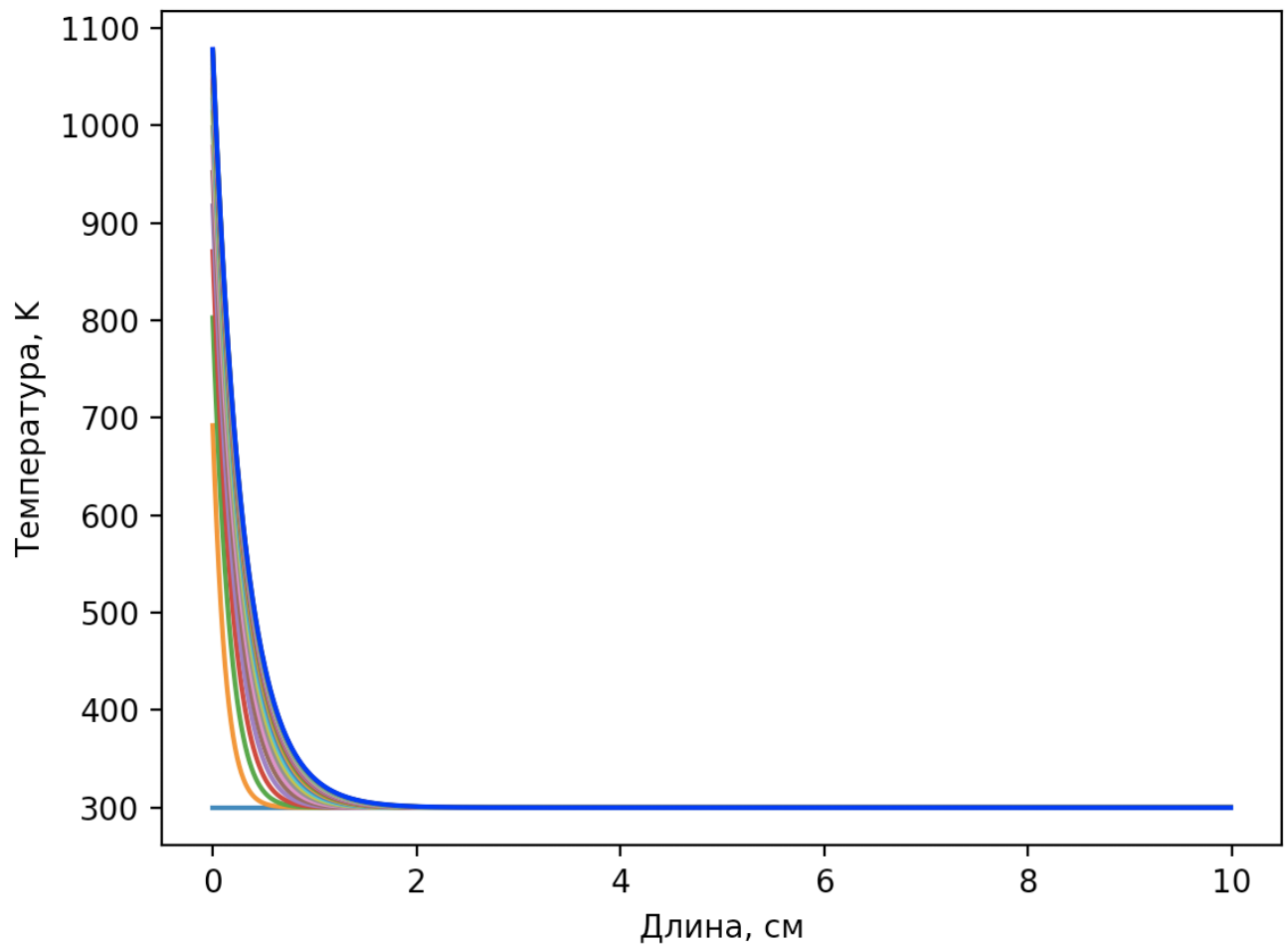
```



```
32         if (check_eps(T, newT) == False):
33             break
34
35         T = newT
36
37     return result
```

## Результаты работы программы.

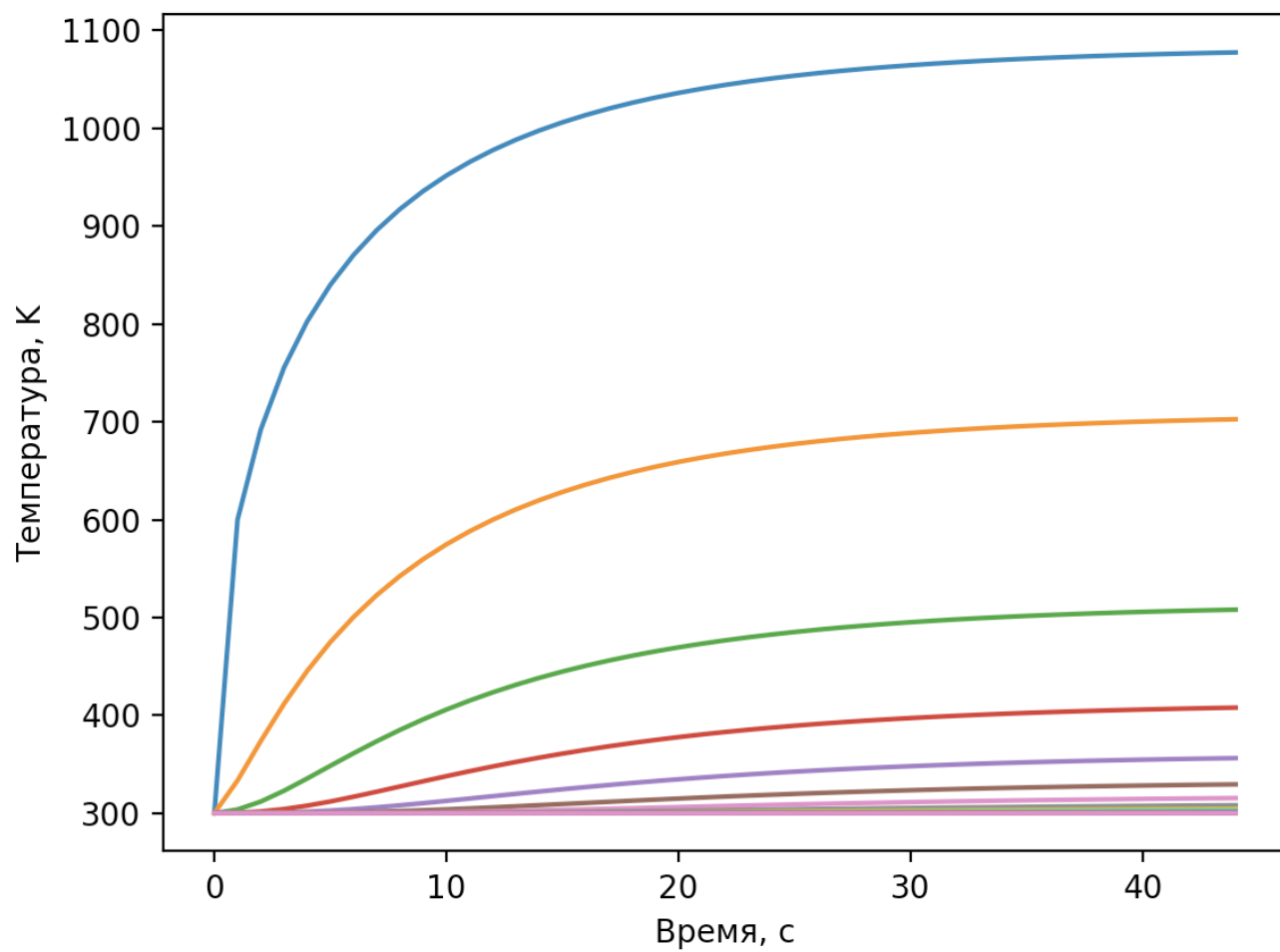
1. График зависимости температуры от координаты при нескольких фиксированных значениях времени.



На рисунке представлены графики зависимости температуры от координаты при фиксированных  $t = 0, 2, 4, \dots$

Последняя – синяя кривая соответствует установившемуся режиму, когда поле перестает меняться с точностью  $1e-3$ .

2. График зависимости температуры от времени при нескольких фиксированных значениях координаты.

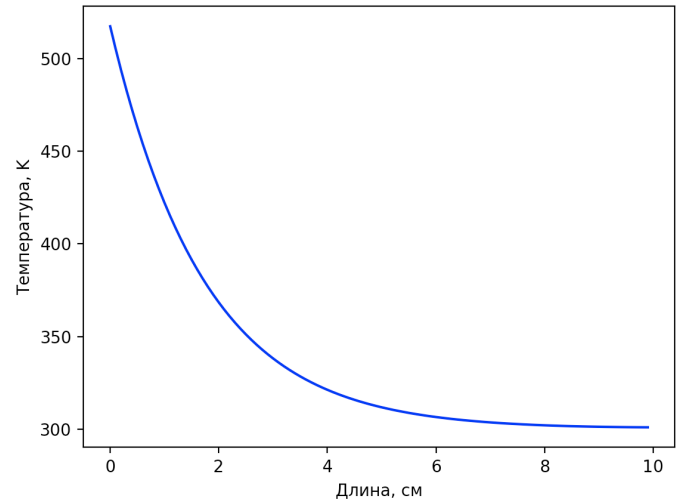
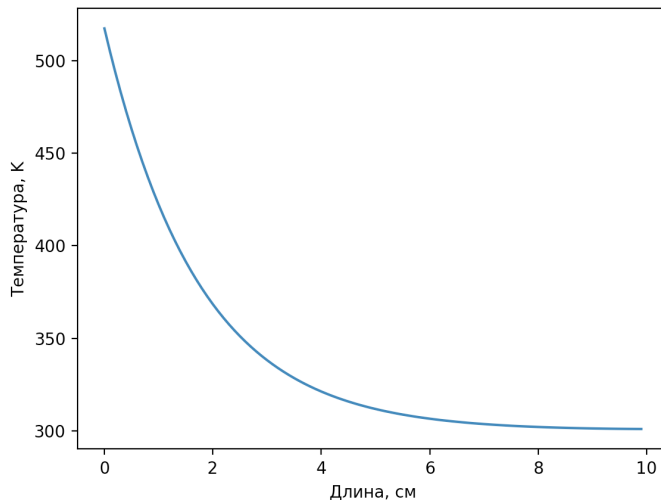


На рисунке представлены графики зависимости температуры от времени при фиксированных  $x = 0, 0.2, 0.4, \dots 3.2$

## Ответы на вопросы.

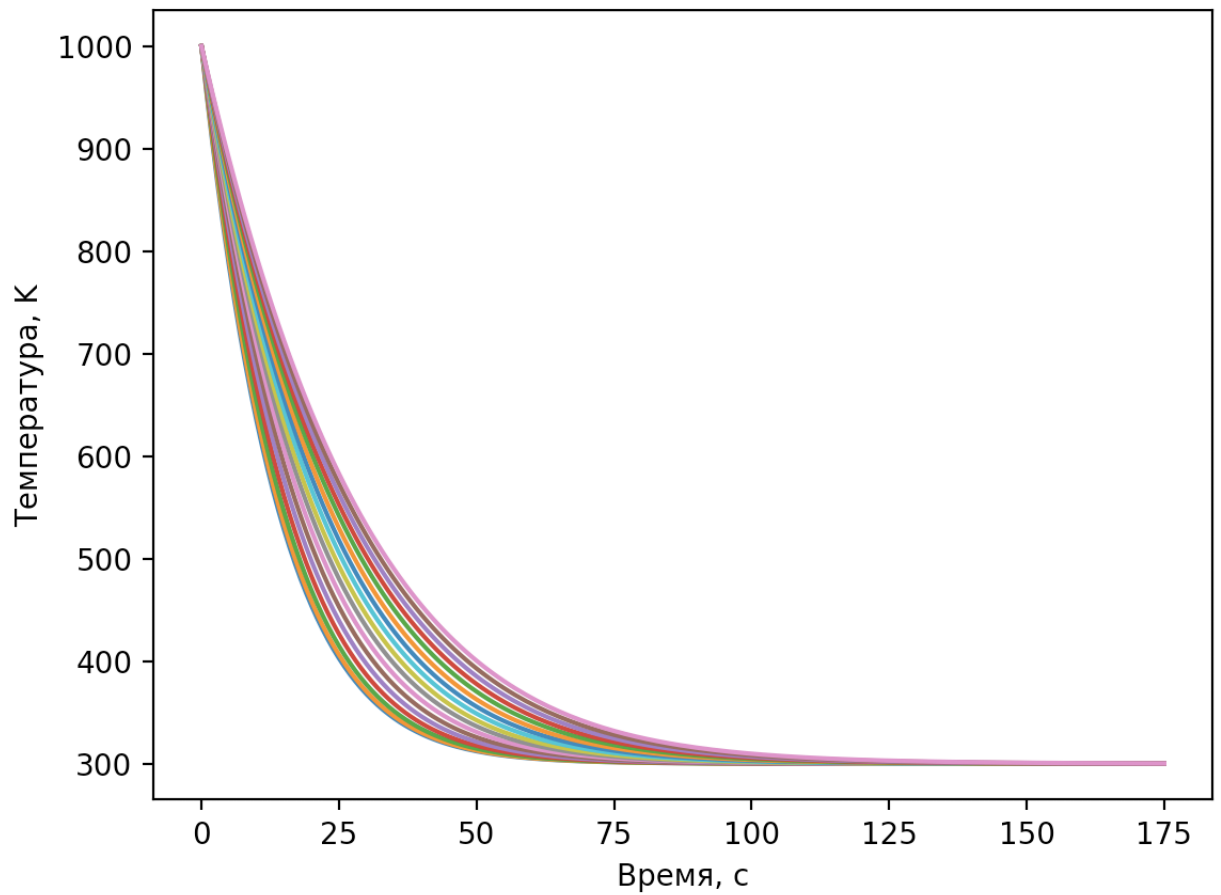
### 1. Тестирование

- (а) Свойства материала стержня привязаны к температуре, теплоемкость и коэффициент теплопроводности зависят от  $T$ , поменяем зависимости установим зависимости  $T$  от  $x$ , а  $s = 0$ . Тогда температурное поле  $T(x,t)$  должно совпасть с температурным распределением  $T(x)$  из лабораторной 3.



На левом рисунке представлен график из 3 работы, а на правом из текущей с измененными параметрами.

- (b) Если после разогрева стержня положить поток  $= 0$ , то будет происходить остывание, пока температура не выровняется по всей длине и не станет равной  $T_0$ .



Нагреем весь стержень до температуры 1000K, поток установим в 0, температура окружающей среды оставим 300K.

На данном рисунке при фиксированных  $x = 0, 0.2, 0.4, \dots, 3.2$  представлены графики зависимости температуры от времени.

2. Выполните, полагая для простоты, что все коэффициенты зависят только от одной переменной  $y_n$ . Приведите линеаризованный вариант уравнения и опишите алгоритм его решения.