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

ФАКУЛЬТЕТ “Информатика и системы управления”

КАФЕДРА "Программное обеспечение ЭВМ и информационные технологии”

ОТЧЁТ

К ЛАБОРАТОРНОЙ РАБОТЕ №4

ПО КУРСУ МОДЕЛИРОВАНИЕ НА ТЕМУ:

“Программно-алгоритмическая реализация моделей на основе ОДУ второго порядка с краевыми условиями II и III рода”

Студент ИУ7-53 БВ _________________ Д.П. Косаревский
(Группа) (Подпись, дата) (И.О.Фамилия)

Преподаватель _________________ В.М. Градов
(Подпись, дата) (И.О.Фамилия)

Москва 2021 г.

Цель работы: Получение навыков разработки алгоритмов решения краевой задачи при реализации моделей, построенных на ОДУ второго порядка.

Теория

Задана математическая модель. Уравнение для функции T(x):

$$\frac{d}{dx}\left(K(x)\frac{dT}{dx}\right) - \frac{2}{R}\alpha(x)T + \frac{2T_0}{R}\alpha(x) = 0 \qquad (1)$$

Краевые условия:

$$\begin{cases} x = 0, \ -k(0)\frac{dT}{dx} = F_0 \\ x = l, \ -k(l)\frac{dT}{dx} = \alpha_N(T(l) - T_0) \end{cases} \qquad (2)$$

Функции k(x), α(x) заданы:

$$k(x) = \frac{a}{x-b} \qquad (3)$$

$$\alpha(x) = \frac{c}{x-d} \qquad (4)$$

Константы a, b следует найти из условий k(0) = k0, k(l) = kN, а константы c, d из условий α(0) = α0, α(l) = αN.

a = −k0b (5)

$$b = \frac{k_N l}{k_N - k_0} \qquad (6)$$

c = −α0d (7)

$$d = \frac{\alpha_0 l}{\alpha_N - \alpha_0} \qquad (8)$$

Разностная схема

Anyn−1 − Bnyn + Cnyn+1 = −Dn,1 ≤ n ≤ N − 1

K0y0 + M0y1 = P0 (9)

KNyN + MNyN−1 = PN , где

$$A_n = \frac{x_{n+\frac{1}{2}}}{h}, \qquad (10)$$

Bn = An + Cn + pnh, (11)

$$C_n = \frac{x_{n-\frac{1}{2}}}{h}, \qquad (12)$$

Dn = fnh (13)

Для вычисления используем метод средних:

$$x_{n\pm\frac{1}{2}} = \frac{k_n + k_{n\pm1}}{2} \qquad (14)$$

Система решается в два прохода: прямой и обратный.

В прямом проходе вычисляем прогоночные коэффициенты ε и η. Начальные значения:

$$\varepsilon_1 = -\frac{M_0}{K_0}$$
$$\eta_1 = \frac{P_0}{K_0} \qquad (15)$$

Вычисляем массивы прогоночных коэффициентов ε,η:

$$\varepsilon_{n+1} = \frac{C_n}{B_n - A_n\varepsilon_n}$$
$$\eta_{n+1} = \frac{D_n + A_n\eta_n}{B_n - A_n\varepsilon_n} \qquad (16)$$

Обратный проход.

Определим yN - значение функции в последней точке:

$$y_N = \frac{P_N - M_N\eta_N}{K_N + M_N\varepsilon_N} \qquad (17)$$

По основной прогоночной формуле находятся все значения неизвестных yn:

yn = εn+1yn+1 + ηn+1 (18)

Таким образом, массив, полученный после прогонки и есть искомый массив T(x).

Краевые условия

Обозначим:

$$F = -k(x)\frac{dT}{dx}$$
$$p(x) = \frac{2}{R}\alpha(x)$$
$$f(x) = \frac{2T_0}{R}\alpha(x)$$
$$p_n = p(x_n),\ f_n = f(x_n) \qquad (19)$$

Разностные аналоги краевых условий при x = 0:

$$\left(x_{\frac{1}{2}} + \frac{h^2}{8}p_{\frac{1}{2}} + \frac{h^2}{4}p_0\right)\cdot y_0 - \left(x_{\frac{1}{2}} - \frac{h^2}{8}p_{\frac{1}{2}}\right)\cdot y_1 = \left(hF_0 + \frac{h^2}{4}(f_{\frac{1}{2}} + f_0)\right) \qquad (20)$$

Разностные аналоги краевых условий при x = l:

Примем простую аппроксимацию:

$$p_{\frac{1}{2}} = \frac{p_0 + p_1}{2}$$
$$f_{\frac{1}{2}} = \frac{f_0 + f_1}{2} \quad (21)$$

Проинтегрируем уравнение (1) на отрезке $[x_{N-\frac{1}{2}}, x_N]$ с учетом замен (19).

$$-\int_{X_{N-\frac{1}{2}}}^{X_N} \frac{dF}{dx}dx - \int_{X_{N-\frac{1}{2}}}^{X_N} p(x)Tdx + \int_{X_{N-\frac{1}{2}}}^{X_N} f(x)dx = 0$$

Второй и третий интегралы вычислим методом трапеций

$$F_{N-\frac{1}{2}} - F_N - \frac{p_{N-\frac{1}{2}}y_{N-\frac{1}{2}} + p_N y_N}{4}h + \frac{f_{N-\frac{1}{2}} + f_N}{4}h = 0$$

Учтем, что

$$F_{N-\frac{1}{2}} = x_{N-\frac{1}{2}}\frac{y_{N-1} - y_N}{h}$$
$$F_N = \alpha_N(y_N - T_0)$$
$$y_{N-\frac{1}{2}} = \frac{y_N + y_{N-1}}{2}$$

После подстановки и приведения подобных получаем:

$$\left(-\frac{x_{N-\frac{1}{2}}}{h} - \alpha_N - \frac{2p_N - 1}{16}h\right)\cdot y_N + \left(\frac{x_{N-\frac{1}{2}}}{h} - \frac{2p_N - 1}{16}h\right)\cdot y_{N-1} = -\alpha_N T_0 - \frac{h}{4}\left(f_{N-\frac{1}{2}} + f_N\right) \quad (22)$$

Заданы начальные параметры:

k_0 = 0.4 Вт/см К,

k_N = 0.1 Вт/см К,

α_0 = 0.05 Вт/см2 К,

α_N = 0.01 Вт/см2 К,

l = 10 см,

T_0 = 300К,

R = 0.5 см,

F_0 = 50 Вт/см2.

Результаты

1. Представить разностный аналог краевого условия при $x = l$ и его краткий вывод интегро-интерполяционным методом.

Проинтегрируем уравнение из лекционных материалов на отрезке $[x_N - \frac{1}{2},\ x_N]$:

$$- \int\limits_{x_N - \frac{1}{2}}^{x_N} \frac{dF}{dx} dx - \int\limits_{x_N - \frac{1}{2}}^{x_N} p(x) u d(x) + \int\limits_{x_N - \frac{1}{2}}^{x_N} f(x) dx = 0$$

2-й и 3-й интеграл вычислим методом трапеций:

$$- (F_{x_N} - F_{x_N - \frac{1}{2}}) - \frac{h}{4} (p_{x_N - \frac{1}{2}} y_{x_N - \frac{1}{2}} + p_{x_N} y_{x_N}) + \frac{h}{4} (f_{x_N - \frac{1}{2}} + f_{x_N}) = 0$$

При подстановке выражения $F_{x_N - \frac{1}{2}}$ при $n = N$ и краевое условие при $x = l\ (x_N)$ получим формулу:

$$y_N = \frac{X_{N-\frac{1}{2}} - \frac{h^2}{8} p_{N-\frac{1}{2}}}{\alpha h + \frac{h^2}{4} p_N + \frac{h^2}{8} p_{N-\frac{1}{2}} + X_{N-\frac{1}{2}}} y_{N-1} + \frac{\frac{h^2}{4}(f_{N-\frac{1}{2}} + f_N) + h\alpha\beta}{\alpha h + \frac{h^2}{4} p_N + \frac{h^2}{8} p_{N-\frac{1}{2}} + X_{N-\frac{1}{2}}},$$ где $\alpha = \alpha_N,\ \beta = T_0$

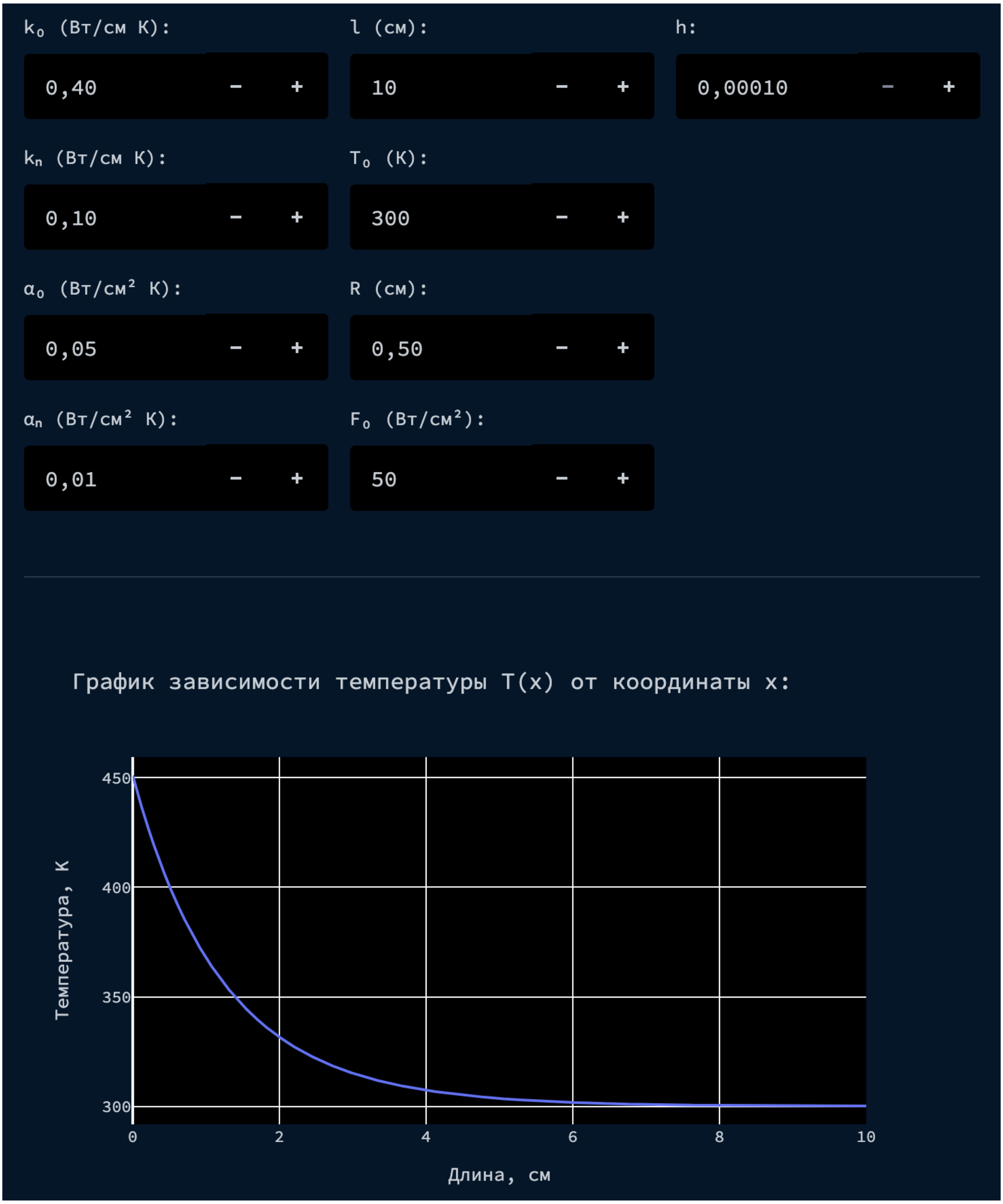
Примем простую аппроксимацию:

$$p_{N-\frac{1}{2}} = \frac{p_{N-1} + p_N}{2},\ f_{N-\frac{1}{2}} = \frac{f_{N-1} + f_N}{2}$$

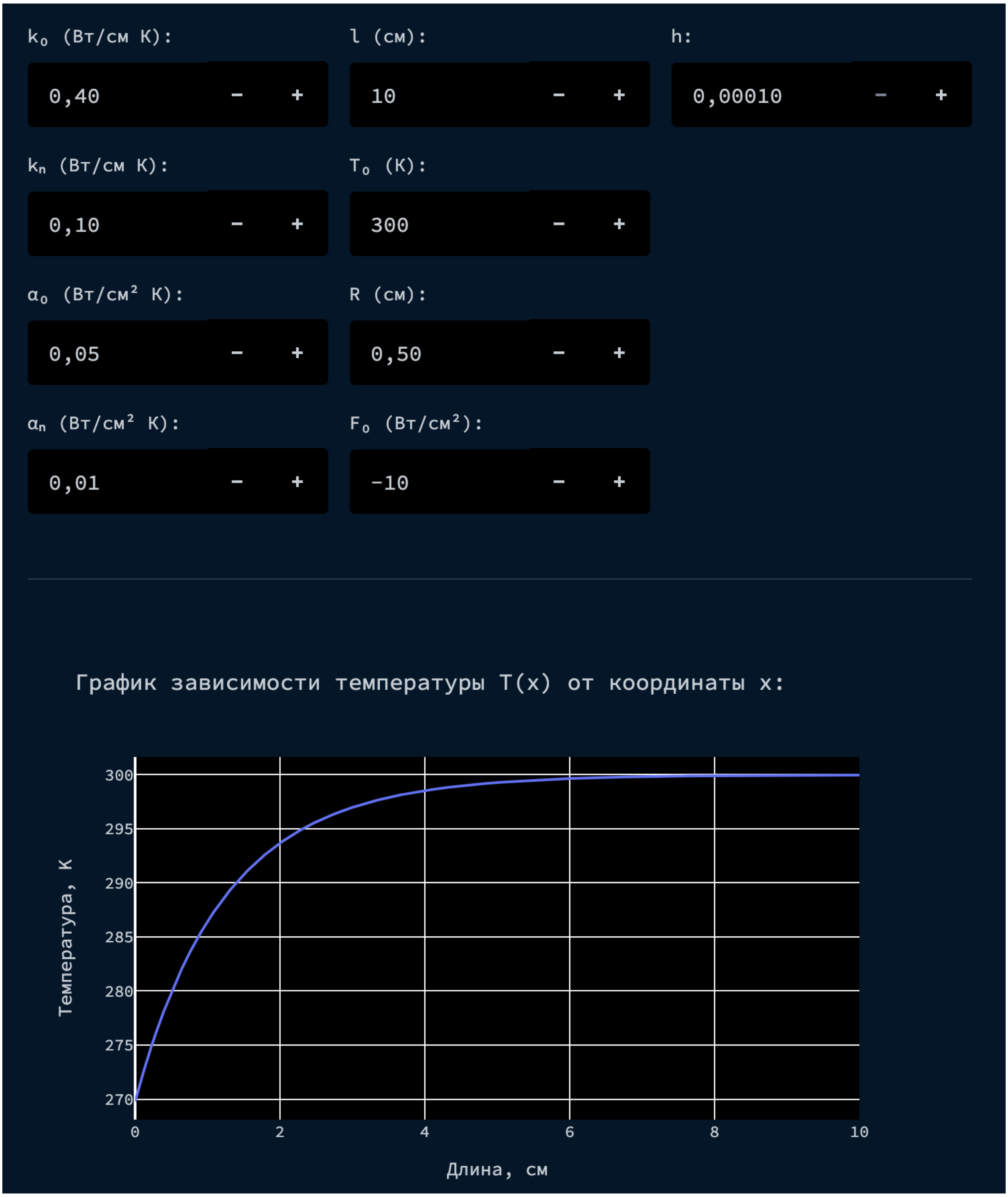
В пределе при $h \to 0$ при уменьшении шага можно пренебречь некоторыми членами и преобразовать ф-лу y_N к следующему виду:

$$y_N = \frac{X_{N-\frac{1}{2}}}{\alpha h + X_{N-\frac{1}{2}}} y_{N-1} + \frac{h\alpha\beta}{\alpha h + X_{N-\frac{1}{2}}}$$

2. График зависимости температуры $T(x)$ от координаты x при заданных выше параметрах.

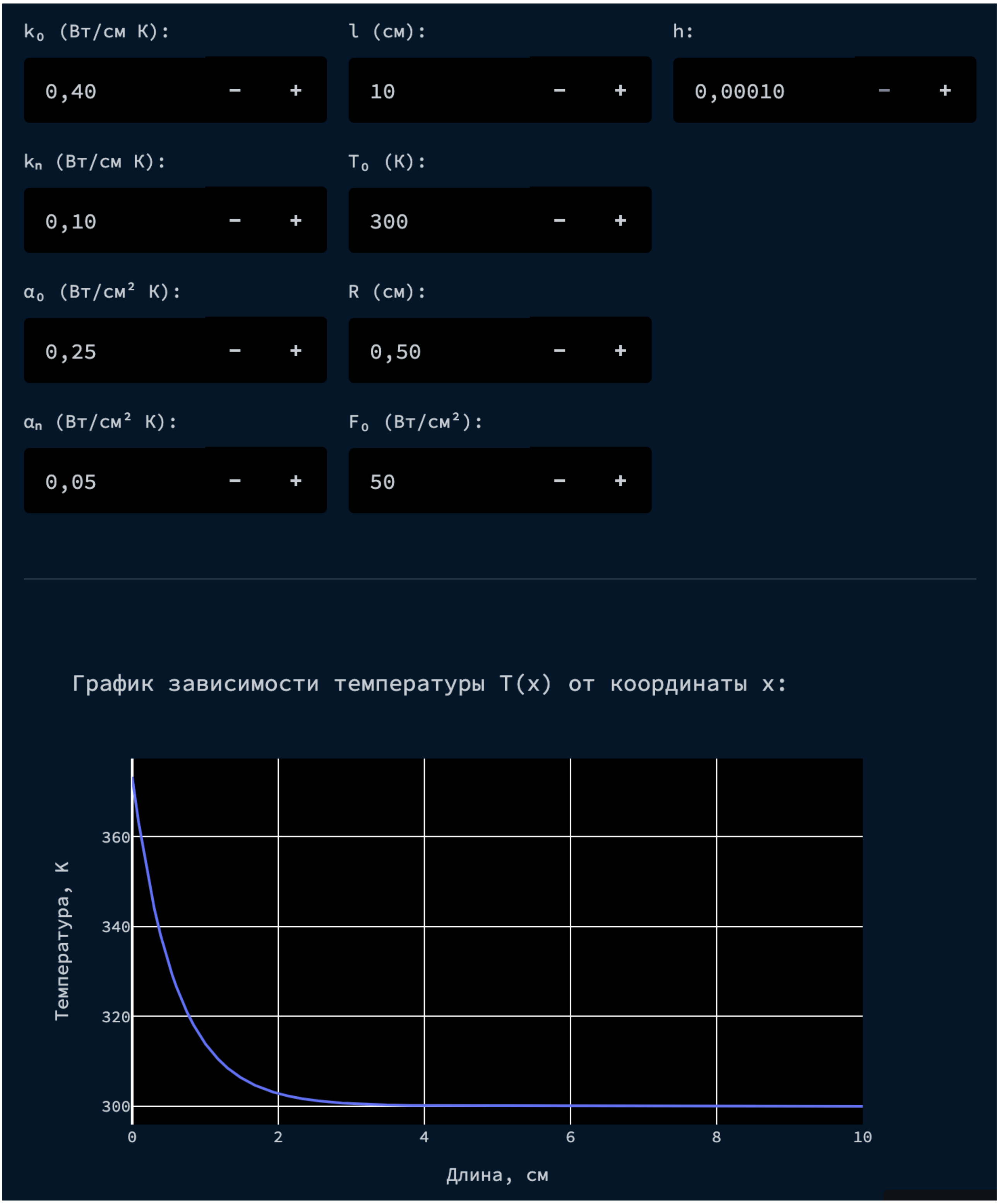


3. График зависимости $T(x)$ при $F_0 = -\ 10$ Вт/см^2.



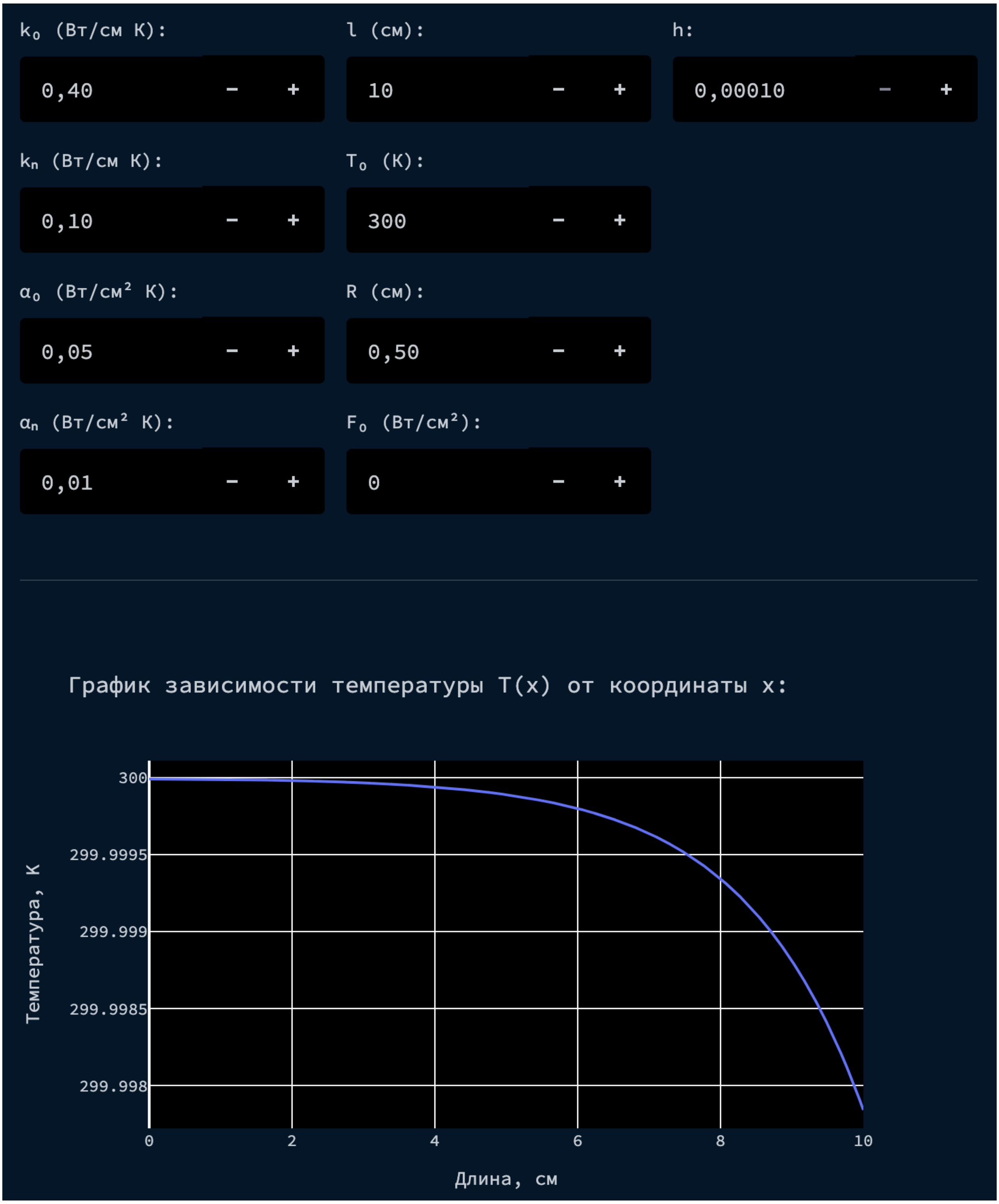
При отрицательном тепловом потоке слева идет съем тепла, поэтому производная $T'(x)$ - положительная.

4. График зависимости $T(x)$ при увеличенных значениях $\alpha(x)$ (например, в 3 раза). Сравнить с п.2.



Увеличено в 5 раз. При увеличении теплосъёма и неизменном потоке F_0 наблюдаем снижение уровня температур $T(x)$ и увеличение градиента.

5. График зависимости $T(x)$ при $F_0 = 0$.



Тепловое нагружение отсутствует, причин для нагрева нет, температура стержня равна температуре окружающей среды T_0 с некоторой погрешностью, определяемой приближённым характером вычислений.

Работающую программу можно увидеть и протестировать через web-интерфейс по следующей ссылке: https://share.streamlit.io/dkosarevsky/math_modelling/main.py

Программа написана на языке программирования Python 3.9.7 с использованием следующих библиотек:

- streamlit
- numpy
- plotly

Код

```
import streamlit as st
import numpy as np
import plotly.graph_objects as go


def k(x, a, b):
    """ коэффициенты теплопроводности и теплоотдачи """
    return a / (x - b)


def alpha(x, c, d):
    return c / (x - d)


def p(x, c, d, R):
    return 2 * alpha(x, c, d) / R


def f(x, c, d, T_0, R):
    return 2 * alpha(x, c, d) * T_0 / R


def plus_half(x, a, b, h):
    """ метод средних """
    return (k(x, a, b) + k(x + h, a, b)) / 2


def minus_half(x, a, b, h):
    """ метод средних """
    return (k(x, a, b) + k(x - h, a, b)) / 2


def A(x, a, b, h):
    """ коэффициенты разностной схемы """
    return plus_half(x, a, b, h) / h


def B(x, a, b, c, d, R, h):
    return A(x, a, b, h) + C(x, a, b, h) + p(x, c, d, R) * h


def C(x, a, b, h):
    return minus_half(x, a, b, h) / h


def D(x, c, d, T_0, R, h):
    return f(x, c, d, T_0, R) * h


def left_boundary_condition(h, a, b, c, d, R, F_0, T_0):
    """ левое граничное условие """
    k_0 = plus_half(0, a, b, h) + h * h * (p(0, c, d, R) + p(h, c, d, R)) / 16 + h * h * p(0, c,
d, R) / 4
    M0 = -plus_half(0, a, b, h) + h * h * (p(0, c, d, R) + p(h, c, d, R)) / 16
    P0 = h * F_0 + h * h / 4 * ((f(0, c, d, T_0, R) + f(h, c, d, T_0, R)) / 2 + f(0, c, d, T_0,
R))
    return k_0, M0, P0
```

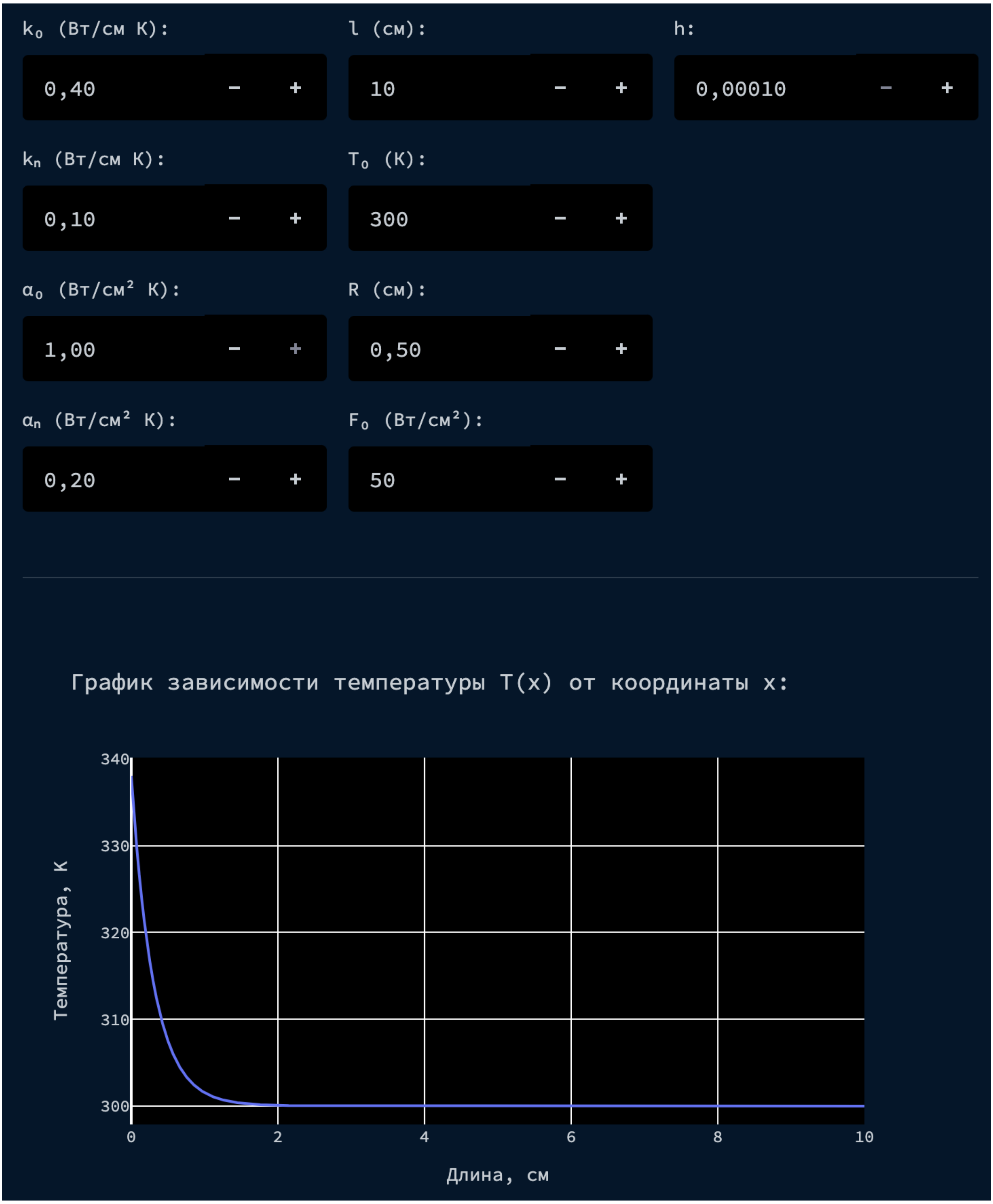
```python
def right_boundary_condition(l, a, b, c, d, h, alpha_n, R, T_0):
    """ правое граничное условие """
    k_n = -minus_half(l, a, b, h) / h - alpha_n - p(l, c, d, R) * h / 4 - ((p(l, c, d, R) + p(l -
h, c, d, R)) * h) / 16
    MN = minus_half(l, a, b, h) / h - ((p(l, c, d, R) + p(l - h, c, d, R)) * h) / 16
    PN = - alpha_n * T_0 - h * (f(l, c, d, T_0, R) + f(l - h, c, d, T_0, R) + f(l, c, d, T_0, R))
/ 8
    return k_n, MN, PN


def plot(x: np.array, t: np.array):
    """ отрисовка графика """
    st.markdown("---")

    fig = go.Figure()
    fig.add_trace(go.Scatter(
        x=x,
        y=t,
        mode='lines',
    ))

    fig.update_layout(
        title_text=f"График зависимости температуры T(x) от координаты x:",
        xaxis_title="Длина, см",
        yaxis_title="Температура, K",
    )

    st.write(fig)
    st.markdown("---")


def main():
    st.markdown("### Лабораторная работа №4")
    st.markdown("""**Тема:**
    Программно-алгоритмическая реализация моделей на основе ОДУ второго порядка с краевыми
условиями II и III рода.""")
    st.markdown("""**Цель работы:**
    Получение навыков разработки алгоритмов решения краевой задачи при реализации моделей,
    построенных на ОДУ второго порядка.""")

    a1, a2, a3 = st.columns(3)
    b1, b2, b3 = st.columns(3)
    c1, c2, c3 = st.columns(3)
    d1, d2, d3 = st.columns(3)

    k_0 = a1.number_input("k₀", min_value=0., max_value=1., value=.4)
    k_n = b1.number_input("k ", min_value=0., max_value=1., value=.1)
    alpha_0 = c1.number_input("α₀", min_value=0., max_value=1., value=.05)
    alpha_n = d1.number_input("α ", min_value=0., max_value=1., value=.01)

    l = a2.number_input("l:", min_value=1, max_value=100, value=10)
    T_0 = b2.number_input("T₀", min_value=1, max_value=1000, value=300)
    R = c2.number_input("R", min_value=0., max_value=1., value=.5)
    F_0 = d2.number_input("F₀", min_value=0, max_value=10, value=0)

    h = a3.number_input("h:", min_value=.00001, max_value=1., value=.0001, format="%.5f")

    # параметры коэффициентов теплопроводности и теплоотдачи
```

```
    b = (k_n * l) / (k_n - k_0)
    a = -k_0 * b
    d = (alpha_n * l) / (alpha_n - alpha_0)
    c = -alpha_0 * d

    k_0, M0, P0 = left_boundary_condition(h, a, b, c, d, R, F_0, T_0)
    k_n, MN, PN = right_boundary_condition(l, a, b, c, d, h, alpha_n, R, T_0)

    # прямой ход
    # массивы прогоночных коэффициентов
    eps = [0]
    eta = [0]

    eps1 = -M0 / k_0
    eta1 = P0 / k_0

    eps.append(eps1)
    eta.append(eta1)

    x = h
    n = 1
    while x + h < l:
        eps.append(C(x, a, b, h) / (B(x, a, b, c, d, R, h) - A(x, a, b, h) * eps[n]))
        eta.append((A(x, a, b, h) * eta[n] + D(x, c, d, T_0, R, h)) / (B(x, a, b, c, d, R, h) -
A(x, a, b, h) * eps[n]))
        n += 1
        x += h

    # обратный ход
    t = [0] * (n + 1)

    # значение функции в последней точке
    t[n] = (PN - MN * eta[n]) / (k_n + MN * eps[n])

    for i in range(n - 1, -1, -1):
        t[i] = eps[i + 1] * t[i + 1] + eta[i + 1]

    x = [i for i in np.arange(0, l, h)]

    plot(x, t[: -1])


if __name__ == "__main__":
    main()
```

Вопросы

1. Какие способы тестирования программы можно предложить?

Можно предложить способ увеличения коэффициента теплоотдачи. Увеличится скорость понижения температуры, стержень будет отдавать больше тепла:



2. Получите простейший разностный аналог нелинейного краевого условия при $x = l$

$$x = l, \ - k(l)\frac{dT}{dx} = a_N(T(l) - T_0) + \varphi(T),$$

где $\varphi(T)$ $-$ заданная функция

Производную аппроксимируйте односторонней разностью.

Метод разностной аппроксимации на равномерной сетке с шагом h.

$$\frac{dT}{dx} = \frac{T(x)-T(x-h)}{h}$$

при $x = l$: $\frac{dT}{dx} = \frac{T_l - T_{l-1}}{h}$

подстановка:

$$- k_l\frac{T_l - T_{l-1}}{h} = \alpha_N(T_l - T_0) + \varphi(T_l)$$

$$- k_lT_l + k_lT_{l-1} = \alpha_N hT_l - \alpha_N hT_0 + \varphi(T_l)h$$

$$- (k_l + \alpha_N h)T_l + k_lT_{l-1} = \varphi(T_l)h - \alpha_N hT_0$$

3. Опишите алгоритм применения метода прогонки, если при $x = 0$ краевое условие линейное (как в настоящей работе), а при $x = l$, как в п.2.

Прямой ход при $x = 0$, начальные прогоночные коэффициенты:

$$\varepsilon_1 = \frac{M_0}{K_0}, \ \eta_1 = \frac{P_0}{K_0}$$

Вычисление прогоночных коэффициентов:

$$\varepsilon_{n+1} = \frac{C_n}{B_n - A_n\varepsilon_n}, \ \eta_{n+1} = \frac{D_n + A_n\eta_n}{B_n - A_n\varepsilon_n}$$

Вычисление значений неизвестных y_n по основной прогоночной ф-ле:

$$y_n = \varepsilon_{n+1}y_{n+1} + \eta_{n+1}$$

Вычисление y_N. Обратным ходом вычисление коэффициентов до y_0.

4. Опишите алгоритм определения единственного значения сеточной функции y_p в одной заданной точке p. Использовать встречную прогонку, т.е. комбинацию правой и левой прогонок. Краевые условия линейные.

$i = p,\ 0 < p < N$

Левая прогонка:

коэффициенты ε_i и η_i на области $p \le i \le N$

$\varepsilon_i = \frac{C_i}{B_i - \varepsilon_{i+1} A_i},\ \eta_i = \frac{A_i \eta_{i+1} + D_i}{B_i - \varepsilon_{i+1} A_i}$

Правая прогонка:

коэффициенты α_i и β_i на области $0 \le i \le p + 1$

$\alpha_{i+1} = \frac{C_i}{B_i - \alpha_i A_i},\ \beta_{i+1} = \frac{A_i \beta_i + D_i}{C_i - \alpha_i A_i}$

при $i = p$ вычислим:

$y_p = \alpha_{p+1} y_{p+1} + \beta_{p+1}$

$y_{p+1} = \varepsilon_{p+1} y_p + \eta_{p+1}$

$y_p = \frac{\beta_{p+1} + \alpha_{p+1}\eta_{p+1}}{1 - \alpha_{p+1}\varepsilon_{p+1}}$