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»

Лабораторная работа № 3

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

Студент: Тимонин Антон

Группа ИУ7-62б

Оценка (баллы) _______________

Преподаватель Градов В. М.

Москва.
2020 г.

Тема

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

Цель работы

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

Задача

Задана математическая модель.

Уравнение для функции $T(x)$

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

Краевые условия

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

Функции $k(x), \alpha(x)$ заданы своими константами

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

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

Константы a, b следует найти из условий $k(0) = k_0$, $k(l) = k_N$, а константы c, d из условий $\alpha(0) = \alpha_0$, $\alpha(l) = \alpha_N$.

$$a = -k_0 b$$

$$b = \frac{k_n l}{k_n - k_0}$$

$$c = -\alpha_0 d$$

$$d = \frac{\alpha_0 l}{\alpha_n - \alpha_0}$$

Для решения исходного уравнения нужно получить разностную схему, их которой получится система из N уравнений, которая решается методом прогонки.

Система уравнений имеет следующий вид:

$$A_n y_{n-1} - B_n y_n + C_n y_{n+1} = -D_n,\ 1 \le n \le N-1$$

$$K_0 y_0 + M_0 = P_0$$

q

где

$$A_n = \frac{x_{n+1/2}}{h},$$

$$B_n = A_n + C_N + p_n h,$$

$$C_n = \frac{x_{n-1/2}}{h},$$

$$D_n = f_n h$$

Для вычисления используем метод средних:

$$x_{n\pm1/2} = \frac{k_n + x_{n\pm1}}{2},$$

В качестве исходных данных в метод прогонки будет передаваться:

- A_n, B_n, C_n, D_n
- Параметры для граничного условия слева: $K_0, M_0,\ P_0$
- Параметры для граничного условия справа: $K_N, M_N,\ P_N$

Система решается в два прохода: прямой и обратный.

В прямом проходе вычисляем прогулочные коэффициенты ε и :

Вычисляем массивы прогулочных коэффициентов ε и :

$$\varepsilon_{n+1} = \frac{C_n}{B_n - A_n \varepsilon_n}$$

$$ {}_{n+1} = \frac{D_n + A_{n}{}_{n}}{B_n - A_n \varepsilon_n} $$

Начальные значения берутся из граничных условий:

$$ \varepsilon_1 = -\frac{M_0}{K_0} $$

$$ {}_1 = -\frac{P_0}{K_0} $$

Обратный проход. По основной прогоночной формуле находятся все значения неизвестных y_n:

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

Начальное значение y_n считается из граничных условий:

$$ y_n = -\frac{P_n - M_{n}{}_{n}}{K_n + M_n \varepsilon_n} $$

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

Краевые условия

Разностные Аналоги краевых условий при x $= 0$.

Проинтегрируем $\frac{dF}{dx} - p(x)u + f(x) = 0$ с учетом $F = -k(x)\frac{du}{dx}$ на отрезке $[0; x_{1/2}]$

$$ -\int_0^{x_{1/2}} \frac{dF}{dx} dx - \int_0^{x_{1/2}} p(x) u\, dx + \int_0^{x_{1/2}} f(x) dx = 0 $$

Второй и третий интеграл посчитаем методом трапеции

$$-\left(F_{1/2}-F_0\right)-\frac{h}{4}\left(p_{1/2}y_{1/2}+p_0y_0\right)+\frac{h}{4}\left(f_{1/2}+f_0\right)\ =\ 0$$

Из полученного при выводе разностной схемы получаем:

$$F_{1/2}=X_{1/2}\frac{y_0-y_1}{h}$$

F_0 берется из заданного граничного условия.

Подставляем значение в уравнение и получаем разностные аналоги краевых условий при $x\ =0$:

$$y_0=\frac{x_{1/2}+\frac{h^2}{8}p_{1/2}}{x_{1/2}+\frac{h^2}{8}p_{1/2}+\frac{h^2}{4}p_0}y_1+\frac{hF_0+\frac{h^2}{4}(f_{1/2}+f_0)}{x_{1/2}+\frac{h^2}{8}p_{1/2}+\frac{h^2}{4}p_0}$$

Разностные аналоги краевых условий при x=1 можно получить аналогичным образом. Проинтегрируем $\dfrac{dF}{dx}-p(x)u+f(x)=0$ с учетом $F=-k(x)\dfrac{du}{dx}$ на отрезке $[x_{n-1/2};x_n]$

$$-\int_{x_{n-1/2}}^{x_n}\frac{dF}{dx}dx-\int_{x_{n-1/2}}^{x_n}p(x)udx+\int_{x_{n-1/2}}^{x_n}f(x)dx=0$$

Второй и третий интеграл вычислим методом трапеций

$$-\left(F_{1/2}-F_0\right)-\frac{h}{4}\left(p_{n-1/2}y_{n-1/2}+p_ny_n\right)+\frac{h}{4}\left(f_{n-1/2}+f_n\right)\ =\ 0$$

Учитывая подстановки

$$F_{n-1/2}=\ x_{n-1/2}\frac{y_{n-1}+y_n}{h}$$

$$F_n=\ \alpha_n(y_n-T_0)$$

$$F_{n-1/2} = \frac{y_n + y_{n-1}}{2}$$

Имеем

$$\left(-\frac{x_{n-1/2}}{h} - \alpha_n - \frac{2p_n - 1}{16}h\right)y_n - \left(-\frac{x_{n-1/2}}{h} - \frac{2p_n - 1}{16}h\right)y_{n-1} = -\alpha_n T_0 - \frac{h}{4}(x_{n-1/2} + f_n)$$

Начальные параметры:

$k_0 = 0.4 Вт/смК$,

$k_n = 0.1 Вт/смК$,

$\alpha_0 = 0.05 Вт/см^2К$,

$\alpha_n = 0.01 Вт/см^2К$,

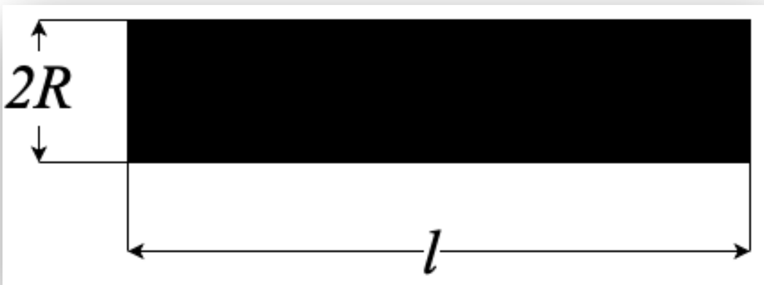
$l = 10\ см$,

$T_0 = 300К$,

$R = 0.5 см$,

$F_0 = 50 Вт/см^2$.

Физическое содержание задачи



Сформулированная математическая модель описывает температурное поле $T(x)$ вдоль цилиндрического стержня радиуса R и длиной l, причем $R \ll l$ и температуру можно принять постоянной по радиусу цилиндра. Ось x направлена вдоль оси цилиндра и начало координат совпадает с левым торцем стержня. Слева при $x = 0$ цилиндр нагружается тепловым потоком F_0. Стержень обдувается воздухом, температура которого равна T_0. В результате происходит съем тепла с цилиндрической поверхности и поверхности правого торца при $x = l$. Функции $k(x), \alpha(x)$ являются, соответственно, коэффициентами теплопроводности материала стержня и теплоотдачи при обдуве.

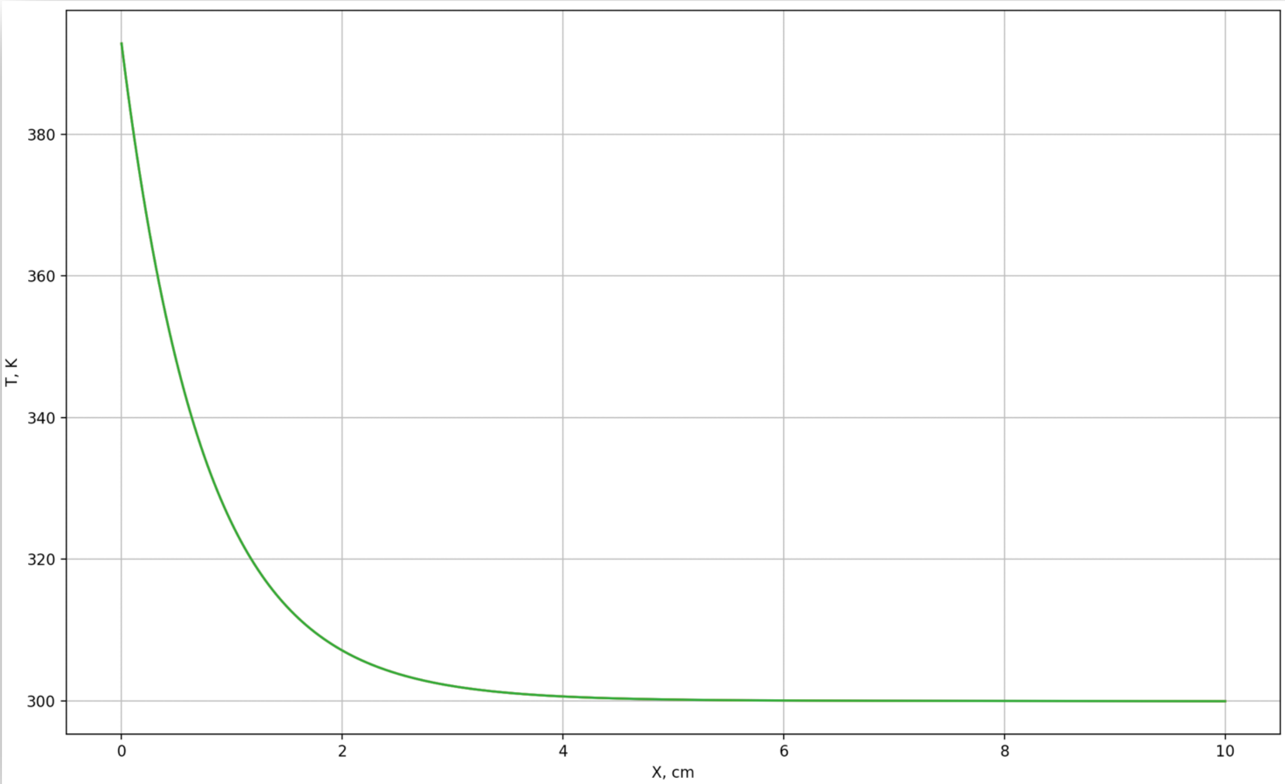
Листинг 1. Программная реализация

```
import matplotlib.pyplot as plt
import numpy as np

def k(x):
    return a / (x - b)

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

def halving(x, sign):
    return (k(x) + k(x + sign * h)) / 2

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

def f(x):
    return 2 * alpha(x) * T0 / R

def A(x):
    return halving(x, 1) / h

def B(x):
    return A(x) + C(x) + p(x) * h

def C(x):
    return halving(x, -1) / h

def D(x):
    return f(x) * h

def left_condition():
    k0 = halving(0, 1) + h * h * (p(0) + p(h)) / 16 + h * h * p(0) / 4
    m0 = -halving(0, 1) + h * h * (p(0) + p(h)) / 16
    p0 = h * F0 + h * h / 4 * ((f(0) + f(h)) / 2 + f(0))
    return k0, m0, p0

def right_condition():
    kN = - halving(l, -1) / h - aN - p(l) * h / 4 - ((p(l) + p(l - h)) *
h) / 16
    mN = halving(l, -1) / h - ((p(l) + p(l - h)) * h) / 16
    pN = - aN * T0 - h * (f(l) + f(l - h) + f(l)) / 8
    return kN, mN, pN
```

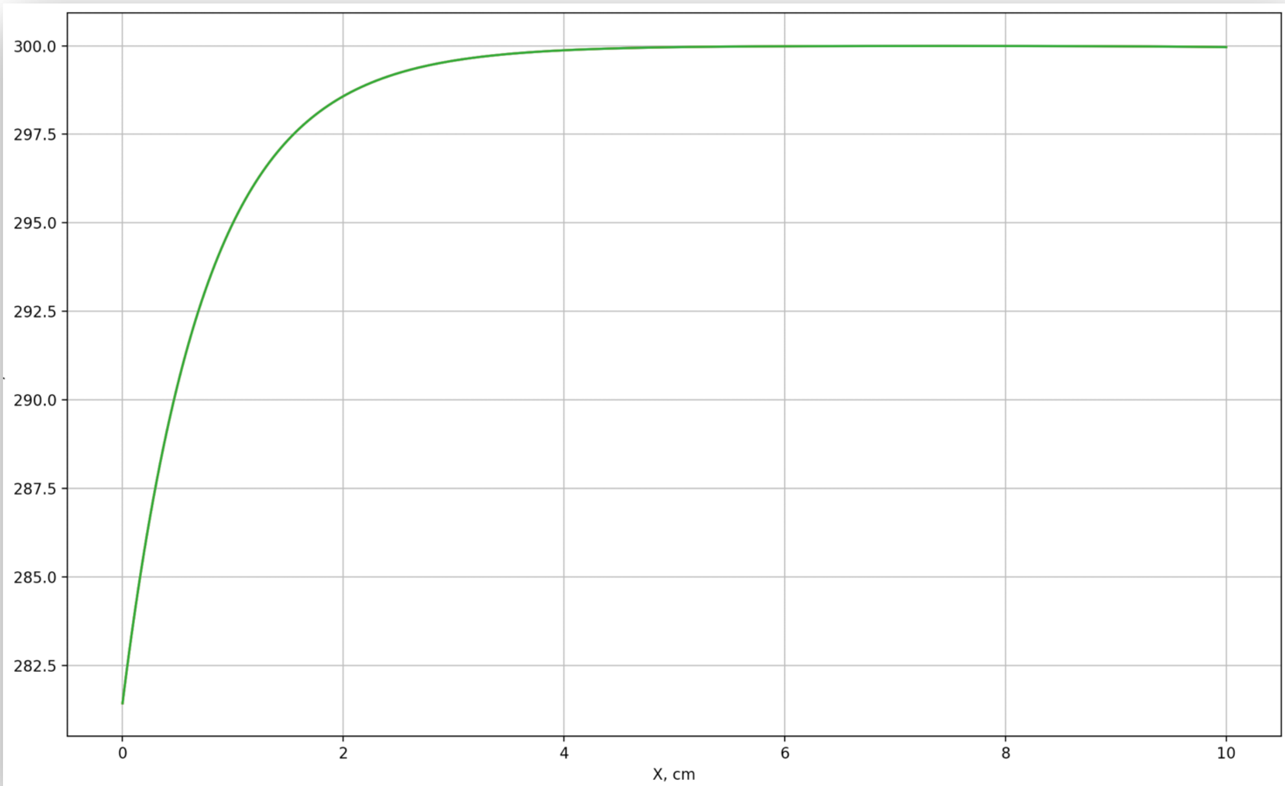
```
def plot(x, t):
    plt.plot(x, t[:-1])
    plt.xlabel("X, cm")
    plt.ylabel("T, K")
    plt.grid()
    plt.show()

if __name__ == "__main__":
    eps = [0]; eta = [0]
    b = (kN * l) / (kN - k0); a = - k0 * b;
    d = (aN * l) / (aN - a0); c = - a0 * d

    k0, m0, p0 = left_condition()
    kN, mN, pN = right_condition()
    eps1 = -m0 / k0; eta1 = p0 / k0
    eps.append(eps1); eta.append(eta1)
    x = h; n = 1
    while x + h < l:
        eps.append(C(x) / (B(x) - A(x) * eps[n]))
        eta.append((A(x) * eta[n] + D(x)) / (B(x) - A(x) * eps[n]))
        n += 1
        x += h

    t = [0] * (n + 1)
    t[n] = (pN - mN * eta[n]) / (kN + 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)
```

Результат работы программы

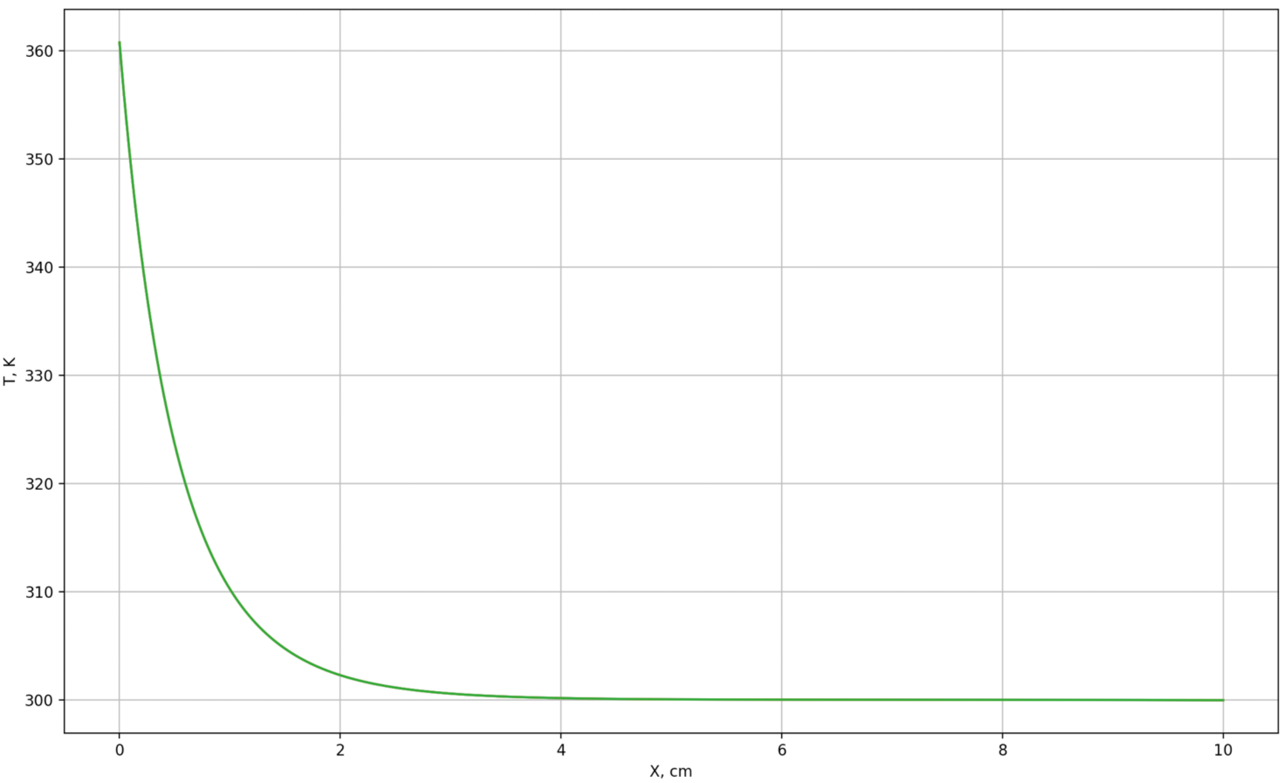
1.График зависимости температуры $T(x)$ от координаты x при заданных выше параметрах.



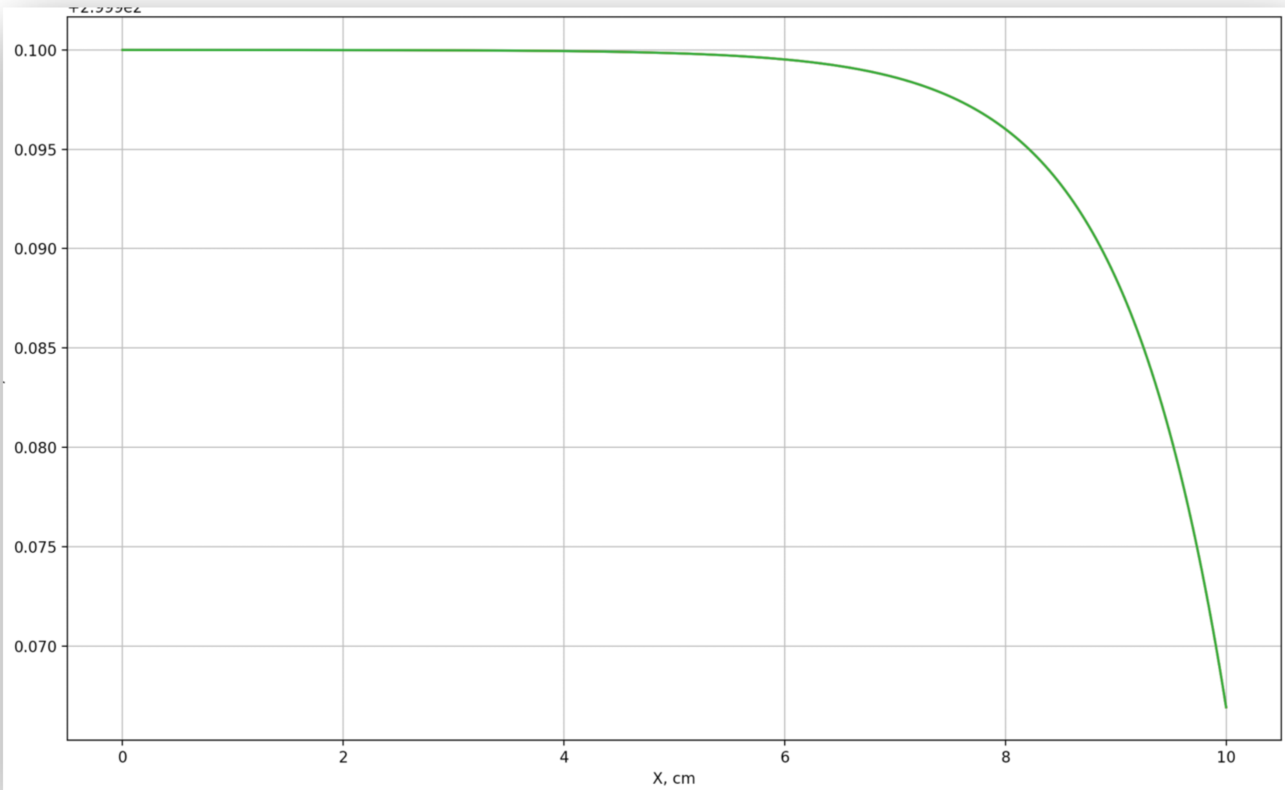
2. График зависимости $T(x)$ при $F_0 = -10 Bm/cм^2$.



3. График зависимости $T(x)$ при $3\alpha(x)$.



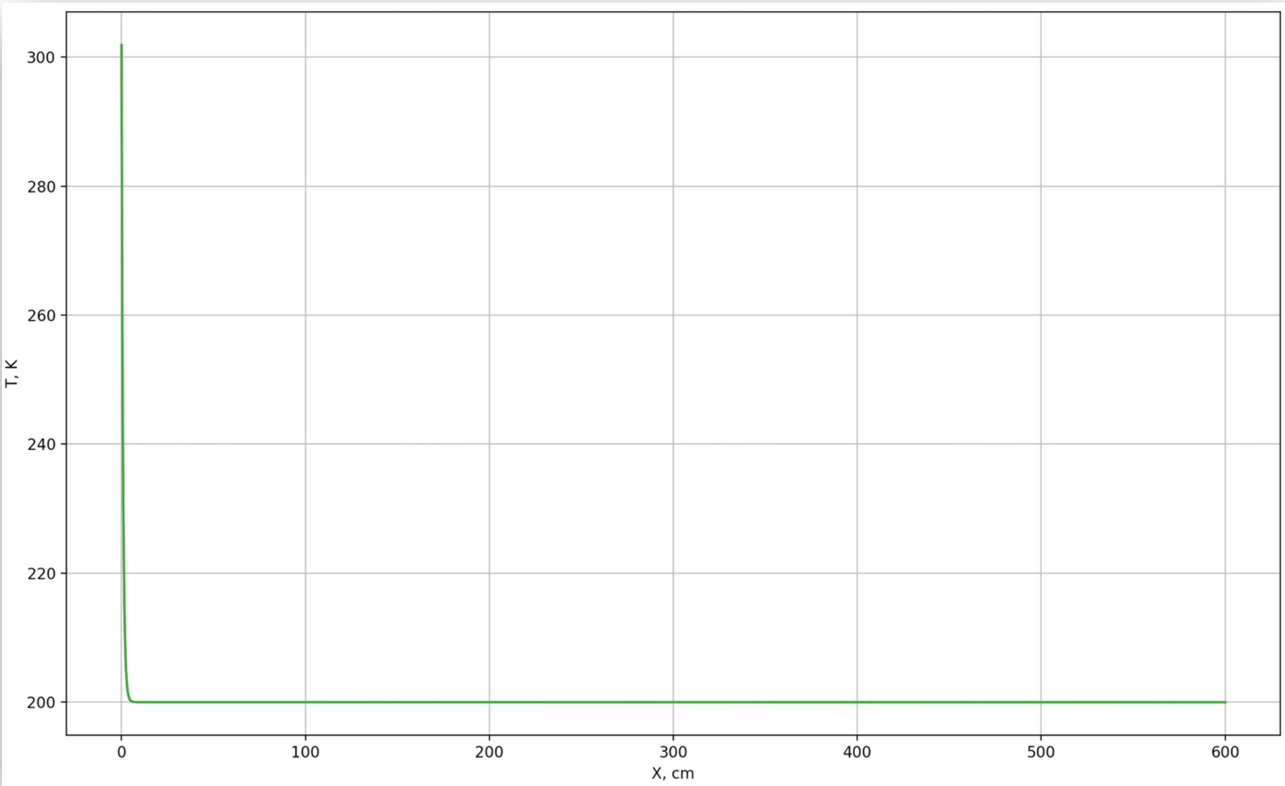
4. График зависимости $T(x)$ при $F_0 = 0$.



Ответы на вопросы

1. Какие способы тестирования программы можно предложить?

Можно увеличить длину стержня, температура должна стремиться к T_0.



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

$$-k(l)\frac{dT}{dx} = \alpha_n\left(T(l) - T_0\right) + \varphi(T)$$

где $\varphi(T)$ заданная функция.

Построим разностную схему методом разностной аппроксимации на равномерной сетке с шагом h.

Аппроксимируем производную:

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

При $x = l$:

$$\frac{dT}{dx} = \frac{T(l) - T(l-1)}{h}$$

Подставим в исходное уравнение:

$$-k(l)\frac{T(l)-T(l-1)}{h} = \alpha_n\Big(T(l)-T_0\Big) + \varphi(T(l))$$

$$-\Big(k(l)+\alpha_n h\Big)T(l) + k(l)T(l-1) = \varphi\Big(T(l)\Big)h - \alpha_n h T_0$$

3. Опишите алгоритм применения метода прогонки если при $x = 0$ краевое условие линейное, а при $x = l$, как во втором пункте.

При $x = 0$

$$\varepsilon_1 = -\frac{M_0}{K_0}$$

$$\eta_1 = -\frac{P_0}{K_0}$$

Прогоночные коэффициенты:

$$\varepsilon_{n+1} = \frac{C_n}{B_n - A_n\varepsilon_n}$$

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

Находим y_n с помощью решения конечного уравнения, полученного во втором пункте.

Обратным ходом найдем все коэффициенты до y_0.

Найдем значение y_n. Сделаем это, решив уравнение

$$h\varphi(y_N) + (k_N + h\alpha_N - k_N\varepsilon_N - k_N\eta_N)y_N - h_{\alpha N}T_0 = 0$$

Значения остальных y находятся по основной прогоночной формуле:

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

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 (лекция №8). Краевые условия линейные.

Пусть $i = p,\ 0 < p < n$

Рассмотрим правую прогонку, область которой $0 \le i \le p + 1$

Прогоночные коэффициенты α_i, β_i:

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

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

Рассмотри левую прогонку, область которой $p \le i \le n$

Прогоночные коэффициенты ε_i, η_i:

$$\varepsilon_i = \frac{C_i}{B_i - \varepsilon_{i+1} A_i}$$

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

Основные формулы левой и правой прогонки:

$$y_i = \alpha_{i+1} y_{\,i+1} + \beta_{i+1}$$

$$y_{i+1} = \varepsilon_{i+1} y\ + \eta_{i+1}$$

Учитывая их, составим систему линейных уравнений и найдем y при $i = p$

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

Заключение

В ходе лабораторной работы были получены навыки по разработке алгоритмов решения краевой задачи при реализации моделей, построенных на ОДУ второго порядка.