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

Дисциплина: Моделирование

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

Студент: Гасанзаде М.А.

Группа ИУ7-66Б

Оценка (баллы) _______________

Преподаватель : Градов В.М.

Москва.
2020 г.

СОДЕРЖАНИЕ

I. **АНАЛИТИЧЕСКАЯ ЧАСТЬ.**

Цель работы

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

Исходные данные

1. Уравнение для функции T(x):

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

2. Краевые условия: $\begin{cases} x=0, -k(0)\dfrac{dT}{dx}=F_0, \\ x=l, -k(l)\dfrac{dt}{dx}=\alpha_N(T(l)-T_0) \end{cases}$

3. Функции α(x), k(x) заданы своими константами.

$$\alpha(x)=\frac{c}{x-d} \quad ; \quad k(x)=\frac{a}{x-b} \quad .$$

Физический смысл задачи.

Сформулированная математическая модель описывает температурное поле T(x) вдоль цилиндрического стержня радиуса R и длиной l, причем R<<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),α(x) являются, соответственно, коэффициентами теплопроводности материала стержня и теплоотдачи при обдуве.

II. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

ЯП был выбран `Python3` из-за простоты работы с графиками и библиотеки `matplotlib`. Ниже на листингах будет представлена реализация программы:

Листинг

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


def plot_maker(masx, masy, xlabel, ylabel):
    plt.plot(masx, masy, color='r')
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    # plt.legend((name1, name2))
    plt.grid(True)
    plt.show()

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

    def alpha(x):
        return 3*x/(x - d)

    def P(Ax):
        return 2 * Ax / R

    def F(Ax):
        return (2 * T0 * Ax)/R

    def Xn_formula(x, h, flag):
        if flag == "+":
            res = 2 * k(x) * k(x + h) / (k(x) + k(x + h))
        if flag == "-":
            res = 2 * k(x) * k(x - h) / (k(x) + k(x - h))
        return res

    def An(x, h):
        res = 2 * k(x) * k(x - h) / (k(x) + k(x - h))
        return res/h

    def Bn(x, h, Ai, Ci):
        return Ai + Ci + P(x) * h

    def Cn(x, h):
        res = 2 * k(x) * k(x + h) / (k(x) + k(x + h))
        return res/h

    def Dn(x, h):
        return F(x) * h

    def get_K0(x0, h):
        pn_1_div_2 = (P(x0) + P(x0 + h)) / 2
        return Xn_formula(x0, h, "+") + h**2 * pn_1_div_2 / 8 + h**2 * P(x0)/4
```

```python
def get_M0(x0, h):
    pn_1_div_2 = (P(x0) + P(x0 + h)) / 2
    return -Xn_formula(x0, h, '+') + h**2 * pn_1_div_2 / 8

def get_P0(x0, h):
    fn_1_div_2 = (F(x0) + F(x0 + h)) / 2
    return h * F0 + h**2 * (fn_1_div_2 + F(x0)) / 4

def get_KN(x, h):
    res = 2 * k(x) * k(x - h) / (k(x) + k(x - h))
    return -P(x)*h/4 - (P(x-h) + P(x))*h/16 - alpha(x) - res/h

def get_MN(x, h):
    res = 2 * k(x) * k(x - h) / (k(x) + k(x - h))
    return res/h - (P(x-h) + P(x))*h/16

def get_PN(xn, h):
    return -alpha(xn) * T0 - h * (3*F(xn) + F(xn - h))/8

def progon(A, B, C, D, K0, M0, P0, KN, MN, PN):
    xi = [0]
    eta = [0]
    xi.append(-M0/K0)
    eta.append(P0/K0)
    for i in range(1, len(A)):
        xi.append(C[i]/(B[i] - A[i]*xi[-1]))
        eta.append((D[i] + A[i] * eta[-1])/(B[i] - A[i]*xi[-2]))
    y = [(PN - MN*eta[-1]) / (KN + MN * xi[-1])]
    for i in range(len(A) - 2, -1, -1):
        y.reverse()
    return y

k0 = 0.4
kN = 0.1
alpha0 = 0.05
alphaN = 0.01
l = 30
T0 = 300
R = 0.5
F0 = 50
h = 1e-3
x0 = 0

b = kN * l / (kN - k0)
a = - k0 * b
d = alphaN * l / (alphaN - alpha0)
c = - alpha0 * d

A = []
B = []
C = []
D = []
xmas = []

 for x in np.arange(x0, l + h, h):
    Ai, Ci, Di = An(x, h), Cn(x, h), Dn(x, h)
    Bi = Bn(x, h, Ai, Ci)
    A.append(Ai)
    B.append(Bi)
    C.append(Ci)
```

```
        D.append(Di)

    K0 = get_K0(x0, h)
    P0 = get_P0(x0, h)
    M0 = get_M0(x0, h)

    KN = get_KN(l, h)
    PN = get_PN(l, h)
    MN = get_MN(l, h)

    dots = progon(A, B, C, D, K0, M0, P0, KN, MN, PN)
    plot_maker(xmas[1:], dots[1:], 'Длина стержня, см', 'Температура, К')
```

Далее, в экспериментальной части, тестирование будет производиться по этим данным:

$k_0 = 0.4\, Вт/см\, K$,

$k_n = 0.1\, Вт/см\, K$,

$\alpha_0 = 0.05\, Вт/см^2\, K$,

$\alpha_N = 0.01\, Вт/см^2\, K$,

$l = 10\, см$,

$T_0 = 300\, K$,

$R = 0.5\, см$,

$F_0 = 50\, Вт/см^2$.

III. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

В данном разделе будет рассмотрен вывод программы и представлены графики зависимостей.

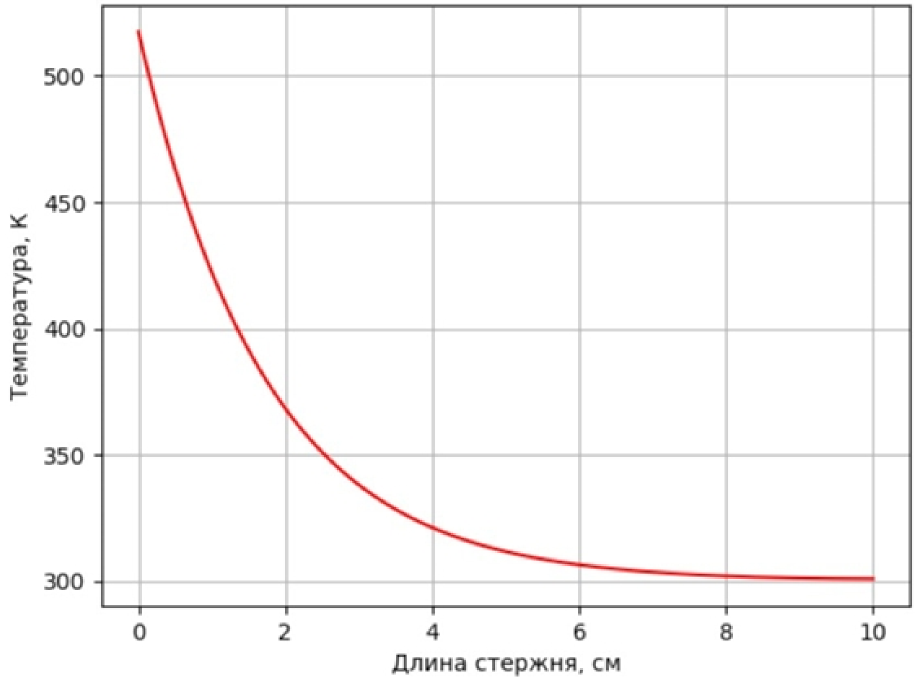

Рисунок 1. График зависимости температуры $T(x)$ от координаты при заданных выше параметрах.

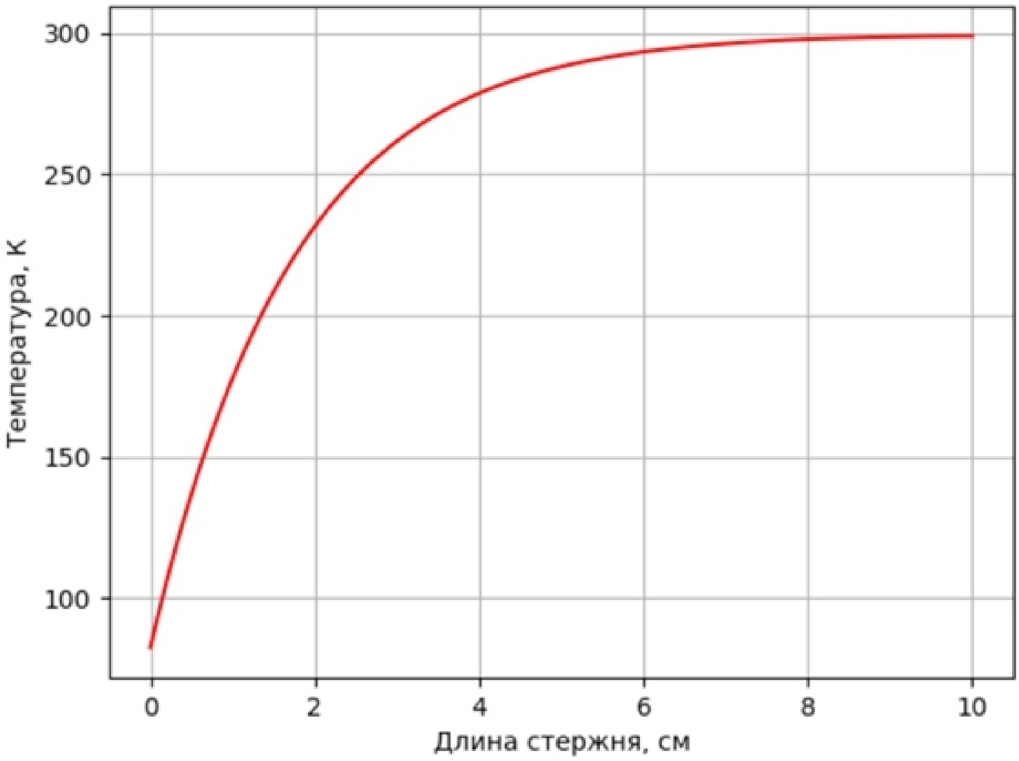

Рисунок 2. при F_0 = -50 Вт/см^2.

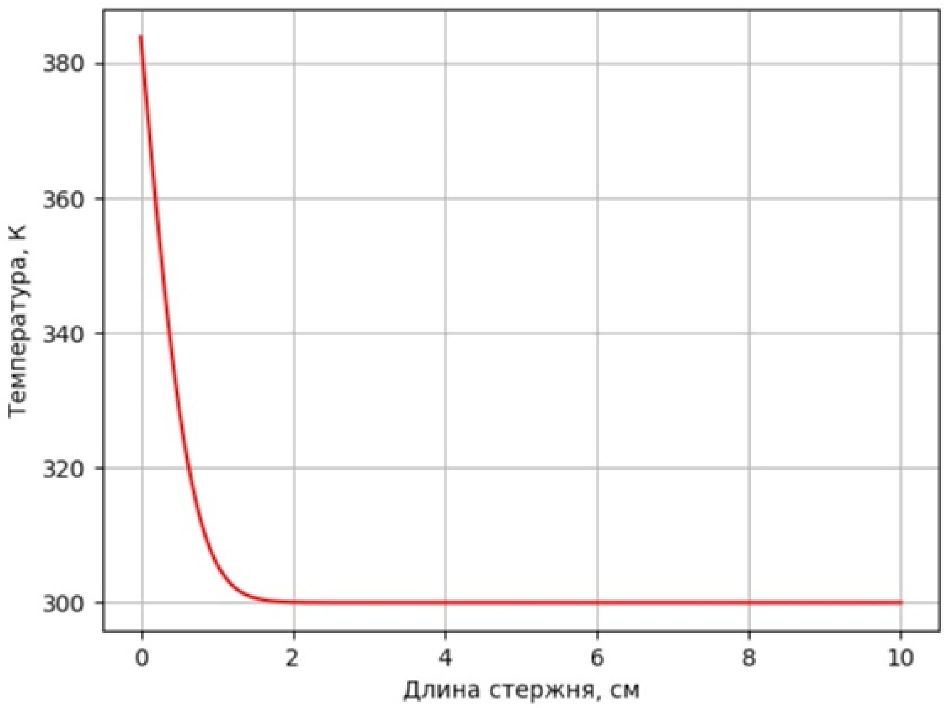

Рисунок 3. График зависимости $T(x)$ при увеличенных значениях $\alpha(x)$ в 3 раза.

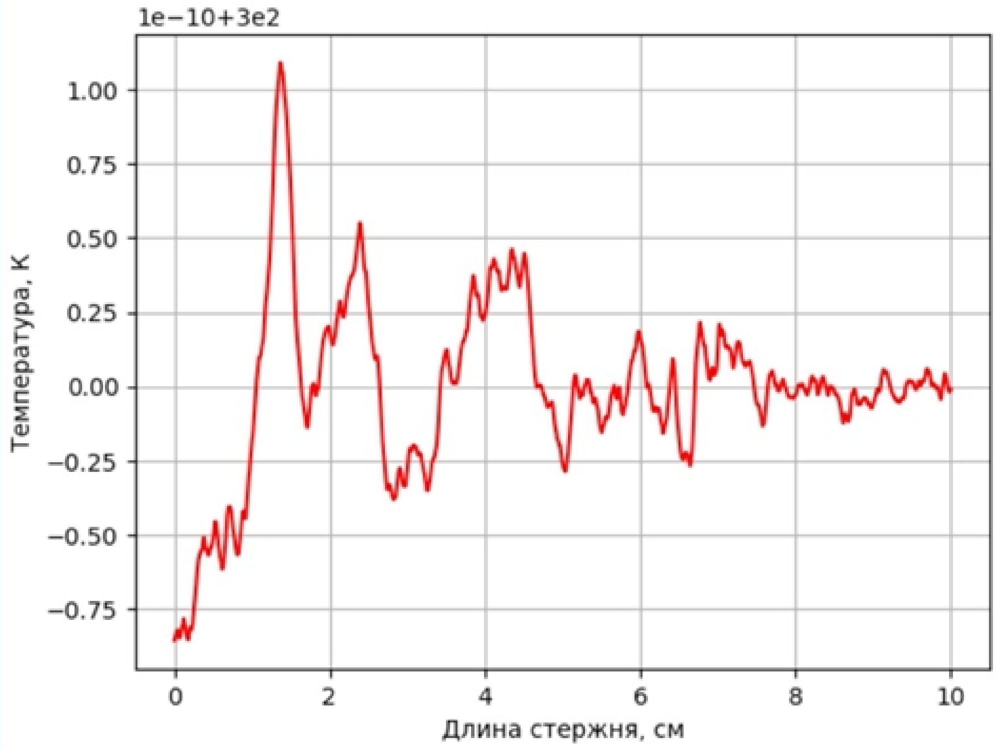

Рисунок 4. График зависимости $T(x)$ при $F_0 = 0$ Вт/см^2.

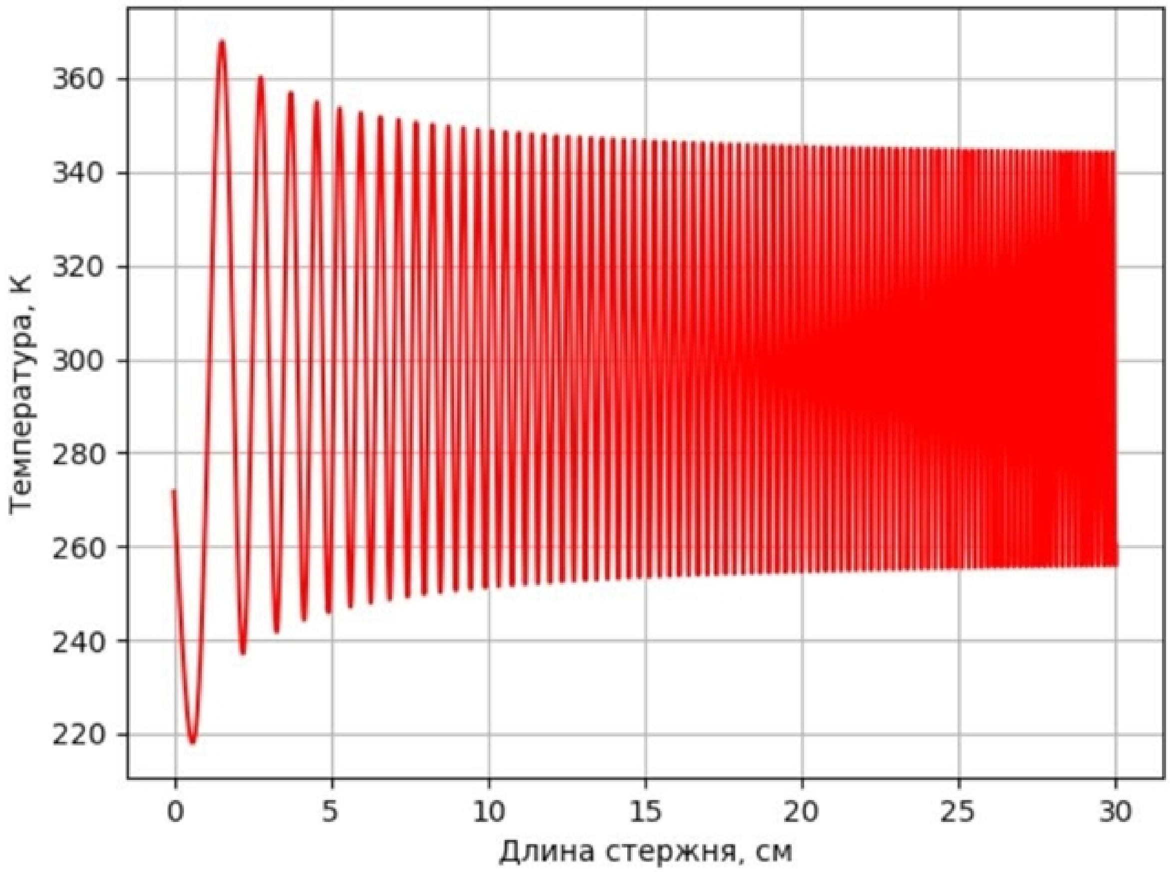

Рисунок 5. Гармонические колебания при R < 0 см и l = 30 см.

IV. ОТВЕТЫ НА ВОПРОСЫ

1. Какие способы тестирования программы можно предложить?

1) При $F_0 = 0$ $T(x) = T_0 \pm \varepsilon$, где ε – погрешность
2) Должна быть положительная производная функции T(x) при $F_0 < 0$
3) При отрицательном радиусе стержня R<0, должны наблюдаться гармонические колебания.

2. Получите простейший разностный аналог нелинейного краевого условия при $\boldsymbol{x=l,\ \ -k(l)\frac{dT}{dx}=\alpha_N(T(l)-T_0)+\phi(T)}$:

Разностная аппроксимация краевого условия

$$\frac{Y_{N-1}-Y_N}{h}k_N=\alpha_N(y_N-T_0)+\phi(y_N)$$

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

Используя простейшую аппроксимацию первых производных односторонними разностями, получим:

$$\xi_1=,\quad \eta_1=\frac{F_0 h}{k_0}$$

Далее, найдем прогоночные коэффициенты:

$$\xi_{n+1}=\frac{C_n}{B_n-A_n\xi_n},\quad \eta_{n+1}=\frac{F_n+A_n\eta_n}{B_n-A_n\eta_n}$$

Учитывая, что $y_{n-1}=\xi_n y_n+\eta_n$, найдём:

$$y_N=\frac{k_N\eta_N+h\alpha\beta-h\phi(y_N)}{k_N(1-\xi_N)+h\alpha}$$

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

$$\eta_1=\frac{F_0}{B_0};\quad \eta_N=\frac{A_N}{B_N}$$

$$\xi_1=\frac{C_0}{B_0};\quad \xi_N=\frac{F_N}{B_N}$$

Прямой ход $(1\le i\le p-1)$:

$$\xi_{i+1}=\frac{C_i}{B_i-A_i\xi_i};$$

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

Обратный ход $\left(p \le i \le N-1\right)$:

$$\hat{\xi}_i = \frac{A_i}{B_i - C_i \hat{\xi}_{i+1}}$$

$$\hat{\eta}_i = \frac{F_i + C_i \hat{\eta}_{i+1}}{B_i - C_i \hat{\xi}_{i+1}}$$

$$\left.\begin{array}{l} y_{p-1} = \xi_p y_p + \eta_p \\ y_{p+1} = \hat{\xi}_p y_p + \hat{\eta} \\ A_p y_{p-1} B_p y_p + C_p y_{p+1} = -P_p \end{array}\right\} \Rightarrow y_p = \frac{F_p + A_p \eta_p + C_p \hat{\eta}_{p+1}}{B_p - A_p \xi_p - C_p \hat{\xi}_{p+1}}$$

ЗАКЛЮЧЕНИЕ

При увеличении тепло съема и неизменном потоке F_0 уровень температур T (x) снижается, а градиент увеличивается (при сравнении рис. 1 и 3). Также на рис. 4 можно наблюдать, что, в отсутствии теплового нагружения, температура стержня равна окружающей температуре, погрешность определяется приближенным характером вычислений.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Градов В.М. Методические указания: «30-03-2020-Задание_на_лаб_раб_№3.doc» (дата обращения 01.05.2020)
2. Matplotlib URL: https://matplotlib.org (дата обращения 10.04.2020)
3. Градов В.М. Компьютерные технологии в практике математического моделирования часть 2 URL:http://ebooks.bmstu.ru/secret/html/bikqxzugca/files/assets/basic-html/page-1.html (дата обращения 01.05.2020)
4. Градов В.М. Лекция №8 «30-03-2020-Лекция_№8_Модели_ОДУ_краевая_задача.pdf» (дата обращения 01.05.2020)