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 Сушина А.Д.

Группа ИУ7-61б

Оценка (баллы) _______________

Преподаватель Градов В.М.

Москва.
2020 г

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

1 Исходные данные

1. 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

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

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

$$k(x)=\frac{a}{x-b}, \quad \alpha(x)=\frac{c}{x-d} \qquad (3, 4)$$

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

Величины k_0, k_N, α_0, α_N задает пользователь.

Получаем:

$$a=-bk_0 \qquad (5)$$

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

$$c=-\alpha_0 d \qquad (7)$$

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

3. Разностная схема с разностным краевым условием при $x=0$.

$$A_n y_{n+1} - B_n y_n + C_n y_{n-1} = -D_n, 1 \le n \le N-1 \quad (9)$$

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

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

$$B_n = A_n + C_n + p_n h$$

$$D_n = f_n h$$

Система (9) совместно с краевыми условиями решается методом прогонки.

Для величин численно вычисляя $\chi_{n\pm 1/2}$ можно получить различные приближенные выражения, интеграл методом трапеций или методом средних. Выражения для приближений:

$$\chi_{n\pm 1/2} = \frac{2k_n k_{n\pm 1}}{k_n + k_{n+1}} \quad X_{n\pm\frac{1}{2}} = \frac{k_n + k_{n\pm 1}}{2} \quad (10)$$

Разностный аналог краевого условия при x=0:

$$y_0 = \frac{X_{\frac{1}{2}} - \frac{h^2}{8} p_{\frac{1}{2}}}{X_{\frac{1}{2}} + \frac{h^2}{8} p_{\frac{1}{2}} + \frac{h^2}{4} p_0} y1 + \frac{h F_0 + \frac{h^2}{4}\left(f_{\frac{1}{2}} + f_0\right)}{X_{\frac{1}{2}} + \frac{h^2}{8} p_{\frac{1}{2}} + \frac{h^2}{4} p_0} \quad (11)$$

Простая аппроксимация:

$$p_{\frac{1}{2}} = \frac{p_0 + p_1}{2} \quad (12)$$

$$f_{\frac{1}{2}} = \frac{f_0 + f_1}{2} \quad (13)$$

Учесть, что поток $F_N = \alpha_N (y_N - T_0)$, а $F_{N-1/2} = \chi_{N-1/2} \frac{y_{N-1} - y_N}{h}$. (14, 15)

4. Значения параметров для отладки (все размерности согласованы)

$k_0 = 0.4$ Вт/см К,

$k_N = 0.1$ Вт/см К,

$\alpha_0 = 0.05$ Вт/см^2 К,

$\alpha_N = 0.01$ Вт/см^2 К,

$l = 10$ см,

$T_0 = 300$К,

$R = 0.5$ 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 $R << l$ 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.

2 Код программы

Код программы представлен на листингах 1-2.

Листинг 1. functions.py

```
from data import data, graph1
import numpy as np

b = data['kN'] * data['l'] / (data['kN'] - data['k0'])
a = - b * data['k0']
d = data['AlphaN'] * data['l'] / (data['AlphaN'] - data['Alpha0'])
c = - b * data['Alpha0']
h = data['h']
N = data['l']

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

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

def P(x):
    return 2/data['R'] * Alpha(x)

def f(x):
    return 2*data['t0']/data['R'] *Alpha(x)

def KHIminus12(x):
    return 2*k(x)*k(x-h)/(k(x) + k(x-h))

def KHIplus12(x):
    return 2*k(x)*k(x+h)/(k(x) + k(x+h))

def A(n):
    return KHIplus12(n)/h

def C(n):
    return KHIminus12(n)/h

def B(n):
    return A(n) + C(n) + P(n)*h

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


def calculate():
    xstart = 0
    xend = data['l']
```

```python
#левые
p0 = P(xstart)
p1 = P(xstart + h)
p12 = (p0+p1)/2

f0 = f(xstart)
f1 = f(xstart+h)
f12 = (f0+f1)/2

M0 = KHIplus12(xstart) - h*h/8*p12
K0 = KHIplus12(xstart) + h*h/8*p12 + h*h/4*p0
P0 = h*data['F0'] + h*h/4*(f12 + f0)

#правые

pn = P(xend)
pn1 = P(xend - h)
pn12 = (pn+pn1)/2

fn = f(xend)
fn1 = f(xend-h)
fn12 = (fn+fn1)/2

Mn = -KHIminus12(xend) + h * h * pn12 / 8
Kn = KHIminus12(xend) + data['AlphaN'] * h + h * h * pn12 / 8 - h * h * pn / 4;
Pn = data['AlphaN'] * data['t0'] * h - h * h * (fn12 + fn) / 4;

eps = [0, M0/K0 ]
eta = [0, P0/K0]

x = h
n = 1
while x + h < N:
    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]

graph1[0] = [i for i in np.arange(0, N, h)]
graph1[1] = t[:-1]
```

Листинг 2. main.py

```python
import matplotlib.pyplot as plt
```

```python
import data as const
from functions import calculate

from tkinter import *
root = Tk()

varList = {
"k0": StringVar(),
"kN": StringVar(),
"Alpha0": StringVar(),
"AlphaN": StringVar(),
"l": StringVar(),
"t0": StringVar(),
"R": StringVar(),
"F0": StringVar(),
"h": StringVar()
}

def create_grid(root):
    i = 0
    for var in varList.keys():
        label = Label(root, text=var)
        label.grid(row=i, column=0, sticky="e")
        entry = Entry(root,width=10,textvariable=varList[var])
        entry.grid(row=i, column=1)
        entry.insert(0, str(const.resetData[var]))
        i+=1

def check_is_num():
    for var in varList.values():
        try:
            float(var.get())
        except ValueError:
            return False
    return True

def clear_graphs():
    const.graph1[0].clear()
    const.graph1[1].clear()

def start_work(Event):
    clear_graphs()
    if not check_is_num():
        print("WARNING NOT DIGIT")
        return
    for var in varList.keys():
        const.data[var] = float(varList[var].get())
    calculate()
    plt.title('T(x)')
    plt.plot(const.graph1[0], const.graph1[1])
```

```
    plt.xlable("x, sm")
    plt.ylabel("T, K")
    plt.show()

if __name__ == '__main__':
    btn = Button(root, text="START")
    create_grid(root)

    btn.bind("<Button-1>", start_work)
    btn.grid(column=1, padx=10, pady=10)
    root.mainloop()
```

3 Результаты работы

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

Обозначим $F = -k(x)\frac{dT}{dx}$.

Запишем (1) в виде:

$$\frac{d}{dx}\left(k(x)\frac{dT}{dx}\right) - p(x)T + f(x) = 0 \quad (16)$$

где

$$p(x) = \frac{2}{R}\alpha(x)$$

$$f(x) = \frac{2T_0}{R}\alpha(x)$$

Проитегрируем (15) на отрезке [$x_{N-1/2}$, x_N]

$$-\int_{X_{N-\frac{1}{2}}}^{X_N} \frac{dF}{dx}dx - \int_{X_{N-\frac{1}{2}}}^{X_N} p(x)u\,dx + \int_{X_{N-\frac{1}{2}}}^{X_N} f(x)dx = 0 \quad (17)$$

Применим метод трапеций. Получим:

$$-\left(F_N - F_{N-\frac{1}{2}}\right) - \frac{h}{4}\left(p_N y_N + p_{N-\frac{1}{2}} y_{N-\frac{1}{2}}\right) + \frac{h}{4}\left(f_N + f_{N-\frac{1}{2}}\right) = 0 \quad (18)$$

Применив (14, 15) к (18), получим

$$-\left(a_N(y_N - T_0) - X_{N-\frac{1}{2}}\frac{y_{N-1} - y_N}{h}\right) - \frac{h}{4}\left(p_N y_N + p_{N-\frac{1}{2}}\frac{y_N + y_{N-1}}{2}\right) + \frac{h}{4}\left(f_N + f_{N-\frac{1}{2}}\right) = 0$$

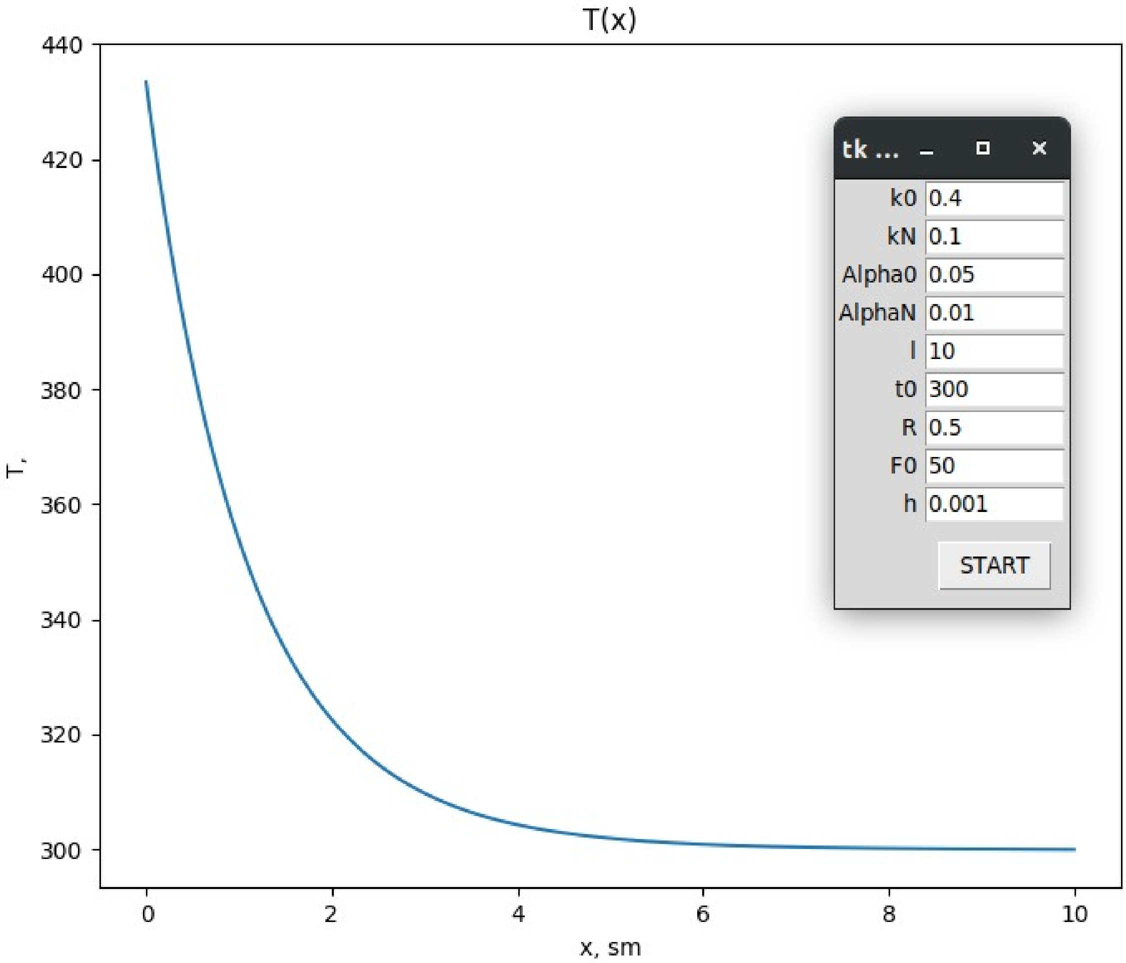
$$-a_N y_N + a_N T_0 + \frac{X_{N-1/2}\,y_{n-1}}{h} - \frac{X_{N-1/2}\,y_n}{h} - \frac{h}{4}p_N y_N - \frac{h}{8}p_{N-1/2}y_N - \frac{h}{8}y_{N-1}p_{N-1/2} - \frac{h}{4}(f_N + f_{N-1/2}) = 0$$

$$-a_N y_N h + a_N T_0 h + X_{N-1/2}\,y_{n-1} - X_{N-1/2}\,y_n - \frac{h^2}{4}p_N y_N - \frac{h^2}{8}p_{N-1/2}y_N - \frac{h^2}{8}y_{N-1}p_{N-1/2} - \frac{h^2}{4}(f_N + f_{N-1/2}) = 0$$

$$\left(X_{1-1/2}-\frac{h^2}{8}p_{N-1/2}\right)y_{N-1}+\left(-a_N h-X_{N-1/2}-\frac{h^2}{4}p_N-\frac{h^2}{8}p_{N-1/2}\right)y_N+a_N T_0 h+\frac{h^2}{4}(f_N+f_{N-1/2})=0$$

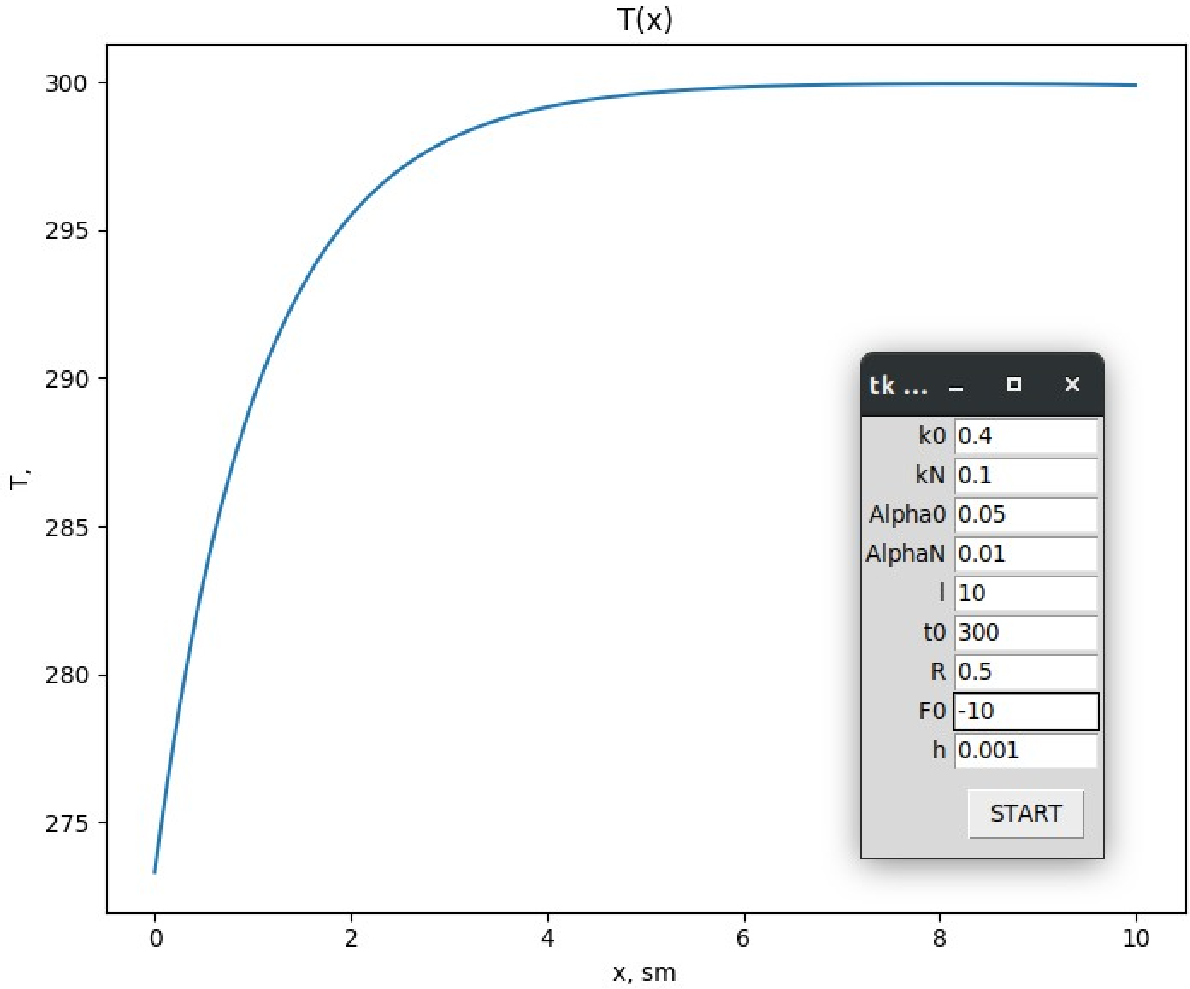
$$y_N=\frac{\left(X_{N-1/2}-\frac{h^2}{8}p_{N-1/2}\right)}{\left(a_N h+X_{N-1/2}+\frac{h^2}{4}p_N+\frac{h^2}{8}p_{N-1/2}\right)}y_{N-1}+\frac{a_N T_0 h+\frac{h^2}{4}(f_N+f_{N-1/2})}{\left(a_N h+X_{N-1/2}+\frac{h^2}{4}p_N+\frac{h^2}{8}p_{N-1/2}\right)} \quad (19)$$

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



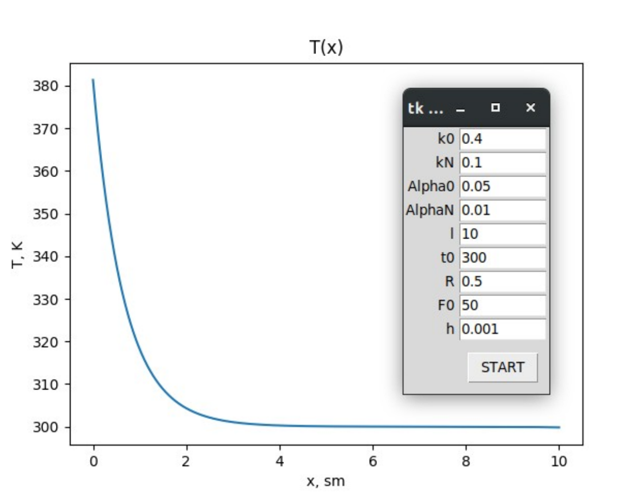
3. График зависимости $T(x)$ при F_0 =-10 Вт/см^2.

Справка. При отрицательном тепловом потоке слева идет съем тепла, поэтому производная $T^{'}(x)$ должна быть положительной.

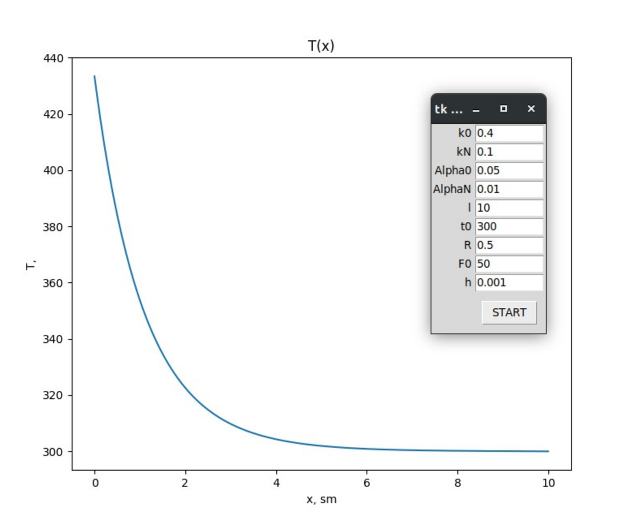


4. График зависимости $T(x)$ при увеличенных значениях $\alpha(x)$ (например, в 3 раза). Сравнить с п.2.

Справка. При увеличении теплосъема и неизменном потоке F_0 уровень температур $T(x)$ должен снижаться, а градиент увеличиваться.



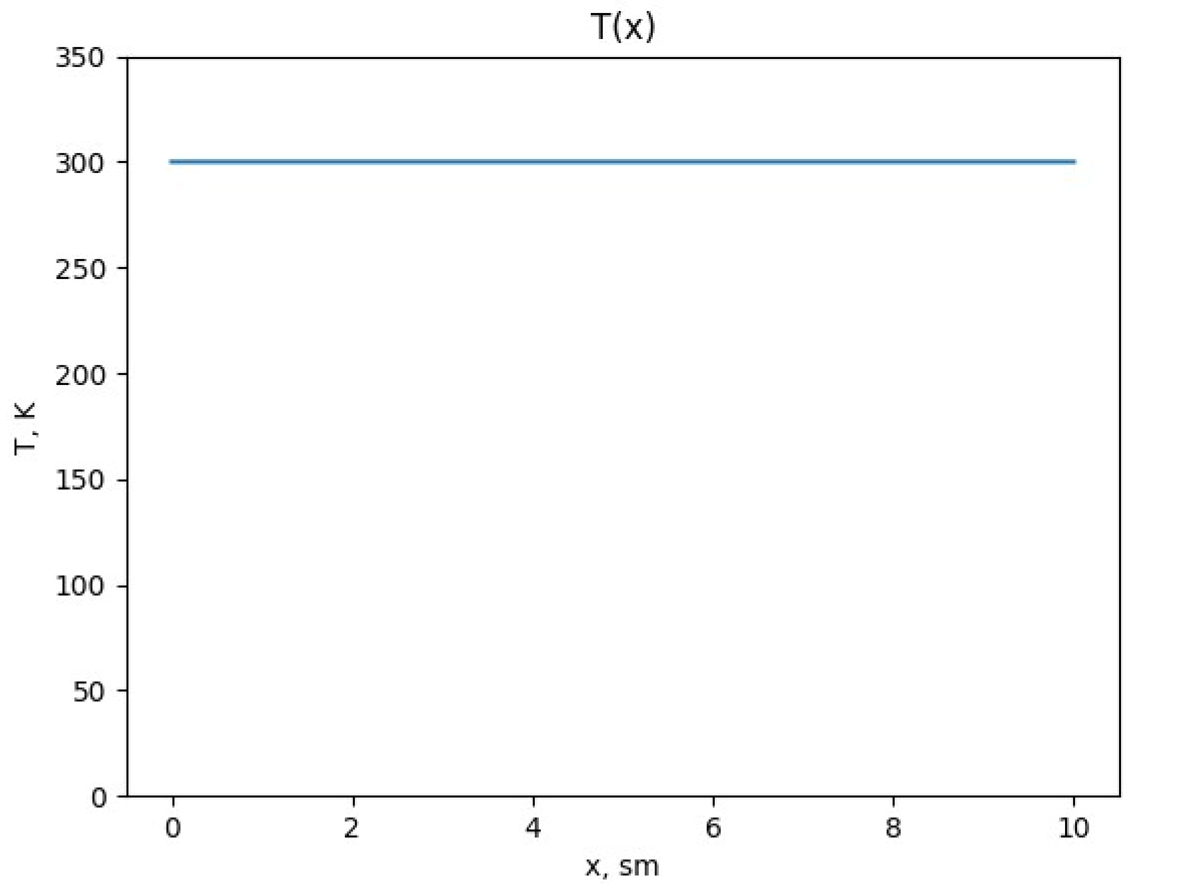
$\alpha(x)$ увеличено в 3 раза



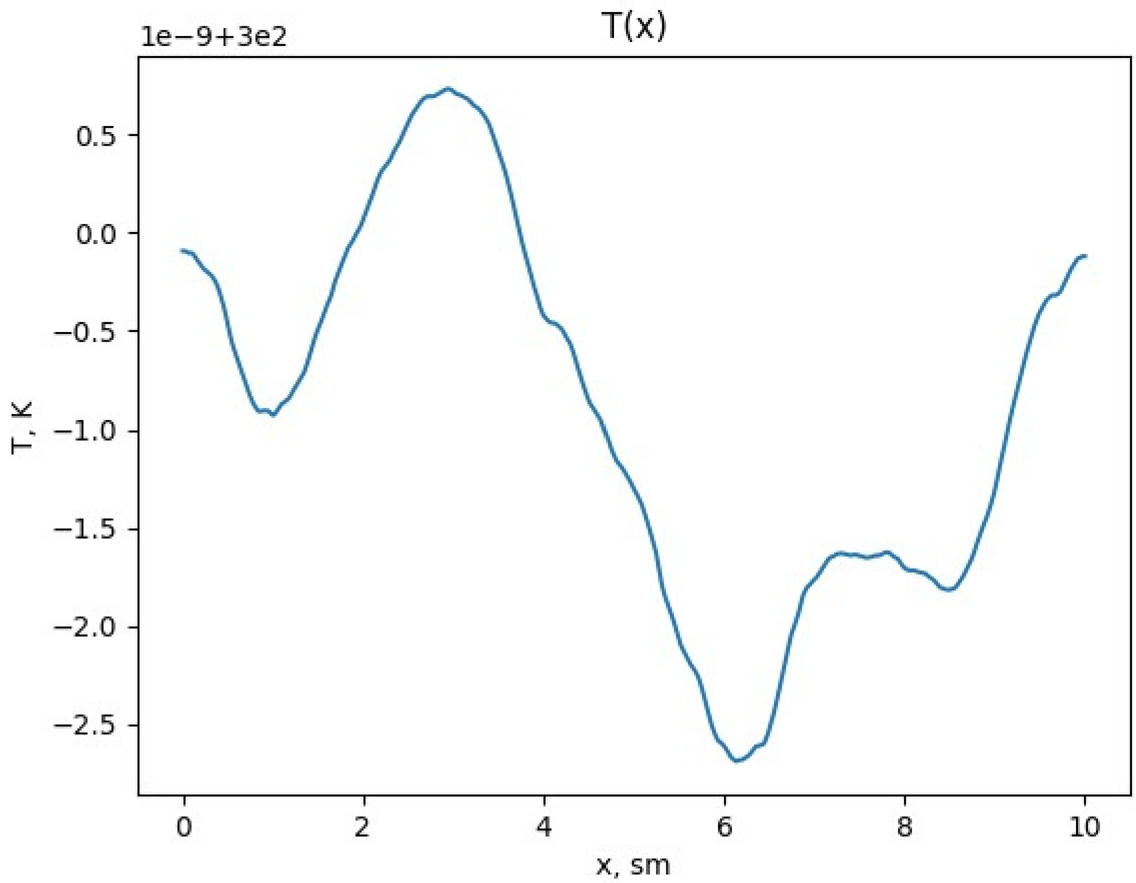
Пункт 2

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

Справка. В данных условиях тепловое нагружение отсутствует, причин для нагрева нет, температура стержня должна быть равна температуре окружающей среды T_0 (разумеется с некоторой погрешностью, определяемой приближенным характером вычислений).



Погрешность:



4 Вопросы при защите лабораторной работы

Ответы на вопросы дать письменно в Отчете о лабораторной работе.

1. Какие способы тестирования программы можно предложить?

Программа должна выводить результаты, соответствующие законам физики. Это можно определить по виду графиков при определенных значениях входных параметров. Отталкиваясь от физического смысла задачи, можно представить как должен вести себя результат. Например, при F0=0 температура должна быть равна температуре окружающей среды T0. При отрицательном тепловом потоке слева идет съем тепла, поэтому производная $T'(x)$ должна быть положительной.

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

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

где $\varphi(T)$ - заданная функция.

Производную аппроксимируйте односторонней разностью.

$$-k_N\frac{y_N-y_{N-1}}{h}=\alpha_N\left(y_N-T_0\right)+\varphi\left(y_N\right) \quad (20)$$

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.

Будет использоваться правая прогонка.

$$y_n=\varepsilon_{n+1}y_{n+1}+\eta_{n+1} \qquad (21)$$

Из краевого условия при x = 0 получим:

$$y_0=y_1+\frac{hF_0}{k_0} \qquad (22)$$

отсюда начальные прогоночные коэффиценты:

$$\varepsilon_1=1 \qquad (23)$$

$$\eta_1=\frac{hF_0}{k_0} \qquad (24)$$

Прогоночные коэффициенты:

$$\varepsilon_{n+1}=\frac{C_n}{B_n-A_n\varepsilon_n} \qquad (25)$$

$$\eta_{n+1}=\frac{D_n+A_n\eta_n}{B_n-A_n\varepsilon_n} \qquad (26)$$

Исходя из того, что

$$y_{n-1}=\varepsilon_n y_n+\eta_n \qquad (27)$$

найдем y_n:

В (20) подставим (27):

$$-k_N\frac{y_N-\left(\varepsilon_N y_N+\eta_N\right)}{h}=\alpha_N y_N-\alpha_N T_0+\varphi\left(y_N\right)$$

$$-k_N y_N+\varepsilon_N k_N y_N+k_N\eta_N=h\alpha_N y_N-h\alpha_N T_0+h\varphi\left(y_N\right)$$

$$y_N\left(-k_N+k_N\varepsilon_N-h\alpha_N\right)+\eta_N+h\alpha_N T_0-h\varphi\left(y_N\right)=0 \qquad (28)$$

Таким образом, два этапа:

Прямой ход: по 23, 24 вычисляем начальные значения коэфицентов, по 25, 26 вычисляем массивы прогоночных коэфицентов.

Обратный ход: по (28) определяем y_N. Далее по 27 вычисяем все значения y.

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.

Левая прогонка:

$$y_n=\alpha_{n-1}y_{n-1}+\beta_{n-1}, 0\le n\le p \tag{29}$$

Прогоночные коэффициенты:

$$\alpha_{n-1}=\frac{C_n}{B_n-A_n\alpha_n} \tag{30}$$

$$\beta_{n-1}=\frac{D_n+A_n\beta_n}{B_n-A_n\alpha_n} \tag{31}$$

Объединив левую и правую (21), (25), (26) прогонки, получим систему:

$$\begin{cases} y_p=\varepsilon_{p+1}y_{p+1}+\eta_{p+1} \\ y_{p+1}=\alpha_p y_p+\beta_p \end{cases} \tag{32}$$

Подставив второе уравнение в первое:

$$y_p=\varepsilon_{p+1}(\alpha_p y_p+\beta_p)+\eta_{p+1}$$

$$y_P-\varepsilon_{p+1}\alpha_p y_p=\varepsilon_{p+1}\beta_p+\eta_{p+1}$$

$$y_p=\frac{\varepsilon_{p+1}\beta_p+\eta_{p+1}}{1-\varepsilon_{p+1}\alpha_p} \tag{33}$$