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»

О Т Ч Е Т

по лабораторной работе № 0 4

Дисциплина: ***Моделирование***

Студент	***ИУ7И-66Б***		**Нгуен Ф. С.**
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			**Градов В. М.**
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2021

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

Исходные данные.

1. Задана математическая модель.

Уравнение для функции $T(x,t)$

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

Краевые условия

$$\left\{\begin{array}{c} t = 0, T(x,0) = T_0 \\ x = 0, -k(T(0))\,\frac{\partial T}{\partial x} = F_0 \\ x = l, -k(T(l))\,\frac{\partial T}{\partial x} = \alpha_N\,(T(l) - T_0) \end{array}\right\} \qquad (2)$$

В обозначениях уравнения (14.1) лекции №14

$$p(x) = \frac{2}{R}\,\alpha(x)\,, f(u) \equiv f(x) = \frac{2T_0}{R}\,\alpha(x) \qquad (3)$$

2. Разностная схема с разностным краевым условием при $x = 0$ ***получена в Лекции может быть использована в данной работе.***

Разностная схема:

$$\hat{A_n} y_{\hat{n-1}} - \hat{B_n} y_{\hat{n}} + \hat{D_n} y_{\hat{n+1}} = -\hat{F_n}, 1 \le n \le N-1 \qquad (4)$$

$$\hat{A_n} = X_{\hat{n-\frac{1}{2}}}\frac{\tau}{h} \qquad (5)$$

$$\hat{B_n} = \hat{A_n} + \hat{D_n} + \hat{c_n} h + p_n h\tau \qquad (6)$$

$$\hat{D_n} = X_{\hat{n+\frac{1}{2}}}\frac{\tau}{h} \qquad (7)$$

$$\hat{F_n} = f_n \tau h + \hat{c_n} y_n h \qquad (8)$$

Разностные аналоги краевых условий при x=0:

$$\left(\frac{h}{8}\hat{c_{\frac{1}{2}}} + \frac{h}{4}\hat{c_0} + \hat{X_{\frac{1}{2}}}\frac{\tau}{h} + \frac{\tau h}{8}p_{\frac{1}{2}} + \frac{\tau h}{4}p_0\right)\hat{y_0} + \left(\frac{h}{8}\hat{c_{\frac{1}{2}}} - \hat{X_{\frac{1}{2}}}\frac{\tau}{h} + \frac{\tau h}{8}p_{\frac{1}{2}}\right)\hat{y_1} =$$

$$= \frac{h}{8}\hat{c_{\frac{1}{2}}}(y_0 + y_1) + \frac{h}{4}\hat{c_0} y_0 + \hat{F}\tau + \frac{\tau h}{4}\left(\hat{f_{\frac{1}{2}}} + \hat{c_0}\right) \qquad (9)$$

Самостоятельно надо получить интегро-интерполяционным методом разностный аналог краевого условия при $x = l$*. Для этого надо проинтегрировать на отрезке [xN-1/2, xN] выписанное выше уравнение (1) и учесть, что поток*

$$F_N = \alpha_N(\hat{y_N} - T_0) \qquad (10)$$

$$F_{N-\frac{1}{2}} = X_{\hat{N-\frac{1}{2}}}\frac{y_{\hat{N-1}} - y_N}{h} \qquad (11)$$

3. Значения параметров для отладки (все размерности согласованы)

$k(T) = a_1(b_1 + c_1 T^{m_1})$, *Вт/см К,*

$c(T) = a_2 + b_2 T^{m_2} - \frac{c_2}{T^2}$, *Дж/см3К.*

a_1=0.0134, b_1=1, c_1=4.35 10-4, m_1=1,

a_2=2.049, b_2=0.563 10-3, c_2=0.528 105, m_2=1.

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

α_0 =0.05 Вт/см2 К,

α_N =0.01 Вт/см2 К,

l = 10 см,

T_0 =300К,

R =0.5 см,

$F(t)$ =50 Вт/см2 (для отладки принять постоянным).

4. Физическое содержание задачи

Постановки задач в данной лабораторной работе и работе №3 во многом совпадают. Отличия заключаются в следующем:

1. Сформулированная в данной работе математическая модель описывает нестационарное температурное поле $T(x,t)$, зависящее от координаты x и меняющееся во времени.

2. Свойства материала стержня привязаны к температуре, т.е. теплоемкость и коэффициент теплопроводности $c(T), k(T)$ зависят от T.

3. При $x = 0$ цилиндр нагружается тепловым потоком $F(t)$, в общем случае зависящим от времени.

Если в настоящей работе задать поток постоянным, т.е. $F(t)$=const, то будет происходить формирование температурного поля от начальной температуры T_0 до некоторого установившегося (стационарного) распределения $T(x,t)$. Это поле в дальнейшем с течением времени меняться не будет Это полезный факт для тестирования программы.

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

При произвольной зависимости потока $F(t)$ от времени температурное поле будет как-то сложным образом отслеживать поток.

***Замечание**. Варьируя параметры задачи, следует обращать внимание на то, что решения, в которых температура превышает примерно 2000К, физического смысла не имеют и практического интереса не представляют.*

5. Разностная схема

$$\begin{cases} \widehat{K}_0\hat{y}_0 + \widehat{M}_0\hat{y}_1 = \hat{P}_0 \\ \hat{A}_n\hat{y}_{n-1} - \hat{B}_n\hat{y}_n + \widehat{D}_n\hat{y}_{n+1} = -\hat{F}_n, \qquad 1 \le n \le N-1 \\ \widehat{K}_N\hat{y}_N + \widehat{M}_N\hat{y}_{N-1} = \hat{P}_N \end{cases} \tag{12}$$

Где

$$\hat{A}_n = \hat{\chi}_{n-\frac{1}{2}}\frac{\tau}{h}$$

$$\widehat{D}_n = \hat{\chi}_{n+\frac{1}{2}}\frac{\tau}{h}$$

$$\hat{B}_n = \hat{A}_n + \widehat{D}_n + \hat{c}_n h + p_n h\tau$$

$$\hat{F}_n = f_n h\tau + \hat{c}_n y_n h$$

Для величин $\hat{\chi}_{n\pm\frac{1}{2}}$ можно получить различные приближенные выражения, численно вычисляя интеграл методом трапеций

$$\hat{\chi}_{n\pm\frac{1}{2}} = \frac{2\hat{k}_n\hat{k}_{n\pm1}}{\hat{k}_n + \hat{k}_{n\pm1}} \tag{13}$$

6. Краевые условия

Обозначим

$$F = -k(T)\frac{\partial T}{\partial x} \tag{14}$$

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

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

<u>Разностные аналоги краевых условий при x = 0</u>

$$\left(\frac{h}{8}\hat{c}_{\frac{1}{2}} + \frac{h}{4}\,\hat{c}_0 + \hat{\chi}_{\frac{1}{2}}\frac{\tau}{h} + \frac{\tau h}{8}\,p_{\frac{1}{2}} + \frac{\tau h}{4}p_0\right)\hat{y}_0 + \left(\frac{h}{8}\,\hat{c}_{\frac{1}{2}} - \hat{\chi}_{\frac{1}{2}}\frac{\tau}{h} + \frac{\tau h}{8}p_{\frac{1}{2}}\right)\hat{y}_1 = \frac{h}{8}\,\hat{c}_{\frac{1}{2}}(y_0 + y_1) + \frac{h}{4}\,\hat{c}_0\,y_0 + \hat{F}\,\tau + \frac{\tau h}{4}\,(\hat{f}_{\frac{1}{2}} + \hat{f}_0)$$

Тогда

$$\begin{cases} \widehat{K}_0 = \frac{h}{8}\hat{c}_{\frac{1}{2}} + \frac{h}{4}\,\hat{c}_0 + \hat{\chi}_{\frac{1}{2}}\frac{\tau}{h} + \frac{\tau h}{8}\,p_{\frac{1}{2}} + \frac{\tau h}{4}p_0 \\ \widehat{M}_0 = \frac{h}{8}\,\hat{c}_{\frac{1}{2}} - \hat{\chi}_{\frac{1}{2}}\frac{\tau}{h} + \frac{\tau h}{8}p_{\frac{1}{2}} \\ \hat{P}_0 = \frac{h}{8}\,\hat{c}_{\frac{1}{2}}(y_0 + y_1) + \frac{h}{4}\,\hat{c}_0\,y_0 + \hat{F}\,\tau + \frac{\tau h}{4}\,(\hat{f}_{\frac{1}{2}} + \hat{f}_0) \end{cases}$$

<u>Разностный аналог краевого условия при x = l</u>

Проинтегрируем уравнение (1) на отрезке $[x_{N-\frac{1}{2}}, x_N]$ и на временном интервале $[t_m, t_{m+1}]$

$$\int_{X_{N-\frac{1}{2}}}^{X_N} dx \int_{t_m}^{t_{m+1}} c(T)\frac{\partial T}{\partial t}dt$$
$$= -\int_{t_m}^{t_{m+1}} dt \int_{X_{N-\frac{1}{2}}}^{X_N} \frac{\partial F}{\partial x}\, dx - \int_{X_{N-\frac{1}{2}}}^{X_N} dx \int_{t_m}^{t_{m+1}} p(x)Tdt + \int_{X_{N-\frac{1}{2}}}^{X_N} dx \int_{t_m}^{t_{m+1}} f(T)dt$$

при вычислении внутренних интегралов по справа в уравнении применен метод правых прямоугольников

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

Интегралы по x в вычислим методом средних, а первый интеграл в правой части (по времени) - по-прежнему, методом правых прямоугольников, получим

$$\frac{h}{4}\left[\hat{c}_N(\hat{y}_N - y_N) + \hat{c}_{N-\frac{1}{2}}\left(\hat{y}_{N-\frac{1}{2}} - y_{N-\frac{1}{2}}\right)\right]$$
$$= -\tau(\hat{F}_N - \hat{F}_{N-\frac{1}{2}}) - \tau\frac{h}{4}(p_N\hat{y}_N + p_{N-\frac{1}{2}}\hat{y}_{N-\frac{1}{2}}) + \tau\frac{h}{4}(\hat{f}_N + \hat{f}_{N-\frac{1}{2}})$$

Подставим в полученное уравнение

$$\hat{y}_{N-\frac{1}{2}} = \frac{\hat{y}_N + \hat{y}_{N-1}}{2}$$

$$y_{N-\frac{1}{2}} = \frac{y_N + y_{N-1}}{2}$$

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

$$\hat{F}_{N-\frac{1}{2}} = \hat{\chi}_{N-\frac{1}{2}}\frac{\hat{y}_{N-1} - \hat{y}_N}{h}$$

Получим

$$\left(\frac{h}{8}\hat{c}_{N-\frac{1}{2}} - \frac{\tau}{h}\hat{\chi}_{N-\frac{1}{2}} + \frac{\tau h}{8}p_{N-\frac{1}{2}}\right)\hat{y}_{N-1} + \left(\frac{h}{4}\hat{c}_N + \frac{h}{8}\hat{c}_{N-\frac{1}{2}} + \tau\alpha_N + \frac{\tau}{h}\hat{\chi}_{N-\frac{1}{2}} + \frac{\tau h}{4}p_N + \frac{\tau h}{8}p_{N-\frac{1}{2}}\right)\hat{y}_N$$
$$= \frac{h}{4}\hat{c}_N y_N + \frac{h}{8}\hat{c}_{N-\frac{1}{2}}y_N + \frac{h}{8}\hat{c}_{N-\frac{1}{2}}y_{N-1} + \tau\alpha_N T_0 + \frac{\tau h}{4}(\hat{f}_N + \hat{f}_{N-\frac{1}{2}})$$

Тогда

$$\begin{cases} \widehat{K}_N = \frac{h}{4}\hat{c}_N + \frac{h}{8}\hat{c}_{N-\frac{1}{2}} + \tau\alpha_N + \frac{\tau}{h}\hat{\chi}_{N-\frac{1}{2}} + \frac{\tau h}{4}p_N + \frac{\tau h}{8}p_{N-\frac{1}{2}} \\ \widehat{M}_N = \frac{h}{8}\hat{c}_{N-\frac{1}{2}} - \frac{\tau}{h}\hat{\chi}_{N-\frac{1}{2}} + \frac{\tau h}{8}p_{N-\frac{1}{2}} \\ \hat{P}_N = \frac{h}{4}\hat{c}_N y_N + \frac{h}{8}\hat{c}_{N-\frac{1}{2}}y_N + \frac{h}{8}\hat{c}_{N-\frac{1}{2}}y_{N-1} + \tau\alpha_N T_0 + \frac{\tau h}{4}(\hat{f}_N + \hat{f}_{N-\frac{1}{2}}) \end{cases}$$

7. Метод простых итераций

Для решения разностной системы исползуется метод простых итераций. Обозначим текующую итерацию s, а предыдущую s-1, тогда процесс организуется по схеме

$$\begin{cases} K_0^{s-1} y_0 + M_0^{s-1} y_1 = P_0^{s-1} \\ A_n^{s-1} y_{n-1}^s - B_n^{s-1} y_n^s + D_n^{s-1} y_{n+1}^s = -F_n^{s-1} \\ K_N^{s-1} y_N + M_N^{s-1} y_{N-1} = P_N^{s-1} \end{cases}$$

Все коэффтциенты берутся на (s-1)-ой итерации, т.е они известны. Получили обычную линейную схему, решение которой осуществляется методом прогонки, показанным ниже.

Итерации прекращаются при условии

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

8. Метод прогонки

Прямой ход : предполагаем, что система уравнений имеет вид

$$A_i y_{i-1} + B_i y_i + C_i y_{i+1} = D_i, \qquad i = 0..N$$

Все прогоночные коэффициенты можно найти по формулам

$$\begin{cases} \xi_1 = -\dfrac{K_0}{M_0} \\ \xi_{i+1} = \dfrac{-C_i}{A_i \xi_i + B_i}, \qquad i = 1..N-1 \end{cases}$$

и

$$\begin{cases} \eta_1 = \dfrac{P_0}{M_0} \\ \eta_{i+1} = \dfrac{D_i - A_i \eta_i}{A_i \xi_i + B_i}, \qquad i = 1..N-1 \end{cases}$$

Обратный ход :

Находим начальное значение x_N

$$y_N = \frac{D_N - A_N \eta_N}{A_N \xi_N + B_N} = \frac{P_N - K_N \eta_N}{K_N \xi_N + M_N}$$

Остальные значения находятся по формуле

$$y_i = \xi_{i+1} y_{i+1} + \eta_{i+1}$$

Код программы

```
import numpy as np
import math
import matplotlib.pyplot as plt
from matplotlib import cm
from mpl_toolkits import mplot3d

#constants
a1 = 0.0134
b1 = 1
c1 = 4.35e-4
m1 = 1
a2 = 2.049
b2 = 0.563e-3
c2 = 0.528e5
m2 = 1

alpha0 = 0.05
alphaN = 0.01
l = 10
d = alphaN * l / (alphaN - alpha0)
cc = -alpha0 * d

T0 = 300
R = 0.5
Ft = 50
eps1 = 10e-4
eps2 = 10e-4

def plot2d(x, y, xlabel, ylabel, title):
    plt.plot(x, y, 'c')
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.title(title)
    plt.grid(True)
    plt.show()

def plot3d(x, y, z):
    fig = plt.figure()
    ax = plt.axes(projection='3d')
    ax.plot_surface(x, y, z, rstride=1, cstride=1, cmap='viridis',
edgecolor='none')
    #ax.set_title("")
    fig.show()
    plt.show()

#ax[n-1]+bx[n]+cx[n+1]=d
def tridiagonal_method(a, b, c, d):
    n = len(a) - 1;
    xi = [None for i in range(n+1)]
    xi[1] = -c[0]/b[0]
    eta = [None for i in range(n+1)]
    eta[1] = d[0]/b[0]
    for i in range(1, n):
```

```
        xi[i+1] = -c[i]/(a[i]*xi[i]+b[i])
        eta[i+1] = (d[i]-a[i]*eta[i])/(a[i]*xi[i]+b[i])

    res = [None for i in range(n+1)]
    res[n] =(d[n]-a[n]*eta[n])/(a[n]*xi[n]+b[n])
    for i in range(n-1, -1, -1):
        res[i] = xi[i+1]*res[i+1]+eta[i+1]
    return res


def k(T):
    return a1 * (b1 + c1 * T**m1)

def c(T):
    return a2 + b2 * T**m2 - c2 / T**2

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

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

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



def solve_equation_system(old_T, prev_T_by_time, tau, N):
    _a = []
    _b = []
    _d = []
    _f = []
    h = l / N

    def cal_plus_half(func, start, end):
        return (func(start) + func(end)) / 2

    def c_(n):
        return c(old_T[n])

    def c_plus(n):
        return cal_plus_half(c, old_T[n], old_T[n+1])

    def chi_plus(n):
        return cal_plus_half(k, old_T[n], old_T[n+1])

    def p_plus(n):
        return (p(n) + p(n+h)) / 2

    def f_plus(n):
        return (f(n) + f(n+h)) / 2

    def p_(n):
        return p(h*n)
```

```
    def f_(n):
        return f(h*n)

    #the left boundary condition x = 0
    _a.append(0)
    _b.append(h/8*c_plus(0)
              + h/4*c_(0)
              + tau/h*chi_plus(0)
              + tau*h/8*p_plus(0)
              + tau*h/4*p(0))
    _d.append(h/8*c_plus(0)
              - tau/h*chi_plus(0)
              + tau*h/8*p_plus(0))
    _f.append(h/8*c_plus(0)*(prev_T_by_time[0]+prev_T_by_time[1])
                + h/4*c_(0)*prev_T_by_time[0]
                + tau*Ft
                + tau*h/4*(f_plus(0) + f(0)))

    #1 <= n <= N-1
    for i in range(1, N):
        _a.append(tau/h*chi_plus(i-1))
        _d.append(tau/h*chi_plus(i))
        _b.append(-(_a[-1] + _d[-1] + c_(i)*h + p_(i)*h*tau))
        _f.append(-(f_(i)*h*tau + c_(i)*prev_T_by_time[i]*h))

    #the right boudary condition x = l
    _a.append(h/8*c_plus(N-1)
              - tau/h*chi_plus(N-1)
              + tau*h/8*p_plus(N-1))
    _b.append(h/4*c_(N)
              + h/8*c_plus(N-1)
              + tau*alphaN
              + tau/h*chi_plus(N-1)
              + tau*h/4*p_(N)
              + tau*h/8*p_plus(N-1))
    _d.append(0)
    _f.append(h/4*c_(N)*prev_T_by_time[N]
              + h/8*c_plus(N-1)*prev_T_by_time[N]
              + h/8*c_plus(N-1)*prev_T_by_time[N-1]
              + tau*alphaN*T0
              + tau*h/4*(f_(N)+f_plus(N-1)))

    return tridiagonal_method(_a, _b, _d, _f)


def check(res1, res2, eps):
    tmp = 0
    for i in range(len(res2)):
        tmp = max(tmp, math.fabs((res2[i]-res1[i])/res2[i]))
    if tmp <= eps:
        return True
    else:
        return False

#simple iteration metho
def solve(tau, N):
```

```
    h = l / N
    time = 0
    old_T = [T0] * (N+1)
    new_T = [None] * (N+1)
    res = [old_T]
    while True:
        old_T = res[-1]
        while True:
            new_T = solve_equation_system(old_T, res[-1], tau, N)
            if check(old_T, new_T, eps1):
                break
            old_T = new_T
        time += tau
        res.append(new_T)
        if check(res[-2], res[-1], eps2):
              break
    return res

if __name__ == "__main__":
    tau = 1
    N = 100
    res = np.array(solve(tau, N))

    #3d surface T(x, t)
    x = np.linspace(0, l, N+1)
    y = np.arange(len(res))
    X, Y = np.meshgrid(x, y)
    Z = res
    plot3d(X, Y, Z)

    #Зависимость температуры от координаты стержня
    x = np.linspace(0, l, N+1)
    for T in res[:10]:
        plt.plot(x, T)
    plt.xlabel("Длина, см")
    plt.ylabel("Температура, K")
    plt.grid(True)
    plt.show()

    #Зависимость температуры от времени
    t = np.arange(len(res))
    for T in res.transpose()[:10]:
        plt.plot(t, T)
    plt.xlabel("Время, сек")
    plt.ylabel("Температура, K")
    plt.grid(True)
    plt.show()
```

Результат работы

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

Для получения разностного аналога краевого условия при $x = l$, надо проинтегрировать на отрезке [xN-1/2, xN] выписанное выше уравнение (1) и учесть (10) и (11)

Обозначим $F = -k(T)\frac{\partial T}{\partial x}$ (15)

Тогда (1) можно записать в виде:

$$c(T)\frac{\partial T}{\partial t} = -\frac{\partial F}{\partial x} - p(x)T + f(x) \quad (16)$$

где

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

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

Проитегрируем (13):

$$\int_{X_{N-\frac{1}{2}}}^{X_N} dx \int_{t_m}^{t_{m+1}} c(T)\frac{\partial T}{\partial t}dt = -\int_{t_m}^{t_{m+1}} dt \int_{X_{N-\frac{1}{2}}}^{X_N} \frac{\partial F}{\partial x}dx - \int_{X_{N-\frac{1}{2}}}^{X_N} dx \int_{t_m}^{t_{m+1}} p(x)Tdt + \int_{X_{N-\frac{1}{2}}}^{X_N} dx \int_{t_m}^{t_{m+1}} f(x)dt \quad (14)$$

$$\int_{X_{N-\frac{1}{2}}}^{X_N} \hat{c_n}\left(\hat{T} - T\right)dx = -\int_{t_m}^{t_{m+1}} \left(F_n - F_{N-\frac{1}{2}}\right)dt - \int_{X_{N-\frac{1}{2}}}^{X_N} p\hat{T}\tau dx + \int_{X_{N-\frac{1}{2}}}^{X_N} \hat{f}\tau dx$$

Вычисляем интегралы

$$\frac{h}{4}\left(\hat{c_N}(\hat{y_N} - y_N) - \hat{c}_{N-\frac{1}{2}}\left(\frac{\hat{y_N} + \hat{y_{N-1}}}{2} - \frac{y_N + y_{N+1}}{2}\right)\right) =$$

$$= -\tau\left(\alpha_N(\hat{y_N} - T_0) - \hat{X_N}\frac{\hat{y_N}+\hat{y_{N-1}}}{h}\right) - \left(p_N\hat{y_N} - p_{n-\frac{1}{2}}\frac{\hat{y_N}+\hat{y_{N-1}}}{2}\right)\frac{\tau h}{4} + \left(\hat{f_N} - \hat{f}_{N-\frac{1}{2}}\right)\frac{\tau h}{4} \quad (17)$$

Приведем к виду $\hat{K_N}\hat{y_N} + \hat{M_N}\hat{y_{N-1}} = \hat{P_N}$. Получим:

$$\left(\frac{h}{4}\hat{c_N} + \frac{h}{8}\hat{c}_{N-\frac{1}{2}} + \tau\alpha_N + \frac{\tau}{h}\hat{X}_{N-\frac{1}{2}} + \frac{h}{4}\tau p_N + \frac{h}{8}\tau p_{N-\frac{1}{2}}\right)\hat{y_N} + \left(\frac{h}{8}\hat{C}_{N-\frac{1}{2}} - \frac{\tau}{h}\hat{X}_{N-\frac{1}{2}} + \frac{h}{8}\tau p_{N-\frac{1}{2}}\right)\hat{y_{N-1}} =$$

$$= \alpha_N\tau T_0 + \frac{h}{4}\hat{C_N}y_N + \frac{h}{8}\hat{c}_{N-\frac{1}{2}}(y_N + y_{N-1}) + \frac{h}{4}\tau\left(\hat{f_N} + \hat{f}_{N-\frac{1}{2}}\right) \quad (16)$$

Для функций c, X, p будет принята простая аппроксимация.

$$p_{N-\frac{1}{2}} = \frac{p_{N-1}+p_N}{2} \quad (18)$$

Из (9) и (16) получим $K_0, M_0, P_0, K_N, M_N, P_N$.

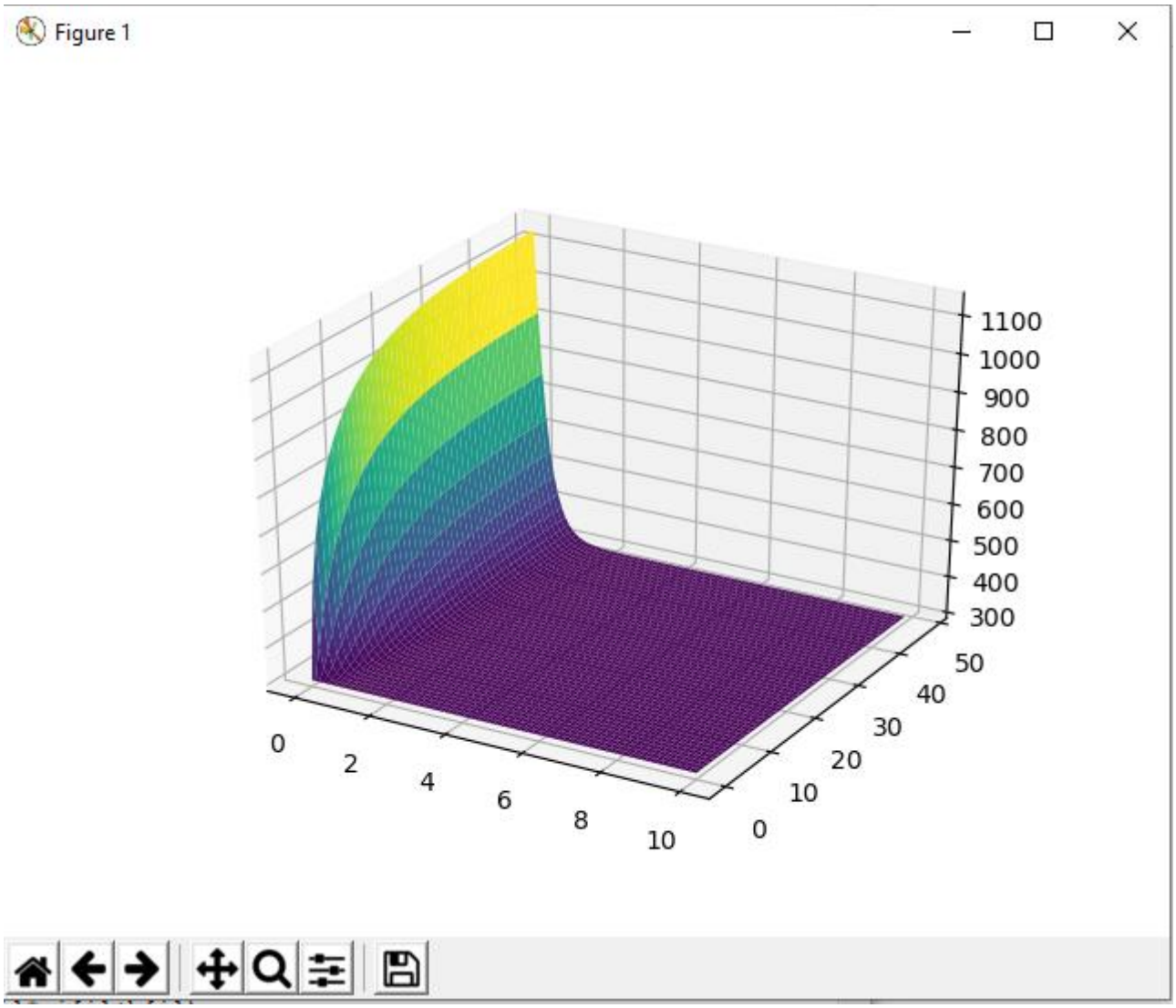
Получим систему:

$$\begin{cases} \hat{K_0}\hat{y_0} + \hat{M_0}\hat{y_1} = \hat{P_0} \\ \hat{A_n}\hat{y_{n-1}} - \hat{B_n}\hat{y_n} + \hat{D_n}\hat{y_{n+1}} = -\hat{F_n}, 1 \le n \le N-1 \\ \hat{K_N}\hat{y_N} + \hat{M_{N-1}}\hat{y_{N-1}} = \hat{P_N} \end{cases} \quad (18)$$

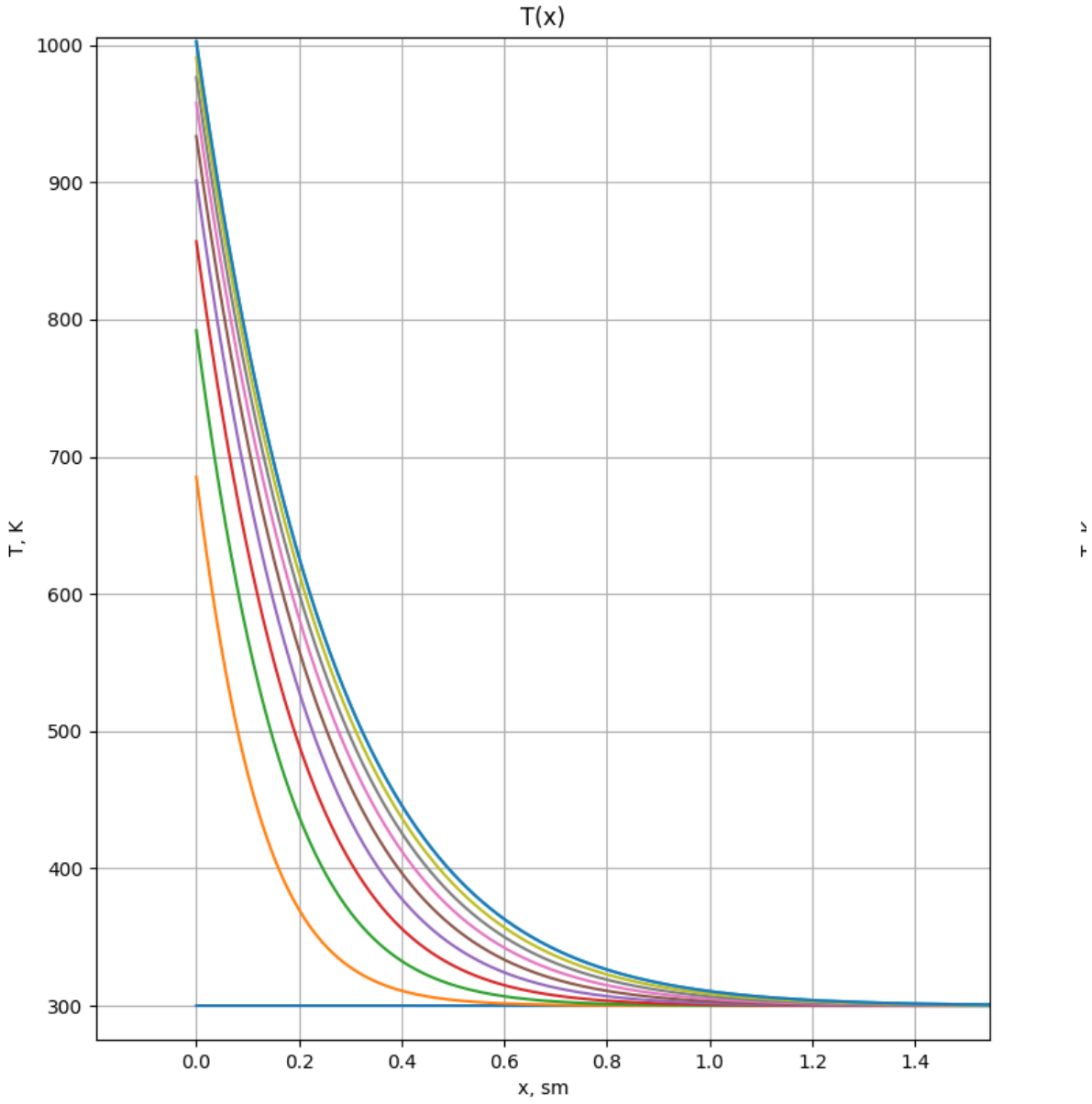
Эту систему можно решить методом итераций. Пусть i — номер итерации

$$A_n^{i-1} y_{n+1}^i - B_n^{i-1} y_n^i + D_n^{i-1} y_{n-1}^i = -F_n^{i-1}$$

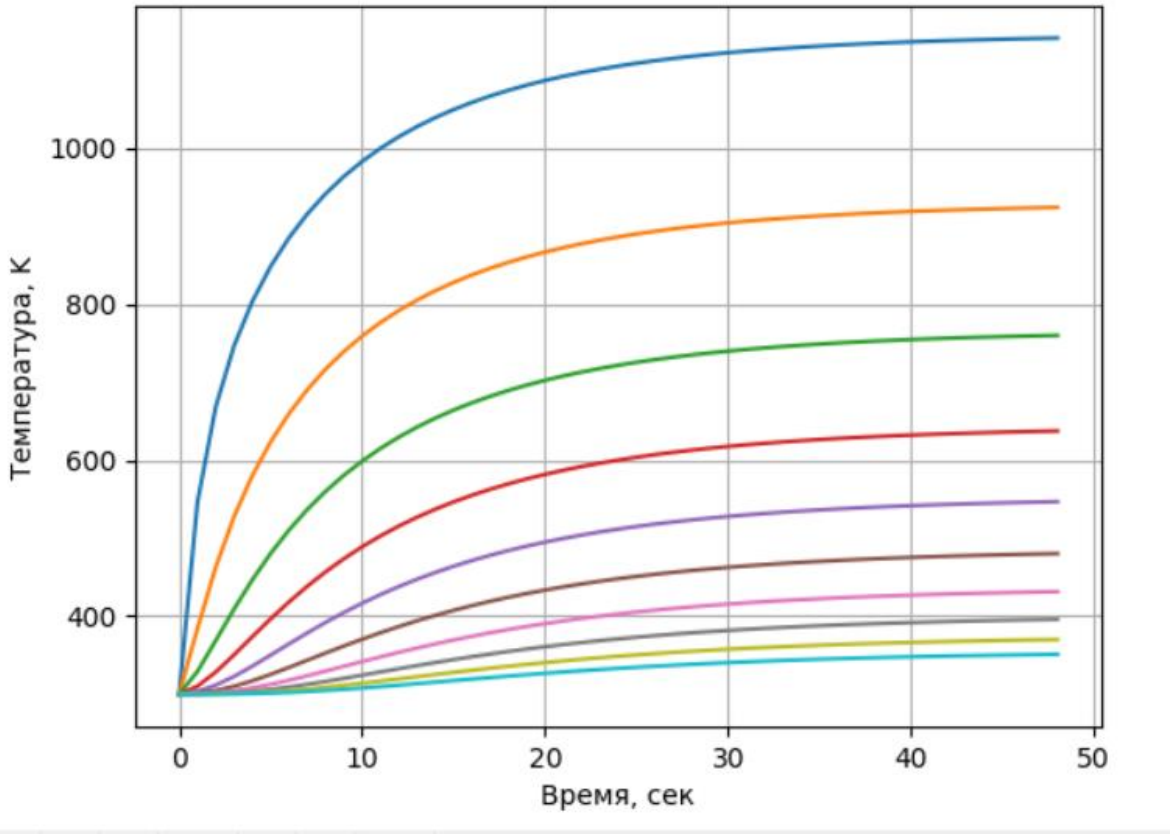
2. График зависимости температуры T(x, t):



3. График зависимости температуры от координаты при нескольких фиксированных значениях времени (аналогично рисунку в лекции) при заданных выше параметрах. Обязательно представить распределение в момент времени, соответствующий установившемуся режиму, когда поле перестает меняться с некоторой точностью (например $\frac{T(t+\tau)-T(t)}{T(t+\tau)} < 10^{-4}$), т.е. имеет место выход на стационарный режим. На этой стадии левая часть дифференциального уравнения близка к нулю



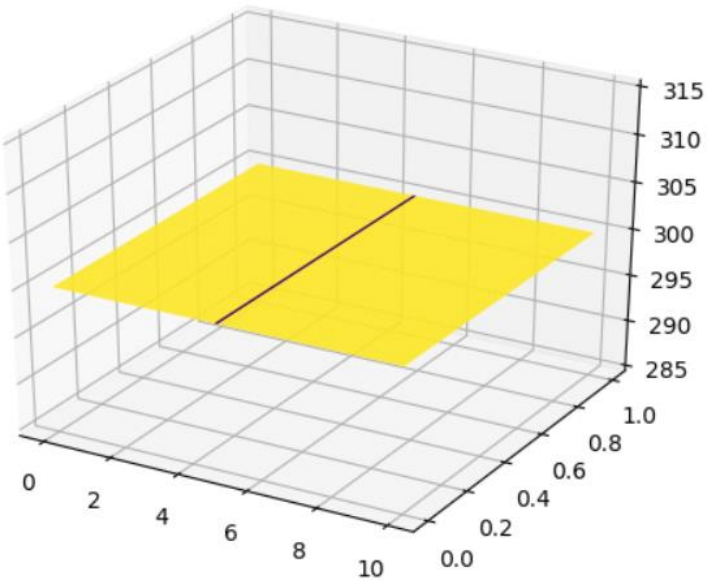
4. График зависимости $\mathbf{T(x_n, t)}$ при нескольких фиксированных значениях координаты $\mathbf{x_n}$. Обязательно представить случай n=0, т.е $x = x_0 = 0$

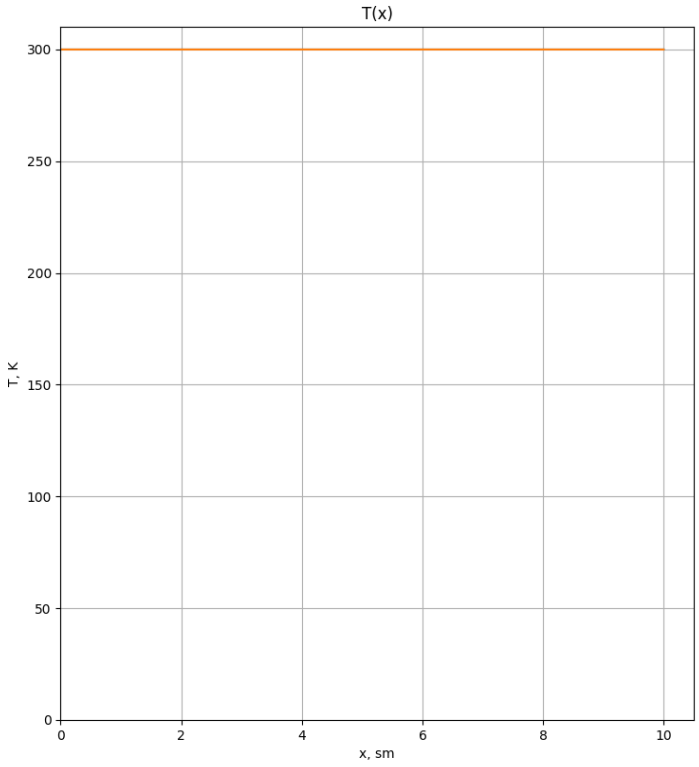


Вопросы

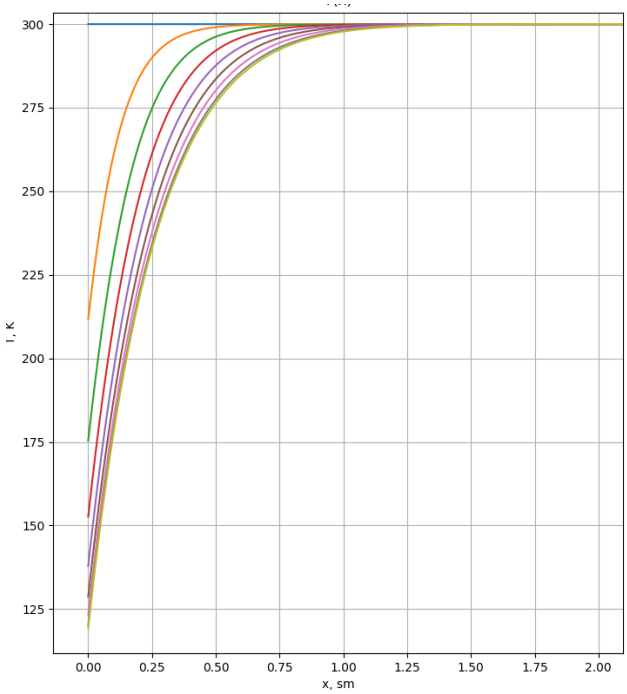
1. Приведите результаты тестирования программы (графики, общие соображения, качественный анализ).

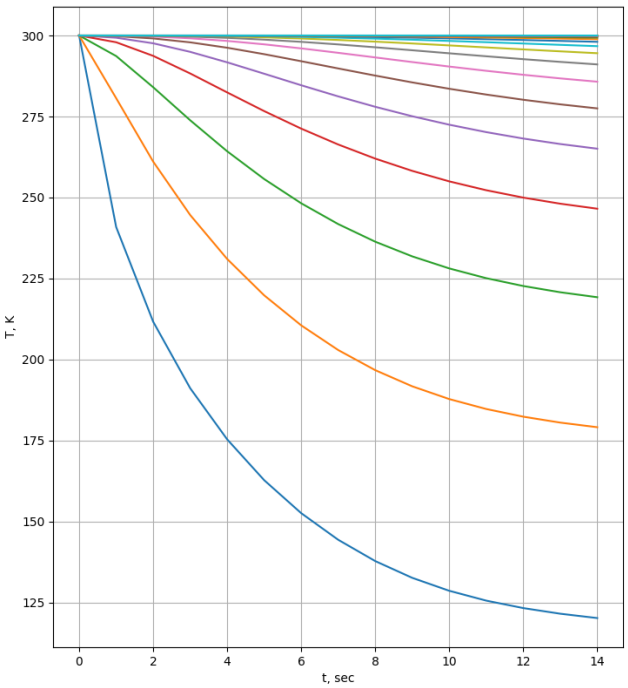
Если принять F0 =0.Тепловое нагружение отсутствует, причин для нагрева нет, температура стержня должна быть равна температуре окружающей среды T0.





Если F_0(тепловой поток) меньше нуля. Это означает, что съем тепла идет слева, поэтому температура будет увеличиваться от 0 до l


300
275
250
225
200
175
150
125
T, K
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
x, sm


300
275
250
225
200
175
150
125
T, K
0
2
4
6
8
10
12
14
t, sec