**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 **ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.03 ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

# О Т Ч Е Т

**По лабораторной работе №** 3

**Дисциплина:** Моделирование

Студент ИУ7-62Б Н.А. Гарасев

(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель В.М. Градов

(Подпись, дата) (И.О. Фамилия)

Москва, 2021

**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.

**Исходные данные:**

1. Дано:

$n_p$ = 1.4 – коэффициент преломления,

$l$ = 0.2 см – толщина слоя,

$T_0$ = 300K – температура окружающей среды,

$\sigma$ = 5.668 $10^{-12}$ Вт/(см$^2$K$^4$)- постоянная Стефана- Больцмана,

$F_0$ = 100 Вт/см2 - поток тепла,

$\alpha$ = 0.05 Вт/(см$^2$ К) – коэффициент теплоотдачи.

2. Уравнение:

$$\frac{d}{dx}\left(\lambda(T)\frac{dT}{dx}\right)-4\cdot k(T)\cdot n_p^2\cdot\sigma\cdot(T^4-T_0^4)=0$$

3. Краевые условия

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

Функции *lambda(T)* и *k(T)* заданы таблицей:

| $T, K$ | $\lambda$, Вт/(см К) | | $T, K$ | $k$, см$^{-1}$ |
|---|---|---|---|---|
| 300 | 1.36 $10^{-2}$ | | 293 | 2.0 $10^{-2}$ |
| 500 | 1.63 $10^{-2}$ | | 1278 | 5.0 $10^{-2}$ |
| 800 | 1.81 $10^{-2}$ | | 1528 | 7.8 $10^{-2}$ |
| 1100 | 1.98 $10^{-2}$ | | 1677 | 1.0 $10^{-1}$ |
| 2000 | 2.50 $10^{-2}$ | | 2000 | 1.3 $10^{-1}$ |
| 2400 | 2.74 $10^{-2}$ | | 2400 | 2.0 $10^{-1}$ |

4. Разностная схема

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

$$M_0 y_0 + K_0 y_1 = P_0,$$

$$K_N y_{N-1} + M_N y_N = P_N.$$

Где,

$$A_n = \frac{\varkappa_{n-\frac{1}{2}}}{h},\ B_n = A_n + C_n + p_n h,\ C_n = \frac{\varkappa_{n+\frac{1}{2}}}{h},\ D_n = f_n h$$

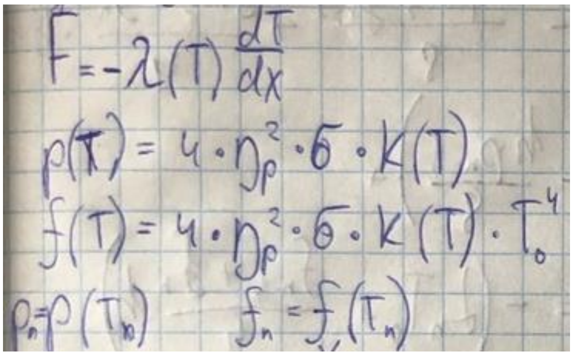
Способ вычисления – метод средних:

$$\varkappa_{n\pm\frac{1}{2}} = \frac{k_n + k_{n\pm 1}}{2}$$

Система совместно с краевыми условиями решается методом прогонки.

5. Краевые условия

Обозначим:



$$F = -\lambda(T)\frac{dT}{dx}$$

$$p(T) = 4 \cdot n_p^2 \cdot \sigma \cdot k(T)$$

$$f(T) = 4 \cdot n_p^2 \cdot \sigma \cdot k(T) \cdot T_0^4$$

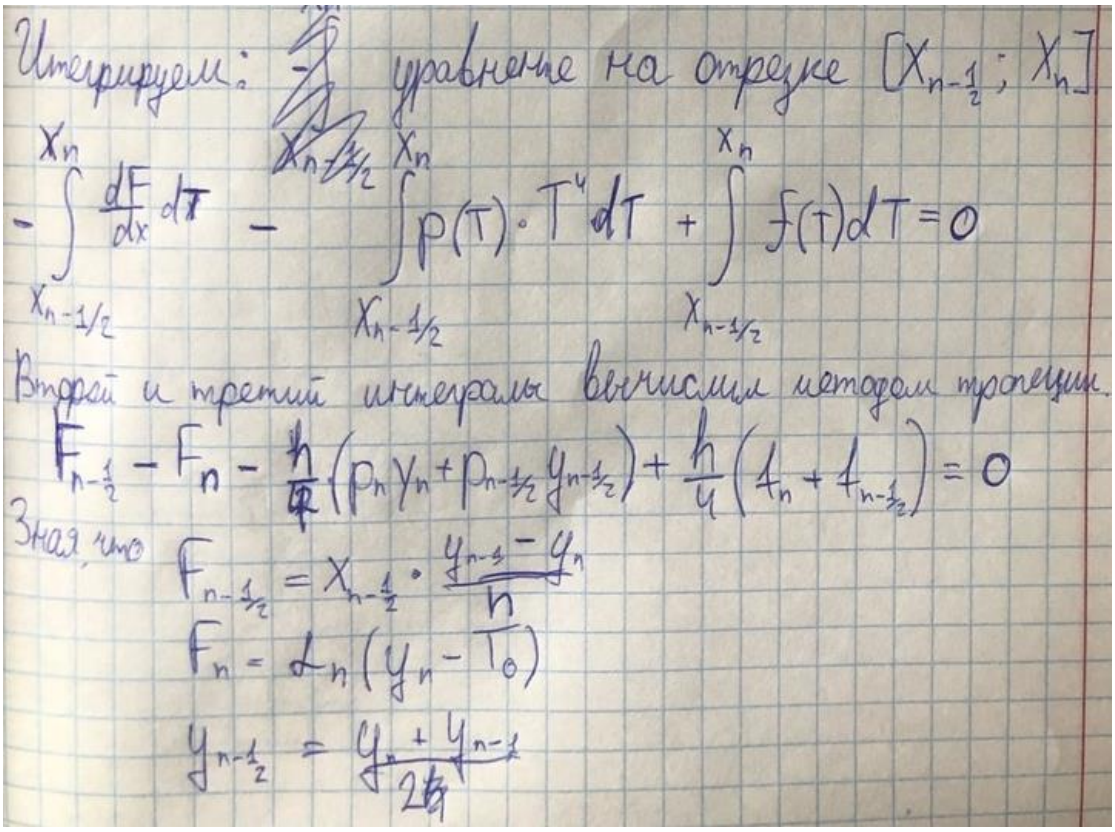
$$p_n = p(T_n) \qquad f_n = f(T_n)$$

Разностные аналоги краевых условий при *x = 0*:

$$-(F_{1/2} - F_0) - \frac{h}{4}(p_{1/2}\, y_{1/2} + p_0\, y_0) + \frac{h}{4}(f_{1/2} + f_0) = 0 \,.$$

, взяты из лекции 7.

Разностные аналоги краевых условий при x = l:



Интегрируем уравнение на отрезке $[x_{n-\frac{1}{2}}; x_n]$

$$-\int_{x_{n-1/2}}^{x_n} \frac{dF}{dx}\, dx - \int_{x_{n-1/2}}^{x_n} p(T)\cdot T^4\, dT + \int_{x_{n-1/2}}^{x_n} f(T)\, dT = 0$$

Второй и третий интегралы вычислим методом трапеций.

$$F_{n-\frac{1}{2}} - F_n - \frac{h}{4}(p_n y_n + p_{n-1/2}\, y_{n-1/2}) + \frac{h}{4}(f_n + f_{n-1/2}) = 0$$

Зная, что

$$F_{n-\frac{1}{2}} = \chi_{n-\frac{1}{2}} \cdot \frac{y_{n-1} - y_n}{h}$$

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

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

Имеем:

$$\frac{X_{n-\frac{1}{2}} y_{n-1}}{h} - \frac{X_{n-\frac{1}{2}} \cdot y_n}{h} - \alpha_n y_n + \alpha_n \cdot T_0 -$$

$$- \frac{h \cdot p_n \cdot y_n}{4\cancel{8}} - \frac{h p_{n-\frac{1}{2}} y_n}{8} - \frac{h \cdot p_{n-\frac{1}{2}} \cdot y_{n-1}}{8} + \frac{f_{n-\frac{1}{2}} + f_n}{4} \cdot h = 0$$

$$y_n\left(-\frac{X_{n-\frac{1}{2}}}{h} - \alpha_n - \frac{h p_n}{4} - \frac{h p_{n-\frac{1}{2}}}{8}\right) +$$

$$+ y_{n-1}\left(\frac{X_{n-\frac{1}{2}}}{h} - \frac{h p_{n-\frac{1}{2}}}{8}\right) = -\left(\alpha_n T_0 + \frac{f_{n-\frac{1}{2}} + f_n}{4} h\right)$$

Опираясь на уравнения краевых условий, находим коэффициенты K0, M0, P0, KN, MN, PN.

6. Решение.

Для решения системы, используем метод прогонки итеративно.

С начала каждой итерации вычисляются начальные значения прогоночных коэффициентов:

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

Далее в каждой итерации вычисляются массивы прогоночных коэффициентов:

$$y_n = \underbrace{\frac{C_n}{B_n - A_n\varepsilon_n}}_{\varepsilon_{n+1}} y_{n+1} + \underbrace{\frac{D_n + A_n\eta_n}{B_n - A_n\varepsilon_n}}_{\eta_{n+1}}$$

Последнее значение функции в последней точке:

$$y_N = \frac{P_N - M_N \eta_N}{K_N + M_N \varepsilon_N}$$

Затем в каждой итерации обратной прогонкой находятся значения:

$$y_N = \frac{P_N - M_N \eta_N}{K_N + M_N \varepsilon_N}$$

# Листинг программы

Листинг 1. Дано.

```
np = 1.4
l = 0.2
T0 = 300
T_CONST = 400
sigma = 5.668 * 10**-12
F0 = 100
alpha = 0.05
h = 10**-4
E = 10**-6
```

Листинг 2. Таблицы.

```
table1 = {
    'T': [300, 500, 800, 1100, 2000, 2400],
    'lambda': [1.36 * 10**-2, 1.63 * 10**-2, 1.81 * 10**-2, 1.92 * 10**-2,
2.5 * 10**-2, 2.74 * 10**-2],
}

table2 = {
    'T': [293, 1278, 1528, 1677, 2000, 2400],
    'k': [2 * 10**-2, 5 * 10**-2, 7.8 * 10**-2, 1 * 10**-1, 1.3 * 10**-1, 2 *
10**-1],
}
```

Листинг 3. Линейная интерполяция.

```
def linear(x, y):
    def p(value):
        i = 1
        while value > x[i] and i < len(x) - 1:
            i += 1
        return y[i - 1] + (y[i] - y[i - 1]) / (x[i] - x[i - 1]) * (value -
x[i - 1])
    return p
l_t = linear(table1['T'], table1['lambda'])
k_t = linear(table2['T'], table2['k'])
```

Листинг 4. Вспомогательные функции.

```
def T(x):
    return T_CONST - (T_CONST - T0) * x / l


def lambda_x(n):
    return l_t(t[n])


# x + 1/2
def x2(n):
    return (lambda_x(n) + lambda_x(n + 1)) / 2


# x - 1/2
def _x2(n):
    return (lambda_x(n) + lambda_x(n - 1)) / 2


def p2(n):
    return (p(n) + p(n + 1)) / 2


def _p2(n):
    return (p(n) + p(n - 1)) / 2


def p(n):
    return 4 * np * np * siqma * k_t(t[n]) * t[n]**3


def f2(n):
    return (f(n) + f(n + 1)) / 2


def _f2(n):
    return (f(n) + f(n - 1)) / 2


def f(n):
    return 4 * np * np * siqma * k_t(t[n]) * T0**4


def a(n):
    return (l_t(t[n]) + l_t(t[n - 1])) / 2 / h


def b(n):
    return a(n) + c(n) + 4 * np * np * siqma * k_t(t[n]) * t[n]**3 * h


def c(n):
    return (l_t(t[n]) + l_t(t[n + 1])) / 2 / h


def d(n):
    return 4 * np * np * siqma * k_t(t[n]) * T0**4 * h
```

Листинг 5. Краевые условия.

```
K0 = x2(0) + h ** 2 / 8 * p2(0) + h ** 2 / 4 * p(0)
M0 = h ** 2 / 8 * p2(0) - x2(0)
P0 = h * F0 + h ** 2 / 4 * (f2(0) + f(0))

KN = - _x2(steps) / h - alpha - h * p(steps) / 4 - h * _p2(steps) / 8
MN = _x2(steps) / h - h * _p2(steps) / 8
PN = - (alpha * T0 + (_f2(steps) + f(steps)) / 4 * h)
```

Листинг 6. Условие выхода из итерации по температуре и по балансу энергии.

```
if max(ys) <= eps1 and max(fs) <= eps2:
    break
```

Листинг 7. Прямой ход.

```
while x + h < l:
    eps.append(c(n) / (b(n) - a(n) * eps[n]))
    eta.append((d(n) + a(n) * eta[n]) / (b(n) - a(n) * eps[n]))
    n += 1
    x += h
```

Листинг 8. Обратный ход.

```
t[n] = (PN - MN * eta[n]) / (KN + MN * eps[n])
for k in range(n - 1, -1, -1):
    t[k] = eps[k + 1] * t[k + 1] + eta[k + 1]
```

## Результат программы

1. Разностный аналог краевого условия при x=l:

Интегрируем: уравнение на отрезке $[X_{n-\frac12}; X_n]$

$$-\int_{X_{n-1/2}}^{X_n} \frac{dF}{dx}\,dT - \int_{X_{n-1/2}}^{X_n} p(T)\cdot T^4\,dT + \int_{X_{n-1/2}}^{X_n} f(T)\,dT = 0$$

Второй и третий интегралы вычислим методом трапеций.

$$F_{n-\frac12} - F_n - \frac{h}{4}\left(p_n y_n + p_{n-1/2}\, y_{n-1/2}\right) + \frac{h}{4}\left(f_n + f_{n-1/2}\right) = 0$$

Зная, что

$$F_{n-1/2} = \varkappa_{n-\frac12}\cdot\frac{y_{n-1} - y_n}{h}$$

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

$$y_{n-\frac12} = \frac{y_n + y_{n-1}}{2}$$

Имеем:

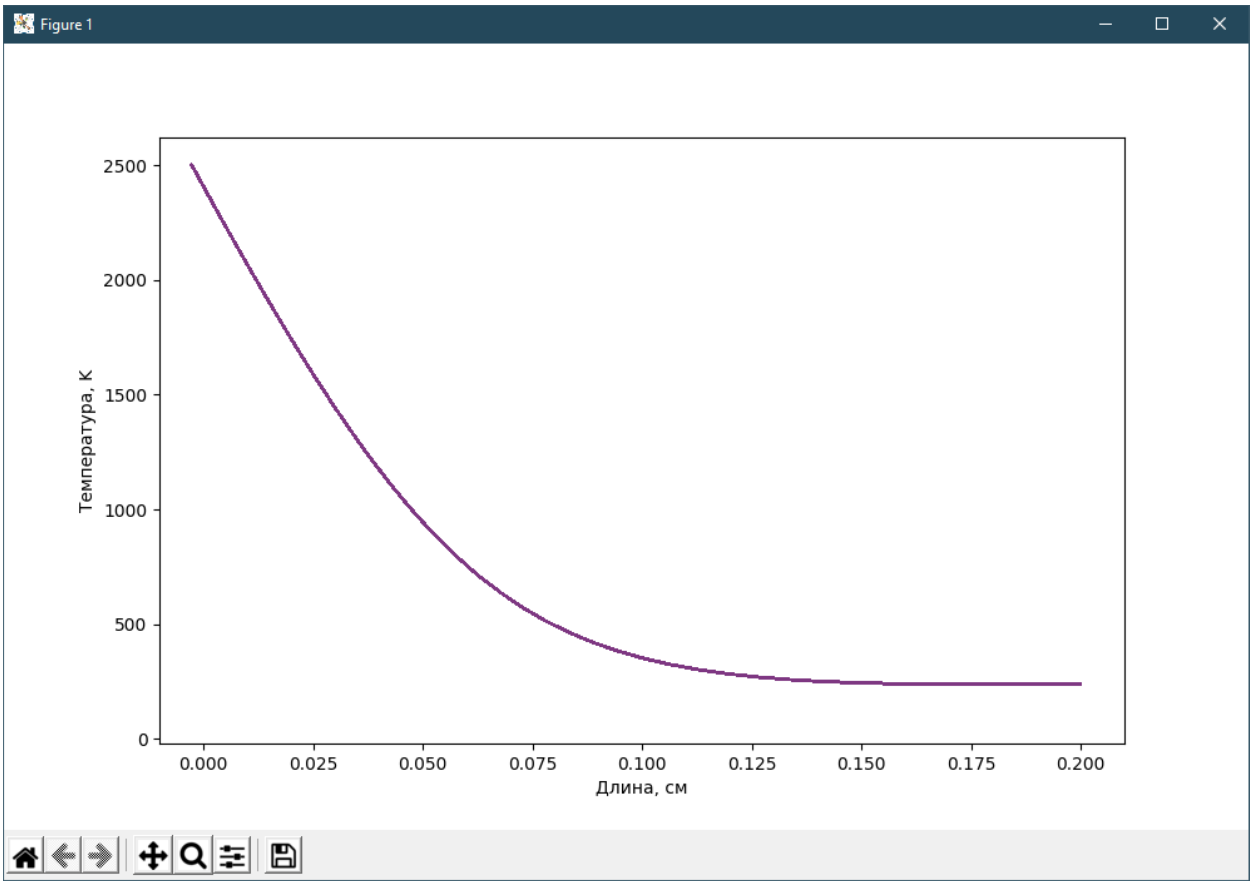
$$\frac{X_{n-\frac{1}{2}}\, y_{n-1}}{h} - \frac{X_{n-\frac{1}{2}} \cdot y_n}{h} - \alpha_n y_n + \alpha_n \cdot T_0 -$$

$$- \frac{h \cdot p_n \cdot y_n}{4\cancel{8}} - \frac{h\, p_{n-\frac{1}{2}}\, y_n}{8} - \frac{h \cdot p_{n-\frac{1}{2}} \cdot y_{n-1}}{8} + \frac{f_{n-\frac{1}{2}} + f_n}{4} \cdot h = 0$$

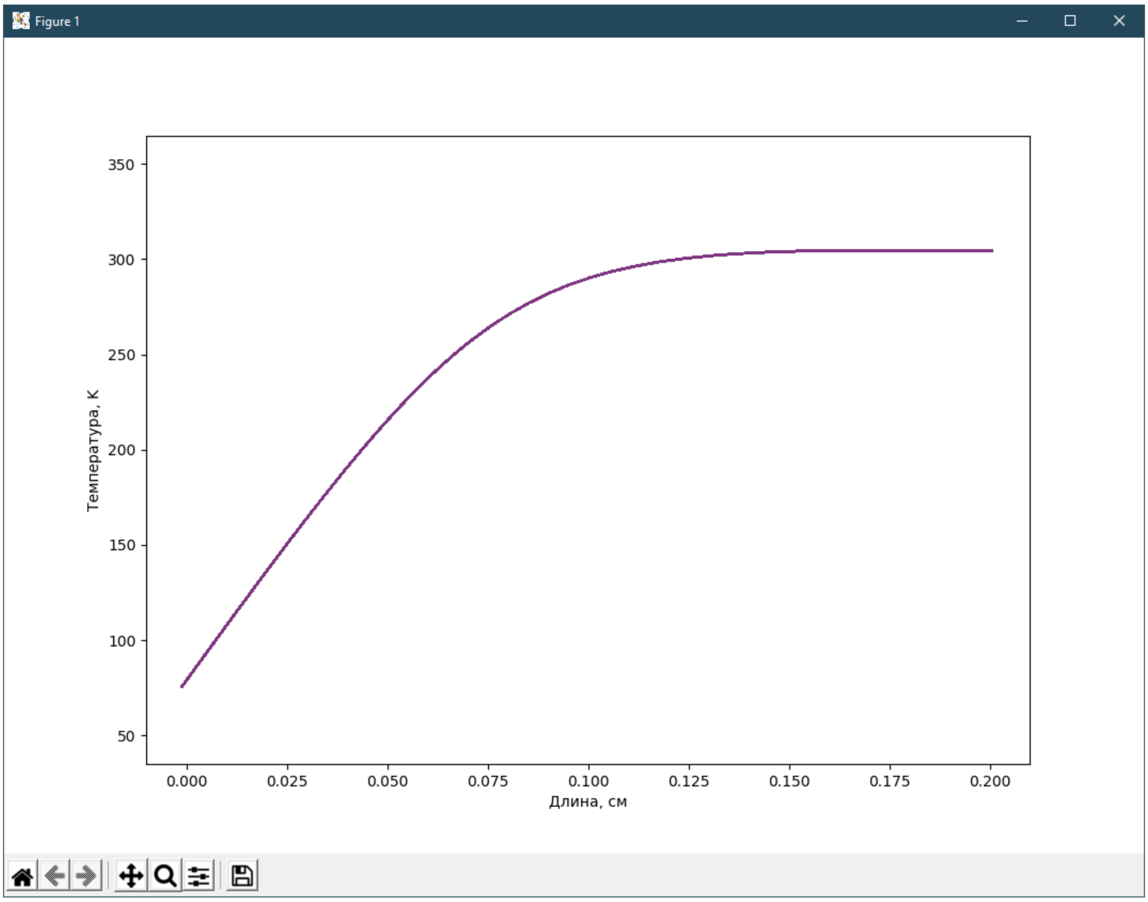
$$y_n\left(-\frac{X_{n-\frac{1}{2}}}{h} - \alpha_n - \frac{h p_n}{4} - \frac{h\, p_{n-\frac{1}{2}}}{8}\right) +$$

$$+ y_{n-1}\left(\frac{X_{n-\frac{1}{2}}}{h} - \frac{h\, p_{n-\frac{1}{2}}}{8}\right) = -\left(\alpha_n T_0 + \frac{f_{n-\frac{1}{2}} + f_n}{4}\, h\right)$$

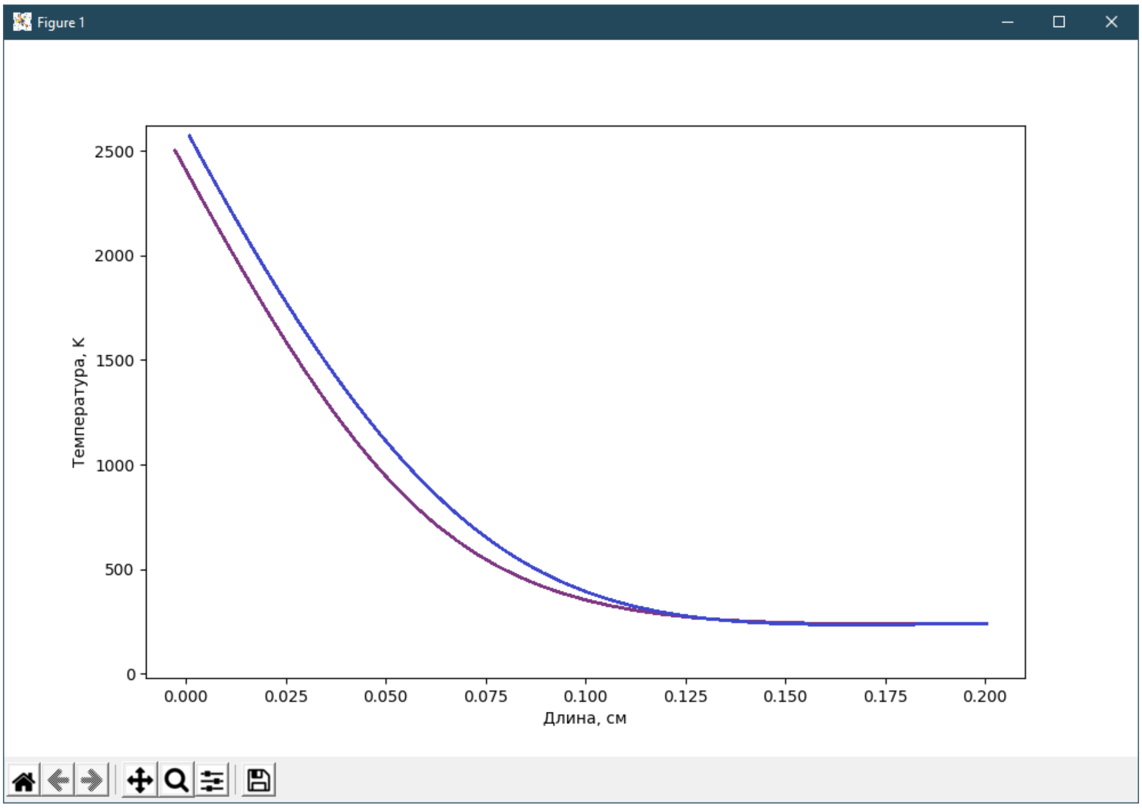
2. График зависимости температуры T(x) от координаты x при заданных условиях.



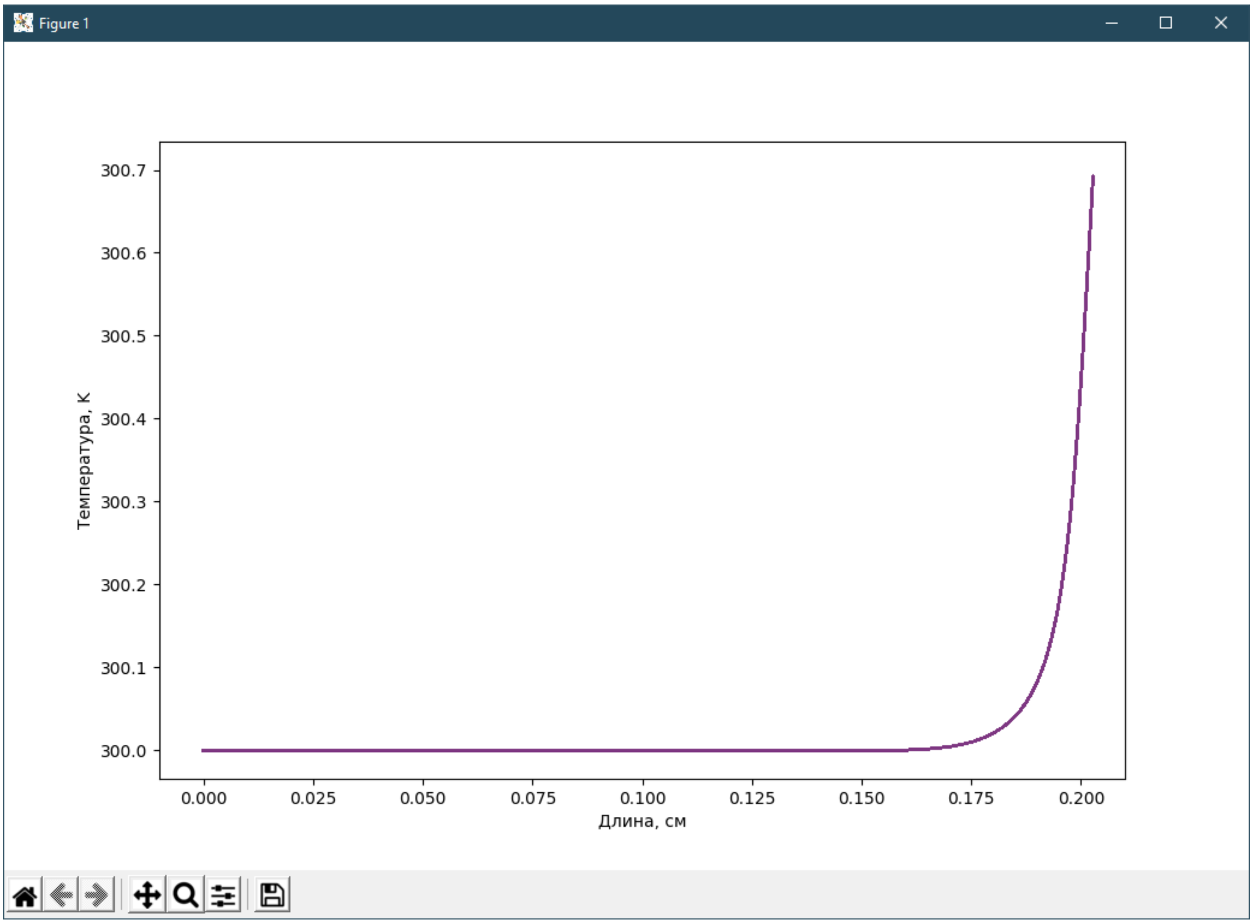
3. График зависимости T(x) при F0 = -10Вт/См$^2$




4. График зависимости T(x) при увеличенных значениях alpha (alpha = 3 * alpha). Сравнить с п.2.



5. График зависимости T(x) при F0 = 0.




6. Для указанного в задании исходного набора параметров привести данные по балансу энергии, т.е. значения величин:

Точность выхода eps_1 (по температуре) = 0,069

Точность выхода eps_2 (по балансу) = 1,12

# Вопросы при защите лабораторной работы.

**1. Какие способы тестирования программы можно предложить?**

Опираясь на физические законы, то при F0 > 0 происходит охлаждение пластины, при F0 < 0 – нагревание пластины. Также при увеличении показателя теплосъема, уровень температур должен снижаться, а градиент увеличиваться.

**2. Получите простейший разностный аналог нелинейного краевого условия $x=l$, $-k(l)dTdx=\alpha N(T(l)-T0)$ + $\varphi(T)$, где $\varphi(T)$ – заданная функция. Производную аппроксимируйте односторонней разностью.**

Аппроксимируем производную:

$$\frac{dT}{dx} = \frac{y_N - y_{N-1}}{h}$$

Подставим в исходное уравнение:

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

Учитывая, что $y_{N-1} = \xi_N y_N + \eta_N$:

$$-k_N(y_N - \xi_N y_N + \eta_N) = \alpha_N(y_N - T_0)h + \varphi(y_N)h$$

Приведем подобные и получим уравнение относительно $y_N$:

$$\varphi(y_N)h + (k_N + \alpha_N h - k_N\xi_N - k_N\eta_N)y_N - h\alpha_N T_0 = 0$$

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

Для прямого хода нужно найти начальные прогоночные коэффициенты по формулам:

$$\xi_1 = \frac{-M_0}{P_0}$$

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

где коэффициенты $M_0, P_0, K_0$ были получены в лекции 7. Затем по формулам находим последующие прогоночные коэффициенты:

$$\xi_{n+1} = \frac{C_n}{B_n - A_n\xi_n}$$

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

Получим $y_N$, решив полученное уравнение в п.2, например, методом дихотомии. Далее по прогоночной формуле можно найти все значения неизвестных $y_n$:

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

**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). Оба краевых условия линейные.**

1. Для начала нужно вычислить начальные прогоночные коэффициенты.

Для правой прогонки они вычисляются по:

$$\xi_1 = \frac{-M_0}{P_0}$$

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

Для левой прогонки по:

$$\alpha_{N-1} = \frac{-M_N}{K_N}$$

$$\beta_{N-1} = \frac{-P_N}{K_N}$$

2. Найдем прогоночные коэффициенты:

Для левой прогонки:

$$\xi_{n+1} = \frac{C_n}{B_n - A_n\xi_n}$$

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

Для правой прогонки:

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

$$\beta_{n-1} = \frac{F_n + C_n\beta_n}{B_n - C_n\alpha_n}$$

3. Левая и правая прогонки:

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

$$y_n = \alpha_{n-1}y_{n+1} + \beta_{n-1}$$

4. Выразим $y_p$:

$$y_{p-1} = \xi_p y_p + \eta_p$$

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

$$y_p = \frac{\xi_{n+1}\beta_n + \eta_{n+1}}{1 - \xi_{n+1}\alpha_n}$$