



Министерства 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» \_\_\_\_\_

ДИСЦИПЛИНА «Моделирование» \_\_\_\_\_

### Лабораторная работа № 3

**Тема** «Программно—алгоритмическая реализация моделей на основе  
ОДУ второго порядка с краевыми условиями II и III рода.»

**Студент** Блохин Д.М.

**Группа** ИУ7 – 62Б

**Оценка (баллы)** \_\_\_\_\_

**Преподаватель** Градов В.М.

Москва.  
2021 г.

## Цель работы:

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

## Исходные данные:

1. Задана математическая модель.

Квазилинейное уравнение для функции  $T(x)$

$$\frac{d}{dx} \left( \lambda(T) \frac{dT}{dx} \right) - 4 \cdot k(T) \cdot n_p^2 \cdot \sigma \cdot (T^4 - T_0^4) = 0 \quad (1)$$

Краевые условия

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

2. Функции  $\lambda(T)$ ,  $k(T)$  заданы таблицей

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

3. Разностная схема с разностным краевым условием при  $x = 0$  была получена в лекции №7:

Имеем систему линейных уравнений с трехдиагональной матрицей

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

где

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

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

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

$$D_n = f_n h.$$

Решается она методом прогонки. Интегрируем на отрезке, выход вычисляем методом трапеций и в конце концов, спустя некоторое количество подстановок, получаем разностный аналог при  $x = 0$ :

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

где

$$M_0 = \chi_{1/2} + \frac{h^2}{8} p_{1/2} + \frac{h^2}{4} p_0,$$

$$K_0 = \frac{h^2}{8} p_{1/2} - \chi_{1/2}, \quad P_0 = h F_0 + \frac{h^2}{4} (f_{1/2} + f_0).$$

Для расчета величин в половинном узле можно принять простую аппроксимацию

$$p_{1/2} = \frac{p_0 + p_1}{2}, \quad f_{1/2} = \frac{f_0 + f_1}{2}.$$

При выполнении ЛР учесть

$$F_N = \alpha_N (y_N - T_0), \text{ а } F_{N-1/2} = \chi_{N-1/2} \frac{y_{N-1} - y_N}{h}. \quad (2),(3)$$

4. Значения параметров для отладки (все размерности согласованы)

$n_p = 1.4$  – коэффициент преломления,

$l = 0.2$  см – толщина слоя,

$T_0 = 300\text{K}$  – температура окружающей среды,

$\sigma = 5.668 \cdot 10^{-12}$  Вт/(см<sup>2</sup>К<sup>4</sup>)- постоянная Стефана- Больцмана,

$F_0 = 100$  Вт/см<sup>2</sup> - поток тепла,

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

5. Выход из итераций организовать по температуре и по балансу энергии, т.е.

$$\max \left| \frac{y_n^s - y_n^{s-1}}{y_n^s} \right| \leq \varepsilon_1, \text{ для всех } n = 0, 1, \dots, N.$$

и

$$\max \left| \frac{f_1^s - f_2^s}{f_1^s} \right| \leq \varepsilon_2,$$

где

$$f_1 = F_0 - \alpha(T(l) - T_0) \text{ и } f_2 = 4n_p^2 \sigma \int_0^l k(T(x))(T^4(x) - T_0^4) dx.$$

**Физическое содержание задачи** (для понимания получаемых результатов при отладке программы).

Сформулированная математическая модель описывает температурное поле  $T(x)$  в плоском слое с внутренними стоками тепловой энергии. Можно представить, что это стенка из полупрозрачного материала, например, кварца или сапфира, нагружаемая тепловым потоком на одной из поверхностей (у нас — слева). Другая поверхность (справа) охлаждается потоком воздуха,

температура которого равна  $T_0$ . Например, данной схеме удовлетворяет цилиндрическая оболочка, ограничивающая разряд в газе, т.к. при больших диаметрах цилиндра стенку можно считать плоской. При высоких температурах раскаленный слой начинает объемно излучать, что описывает второе слагаемое в (1) (закон Кирхгофа). Зависимость от температуры излучательной способности материала очень резкая. При низких температурах стенка излучает очень слабо, второе слагаемое в уравнении (1) практически отсутствует. Функции  $\lambda(T)$ ,  $k(T)$  являются, соответственно, коэффициентами теплопроводности и оптического поглощения материала стенки.

Ниже приведен листинг программы.

### Листинг 1. Основная функция моделирования

```
def modeling():
    while (count < 200 and max_abs_y(y_n, count) and max_abs_f(count)):
        k0, M0, P0 = left()
        kN, MN, PN = right()
        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]))
            eps.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])
            y_n[count] = t[n]
            count += 1
            h = h / 2
        x = [i for i in np.arange(0, N, h)]
        plt.plot(x, t[:-1])
        plt.xlabel("x, cm")
        plt.ylabel("T, K")
        plt.grid()
        plt.show()
```

Листинг 2. Вспомогательные функции.

```
def left():
    k0 = approx1_2(getK, 0) + (h * h * approx1_2(p, 0) / 8) + \
        ((h * h * p(0)) / 4)
    M0 = -approx1_2(getK, 0) + (h * h * approx1_2(p, 0) / 8)
    P0 = h * F0 + (h * h / 4) * (approx1_2(f, 0) + f(0))
    return k0, M0, P0

def right():
    kN = (approx1_2(getK, N) / h) - (h * approx1_2(p, N) / 8)
    MN = -getLm(N) - (approx1_2(getK, N) / h) - \
        (h * p(N) / 4) - (h * approx1_2(p, N) / 8)
    PN = -(h / 4) * (f(N) + approx1_2(f, N)) - T0 * getLm(N)
    return kN, MN, PN

def simpson(func, a, b, N=50):
    dx = (b - a) / N
    even = 0.0
    odd = 0.0
    for i in range(1, N):
        if i % 2 == 0:
            even += func(left + dx * i)
        else:
            odd += func(left + dx * i)

    S = dx / 3 * (2 * even + 4 * odd + func(left) + func(right))

    return S
```

## Результаты работы программы:

1. Представить разностный аналог краевого условия при  $x = 1$  и его краткий вывод интегроинтерполяционным методом.

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

$$p(x) = 0$$

$$f(x) = -4k(T)n_p^2 * \sigma * (T^4 - T_0^4)$$

Тогда (3) можно записать как

$$\frac{dF}{dx} - p(x) + f(x) = 0$$

Причем  $p(x) = 0$ . Следовательно:

$$\frac{dF}{dx} + f(x) = 0$$

Проинтегрируем последнее на отрезке:  $[x_{N-\frac{1}{2}}; x_N]$

$$\int_{x_{N-\frac{1}{2}}}^{x_N} \frac{dF}{dx} dx + \int_{x_{N-\frac{1}{2}}}^{x_N} f(x) dx = 0$$

Применяем метод трапеций ко второму и третьему интегралу, получаем:

$$-(F_N - F_{N-\frac{1}{2}}) + \frac{h}{4}(f_N + f_{N-\frac{1}{2}}) = 0$$

Применив 2,3 к последнему, получим:

$$\alpha_N(y_N - T_0) - X_{N-\frac{1}{2}} * \frac{y_{N-1} - y_N}{h} + \frac{h}{4}(f_N + f_{N-\frac{1}{2}}) = 0$$

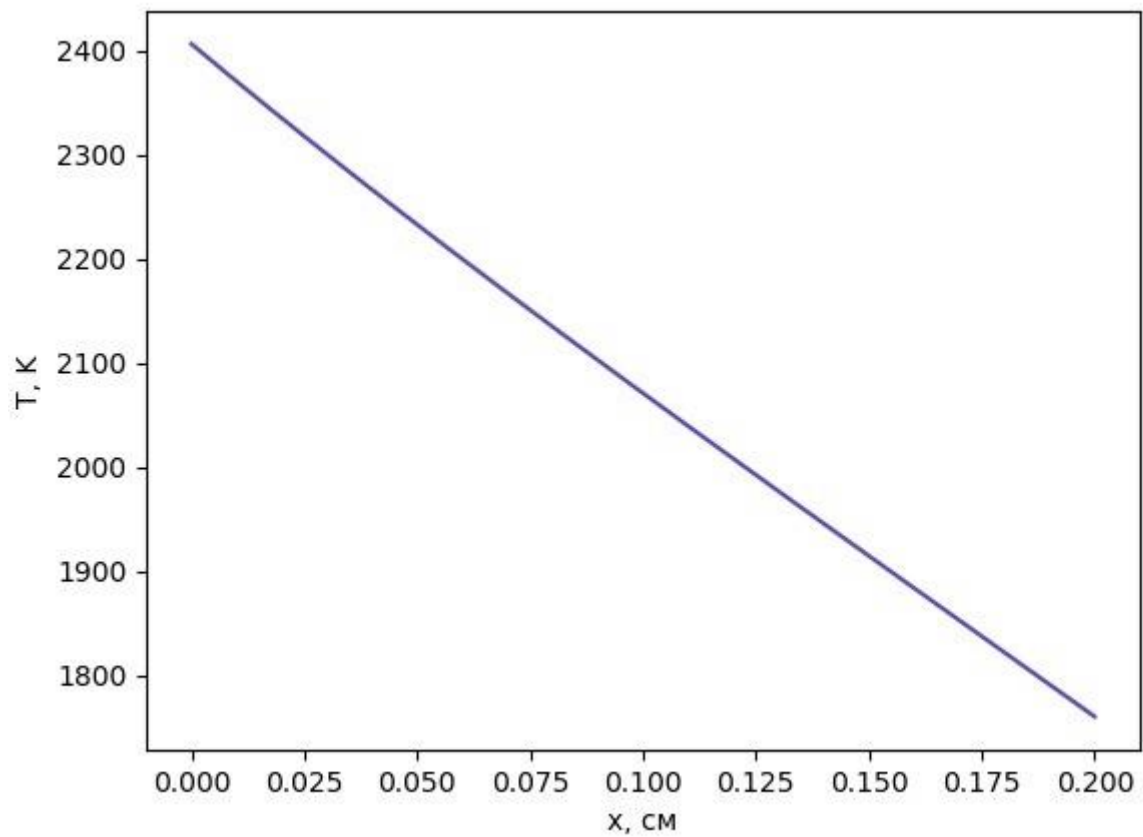
$$T_0 \alpha_N + \frac{h}{4}(f_N + f_{N-\frac{1}{2}}) = y_N(\alpha_N + \frac{X_{N-\frac{1}{2}}}{h}) - y_{N-1} \frac{X_{N-\frac{1}{2}}}{h}$$

$$-T_0 \alpha_N - \frac{h}{4}(f_N + f_{N-\frac{1}{2}}) = y_N(-\alpha_N - \frac{X_{N-\frac{1}{2}}}{h}) + y_{N-1} \frac{X_{N-\frac{1}{2}}}{h}$$

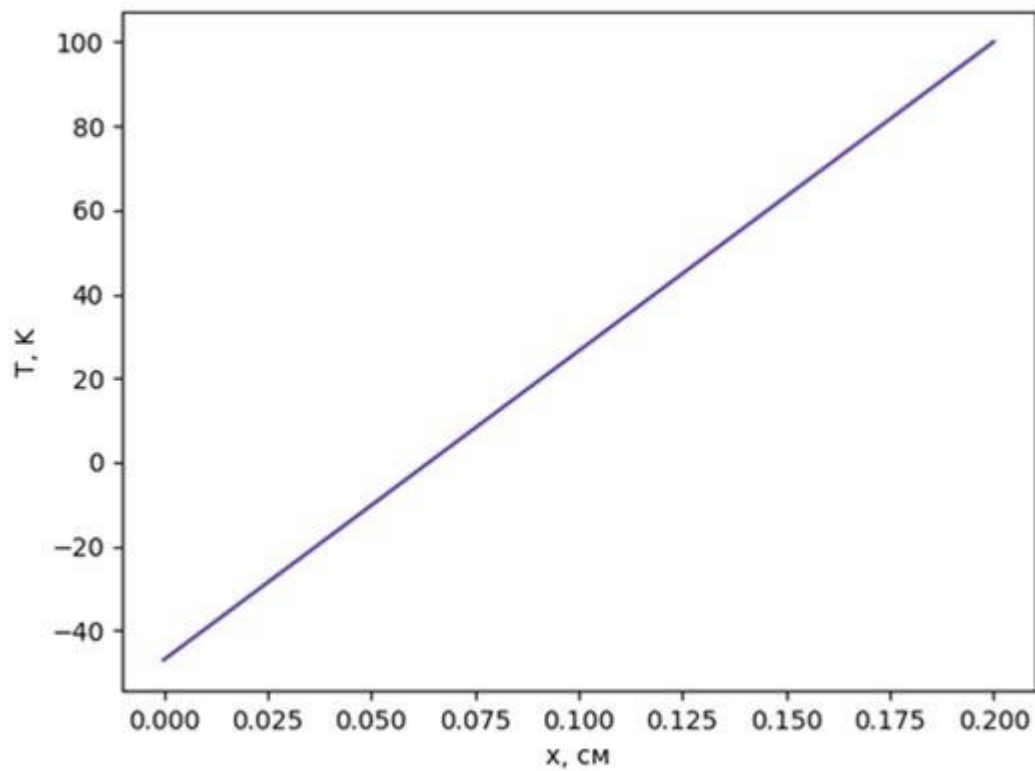
Откуда

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

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

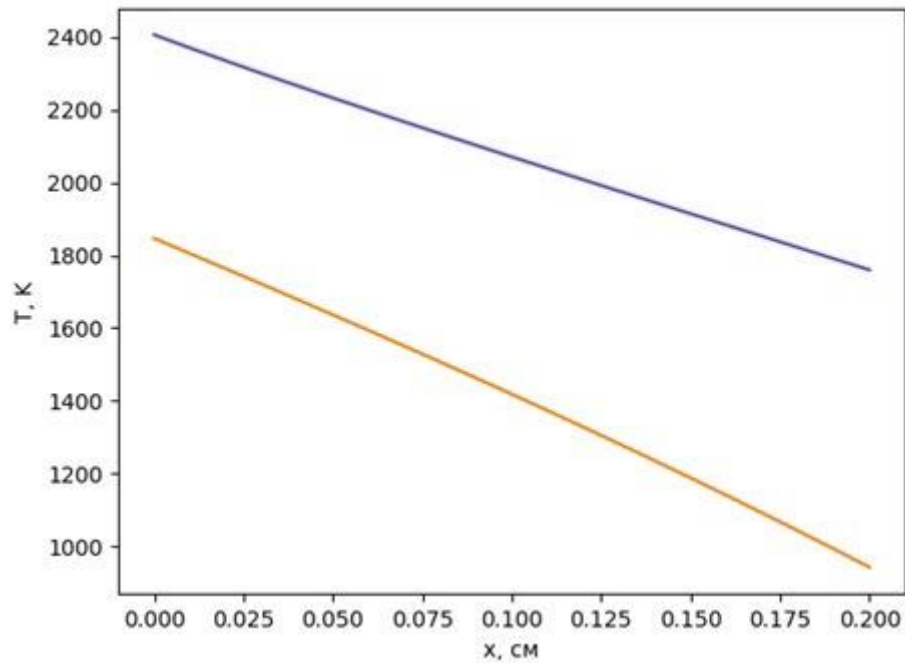


3. График зависимости  $T(x)$  при  $F_0 = -10$  Вт/см<sup>2</sup>



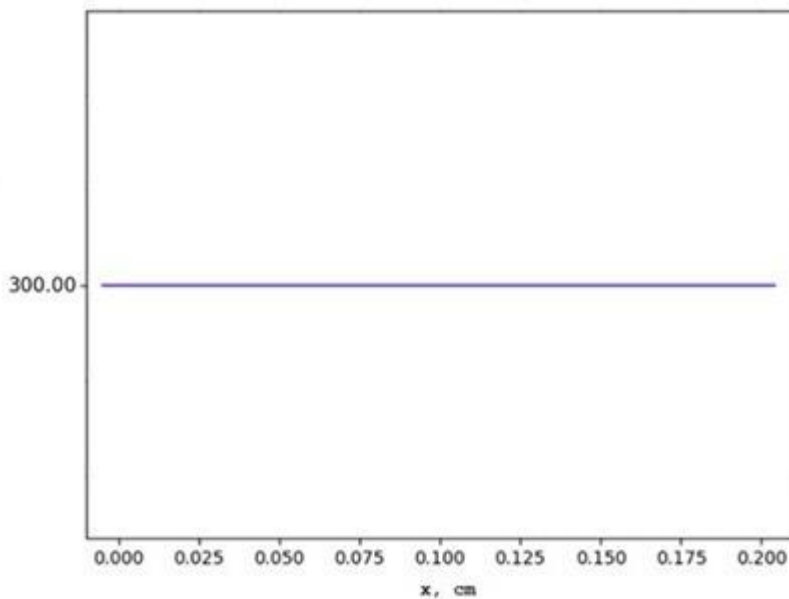


4. График зависимости  $T(x)$  при увеличенных значениях  $\alpha$  (например, в 3 раза). Сравнить с п.2



Фиолетовый график – график из пункта 2, желтый график – график при увел. коэф.

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



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

$$f_1 = F_0 - \alpha(T(l) - T_0) \text{ и } f_2 = 4n_p^2 \sigma \int_0^l k(T(x))(T^4(x) - T_0^4) dx.$$

Каковы использованные в работе значения точности выхода из итераций  $\varepsilon_1$  (по температуре) и  $\varepsilon_2$  (по балансу энергии)?

Точность выхода  $\varepsilon_1$  (по температуре) = 0.087

Точность выхода  $\varepsilon_2$  (по балансу энергии) = 1.17

### Вопросы при защите лабораторной работы:

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

Программа должна работать, не нарушая законам физики, и результат не должен им противоречить. То есть отталкиваясь от физического смысла задачи, можно представить, как должен вести себя результат.

К примеру, при изменении значения  $F_0$  температура будет изменяться. При удалении от начала стержня она будет становиться меньше.

Также можно увеличить или уменьшить длину стержня. Температура будет уменьшаться или увеличиваться соответственно. При увеличении стержня температура будет стремиться к  $T_0$ .

#### 2. Получите простейший разностный аналог нелинейного краевого условия при $x = l$ . Производную аппроксимируйте односторонней разностью.

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

где  $\phi(T)$  - заданная функция.

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

Или:

$$-k_N * (y_N - y_{N-1}) - h(\alpha_N (y_N - T_0) + \phi(y_N)) = 0$$

Приводим подобные и получаем:

$$y_N(-k_N - h\alpha_N) + k_N y_{N-1} + \alpha_N h T_0 + \phi(y_N) = 0$$

#### 3. 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.

Вычисляем начальные прогоночные коэффициенты.

Для правой по:

$$\xi_1 = \frac{-M_0}{P_0}; \eta_1 = \frac{-K_0}{P_0}$$

Для левой по:

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

Находим остальные прогоночные коэффициенты.

Для правой:

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

Для левой:

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

--

Получаем формулы:

$$\begin{cases} y_n = \xi_{n+1} y_{n+1} + \eta_{n+1} \\ y_n = \alpha_{n-1} y_{n+1} + \beta_{n-1} \end{cases}$$

Далее составляем систему уравнений, из которой выражаем  $y_p$ :

$$\begin{cases} y_{p-1} = \xi_p y_p + \eta_p \\ y_p = \alpha_{p-1} y_{p-1} + \beta_{p-1} \end{cases}$$

В итоге:

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

