МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное агентство по образованию

НАЦИОНАЛЬНЫ ИССЛЕДВАТЛЬСКИЙ УНИВЕРСИТЕТ "МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОННОЙ ТЕХНИКИ"

Кафедра высшей математики 1

Пономарев Александр Олегович

РЕФЕРАТ

Решение практической задачи расчета системы охлаждения современного центрального процессора в персональном компьютере посредством системы с воздушным охлаждением

3 курс, группа ПМ-31

Преподаватель
__________________ М. А. Гурьянов
«____» ___________ 2020 г.

Зеленоград, 2020 г.

Содержание

.

1 Введение

В данной работе решаем практическую задачу расчета системы охлаждения современного центрального процессора в персональном компьютере посредством системы с воздушным охлаждением.

Мы выбрали для себя реально существующую систему воздушного охлаждения для проведения работы и реально существующую модель центрального процессора. На основании их построили математическую модель, состоящую из уравнений теплопроводности, граничных и начальных условий. Решили их численно, рассмотрев различные варианты разностной аппроксимации линейного одномерного по пространству уравнения теплопроводности.

2 Выбор системы охлаждения центрального процессора

2.1 Выбор модели

В качестве системы воздушного охлаждения мы выбрали систему охлаждения с тепловыми трубками: SHADOW ROCK SLIM. Он содержит в себе теплорастекательный элемент формы пластина и испарительный элемент типа стержень.

Характеристики системы охлаждения:

- Размер: 135 x 135 x 22 мм;
- Максимальная скорость вращения: 1400 rpm;
- Максимальный воздушный поток: 113.8 м3/ч;
- Максимальное воздушное давление: 2.1/1.23 мм водного столба;
- Количество тепловых трубок: 8 шт;
- Диаметр тепловой трубки: 6 мм;
- Длина трубки: 458 мм;
- Расстояние между пластинами: 2 мм;
- Ширина пластины: 0,35 мм;

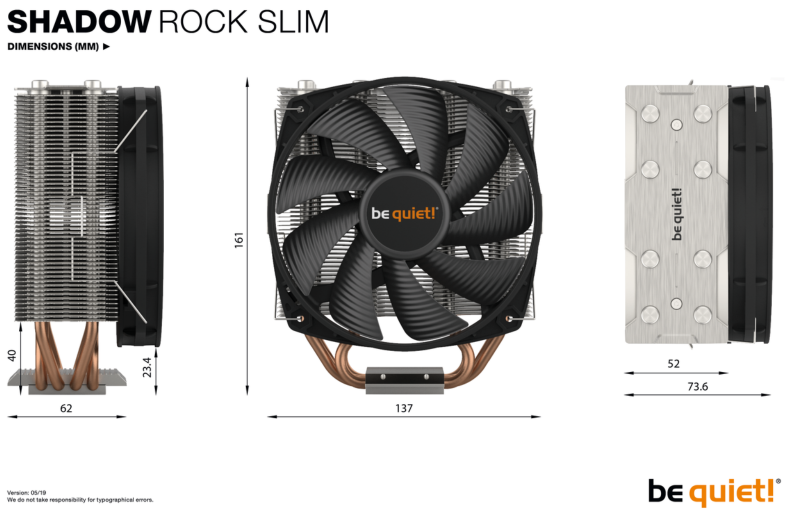

Рис. 1: Схема процессора

Любой экземпляр данного типа можно условно разбить на группы из однотипных элементов. В данном случае это пластина (их 52 одинаковых: 52 мм (Длина) x 130 мм (Ширина) x 161 мм (Высота)) и тепловая трубка (их 8 одинаковых по 6 мм).

При этом пластину можно представить в виде прямоугольника. На пластине можно выделить точки, по которым в пластину втекает тепло (точечные источники тепла).

Стержень, с другой стороны, можно представить в виде одномерной протяженной структуры, в центре у которой расположен точечный источник тепла (процессор), а по краям расположены точки крепления к пластинам радиатора (X1...X52). Стержень, очевидно, симметричен, так что для простоты решения удобно перенести точку начала координат в точку симметрии. Результат показан на Рис. 3.

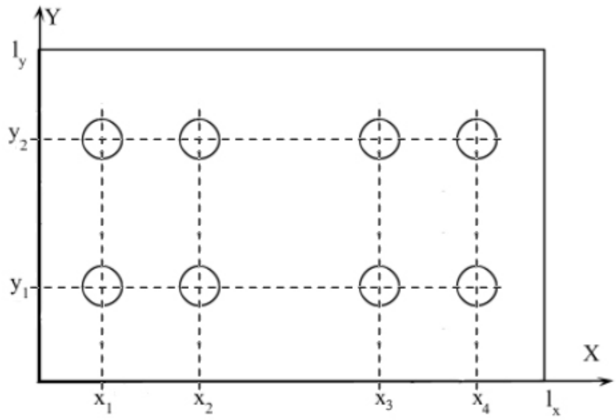

Рис. 2: Сеточная область пластины

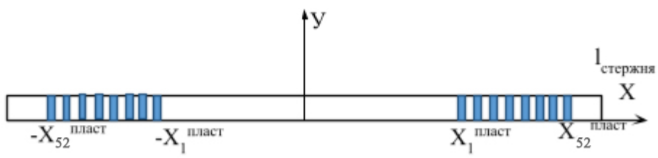

Рис. 3: Сеточная область трубки

Параметр	Значение, м
l_{tube}	$458 * 10^{-3}$
l_x	$130 * 10^{-3}$
l_y	$52 * 10^{-3}$
x_1	$14 * 10^{-3}$
x_2	$42 * 10^{-3}$
x_3	$88 * 10^{-3}$
x_4	$113 * 10^{-3}$
y_1	$12 * 10^{-3}$
y_2	$26 * 10^{-3}$
$X_1^{plate}...X_{52}^{plate}$	$2 * 10^{-3}$

3 Теоретическая постановка задачи

3.1 Вывод уравнения теплопроводности

Рассмотрим физическое тело, температура которого в точке с координатами (x, y, z) в момент времени t определяется функцией $u(x, y, z, t)$. Если температура различных частей тела различна, то в теле будет происходить перенос тепла от более нагретых участков тела к менее нагретым. Рассмотрим какую-нибудь поверхность S внутри тела и обозначим ее малый элемент Δs. Тогда количество тепла, проходящего через элемент поверхности Δs за время Δt, будет определяться выражением

$$\Delta Q = -k\frac{\partial u}{\partial \vec{n}}\Delta s \Delta t = -k\Delta s\Delta t\nabla_n u \tag{1}$$

где $k > 0$ – коэффициент теплопроводности, а $\vec{n}$ – нормаль к элементу поверхности Δs в направлении распространения тепла. Здесь и далее $\nabla_n u = (\nabla u * \vec{n})$, а ∇u означает градиент функции u. Будем считать, что тело изотропно в отношении теплопроводности, т. е. коэффициент теплопроводности не зависит от направления нормали, а зависит только от координат точки. Кроме того, обозначим поток тепла через единичную площадку через q т. е.

$$q = -k\frac{\partial u}{\partial \vec{n}} \tag{2}$$

Тогда для произвольного объема V , ограниченного замкнутой поверхностью S, изменение количества тепла за промежуток времени $(t1, t2)$ будет определяться интегралом

$$Q_1 = -\int\limits_{t_1}^{t^2} dt \iint\limits_{S} k(x, y, z)\frac{\partial u}{\partial \vec{n_0}} dS \tag{3}$$

где $\vec{n0}$ – внутренняя нормаль к поверхности S.

С другой стороны, количество тепла, необходимое для изменения температуры выделенного объема тела на $\Delta u = u(x, y, z, t_2) - u(x, y, z, t_1)$, равно

$$Q_2 = -\int\limits_{t_1}^{t^2} dt \iiint\limits_{V} \rho\gamma\frac{\partial u}{\partial t} dV \tag{4}$$

где ρ и γ – плотность и теплоемкость данного физического тела.

Если внутри рассматриваемого тела имеются источники тепла плотности $F(x, y, z, t)$, то количество тепла, выделяемого (или поглощаемого) ими в объеме V за промежуток времени (t_1, t_2), будет равно

$$Q_3 = \int\limits_{t_1}^{t^2} dt \iiint\limits_{V} F(x, y, z, t) dV \tag{5}$$

Очевидно, что баланс тепла для объема V определяется соотношением $Q_2 = Q_1 + Q_3$, т. е.

$$\int\limits_{t_1}^{t^2} dt \iiint\limits_{V} \rho\gamma\frac{\partial u}{\partial t} dV = -\int\limits_{t_1}^{t^2} dt \iint\limits_{S} k(x, y, z)\frac{\partial u}{\partial \vec{n_0}} dS + \int\limits_{t_1}^{t^2} dt \iiint\limits_{V} F(x, y, z, t) dV \tag{6}$$

Применяя к поверхностному интегралу формулу Гаусса-Остроградского, получим

$$\int_{t_1}^{t_2} dt \iiint_V \left[\rho\gamma\frac{\partial u}{\partial t} - div(k\, grad\, u) - F(x,y,z,t)\right] dV = 0 \tag{7}$$

где $grad\, u = \nabla u = \frac{\partial u}{\partial x}\vec{i} + \frac{\partial u}{\partial y}\vec{j} + \frac{\partial u}{\partial z}\vec{k}$.

Так как подынтегральная функция непрерывна, а объем V и промежуток времени (t_1, t_2) произвольны, то для любой точки (x, y, z) и любого момента времени справедливо соотношение

$$\rho\gamma\frac{\partial u}{\partial t} = div\,(k\, grad\, u) + F(x,y,z,t) \tag{8}$$

Это уравнение называется уравнением теплопроводности неоднородного изотропного тела. Если тело однородно, то коэффициенты γ, ρ, k являются постоянными и уравнение теплопроводности принимает вид

$$\frac{\partial u}{\partial t} = a^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + f(x,y,z,t) \tag{9}$$

где $div\,(grad\, u) = \Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$, $f(x,y,z,t) = \frac{F(x,y,z,t)}{\gamma\rho}$, $a = \sqrt{\frac{k}{\gamma\rho}}$

3.2 Постановка начально-краевых задач для одномерного уравнения теплопроводности

Как правило, для нахождения неизвестной функции $u(x,t)$, удовлетворяющей уравнению теплопроводности, когда пространственная переменная x принадлежит некоторому интервалу $(0, l)$, а время t больше нуля, необходимо задать начальное условие (при $t = 0$) и краевые условия (при $x = 0$, $x = l$).

Существует несколько стандартных постановок начально-краевых задач для уравнения теплопроводности на отрезке.

Математическая постановка первой начально-краевой задачи имеет вид:

$$u_t = a^2 u_{xx} + f(x,t), \quad x \in (0,l), \quad t > 0 \tag{10}$$

$$u|_{t=0} = \varphi(x)$$

$$u(0,t) = \mu_1(t)$$

$$u(l,t) = \mu_2(t)$$

Здесь $u(x,t)$ подлежит определению, функции $f(x,t)$, $\varphi(x)$, $\mu_1(t)$, $\mu_2(t)$ считаются известными.

Если речь идет о процессах теплопроводности, то физическая формулировка первой краевой задачи выглядит так: найти распределение температуры $u(x,t)$ в стержне $x \in [0,l]$, если по стержню непрерывно распределены источники тепла плотности $f(x,t)$, начальное распределение температуры $\varphi(x)$ задано, на левом конце стержня температура равна $\mu_1(t)$, а на правом - $\mu_2(t)$.

Вторая краевая задача отличается от первой тем, что вместо температура на концах стержня задаются тепловые потоки:

$$-k u_x(0,t) = q_1(t)$$

$$ku_x(l,t) = q_2(t)$$

Так как коэффициент теплопроводности k и величины потоков $q_1(t)$, $q_2(t)$ считаются известными, то задание потоков равносильно заданию значений первых производных функции $u(x,t)$ по переменной x на концах стержня.

Полностью математическая постановка второй начально-краевой задачи выглядит так:

$$u_t = a^2 u_{xx} + f(x,t), \quad x \in (0,l), \quad t > 0; \tag{11}$$

$$u|_{t=0} = \varphi(x);$$

$$u_x(0,t) = v_1(t);$$

$$u_x(l,t) = v_2(t)$$

3.3 Численное решение уравнения теплопроводности

Рассмотрим простейшие примеры сеток.

Пусть область изменения аргумента x есть отрезок $0 \le x \le l$. Разобъем отрезок $0 \le x \le l$ точками $x_i = ih$ $(i = 0, 1, 2..., N; h > 0)$ на N равных частей длины $h = \frac{l}{N}$ каждая. Множеств точек

$$x_i = ih \quad (i = 0, 1, 2..., N; h > 0) \tag{12}$$

называется разностной сеткой на отрезке $0 \le x \le l$ и обначается $\bar{\omega}_h = ih, i = 0, 1, 2..., N$, а число h - расстояние между точками (узлами) сетки $\bar{\omega}_h$ - называется шагом сетки.

Отрезок $[0,l]$ можно разбить на N частей, вводя произвольные точки $x_1 \le x_2 \le ... \le x_{N-1} \le l$. Тогда получим сетку

$$\bar{\omega_h} = x_i, i = 0.1, ..., N, \, x_0 = 0, \, x_N = l \tag{13}$$

с шагом $h_i = x_i - x_{i-1}$, который зависит от номера i узла x_i. Если $h_i \neq h_{i+1}$ хотя бы для дно номера i, то сетка $\bar{\omega}_h = \bar{\omega}_h^*$ называется неравномерной. Если $h_i = const = h = \frac{l}{N}$ для всех $i = 1, 2, ..., N$, то мы получаем построенную выше равномерную сетку.

На бесконечной прямой $-\infty < x < \infty$ можно рассматривать сетку

$$\omega_{h,x} = x + oh, i = 0, \pm 1, \pm 2, ... \tag{14}$$

с началом в любой точке x, состоящую из бесконечного числа узлов.

Функцию $y_i = y(x_i)$ дискретного аргумента x_i, $i = 0, 1, ..., N$, называют сеточной функцией, определенной на сетке $\overline{\omega}_h$.

Всякой непрерывной функции $f(x)$ можно поставить в соответствие сеточную функцию f_i^h, плагая, например, $f_i^h = f(x_i)$. Впрочем, в некоторых случаях удобнее устанавливать это соответствие другими способами.

Пусть область изменения аргументов (x,t) есть прямоугольник $\bar{D} = (0 \le x \le 1, 0 \le t \le T)$. Построим на отрезке $0 \le x \le 1$ сетку $\bar{\omega}_h = x_i = ih, i = 0, 1, ..., N$ с шагом $h = \frac{l}{N}$ и сетку $\bar{\omega}_\tau = t_i = j\tau, j = 0, 1, ..., N_0$ с шагом $\tau = \frac{T}{N_0}$ на отрезке $0 \le t \le T$. Множество узлов (x_i, t_j) с координатами $x_i = ih$ и $t_j = j\tau$ назовем сеткой в прямоугольнике $\bar{D}$ и обозначим

$$\bar{}_{h\tau} = (x_i = ih, t = j\tau),\ i = 0, 1, ..., N, = 0, 1, ..., N_0 \tag{15}$$

Эта сетка равноомерна по каждму и прменных x и t. Если хотя бы дна и сетк $\bar{\omega}_{h\tau}$ называется неравномерной. Сетка $\bar{\omega}_{h\tau}$, очевидно, состоит из точек пересения прямых $x = x_i$, $i = 0, 1, ..., N$ и прямых $t = t_j$, $= 0, 1, ..., N_0$.

Пусть y - сеточная функция, заданная на $\bar{\omega}_{h\tau}$. Будем обозначать $y_i^j = y(x_i, y_j)$ значение сеточной функции y в узл x_i, t_j сетки $\bar{\omega}_{h\tau}$.

Непрерывной функции $u(x,t)$, где (x,t) - точка из $\bar{D}$, будм ставить в соответствие сеточную функцию

$$u_i^j = u_{i,h,\tau}^j = u(x_i, t_j) \tag{16}$$

3.4 Явная схема

Уравнение теплопроводности:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(x,t),\ x \in (0,l),\ t \in (0,T] \tag{17}$$

Для аппроксимации оператора $L = \frac{\partial}{\partial t} - \frac{\partial^2}{\partial x^2}$ в уравнении (17) используем шаблон, приведенный на рис.4

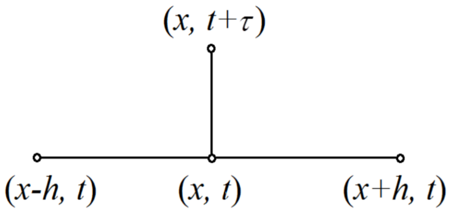

Рис. 4: Шаблон явной схемы для уравнения теплопроводности

Соответствующий разностный оператор $L_{h\tau}^{(0)}u$ и имеет вид:

$$L_{h\tau}^{(0)}u = \frac{u(x,t) + \tau - u(x,t)}{\tau} - \frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2} \tag{18}$$

Далее для кратности будем использовать следующие стандартные обозначения:

$$u = u(x,t); \quad u^* = u(x, t+\tau)$$

Тогда:

$$u_t = \frac{u^* - u}{\tau}, \quad L_{h\tau}^{(0)}u = u_t - u_{\bar{x}x}$$

Найдем погрешность аппроксимации разностным оператором $L_{h\tau}^{(0)}$ исходного дифференциального оператора L в точке (x,t). В случае достаточно гладкой функции $u(x,t)$ при достаточно малых шагах h и τ имеем:

$$u_t = \frac{u(x,t+\tau) - u(x,t)}{\tau} = \frac{\partial u(x,t)}{\partial t} + O(\tau) \tag{19}$$

$$u_{\bar{x}x} = \frac{\partial^2 u(x,t)}{\partial x^2} + O(h^2) \tag{20}$$

Следовательно, разностный оператор $L_{h\tau}^{(0)}$ аппроксимирует дифференциальный оператор L с погрешностью $O(\tau + h^2)$ в точке (x,t):

$$L_{h\tau}^{(0)} u = \underbrace{\frac{\partial u(x,t)}{\partial t} - \frac{\partial^2 u(x,t)}{\partial x^2}}_{L[u(x,t)]} + O(\tau + h^2) \tag{21}$$

Введем сеточную функцию $\phi = \varphi(x_i, t_j)$, аппроксимирующую правую часть $f(x,t)$ уравнения (17) на всех внутренних узлах (x_i, t_j) сетки с погрешностью $O(\tau + h_2)$. В качестве φ можно взять, например $\varphi(x_i, t_j) = f(x_i, t_j)$. Тогда разностное уравнение

$$L_{h\tau}^{(0)} y = \varphi \tag{22}$$

будет аппроксимировать исходное дифференциальное уравнение теплопроводности (17) с первым порядком погрешности по τ и вторым по h.

3.5 Неявная схема

Используем для аппроксимации оператора $L = \frac{\partial}{\partial t} - \frac{\partial^2}{\partial x^2}$ в уравнении (17) шаблон, приведенный на Рис.5

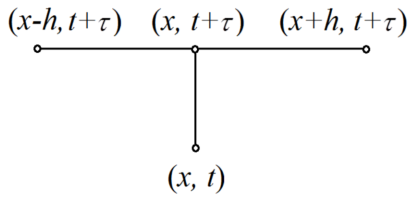

Рис. 5: Шаблон ytявной схемы для уравнения теплопроводности

Тогда разностная аппроксимация оператора L уравнения теплопроводности будет выглядеть следующим образом:

$$L_{h\tau}^{(1)} u = \frac{u(x, t+\tau) - u(x,t)}{\tau} - \frac{u(x+h, t+\tau) - 2u(x, t+\tau) + u(x-h, t+\tau)}{h^2} = u_t - u^*_{\bar{x}x} \tag{23}$$

Рассмотрим погрешность аппроксимации разностным оператором $L_{h\tau}^{(1)}$ исходного дифференциального оператора L в точках (x,t), $(x, t+\tau)$. Так как для достаточно гладкой функции $u(x,t)$ справедливы равенства

$$u^*_{\bar{x}x} = \frac{\partial^2 u(x, t+\tau)}{\partial x^2} + O(h^2) = \frac{\partial^2 u(x,t)}{\partial x^2} + O(\tau + h^2) \tag{24}$$

то с учетом (19) получаем, что оператор $L_{h\tau}^{(1)}$ аппроксимирует дифференциальный оператор L в уравнении (17) с погрешностью $O(\tau + h^2)$ в точках (x,t) и $(x, t+\tau)$:

$$L_{h\tau}^{(1)} u = \frac{\partial u(x,t)}{\partial t} + O(\tau) - \frac{\partial^2 u(x,t)}{\partial x^2} + O(\tau + h^2) = \underbrace{\frac{\partial u(x,t)}{\partial t} - \frac{\partial^2 u(x,t)}{\partial x^2}}_{L[u(x,t)]} + O(\tau + h^2) \tag{25}$$

$$L_{h\tau}^{(1)}u = \frac{\partial u(x,t+\tau)}{\partial t} + O(\tau) - \frac{\partial^2 u(x,t+\tau)}{\partial x^2} + O(h^2) = \underbrace{\frac{\partial u(x,t+\tau)}{\partial t} - \frac{\partial^2 u(x,t+\tau)}{\partial x^2}}_{L[u(x,t+\tau)]} + O(\tau + h^2) \quad (26)$$

Беря в качестве сеточной аппроксимации правой части уравнения (15), например, функцию $\varphi(x_i, t_j) = f(x_i, t_{j+1})$, получим разностное уравнение

$$L_{h\tau}^{(1)}y = \varphi \quad (27)$$

аппроксимирующее (17) с погрешностью $O(\tau + h^2)$.

4 Практическая построение математической модели

4.1 Построение математической модели

При построении математической модели нужно учитывать следующие вещи:
Для пластины:

1. Коэффициент a опрделяется материалом
2. Чем горячее пластина, тем больше скорость теплоотдачи
3. Функции равна нулю везде кроме $x = x_j$
4. Кулер пытается охладить пластину
5. Теплоперенос пропорционален разнице температур
6. Коеффициент теплопроводности зависимост от способа крепления пластин
7. Температура в комнате $+24^oC$
8. На границах пластины отсутствует теплообмен
9. Охлаждение пропорционально скорости вращения кулера

Теперь адптируем дифференциальное уравнение теплопроводности:

$$V_t = a^2 V_{xx} - \sum_{x_j} \sigma(x - x_j) * k * (V(x_j, t) - U_j) - F \tag{28}$$

$$F = r * RPM$$

$$V'(0, t) = 0$$

$$V'(l, t) = 0$$

$$V(x, 0) = 24$$

Для стержня:

1. Коэффициент a определяется материалом
2. Чем горячее стержень, тем больше скорость теплоотдачи
3. Функции равна нулю везде кроме $x = x_i$
4. Теплоперенос пропорционален разнице температур
5. Коэффициент теплопроводности зависит от способа крепления пластин
6. Температура в комнате $+24^oC$
7. В другом конце стержня движение тепла отсутствует
8. В начале стержня непрерывный источник тепла

Теперь адптируем дифференциальное уравнение теплопроводности:

$$U_t = a^2 U_{xx} - \sum_{x_i} \sigma(x - x_i) * k * (U(x_i, t) - V_i) \tag{29}$$

$$U'(0,t) = +p/4$$

$$U'(l,t) = 0$$

$$U(x,0) = 24$$

Перейдем к составлению математической модели системы охлаждения современного центрального процессора:

Краевая задача 1:

$$U'_{j_t}(x_n, t_m) = a^2 U''_{j_{xx}}(x_n, t_m) - \sum_{x_i} \sigma(x_n - x_i) * k * (U_j(x_n, t_{m-1} - V_i(x_i^v, t_{m-1})) \tag{30}$$

$$U'_j(0,t) = +p/4$$

$$U'_j(l,t) = 0$$

$$U_j(x,0) = 24$$

$$x_n = x \times dx, \quad n \in \left[1, \frac{l}{dx}\right]$$

$$t_m = m \times dt, \quad m \in N$$

$$j = 1, 2, ..., 8$$

$$j = 1, 2, ..., 52$$

Краевая задача 2:

$$V'_{i_t}(x_n, t_m) = a^2 V''_{i_{xx}}(x_n, t_m) - \sum_{x_j} \sigma(x_n - x_j) * k * (V_i(x_n, t_{m-1}) - U_j(x_n^U, t_{m-1})) - F \tag{31}$$

$$F = r * RPM$$

$$V'(0,t) = 0$$

$$V'(l,t) = 0$$

$$V(x,0) = 24$$

$$x_n = x \times dx, \quad n \in \left[1, \frac{l}{dx}\right]$$

$$t_m = m \times dt, \quad m \in N$$

$$j = 1, 2, ..., 8$$

$$j = 1, 2, ..., 52$$

4.2 Факторы внешней среды

Теперь необходимо описать внешние условия:

Для задачи 1 (Стержень) факторами внешней среды (правая часть уравнения) будут: Тепло передаваемое процессором (Граничное условие №1 следовательно будет источником тепла) тепло отнимаемое пластинами – Задача 1 и задача 2 связаны.

Для задачи 2 (Пластина) факторами внешней (правая часть уравнения) среды будут: тепло.

4.3 Недостатки математичкой модели

1. Не учитываем толщину пластины.

   Верхня и нижняя грань в реальности будут иметь разную температуру, так как тепло передается внутри пластины не мгновенно.Также при изготовлении металических пластин, если оборудованние на заводе изношенно, то пластины получатся с подогнутыми краями, и воздух будет скапливаться за преградами, так как там зона пониженного давления.

2. Для одномерного случая не учитывается ширина пластин.

   Так как скорость передачи тепла внутри пластины не бесконечно большая, пластина в трехмерном случае будет иметь разную температуру в каждой точке пластины, что наша модель не учитывает.

3. Мы не учитываем толщину стержня и рассматриваем его, как одномерный случай.

4. Не учитываем турбулентность и охлаждение пластин и трубок окружающей средой.

   Но его толщиной можно пренебречь, потому что охлаждение окружающей средой на порядки меньше чем охлаждение куллером.

5. Математическая модель не учитывает того, что коэффициент теплопроводности, удельная теплоемкость и плотность зависят от температуры.

5 Моделирование средствами MATLAB

5.1 Реализация явной схемы

Рассмотрим начально-краевую задачу на отрезке $x \in [0, l]$:

$$\begin{cases} \dfrac{\partial u}{\partial t} = a^2 \dfrac{\partial^2 u}{\partial x^2} + f \\ u(x,0) = 24 \\ u'(0,t) = 0 \\ u'(0,l) = 0 \end{cases}$$

Для того, чтобы получить численное решение, введем в расчетной области равномерную сетку:

$$x_n = nh, \quad n = 0, 1, ..., N, \quad hN = l, \quad t_j = j\tau, \quad j = 0, 1, ..., J, \quad \tau J = l \tag{32}$$

такую что $\tau \leq h^2/2$. Для этого достаточно N задать произвольно, а J выбрать так, чтобы выполнялось неравенство $J \geq 2N^2$.

Построим разностную аппроксимацию уравнения в соответствии с явной схемой:

$$\frac{y_n^{j+1} - y_n^j}{\tau} = \frac{y_{n-1}^j - 2y_n^j + y_{n+1}^j}{h^2} + f_n, \quad n = 1, 2, ..., N-1, \quad j = 0, 1, ..., J-1 \tag{33}$$

Это разностное уравнение необходимо дополнить соответствующими начальными и граничными условиями на сетке. Начальное условие и граничное условие второго рода аппроксимируются точно:

$$y_n^0 = 24$$

Граничное условие при $x = 1$ содержит производную $\frac{\partial u}{\partial x}$. Если ее просто заменить односторонней разностной производной, то уравнение

$$\frac{y_N^j - y_{N-1}^j}{h} = 0, \quad j = 0, 1, ..., J \tag{34}$$

$$\frac{y_2^j - y_1^j}{h} = 0, \quad j = 0, 1, ..., J \tag{35}$$

будет аппроксимировать соответствующее граничное условие с первым порядком погрешности аппроксимации по h. Это означает, что и для всей разностной схемы порядок погрешности аппроксимации по h будет первым.

Для того, чтобы для всей схемы сохранить погрешность аппроксимации $O(\tau) + O(h^2)$, можно использовать различные подходы. Например, можно аппроксимировать граничное условие с помощью трехточечной односторонней производной.

В результате получим следующую разностную схему:

$$\begin{cases} \frac{y_n^{j+1-y_n^j}}{\tau} = \frac{y_{n-1}^j - 2y_n^j + y_{n+1}^1}{h^2} + f_n, \quad n = 1, ..., N-1, \quad j = 0, ..., J-1 \\ \frac{-3y_0^{j+1} + 4y_1^j j+1 - y_2^{j+1}}{2h} = 0; \quad \frac{3y_N^{j+1} - 4y_{N-1}^j j+1 - y_{N-2}^{j+1}}{2h} = 0, \quad n = 1, ..., N-1, \quad j = 0, ..., J-1 \\ y_n^0 = 24, \quad n = 0, 1, ..., N \end{cases}$$

Рассмотрим алгоритм решения этой системы. При j $=$ 0 значения y_n^0 известны из начального условия, а значения y_n^1 неизвестны и должны быть найдены для всех $n = 0, 1, ..., N$. Когда найдены все значения y_n^1, нужно найти y_n^2 и т.д. Следовательно, при каждом фиксированном $j = 0, 1, ..., J-1$ неизвестными являются значения y_n^{j+1}. Найти их можно следующим образом:

1) При $n = 1, 2, ..., N-1$ из (1):

$$y_n^{j+1} = y_n^j + \frac{\tau}{h^2}(y_{n+1}^j - 2y_n^j + y_{n-1}^j) + \tau f_n \tag{36}$$

2) При $n = 0$ и $n = N$:

$$y_0^{j+1} = \frac{4}{3}y_1^{j+1} - \frac{1}{3}y_2^{j+1} \tag{37}$$

$$y_N^{j+1} = \frac{4}{3}y_{N-1}^{j+1} - \frac{1}{3}y_{N-2}^{j+1} \tag{38}$$

3) Переходим на новый слой по времени, увеличивая j на единицу и повторяем действия 1) и 2).

5.2 Визуализация в MATLAB

Для визуального представлениям явной схемы смоделируем подзадачу похожую на нашу краевую задачу с пластиной. Отличие будем в том, что передавать будут не стержни, а некоторая функция в специальных точках для наглядности.

Возьмем пластину единичной длины и разобьем ее на некоторые элементарные участки. И на те участки, которые кратны 5, 6 и 7 будем передавать значения нашей функции, но без охлаждения для того, чтобы проследить теплопередачу от стрежня к пластине. Естественно, зададим начальную температуру 24^oC:

```
clear;
n = 50; % Number of space steps
L = 1; % Length of the wire
T =20; % Final time
maxk = 300000; % Number of time steps
dt = T/maxk;
dx = (L)/n;
b = dt/(dx*dx); % Stability parameter (b=<1)
for i = 1:n+1
    if(mod(i,5)==0)
        f(i) =50;
    else
        f(i)=0;
    end
end
for i = 1:n+1
    if(mod(i,6)==0)
        f(i) =50;
    else
        f(i)=0;
    end
end
for i = 1:n+1
    if(mod(i,7)==0)
        f(i) =50;
    else
```

```
        f(i)=0;
    end
end
for i = 1:n+1
x(i) =(i-1)*dx;
u(i,1) =24;
end
for k=1:maxk+1
time(k) = (k-1)*dt;
end

for k=1:maxk % Time Loop
for i=2:n % Space Loop
u(i,k+1) =u(i,k) + 0.001*b*(u(i-1,k)+u(i+1,k)-2.*u(i,k))+0.5*dt*f(i);
end
u(1,k+1)=4/3*u(2,k+1)-1/3*u(3,k+1);
u(n+1,k+1) =4/3*u(n,k+1)-1/3*u(n-1,k+1);
end
% Graphical representation of the temperature at different selected times
figure(1)
grid on
plot(x,u(:,1),'-',x,u(:,100),'-',x,u(:,300),'-',x,u(:,1000),'-',x,u(:,2000),'-')
title('Temperature analysis in time moments t=0.001, t=0.1, t=0.3, t=1, t=2')
xlabel('X')
ylabel('T')
figure(2)
grid on
mesh(x,time,u')
title('Visual representation of the temperature distribution on the plate')
xlabel('X')
ylabel('Temp')
```

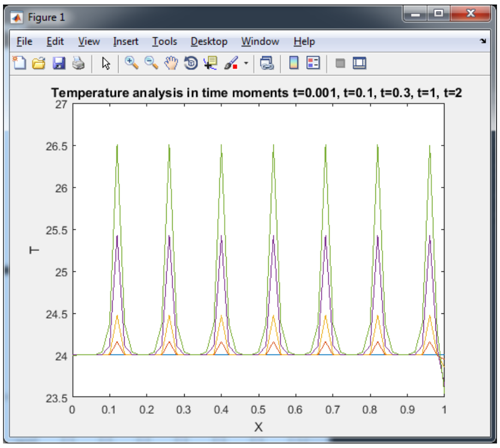

Рис. 6:

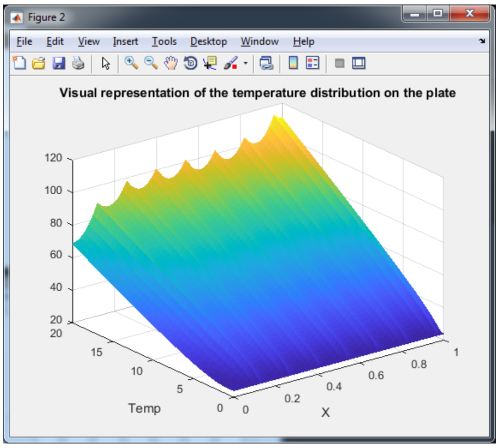

Рис. 7:

На графике видно, что значения смещаются к концу пластины. Это происходит из-за того, что мы запускаем итерационный процесс от 0 к l, при этом в соответствии с исходной задачей у нас граничные условия второго рода:$V'(0,t) = 0$, $V'(l,t) = 0$, что говорит о том, что теплопотери на концах нулевые, из-за чего и происходит смещение.

6 Моделирование двумерного случая средствами MATLAB

6.1 Вывод сеточной функции для двумерного случая

Рассмотрим решения уравнений параболического типа рассмотрим нестационарное уравнение теплопроводности

$$\rho(x,y)X(x,y)\frac{\partial T}{\partial t} - \Delta(k(x,y)\Delta T) = f(x,y) \tag{38}$$

где t – время; x, y – координаты; $T(x,y)$ – искомая функция распределения абсолютной температуры по координатам; $\rho(x,y)$ – плотность вещества; $C(x,y)$ – удельная теплоемкость вещества; $k(x,y)$ – коэффициент теплопроводности вещества; $f(x,y)$ – плотность мощности источников тепла на прямоугольной области с граничными условиями Дирихле или Неймана на границах $x = x_{min}$, $x = x_{max}$, $y = y_{min}$, $y = y_{max}$ и начальными условиями первого или второгорода н8а отрезке времени $[t_{min}, t_{max}]$. Зададим на отрезке $[x_{min}, x_{max}]$ равномерную координатную сетку с шагом Δx:

$$x = x_i | i = 1, 2, ..., n \tag{40}$$

на отрезке y_{min}, y_{max} – равномерную координатную сетку с шагом Δy:

$$y = y_j | j = 1, 2..., m \tag{41}$$

на отрезке t_{min}, t_{max} – равномерную координатную сетку с шагом Δt:

$$y = t_l | l = 1, 2..., s \tag{42}$$

Векторы, заданные выражениями (40) – (42), определяют на прямоугольной области равномерную пространственно-временную сетку:

$$G = x_i, y_j, t_l | i = 1, 2, ..., n, j = 1, 2, ..., m, t = 1, 2, ..., s \tag{43}$$

Граничные условия второго рода (Неймана) для рассматриваемой задачи могут быть представлены в виде:

$$\left.\frac{\partial T}{\partial x}\right|_{x_1,y,t} = g_1(y) \tag{44}$$

$$\left.\frac{\partial T}{\partial x}\right|_{x_n,y,t} = g_2(y) \tag{45}$$

$$\left.\frac{\partial T}{\partial y}\right|_{x,y_1,t} = g_3(y) \tag{46}$$

$$\left.\frac{\partial T}{\partial y}\right|_{x,y_n,t} = g_4(y) \tag{47}$$

Начальные условия первого рода для рассматриваемой задачи могут быть представлены в виде:

$$T(x, y, t_1) = g_t(x, y) \tag{48}$$

где t_1 - начальный момент времени; $g_t(x, y)$ некоторая непрерывная функция соответствующих координат.

6.2 Реализация в MATLAB

Рассмотриим реализацию двумерного уравнения теплопроводности на языке MATLAB.

```
% Heat equation r(x,y)C(x,y)dT/dt-d/dx(k(x,y)dT/dx)- d/dy(k(x,y)dT/dy)=f(x,y)
% on a rectangular domain with Dirichlet and / or Neumann boundary conditions

% Output parameters:
% x - row vector of the coordinate grid along the x axis with dimension 1 x n;
% y - the row vector of the coordinate grid along the y-axis with
%a dimension of 1 x m;
% t - grid line vector along the time axis with dimension 1 x s;
% T is a matrix of the resulting temperature values at the nodes of the
% coordinate grid with dimensions n x m x s.

function[x, y, t, T] = termo_2d(
    t0 = 0, % t0 is the initial time, c;
    ts = 6, % ts-end time, c;
    s = 6, % s is the number of grid points along the time axis t;
    x0 = 0, % x0 is the initial coordinate of the solution area on
    % the x-axis, m;
    xn = 0.05, % xn is the final coordinate of the solution area on
    % the x-axis, m;
    n = 10, % n is the number of grid points along the x-axis;
    y0 = 0, % y0 is the initial coordinate of the solution area on
    % the y-axis, m;
    ym = 0.08, % ym is the final coordinate of the solution area on
    % the y axis, m;
    m = 20, % m is the number of grid points along the y-axis;
    r = '2712', % r is the density function of the substance, given by
    % a string of characters enclosed in single quotes;
    c = '897', % c is a function of the heat capacity of the substance,
    % given by a string of characters enclosed in single quotes, J/(kg K);
    k = '232', % k is a function of the thermal conductivity of the substance,
    % given by a string of characters enclosed in single quotes, W/(m K);
    f = '0.01*1400', % f is the power density function of heat sources,
    % given by a string of characters enclosed in single quotes, W/m^3;
    vt = 1, % vt-parameter whose value determines the type of
    % initial condition (1-Dirichlet, 2-Neumann);
    gt1 = '24', % gt1-function on the right side of the initial condition,
    % specified by a string of characters enclosed in single quotes, K or K/c;
    v1 = 1, % v1 is a parameter whose value determines the type of boundary
    % condition on the first boundary of the domain x = x(1)
    % (1-Dirichlet GU, 2-Neumann GU);
    g1 = '0', % g1 is a function on the right side of the boundary condition
    % on the first boundary, defined by a string of characters enclosed
    % in single quotes, K or K/m;
    v2 = 2, % v2 is a parameter whose value determines the type of boundary
    % condition on the second boundary of the domain x = x(n)
    % (1-Dirichlet GU, 2-Neumann GU);
    g2 = '0', % g2 is a function on the right side of the boundary condition
    % on the second boundary, defined by a string of characters enclosed
```

```
    % in single quotes, K or K/m;
    v3 = 1, % v3 is a parameter whose value determines the type of boundary
    % condition on the third boundary of the domain y = y(1)
    % (1-Dirichlet GU, 2-Neumann GU);
    g3 = '0', % g3 is a function on the right side of the boundary condition
    % on the third boundary, defined by a string of characters enclosed in
    % single quotes, K or K/m;
    v4 = 1, % v4 is a parameter whose value determines the type of boundary
    % condition on the fourth boundary of the domain y = y(m)
    % (1-Dirichlet GU, 2-Neumann GU);
    g4 = '0' % g4 is a function on the right side of the boundary condition
    % on the fourth boundary, defined by a string of characters enclosed
    % in single quotes, K or K/m.
)
    % Setting a uniform grid
    x  = x0:(xn-x0)/(n-1):xn;
    dx = x(2) - x(1);

    y  = y0:(ym-y0)/(m-1):ym;
    dy = y(2) - y(1);

    t  = t0:(ts-t0)/(s-1):ts;
    dt = t(2) - t(1);

    % Calculation of the values of functions defined symbolically in
    % the nodes of the coordinate grid
    F  = inline(f, 'x', 'y');
    R  = inline(r, 'x', 'y');
    C  = inline(c, 'x', 'y');
    K  = inline(k, 'x', 'y');
    GT = inline(gt1, 'x', 'y');
    G1 = inline(g1, 'y');
    G2 = inline(g2, 'y');
    G3 = inline(g3, 'x');
    G4 = inline(g4, 'x');

    % Determination of System of linear equations dimension
    N = s * n * m;
    % Setting a System of linear equations coefficient matrix of
    % dimension N x N, all elements of which are equal to 0
    a = zeros(N, N);
    % Specifying a matrix-a string of free System of linear equations
    % terms of dimension 1 x N, all elements of which are equal to 0
    b = zeros(1, N);
    % Determination of the coefficients and free terms of the SLA
    % corresponding to the boundary conditions, and verification of
    % the correctness of the values of the parameters vt, v1, v2, v3, v4
    for (i = 1:n)
        for (j = 1:m)
            b(m*(i-1)+j) = GT(x(i), y(j));
            if (vt == 1)
```

```
            a(m*(i-1)+j, m*(i-1)+j) = 1;
        elseif (vt == 2)
            a(m*(i-1)+j, m*(i-1)+j) = -1/dt;
            a(m*(i-1)+j, n*m+m*(i-1)+j) = 1/dt;
        else
            error('Parameter vt have incorrect value');
        end
    end
end

for (l = 1:s)
    for (j = 1:m)
        b(n*m*(l-1)+j) = G1(y(j));
        if (v1 == 1)
            a(n*m*(l-1)+j, n*m*(l-1)+j) = 1;
        elseif (v1 == 2)
            a(n*m*(l-1)+j, n*m*(l-1)+j) = -1/dx;
            a(n*m*(l-1)+j, n*m*(l-1)+m+j) = 1/dx;
        else
            error('Parameter v1 have incorrect value');
        end

        b(n*m*(l-1)+m*(n-1)+j) = G2(y(j));
        if (v2 == 1)
            a(n*m*(l-1)+m*(n-1)+j, n*m*(l-1)+m*(n-1)+j) = 1;
        elseif (v2 == 2)
            a(n*m*(l-1)+m*(n-1)+j, n*m*(l-1)+m*(n-1)+j) = 1/dx;
            a(n*m*(l-1)+m*(n-1)+j, n*m*(l-1)+m*(n-2)+j) = -1/dx;
        else
            error('Parameter v2 have incorrect value');
        end
    end

    for (i = 2:n-1)
        b(n*m*(l-1)+m*(i-1)+1) = G3(x(i));
        if (v3 == 1)
            a(n*m*(l-1)+m*(i-1)+1, n*m*(l-1)+m*(i-1)+1) = 1;
        elseif (v3 == 2)
            a(n*m*(l-1)+m*(i-1)+1, n*m*(l-1)+m*(i-1)+1) = -1/dy;
            a(n*m*(l-1)+m*(i-1)+1, n*m*(l-1)+m*(i-1)+2) = 1/dy;
        else
            error('Parameter v3 have incorrect value');
        end

        b(n*m*(l-1)+m*(i-1)+m) = G4(x(i));
        if (v4 == 1)
            a(n*m*(l-1)+m*(i-1)+m, n*m*(l-1)+m*(i-1)+m) = 1;
        elseif (v4 == 2)
            a(n*m*(l-1)+m*(i-1)+m, n*m*(l-1)+m*(i-1)+m) = 1/dy;
            a(n*m*(l-1)+m*(i-1)+m, n*m*(l-1)+m*(i-1)+m-1) = -1/dy;
        else
```

```
            error('Parameter v4 have incorrect value');
        end
    end
end

% Determination of the coefficients and free terms of the System
% of linear equations corresponding to the interior points of the domain
for (l = 2:s)
    for (i = 2:n-1)
        for (j = 2:m-1)
            a(n*m*(l-1)+m*(i-1)+j, n*m*(l-1)+m*(i-1)+j) =...
            +R(x(i),y(j))*C(x(i),y(j))/dt + ...
            (K(x(i),y(j))+K(x(i-1),y(j)))/dx^2 + ...
            (K(x(i),y(j))+K(x(i),y(j-1)))/dy^2;
            a(n*m*(l-1)+m*(i-1)+j, n*m*(l-1)+m*i+j) =...
            -K(x(i),y(j))/dx^2;
            a(n*m*(l-1)+m*(i-1)+j, n*m*(l-1)+m*(i-2)+j) =...
            -K(x(i-1),y(j))/dx^2;
            a(n*m*(l-1)+m*(i-1)+j, n*m*(l-1)+m*(i-1)+j+1) =...
            -K(x(i),y(j))/dy^2;
            a(n*m*(l-1)+m*(i-1)+j, n*m*(l-1)+m*(i-1)+j-1) =...
            -K(x(i),y(j-1))/dy^2;
            a(n*m*(l-1)+m*(i-1)+j, n*m*(l-2)+m*(i-1)+j) =...
            -R(x(i),y(j))*C(x(i),y(j))/dt;
            b(n*m*(l-1)+m*(i-1)+j) = F(x(i),y(j));
        end
    end
end

% System of linear equations solution
u = b/a';
% Conversion of the row vector of the values of the desired function
% at the nodes of the coordinate grid into an n x m matrix, convenient
% for presenting the results in graphical form
for (l = 1:s)
    for (i = 1:n)
        for (j = 1:m)
            T(i,j,l) = u(n*m*(l-1)+m*(i-1)+j);
        end
    end
end

% Plotting the required function T (x, y, t)
for (l = 1:s)
    figure
    surf(y,x,T(:,:,l))
    xlabel('y, [U+FFFD]')
    ylabel('x, [U+FFFD]')
    zlabel('T, K')
    grid on
    colormap('cool')
```

```
        axis([min(y), max(y), min(x), max(x),...
        min(min(T(:,:,1))), max(max(T(:,:,1)))])
        pause(0.1)
        M(l) = getframe;
    end

    figure
    movie(M,10,3)

end
```

Результат работы скрипта:

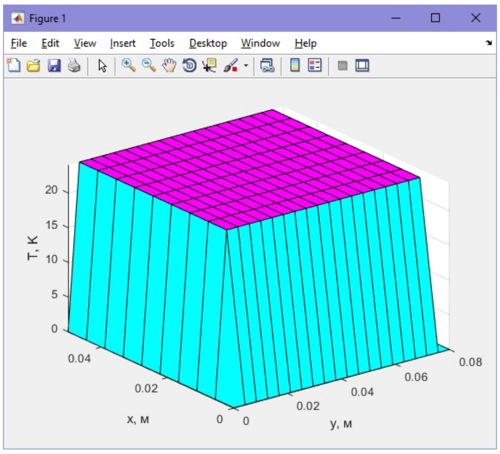

Рис. 8:

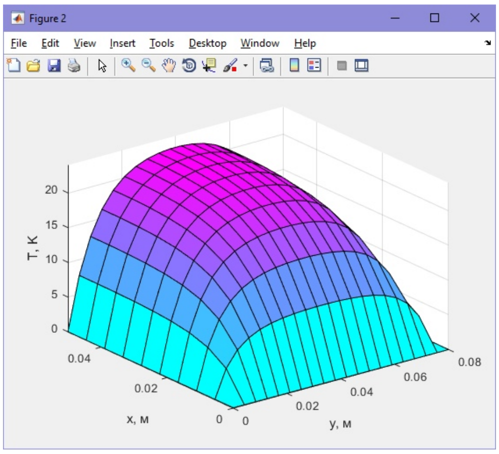

Рис. 9:

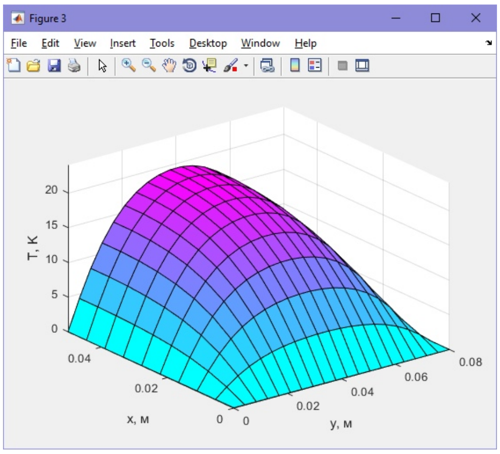

Рис. 10:

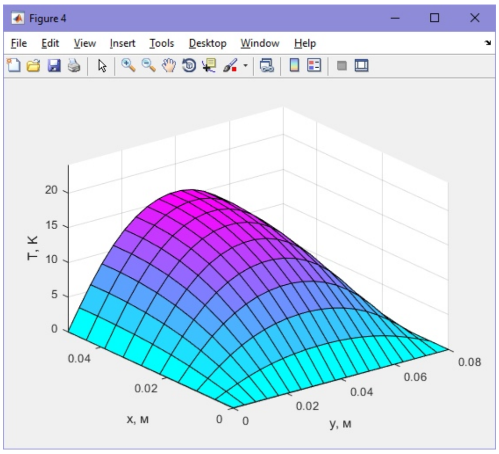

Рис. 11:

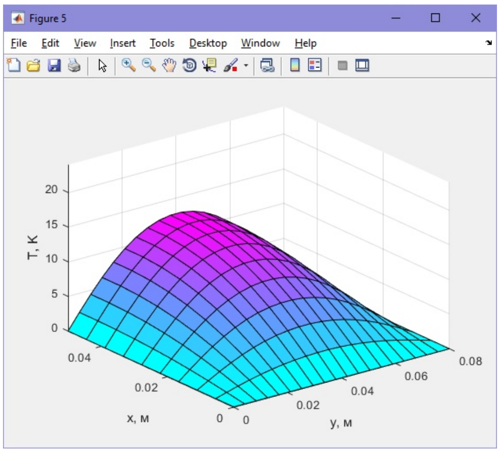

Figure 5
File Edit View Insert Tools Desktop Window Help
T, K
20
15
10
5
0
0.04
0.02
0
x, м
0
0.02
0.04
0.06
0.08
y, м

Рис. 12:

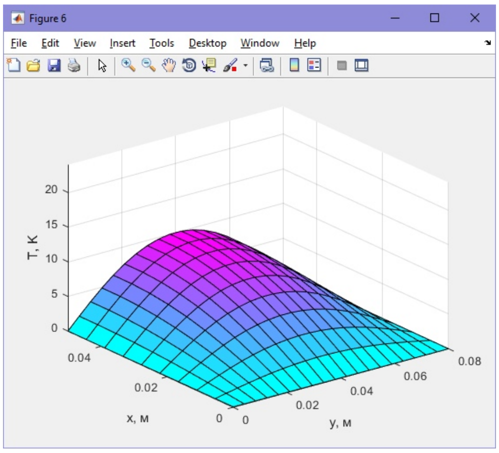

Figure 6
File Edit View Insert Tools Desktop Window Help
T, K
20
15
10
5
0
0.04
0.02
0
x, м
0
0.02
0.04
0.06
0.08
y, м

Рис. 13:

7 Решение в MATLAB и проверка

7.1 Реализация

Результируя все выводы полученные выше, реализуем код в MATLAB:

```
clear;

plane_num = 52;
tube_num = 8;

p = 100;
T_simulation = 10; % time of simulation
T = 5000; % num of dt
NP = 5; % num of dx in plane
NT = 5; % num of dx in tube

lp = 0.458; % length of plane
lt = 0.13; % length of tube

dt = T_simulation / T;
dx_p = lp / NP;
dx_t = lt / NT;
d_tube = 0.006;
dt / dx_p / dx_p
dt / dx_t / dx_t

for (j = 1:plane_num)
    for (n = 1:NP)
        U(j, n, 1) = 24;
    end
end


for (i = 1:tube_num)
    for (n = 1:NT)
        V(i, n, 1) = 24;
    end
end

f_tube(1) = 0;
f_plate(1) = 0;

width_plane = 0.00035;
diff_plane = 0.0002; % distance between plates


% find field where plane touch tube
j = 1;
for (i = 0:dx_t:lt)
    target_tube(j) = 0;
```

```
    current_plane = mod(i, width_plane + diff_plane);
    if (current_plane < width_plane)
        target_tube(j) = 1;
    end

    j = j + 1;
end

% find field where tube touch plane
x_center = [0.014, 0.042, 0.088, 0.114];
j = 1;
for (i = 0:dx_p:lp)
    target_plane(j) = 0;
    for (k = 1:length(x_center))
        if (i > x_center(k) - d_tube / 2 && i < x_center(k) + d_tube / 2)
            target_plane(j) = 1;
        end
    end

    j = j + 1;
end

k_diff = 2;
F_cooler = 1/100000;
rpm = 1400;

for (t = 1:T-1) % Time
    t
    V_mean = mean(mean(V, 1), 2);
    U_mean = mean(mean(U, 1), 2);

    %%% calculate tube
    for (i = 1:tube_num)
        for (n = 2:NT-1)
            if (t ~= 1)
                f_tube(t) = -k_diff*(V_mean(t) - U_mean(t-1))*target_tube(n);
            end
            V(i, n, t+1) = V(i, n, t) + 2*dt/dx_p^2*(V(i, n+1, t) - 2*V(i, n, t) + V(i, n-1, t)) +
                dt*f_tube(t);
        end
        V(i, 1, t+1) = 4/3*V(i, 2, t+1) - 1/3*V(i, 3, t+1);
        V(i, NT, t+1) = 4/3*V(i, NT-1, t+1) - 1/3*V(i, NT-2,t+1) + 2/3*dx_t*p/4;
    end


    V_mean = mean(mean(V, 1), 2);
    U_mean = mean(mean(U, 1), 2);

    %%% calculate plane
    for (j = 1:plane_num)
        for (n = 2:NP-1)
```

```
            if (t ~= 1)
                f_plate(t) = k_diff*(V_mean(t) - U_mean(t-1))*target_plane(n) + F_cooler*rpm;
            end
            U(j, n, t+1) = U(j, n, t) + 2*dt/dx_p^2*(U(j, n+1, t) - 2*U(j, n, t) + U(j, n-1, t)) +
                dt*f_plate(t);
        end
        U(j, 1, t+1) = 4/3*U(j, 2, t+1) - 1/3*U(j, 3, t+1);
        U(j, NP, t+1) = 4/3*U(j, NP-1, t+1) - 1/3*U(j, NP-2,t+1);
    end


end

U_mean = mean(mean(U, 1), 2);
V_mean = mean(mean(V, 1), 2);

figure(1);
plot(0:dt:T_simulation-dt, U_mean);
title('mean temperature of planes')
xlabel('t, c');
ylabel('T, C');
colormap winter;

figure(2);
plot(0:dt:T_simulation-dt, V_mean);
title('mean temperature of tubes')
xlabel('t, c');
ylabel('T, C');
colormap winter;

figure(3);
[X, Y] = meshgrid(0:dt:T_simulation-dt, 0:dx_p:lp-dx_p);
mesh(X, Y, squeeze(mean(U, 1)));
title('mean temperature of planes in each point')
xlabel('t, c');
ylabel('x, m');
zlabel('T, C');
colormap winter;



figure(4);
[X, Y] = meshgrid(0:dt:T_simulation-dt, 0:dx_t:lt-dx_t);
mesh(X, Y, squeeze(mean(V, 1)));
title('mean temperature of planes in each point')
xlabel('t, c');
ylabel('x, m');
zlabel('T, C');
colormap winter;
```

При $t = 1$ получили графики:

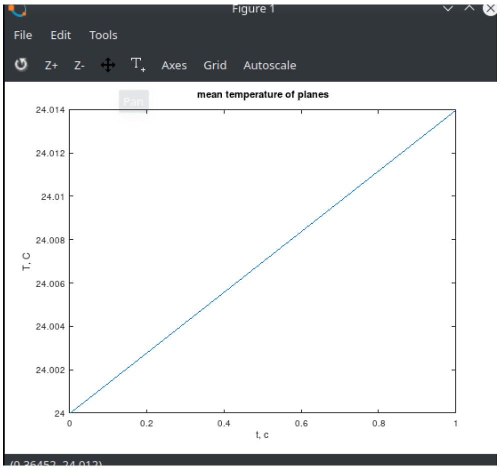

Рис. 14:

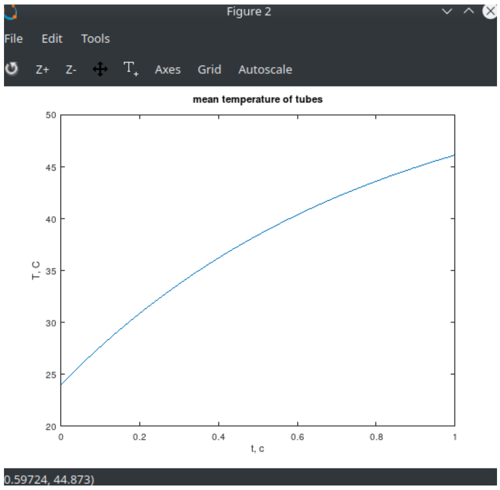

Рис. 15:

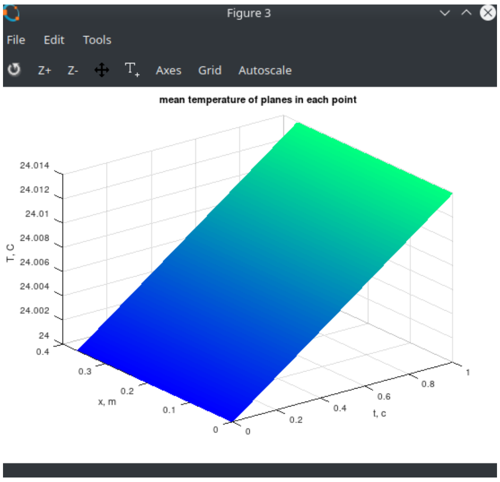

Рис. 16:

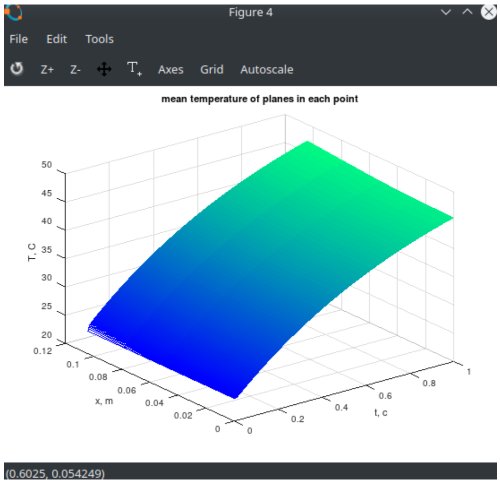

Рис. 17:

При $t = 10$ получили графики:

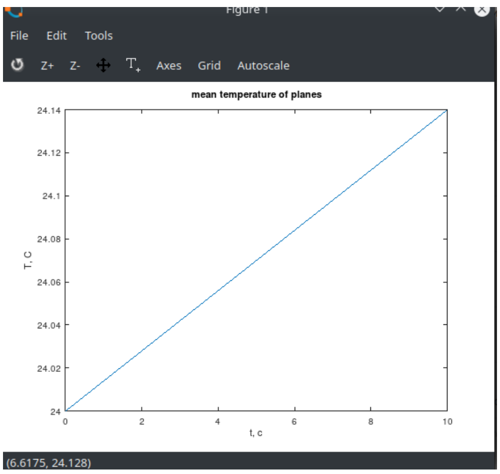

Рис. 18:

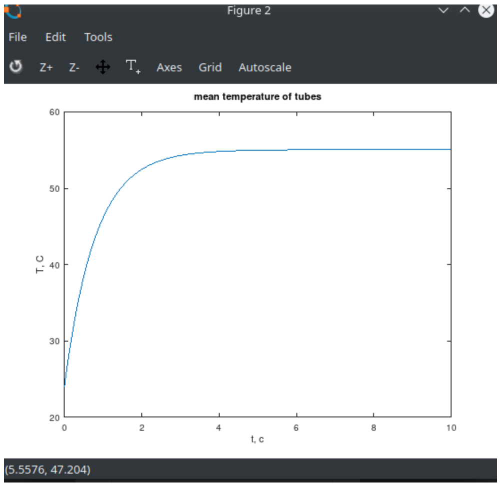

Рис. 19:

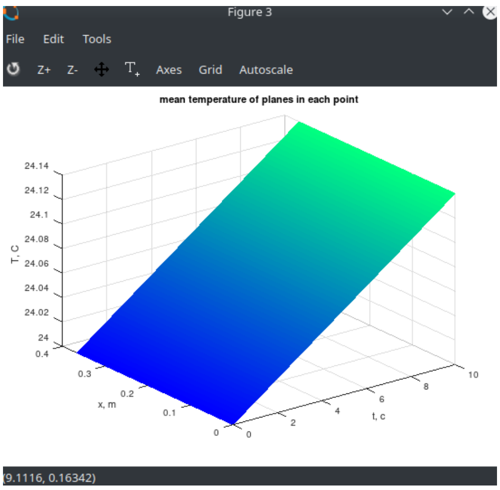

Рис. 20:

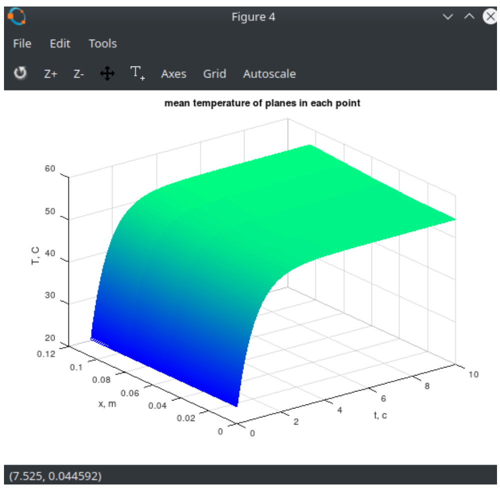

Рис. 21:

7.2 Тесты

Наша конфигурация:

1. Processor - Intel Core i7-3930K
2. Motherboard - Gigabyte X79-UP4
3. Memory - 8GB Corsair Vengeance LP 1600MHz CL9
4. Video Card - Sapphire Radeon HD 6850 1GB
5. Power Supply - NZXT Hale90 650W
6. Storage Drive - Kingston HyperX 240GB SATA III SSD
7. Case - Corsair Carbide 760T

8. Monitor ASUS ROG SWIFT PG278Q

9. The system used is as follows and tests are performed 4.4 GHz frequencies

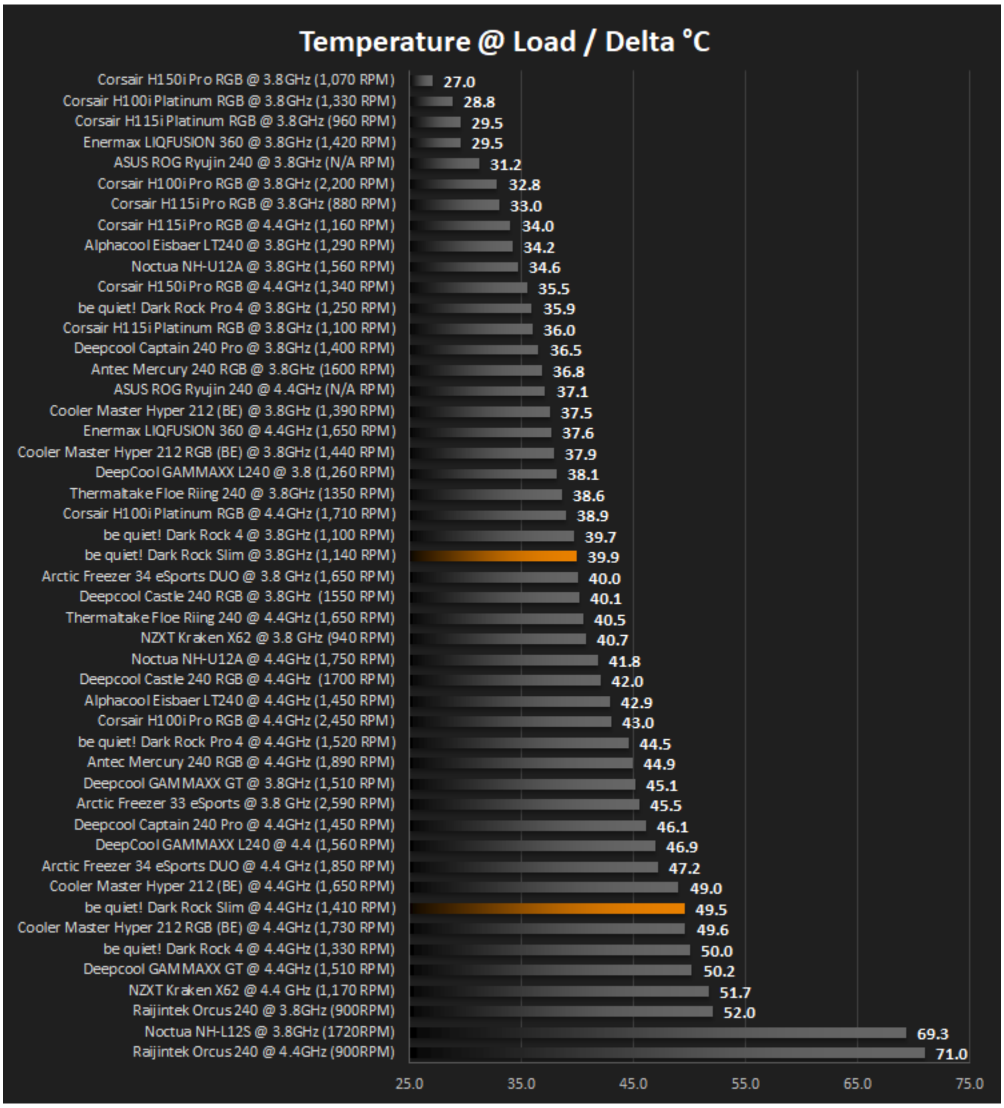

Рис. 22: Устоявшаяся температура

8 Вывод

Мы рассмотрели математическую модель системы охлаждения центрального процессора, разбив ее на составляющиее компоненты, а именно: трубки и пластины. Данная математическая модель имеет ряд недостатков, описанные в нашей работе, но несмотря на это, мы смогли получить достаточно точный результат, описывающий данную систему. Расхождение составило порядка $5 - 6^oC$, что является неплохим результатом, учитывая наши грубые предположения.

Список литературы

[1] В.В. Лесин учебное пособие "Уравнение математической физики"

[2] https://www.bequiet.com/en/cpucooler/479

[3] https://www.kitguru.net/components/silas-newman/be-quiet-dark-rock-slim-review/5/

[4] http://www.mmcs.sfedu.ru/jdownload/finish/16-kafedra-vychislitelnoj-matematiki-i-matematicheskoj-fiziki/1419-uravneniya-matematicheskoj-fiziki-zadachi-i-resheniya-s-v-revina-l-i-sazonov-o-a-tsyvenkova

[5] https://www.vortez.net/articles_pages/be_quiet_dark_rock_slim_review,7.html