Библиотека СОК

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение высшего профессионального образования
"ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Г.В. Кузнецов, М.А. Шеремет

РАЗНОСТНЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ТЕПЛОПРОВОДНОСТИ

Учебное пособие

Издательство
Томского политехнического университета
Томск 2007

ББК 22.36:22.19я73
УДК 536.2:519.876(075.8)
К 891

Кузнецов Г.В., Шеремет М.А.
К 891 Разностные методы решения задач теплопроводности: учебное пособие. / Г.В. Кузнецов, М.А. Шеремет. – Томск: Изд-во ТПУ, 2007. – 172 с.

В пособии приведены основные сведения по методам, алгоритмам и программам решения на ЭВМ основных задач теплопроводности. Приведены примеры решения линейных, нелинейных, одномерных, двумерных, нестационарных задач. Пособие предназначено для студентов всех специальностей и аспирантов теплоэнергетического факультета Томского политехнического университета.

УДК 536.2:519.876(075.8)

Рекомендовано к печати Редакционно-издательским советом
Томского политехнического университета

Рецензенты

Доктор физико-математических наук, профессор ТГУ
А.В. Герасимов

Доктор физико-математических наук, профессор ТГАСУ
В.Н. Барашков

© Томский политехнический университет, 2007
© Оформление. Издательство ТПУ, 2007

О Г Л А В Л Е Н И Е

Введение 4
1. Метод конечных разностей (МКР) 8
2. Линейные задачи теплопроводности 8
2.1. Одномерное уравнение теплопроводности 8
2.2. Конечно-разностная аппроксимация граничных условий второго и третьего рода 30
2.3. Примеры краевых задач с различными граничными условиями 35
2.4. Двухслойная пластина 47
2.5. Задача теплопроводности с внутренними источниками 57
2.6. Двумерная задача теплопроводности для однородного тела 64
2.7. Двумерная задача теплопроводности для неоднородного тела 73
3. Нелинейные задачи теплопроводности 89
3.1. Одномерное уравнение теплопроводности с зависящим от температуры коэффициентом теплопроводности 89
3.2. Одномерное уравнение теплопроводности с нелинейными граничными условиями (излучение на границе) 101
3.3. Одномерное уравнение теплопроводности с фазовым переходом на границе (испарение материала) 111
3.4. Одномерное уравнение теплопроводности с химической реакцией в материале (термическое разложение) 122
3.5. Одномерное уравнение теплопроводности с подвижной границей (промерзание влажного грунта) 128
3.6. Двумерное уравнение теплопроводности с излучением на границах 140
3.7. Двумерное уравнение теплопроводности с фазовым переходом на одной из границ 150
3.8. Квазитрехмерная модель (излучение и конвекция по третьей координате) 159
Список литературы 171

ВВЕДЕНИЕ

Изучение процессов теплообмена всегда играло видную роль в развитии техники и естествознания. В конце позапрошлого и в начале прошлого веков исследования в этой области стимулировались главным образом потребностями возникшей в то время теплоэнергетики. В годы после второй мировой войны развитие авиации, атомной энергетики, ракетно-космической техники выдвинуло новые постановки задач теплообмена и вместе с тем – новые, более жесткие требования к полноте и надежности прогностических возможностей теории.

За последние десятилетия сфера интенсивного исследования и применения явлений теплообмена чрезвычайно расширилась. Она включает как ведущие направления техники (химическая технология, металлургия, строительное дело, нефтеразработка, машиностроение, агротехника и т.д.), так и основные естественные науки (биология, геология, физика атмосферы и океана и другие). Теоретическое исследование процессов теплообмена в настоящее время в значительной степени базируется на их численном моделировании с использованием ЭВМ. Это стало возможным благодаря значительному прогрессу в развитии вычислительных методов решения задач для уравнений в частных производных и увеличению мощности современных вычислительных машин.

Нужно особо отметить еще следующие обстоятельства. Численное моделирование процессов теплообмена в настоящее время приобретает все более значительную роль в связи с тем, что для современной науки и техники необходим достоверный прогноз таких процессов, экспериментальное изучение которых в лабораторных или натурных условиях очень сложно и дорого, а в некоторых случаях просто невозможно. Численное моделирование процессов теплопереноса все успешнее входит в практику работы различных научно-исследовательских, проектно-конструкторских и производственных учреждений.

Как известно, существуют три механизма передачи тепла: теплопроводность, конвекция и излучение. В настоящем пособии мы сконцентрируем внимание на основном механизме переноса тепла – теплопроводности.

Теплопроводностью называется молекулярный перенос теплоты в сплошной среде. Этот процесс возникает при неравномерном распределении температур. В этом случае теплота передается за счет непосредственного соприкосновения частиц, имеющих различную

температуру, что приводит к обмену энергией между молекулами, атомами или свободными электронами.

Теплопроводность зависит от агрегатного состояния вещества, его состава, чистоты, температуры, давления и других характеристик. Так, в большинстве случаев теплопроводность вещества в жидком состоянии примерно в 10 раз больше, чем теплопроводность в газообразном состоянии. Для твердого тела она значительно выше, чем для жидкости около точки плавления (за исключением жидких висмута, олова, теллура).

На практике часто случается, что теплопроводность внутри тела и вблизи его границ различна. Это различие обусловлено как изменением условий протекания процессов теплопереноса, так и изменением структуры вещества (в результате термообработки, наклепа, старения, износа и т.д.).

Существенное влияние на теплопроводность могут оказывать внешние факторы, например облучение, изменение давления, магнитное поле.

В полупрозрачных средах теплопроводность сопровождается радиационным теплопереносом. Наблюдаемая эффективная теплопроводность таких сред есть сумма собственно теплопроводности и радиационного теплопереноса. Вклад радиационной составляющей комбинированного теплопереноса увеличивается с повышением температуры и становится существенным при температурах, составляющих несколько сотен градусов Цельсия.

Нестационарный перенос тепла теплопроводностью описывается следующим уравнением, записанным в декартовой системе координат:

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(\lambda \frac{\partial T}{\partial z}\right) + Q_w(x, y, z, t, T). \quad (1)$$

Это уравнение (уравнение Фурье – Кирхгофа) устанавливает связь между временным и пространственным изменением температуры в любой точке тела. Здесь ρ – плотность, c – удельная теплоемкость, λ – коэффициент теплопроводности, $Q_w(x, y, z, t, T)$ – мощность внутренних источников тепловыделения.

Уравнение (1) описывает множество вариантов развития процесса кондуктивного теплопереноса (теплопроводности). Чтобы из бесчисленного количества этих вариантов выбрать один и дать его полное математическое описание, к соотношению (1) необходимо добавить условия однозначности, которые содержат геометрические, физические, начальные и граничные условия.

Геометрические условия определяют форму и размеры тела, в котором протекает изучаемый процесс. Физические условия определяют теплофизические характеристики тела λ, ρ, *c*. Временные (начальные) условия содержат распределение температуры в теле в начальный момент времени:

$$t = 0:\ T = f(x, y, z) - \text{в общем виде.}$$

При равномерном распределении температуры в теле начальное условие упрощается: $t = 0:\ T = T_0 = \text{const}$. Граничные условия определяют особенности протекания процесса на поверхности тела и могут быть заданы несколькими способами.

- Граничные условия первого рода – задается распределение температуры на поверхности (или границе) тела для каждого момента времени:

$$T = T_w(x, y, z, t),$$

где T_w – температура на поверхности тела. Во многих практически значимых вариантах $T_w = \text{const}$.

- Граничные условия второго рода – задается значение теплового потока для каждой точки поверхности (или границы) тела в любой момент времени:

$$-\lambda\left(\frac{\partial T}{\partial \vec{n}}\right)_w = q_w(x, y, z, t),$$

где $\vec{n}$ – нормаль к поверхности тела. Наиболее часто используется условие $q_w = \text{const}$.

Такой вариант теплообмена имеет место, например, при нагревании различных изделий в высокотемпературных печах.

- Граничные условия третьего рода – задается взаимосвязь между потоком тепла за счет теплопроводности от твердой стенки и тепловым потоком из окружающей среды за счет температурного напора (закон Ньютона – Рихмана):

$$-\lambda\left(\frac{\partial T}{\partial \vec{n}}\right)_w = \alpha\left(T_w - T^{\text{e}}\right),$$

где α – коэффициент теплообмена. Это наиболее широко применяемое условие в задачах теплотехники.

- Граничные условия четвертого рода – для определения теплового взаимодействия между элементами, имеющими различные теплофизические характеристики, задают условия равенства температур и тепловых потоков по обе стороны от границы раздела:

$$\begin{cases} -\lambda_1\left(\dfrac{\partial T_1}{\partial \vec{n}}\right)_\Gamma = -\lambda_2\left(\dfrac{\partial T_2}{\partial \vec{n}}\right)_\Gamma; \\ T_1(x_\Gamma, y_\Gamma, z_\Gamma, t) = T_2(x_\Gamma, y_\Gamma, z_\Gamma, t). \end{cases}$$

где $x_\Gamma, y_\Gamma, z_\Gamma$ – координаты границы раздела сред; T_1, T_2 – температуры соприкасающихся сред. Это условие применяется, например, при решении задач теплопроводности для многослойных пластин.

Дифференциальное уравнение (1) вместе с условиями однозначности дает полную математическую формулировку краевой задачи теплопроводности.

При решении конкретных краевых задач нестационарной теплопроводности можно, применяя методы математического моделирования, добиться существенного упрощения общей математической постановки. Так, если для рассматриваемого процесса:

$$\frac{\partial^2 T}{\partial x^2} >> \frac{\partial^2 T}{\partial y^2} \text{ и } \frac{\partial^2 T}{\partial x^2} >> \frac{\partial^2 T}{\partial z^2},$$

то можно вместо уравнения (1) ограничиться одномерным нестационарным уравнением кондуктивного теплопереноса

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x}\right) + Q_w(x, t, T), \qquad (2)$$

которое вместе с условиями однозначности дает более простую математическую формулировку краевой задачи. Есть много практически значимых случаев, когда решение уравнения (2) достаточно для полного описания рассматриваемого процесса.

В практике теплотехнических расчетов часто возникают одномерные задачи с цилиндрической или сферической симметрией. Например, цилиндрическая симметрия имеется в задачах об остывании длинного цилиндра или при анализе теплового состояния в трубчатых каналах.

Естественной системой координат в таких задачах является, соответственно, цилиндрическая (r, φ) или сферическая (r, θ, φ). Вследствие одномерности все величины не будут зависеть от углов θ, φ. Тогда параболическое уравнение (2) с переменными коэффициентами в соответствующих координатах примет вид:

$$\rho c \frac{\partial T}{\partial t} = \frac{1}{r^\nu}\frac{\partial}{\partial r}\left(\lambda r^\nu \frac{\partial T}{\partial r}\right) + Q_w(r, t, T),$$

где r – радиальная координата, ν – показатель симметрии, равный 0, 1, 2 соответственно для плоского, цилиндрического и сферического случаев.

1. МЕТОД КОНЕЧНЫХ РАЗНОСТЕЙ (МКР)

Сформулированное уравнение (1) с соответствующими краевыми условиями (начальными и граничными) будем решать численно, т.е. воспользуемся возможностями ЭВМ. Численным решением называется решение, полученное в виде таблицы чисел.

При решении дифференциального уравнения в частных производных наиболее часто используется метод конечных разностей (МКР) [1]. Идея МКР решения краевых задач весьма проста и видна уже из самого названия: вместо производных в дифференциальном уравнении используются их конечноразностные аппроксимации. При построении дискретных аппроксимаций краевых дифференциальных задач нужно стремиться увязать две, возможно, противоречивые цели: хорошее качество аппроксимации и эффективное устойчивое решение получающихся при этом алгебраических систем.

При использовании МКР для задач теплопроводности твердое тело представляют в виде совокупности узлов. Аппроксимируя (заменяя) частные производные дифференциального уравнения (1) конечными разностями получают систему линейных алгебраических уравнений для определения температуры, как локальной характеристики в каждом узле сетки. Полученная система является незамкнутой, для ее замыканию используют разностное представление граничных условий. В результате получают замкнутую систему линейных алгебраических уравнений, которую решают численными методами с помощью ЭВМ.

2. ЛИНЕЙНЫЕ ЗАДАЧИ ТЕПЛОПРОВОДНОСТИ
2.1. ОДНОМЕРНОЕ УРАВНЕНИЕ ТЕПЛОПРОВОДНОСТИ

В качестве примера применения метода конечных разностей рассмотрим краевую задачу на основе одномерного уравнения теплопроводности. Анализируется теплопередача через плоскую бесконечную пластину или изолированный стержень (рис. 1). На одной границе пластины поддерживается постоянная температура $T_л$, на другой границе – температура $T_п$. Начальная температура равна T_0, источники тепловыделения внутри пластины отсутствуют.

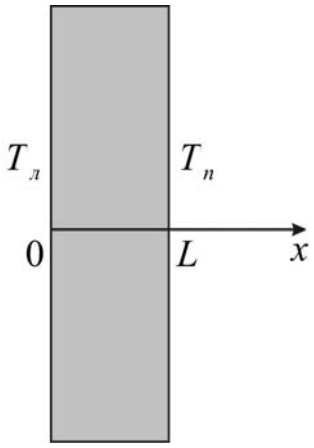


Рис. 1. Геометрия задачи

При заданных условиях температура будет изменяться только в направлениях, перпендикулярных границе пластины. Если ось O*x* направить, как показано на рис. 1, то температура в направлении O*y* и O*z* может считаться постоянной. Также предположим, что теплофизические характеристики не зависят от температуры. В связи с этим дифференциальное уравнение (1) преобразуется к виду:

$$\rho c \frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2}, \quad 0 < x < L\,. \tag{3}$$

Начальные и граничные условия запишутся следующим образом:

$$\begin{aligned} &t = 0:\ T = T_0,\ 0 \le x \le L; \\ &x = 0:\ T = T_л, t > 0; \\ &x = L:\ T = T_п, t > 0. \end{aligned} \tag{4}$$

Для того чтобы дать полное математическое описание рассматриваемой задачи, необходимо еще задать физические условия однозначности. Если пластина изготовлена из стали, то λ = 46 Вт/(м·ºС), ρ = 7800 кг/м^3, c = 460 Дж/(кг·ºС).

Эту задачу в полной математической постановке будем решать методом конечных разностей на равномерной сетке. Для этого разобьем пластину по толщине на N–1 равных промежутков, т.е. построим конечно-разностную сетку (рис. 2):

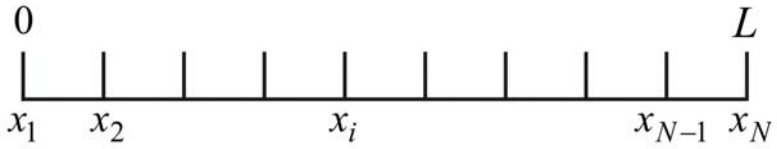


Рис. 2. Конечно-разностная сетка:
$x_2, x_3, \ldots, x_{N-1}$ – координаты внутренних узлов;
x_1, x_N – координаты граничных узлов

Определим значение температуры в *i*-ом узле в момент времени $t = t_n = n \cdot \tau$ как $T(x_i, t_n) = T_i^n$. Здесь τ – шаг интегрирования по временной координат, *n* – номер шага по времени.

Далее заменим дифференциальные операторы в (3) на их конечно-разностные аналоги. Будем пользоваться неявной схемой.

$$\frac{\partial T}{\partial t} = \frac{T_i^{n+1} - T_i^n}{\tau},$$

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1}^{n+1} - 2 \cdot T_i^{n+1} + T_{i-1}^{n+1}}{h^2}.$$

В результате аппроксимации частных производных соответствующими конечными разностями получаем следующую систему линейных алгебраических уравнений (СЛАУ):

$$\rho \cdot c \cdot \frac{T_i^{n+1} - T_i^n}{\tau} = \lambda \cdot \left(\frac{T_{i+1}^{n+1} - 2 \cdot T_i^{n+1} + T_{i-1}^{n+1}}{h^2} \right), \quad i = 2, \ldots, N-1, \quad n \geq 0. \quad (5)$$

Выбранную схему аппроксимации частных производных можно графически представить следующим образом:

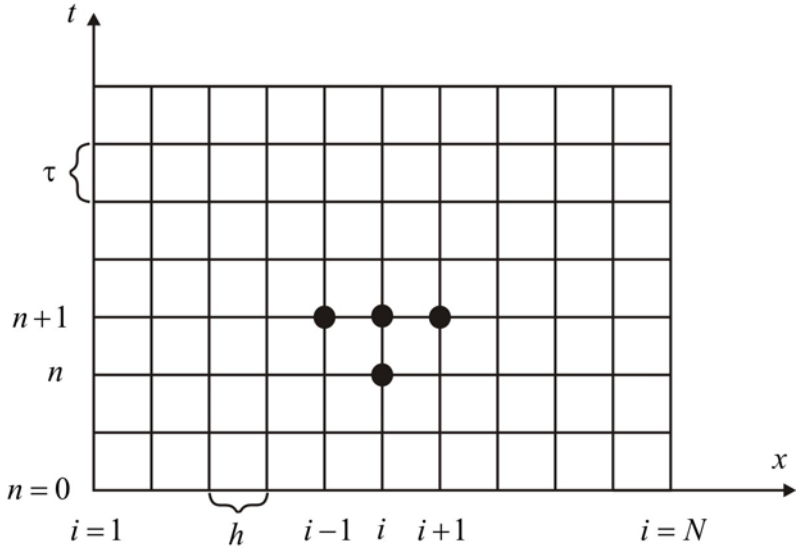


Рис. 3. Шаблон неявной четырехточечной разностной схемы

Рис. 3 наглядно демонстрирует, что используется четырехточечная разностная схема – три точки берутся на новом временном слое и одна со старого временного слоя.

Сформулированный выше способ аппроксимации производных называется неявным потому, что поле температуры на новом временном слое представлено неявно, т.е. для его определения необходимо решать систему уравнений (5).

Полученную систему можно свести к наиболее общему виду:

$$A_i \cdot T_{i+1}^{n+1} - B_i \cdot T_i^{n+1} + C_i \cdot T_{i-1}^{n+1} = F_i, \qquad (6)$$

где

$$A_i = C_i = \frac{\lambda}{h^2}, \quad B_i = \frac{2 \cdot \lambda}{h^2} + \frac{\rho c}{\tau}, \quad F_i = -\frac{\rho c}{\tau} T_i^n.$$

Такие уравнения называют трехточечными разностными уравнениями второго порядка. Система (6) имеет трехдиагональную структуру. В связи с тем, что рассматривается нестационарная задача, систему (6) необходимо решать на каждом шаге по времени.

Предположим, что существуют такие наборы чисел α_i и β_i $\left(i = \overline{1, N-1}\right)$, при которых

$$T_i^{n+1} = \alpha_i \cdot T_{i+1}^{n+1} + \beta_i, \qquad (7)$$

т.е. трехточечное уравнение второго порядка (6) преобразуется в двухточечное уравнение первого порядка (7). Уменьшим в связи (7) индекс на единицу и полученное выражение $T_{i-1}^{n+1} = \alpha_{i-1} \cdot T_i^{n+1} + \beta_{i-1}$ подставим в данное уравнение (6):

$$A_i \cdot T_{i+1}^{n+1} - B_i \cdot T_i^{n+1} + C_i \cdot \alpha_{i-1} \cdot T_i^{n+1} + C_i \cdot \beta_{i-1} = F_i,$$

откуда получаем

$$T_i^{n+1} = \frac{A_i}{B_i - C_i \cdot \alpha_{i-1}} T_{i+1}^{n+1} + \frac{C_i \cdot \beta_{i-1} - F_i}{B_i - C_i \cdot \alpha_{i-1}}.$$

Последнее равенство имеет вид (7) и будет точно с ним совпадать, если при всех $i = 2,3,\ldots,N-1$ выполняются соотношения

$$\alpha_i = \frac{A_i}{B_i - C_i \cdot \alpha_{i-1}}, \qquad \beta_i = \frac{C_i \cdot \beta_{i-1} - F_i}{B_i - C_i \cdot \alpha_{i-1}}. \qquad (8)$$

Для определения α_i и β_i по (8) необходимо знать α_1 и β_1, которые находятся из левого граничного условия.

Далее по формулам (7) последовательно находятся $T_{N-1}^{n+1}, T_{N-2}^{n+1}, \ldots, T_2^{n+1}$, при условии, что T_N^{n+1} найдено из правого граничного условия.

Таким образом, решение уравнений вида (6) описываемым способом, называемым методом прогонки, сводится к вычислениям по трем формулам: нахождение так называемых прогоночных коэффициентов α_i, β_i по формулам (8) при $i = \overline{2, N-1}$ (прямая прогонка) и затем получение неизвестных T_i^{n+1} по формуле (7) при $i = N-1, N-2, \ldots, 2$ (обратная прогонка).

Для успешного применения метода прогонки нужно, чтобы в процессе вычислений не возникло ситуаций с делением на нуль, а при больших размерностях систем не должно быть быстрого роста погрешностей округлений.

Будем называть прогонку корректной, если знаменатели прогоночных коэффициентов (8) не обращаются в нуль, и устойчивой, если $|\alpha_i| < 1$ при всех $i = \overline{1, N-1}$.

В [1] доказана теорема, представляющая достаточные условия корректности и устойчивости прогонки уравнений (6):

$$|B_i| > |A_i| + |C_i| \quad \forall i = \overline{2, N-1} \text{ и } |\alpha_1| < 1 \Rightarrow |\alpha_i| < 1, \qquad (9)$$

которые во многих приложениях метода выполняются автоматически.

Возвращаясь к системе (5), определим прогоночные коэффициенты и воссоздадим полный алгоритм решения полученной системы.

Поскольку при $x = 0 \;\; T = T_л$,

то

$$T_1^{n+1} = \alpha_1 \cdot T_2^{n+1} + \beta_1 = T_л,$$
$$\alpha_1 = 0, \; \beta_1 = T_л$$

а при

$$x = L \qquad T = T_n,$$
$$T_N^{n+1} = T_n.$$

Прогоночные коэффициенты вычисляются по формулам (8).

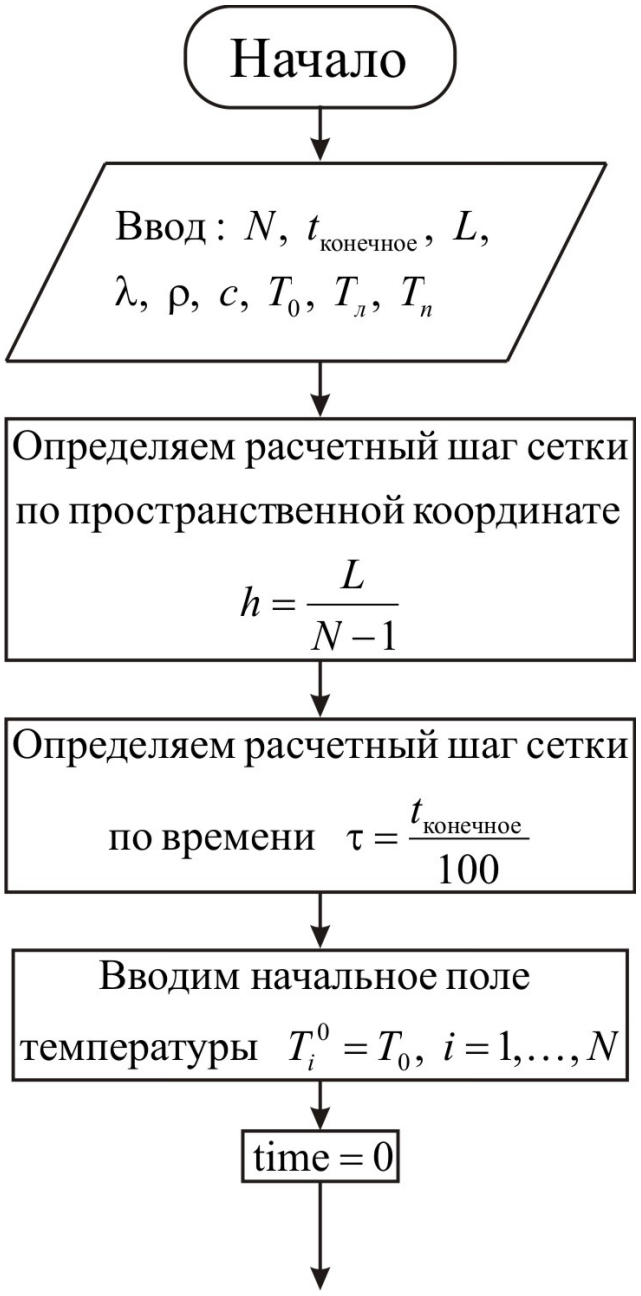
Таким образом, разностные соотношения, аппроксимирующие дифференциальную задачу (3), (4), имеют следующий вид:

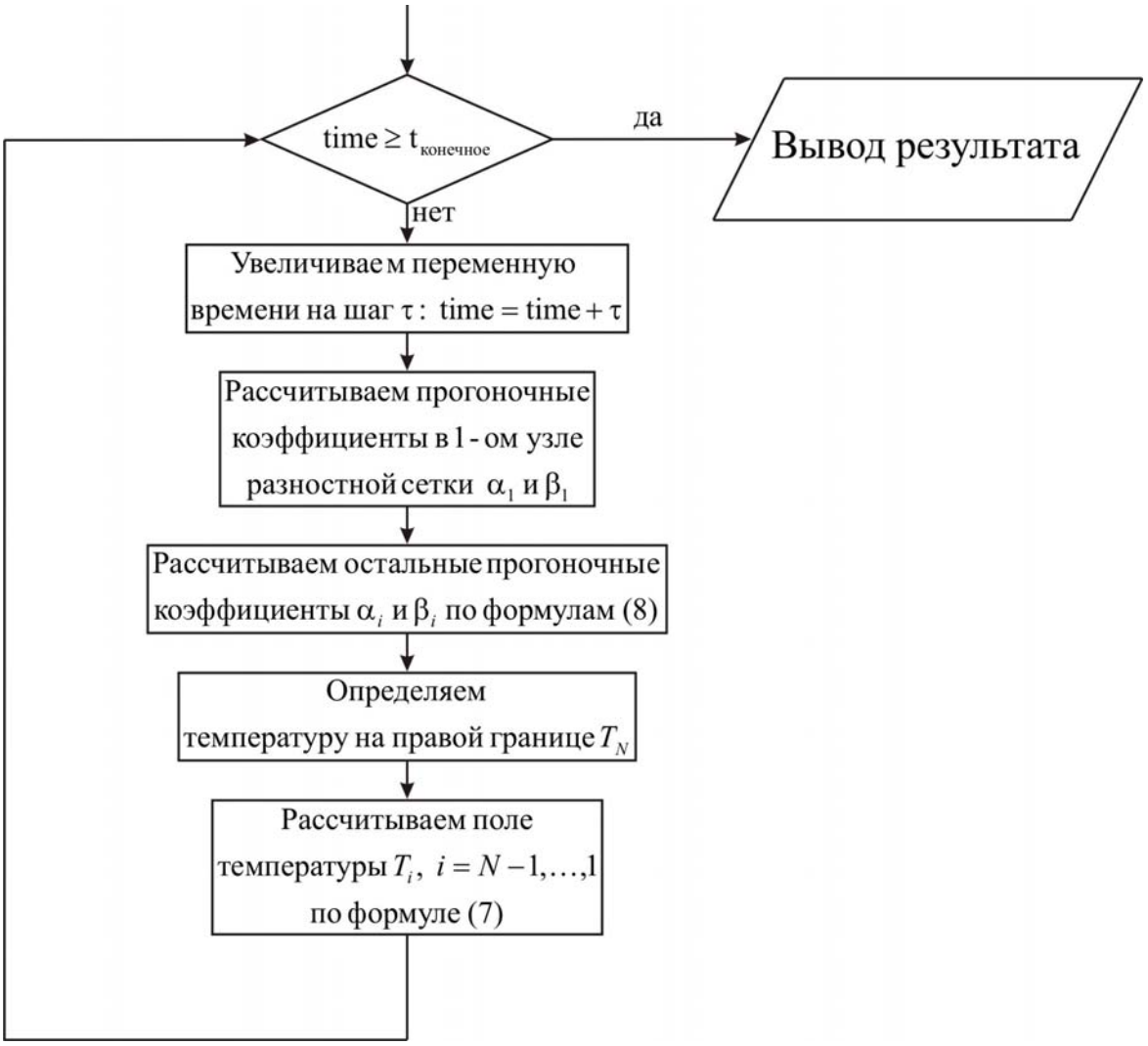
$$\rho \cdot c \cdot \frac{T_i^{n+1} - T_i^n}{\tau} = \lambda \cdot \left(\frac{T_{i+1}^{n+1} - 2 \cdot T_i^{n+1} + T_{i-1}^{n+1}}{h^2} \right), \quad i = 2, \ldots, N-1, \quad n \geq 0 \qquad (10)$$

$$T_i^0 = T_0, \quad i = 2, \ldots, N-1;$$
$$T_1^n = T_л, \quad n > 0; \qquad (11)$$
$$T_N^n = T_n, \quad n > 0.$$

Аппроксимация дифференциальной задачи (3), (4) конечно-разностной (10), (11) выполнена с первым порядком точности по времени t и вторым по пространственной координате h. При этом неявная разностная схема является абсолютно устойчивой, т.е. можно проводить интегрирование краевой задачи (3), (4) с любым разностным шагом по времени. Шаг по времени выбирается таким образом, чтобы весь интервал времени разбивался хотя бы на 10 шагов (желательно больше).

Блок-схема к рассматриваемой задаче имеет вид:




time ≥ t конечное
да
Вывод результата
нет
Увеличиваем переменную времени на шаг τ : time = time + τ
Рассчитываем прогоночные коэффициенты в 1 - ом узле разностной сетки α1 и β1
Рассчитываем остальные прогоночные коэффициенты αi и βi по формулам (8)
Определяем температуру на правой границе TN
Рассчитываем поле температуры Ti, i = N −1,…,1 по формуле (7)

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
type
    vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N                        : integer;
 T, alfa, beta                  : vector;
 ai, bi, ci, fi                 : real;
 lamda, ro, c, h, tau           : real;
 Tl, T0, Tr, L, t_end, time     : real;
 f, g                           : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите толщину пластины, L');
 Readln(L);
 Writeln('Введите коэффициент теплопроводности материала пластины, lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
 Writeln('Введите температуру на границе x=0, Tl');
 Readln(Tl);
 Writeln('Введите температуру на границе x=L, Tr');
 Readln(Tr);
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
```

```
{определяем поле температуры в начальный момент времени}
 for i:= 1 to N do
  T[i]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
   time:=time+tau;
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия}
   alfa[1]:=0.0;
   beta[1]:=Tl;
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
   for i:= 2 to N-1 do
    begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
     ai:=lamda/sqr(h);
     bi:=2.0*lamda/sqr(h)+ro*c/tau;
     ci:=lamda/sqr(h);
     fi:=-ro*c*T[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
     alfa[i]:=ai/(bi-ci*alfa[i-1]);
     beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
    end;
{определяем значение температуры на правой границе}
   T[N]:=Tr;
{используя соотношение (7) определяем неизвестное поле
температуры}
   for i:= N-1 downto 1 do
    T[i]:=alfa[i]*T[i+1]+beta[i];
  end; {цикл с предусловием окончен}
{выводим результаты в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Толщина пластины L = ',L:6:4);
 Writeln(f,'Число узлов по координате N = ',N);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
```

```
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура  T0 = ',T0:6:4);
 Writeln(f,'Температура на границе x = 0, Tl = ',Tl:6:4);
 Writeln(f,'Температура на границе x = L, Tr = ',Tr:6:4);
 Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):6:3,'   ',T[i]:8:5);
 close(g);
end.
```

Результаты расчетов по приведенной программе при L = 0,1 м, λ = 46 Вт/(м·ºС), ρ = 7800 кг/м^3, c = 460 Дж/(кг·ºС), T_0=20 °С, $T_л$ = 300 °С, $T_п$ = 100 °С через 60 секунд процесса нагрева приведены на рис. 4.

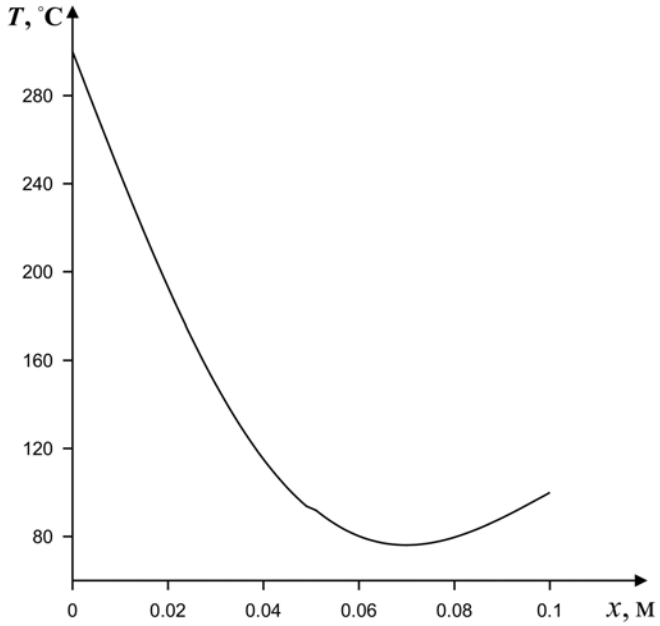


Рис. 4. Распределение температуры по толщине пластины в момент времени t = 60 с

Как отмечалось выше, рассмотренная расчетная схема является неявной, т.е. для определения поля температуры приходится решать систему линейных алгебраических уравнений. Но помимо

предложенной схемы существует также и явная схема. В такой схеме явно определяется поле температуры и не нужно решать систему уравнений для определения прогоночных коэффициентов α_i и β_i. Рассмотрим ту же задачу, но уже с использованием явной схемы.

Отличие явной схемы от неявной заключается в аппроксимации диффузионного слагаемого, а именно, во временном слое на котором рассматривается неизвестное поле температуры:

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1}^n - 2 \cdot T_i^n + T_{i-1}^n}{h^2}.$$

Таким образом, в результате аппроксимации частных производных соответствующими конечными разностями получаем следующее соотношения для определения поля температуры:

$$\rho \cdot c \cdot \frac{T_i^{n+1} - T_i^n}{\tau} = \lambda \cdot \left(\frac{T_{i+1}^n - 2 \cdot T_i^n + T_{i-1}^n}{h^2} \right), \quad i = 2, \ldots, N-1, \quad n \geq 0.$$

Графически явную разностную схему можно представить следующим образом:

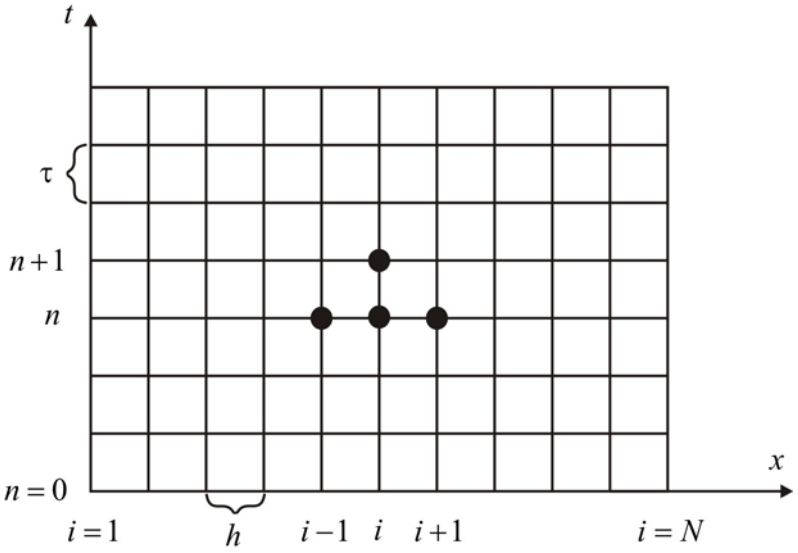


Рис. 5. Шаблон явной четырехточечной разностной схемы

Из шаблона (рис. 5) видно, что для определения неизвестного поля температуры никакой системы уравнений для α_i и β_i решать не требуется.

$$T_i^{n+1} = T_i^n + \frac{\lambda \cdot \tau}{\rho \cdot c} \cdot \left(\frac{T_{i+1}^n - 2 \cdot T_i^n + T_{i-1}^n}{h^2} \right), \quad i = 2, \ldots, N-1, \quad n \geq 0 \quad (12)$$

и аналогичные разностные аналоги краевых условий:

$$T_i^0 = T_0, \quad i = 2,\ldots,N-1;$$
$$T_1^n = T_л, \quad n > 0; \qquad (13)$$
$$T_N^n = T_n, \quad n > 0.$$

Таким образом, мы получили простую систему линейных алгебраических уравнений для нахождения распределения температуры в пластине в различные моменты времени. Аппроксимация дифференциальной задачи (3), (4) конечно-разностной (12), (13) выполнена также с первым порядком по времени t и вторым по пространственной координате h. Но чтобы решение конечно-разностной задачи (12), (13) сходилось к решению дифференциальной задачи, достаточно выполнение следующего условия (условие устойчивости разностной схемы):

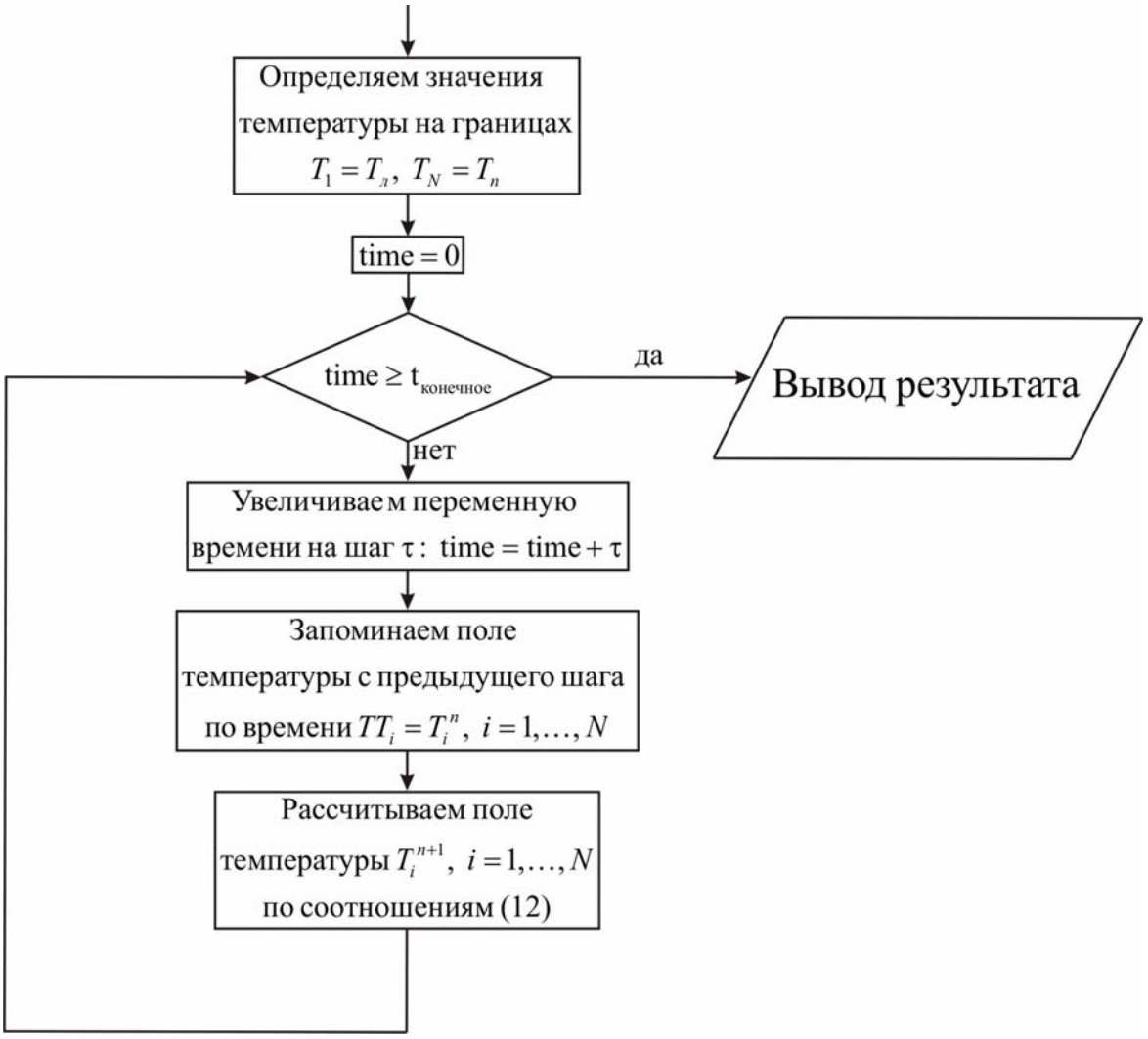
$$\tau < \frac{\rho \cdot c \cdot h^2}{2 \cdot \lambda}.$$

Из этого условия определяется шаг интегрирования по временной координате.

Таким образом, явная разностная схема является условно устойчивой и требует специальных мероприятий по оценке возможности ее использования.

Блок-схема к рассматриваемой задаче (явная схема) имеет вид:




Определяем значения температуры на границах $T_1 = T_л,\ T_N = T_п$
time = 0
time ≥ t конечное
да
Вывод результата
нет
Увеличиваем переменную времени на шаг τ : time = time + τ
Запоминаем поле температуры с предыдущего шага по времени $TT_i = T_i^n,\ i = 1,\ldots,N$
Рассчитываем поле температуры $T_i^{n+1},\ i = 1,\ldots,N$ по соотношениям (12)

Ниже приведен листинг программы (явная схема) для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
type
    vector=array[0..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N                        : integer;
 T, TT                          : vector;
 a, lamda, ro, c, h, tau        : real;
 Tl, T0, Tr, L, t_end, time     : real;
 f, g                           : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество узлов по пространственной координате, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите толщину пластины, L');
 Readln(L);
 Writeln('Введите коэффициент теплопроводности материала пластины, lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
 Writeln('Введите температуру на границе x=0, Tl');
 Readln(Tl);
 Writeln('Введите температуру на границе x=L, Tr');
 Readln(Tr);
{определяем коэффициент температуропроводности}
 a:=lamda/(ro*c);
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
```

```
{определяем расчетный шаг сетки по времени, руководствуясь
условием устойчивости}
 tau:=0.25*sqr(h)/a;
{определяем поле температуры в начальный момент времени}
 for i:= 2 to N-1 do
  T[i]:=T0;
{определяем значения температуры на границе}
 T[1]:=Tl;
 T[N]:=Tr;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг}
   time:=time+tau;
{запоминаем поле температуры на предыдущем слое по времени}
   for i:= 1 to N do
    TT[i]:=T[i];
{определяем неизвестное поле температуры по соотношениям (12)}
   for i:= 2 to N-1 do
    T[i]:=TT[i]+a*tau/sqr(h)*(TT[i+1]-2.0*TT[i]+TT[i-1]);
  end;
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Толщина пластины L = ',L:6:4);
 Writeln(f,'Число узлов по пространственной координате  N = ',N);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура  T0 = ',T0:6:4);
 Writeln(f,'Температура на границе x = 0, Tl = ',Tl:6:4);
 Writeln(f,'Температура на границе x = L, Tr = ',Tr:6:4);
 Writeln(f,'Результат получен с шагом по пространственной координате
x, h = ',h:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
```

```
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):6:3,'   ',T[i]:8:5);
 close(g);
end.
```

Результаты расчетов по приведенной программе при L = 0,1 м, λ = 46 Вт/(м·ºС), ρ = 7800 кг/м^3, c = 460 Дж/(кг·ºС), T_0=20 °С, T_1 = 300 °С, T_2 = 100 °С через 60 секунд процесса нагрева приведены на рис. 6.

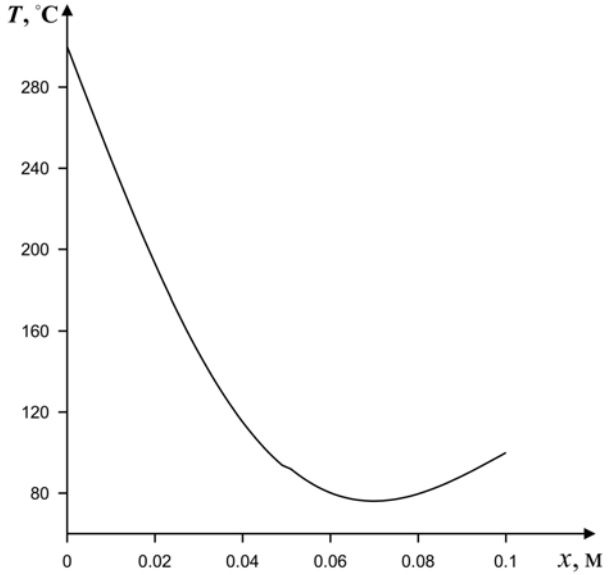
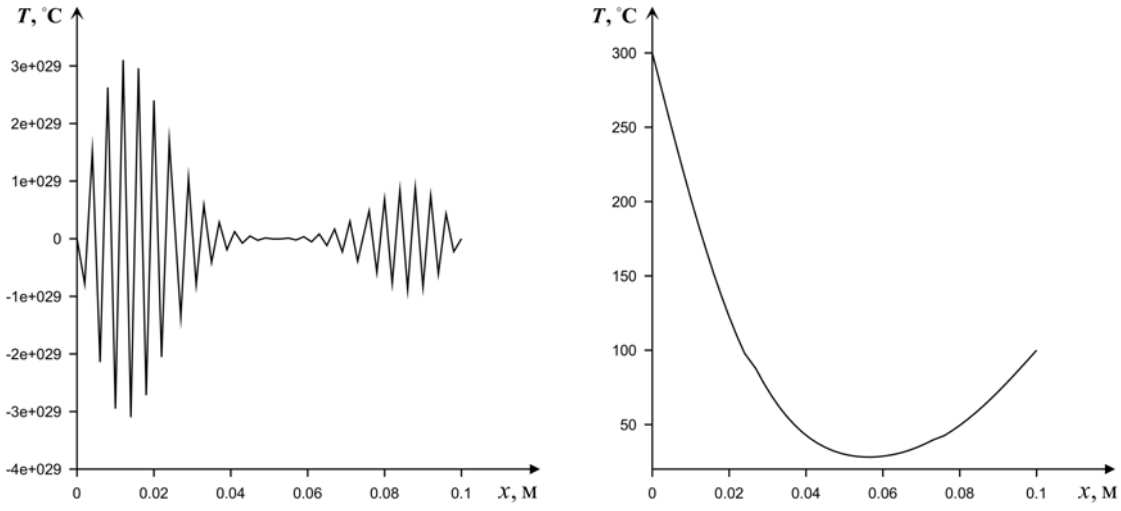


Рис. 6. Распределение температуры по толщине пластины в момент времени t = 60 с (с использованием явной разностной схемы)

Анализируя рисунки 4 и 6, видим, что полученные поля температур идентичны. Покажем, в чем же заключается недостаток явных схем. При использовании данной схемы $\tau < \frac{\rho \cdot c \cdot h^2}{2 \cdot \lambda}$. Рассмотрим случай $\tau = \frac{\rho \cdot c \cdot h^2}{\lambda}$ как для явной, так и для неявной схемы. С помощью программа, построенной на основе явной схемы, рассчитан только случай N = 50, t = 20. При N > 50 и t > 20 происходит ошибка (в случае среды *Turbo Pascal* – переполнение операции с плавающей точкой).



а *б*

Рис. 7. Распределение температуры по толщине пластины в момент времени t = 20 с: а – явная схема, б – неявная схема

Рис. 7,*а* наглядно демонстрирует появление осцилляций при использовании явной схемы, что характеризует её неустойчивость. Неявная же схема напротив дает корректный результат.

Поэтому дальнейшее изложение материала будем проводить с использованием неявной разностной схемы.

Рассмотрим одномерное уравнение теплопроводности в цилиндрических координатах.

Определим температурное поле в бесконечном цилиндре радиуса $R = 0.1\,\text{м}$ через 5, 10, 30 и 60 секунд. Материал цилиндра – кирпич (λ = 0,7 Вт/(м·ºС), ρ = 1500 кг/м^3, c = 750 Дж/(кг·ºС)). Начальная температура $T_0 = 20^0\,\text{C}$. На границе $r = R$ поддерживается постоянная температура $T_h = 50\ ^0\text{C}$.

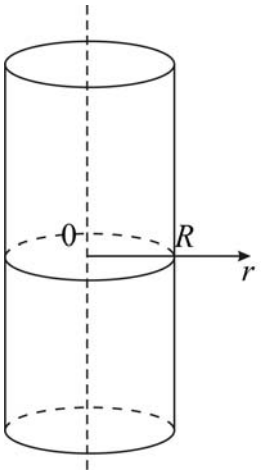


Рис. 8. Геометрия задачи

Математическая постановка задачи будет иметь вид:

$$\rho c \frac{\partial T}{\partial t} = \frac{\lambda}{r} \frac{\partial}{\partial r}\left(r \frac{\partial T}{\partial r} \right), \quad 0 < r < R. \tag{14}$$

Начальные и граничные условия запишутся следующим образом:

$$t = 0: \; T = T_0, 0 \le r \le R;$$

$$r = 0: \; \frac{\partial T}{\partial r} = 0, \; t > 0; \tag{15}$$

$$r = R: \; T = T_h, \; t > 0.$$

Для решения сформулированной краевой задачи применим метод конечных разностей на основе неявной четырехточечной схемы. Сначала введем равномерную пространственную сетку:

$$r_i = (i-1) \cdot h, \quad i = 1,\ldots,N;$$

$$r_1 = 0,\ldots,r_N = R;$$

$$h = \frac{R}{N-1}.$$

Аналогично вводится временная сетка:

$$t_n = n \cdot \tau, \quad n = 0,1,\ldots,M;$$

$$t_0 = 0,\ldots,t_M = t_{\text{конечное}};$$

$$\tau > 0.$$

Заменим дифференциальные операторы в уравнении (14) на их конечно-разностные аналоги:

$$\frac{\partial T}{\partial t} = \frac{T_i^{n+1} - T_i^n}{\tau},$$

$$\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right)=\frac{1}{h^2}\left[r_{i+\frac{1}{2}}\cdot T_{i+1}^{n+1}-\left(r_{i-\frac{1}{2}}+r_{i+\frac{1}{2}}\right)\cdot T_i^{n+1}+r_{i-\frac{1}{2}}\cdot T_{i-1}^{n+1}\right],$$

где $r_{i-\frac{1}{2}}=\frac{r_{i-1}+r_i}{2}$, $r_{i+\frac{1}{2}}=\frac{r_i+r_{i+1}}{2}$.

Таким образом, в результате аппроксимации частных производных соответствующими конечными разностями получаем следующую систему линейных алгебраических уравнений:

$$\rho\cdot c\cdot\frac{T_i^{n+1}-T_i^n}{\tau}=\frac{\lambda}{r_i\cdot h^2}\left[r_{i+\frac{1}{2}}\cdot T_{i+1}^{n+1}-\left(r_{i-\frac{1}{2}}+r_{i+\frac{1}{2}}\right)\cdot T_i^{n+1}+r_{i-\frac{1}{2}}\cdot T_{i-1}^{n+1}\right],$$

$$i=2,\ldots,N-1,\quad n=0,1,\ldots,M.$$

Полученную систему можно свести к наиболее общему виду:

$$A_i\cdot T_{i+1}^{n+1}-B_i\cdot T_i^{n+1}+C_i\cdot T_{i-1}^{n+1}=F_i,$$

где

$$A_i=\frac{\lambda}{h^2}\cdot\frac{r_{i+\frac{1}{2}}}{r_i},\quad B_i=\frac{\lambda}{h^2}\cdot\frac{r_{i-\frac{1}{2}}+r_{i+\frac{1}{2}}}{r_i}+\frac{\rho c}{\tau},\quad C_i=\frac{\lambda}{h^2}\cdot\frac{r_{i-\frac{1}{2}}}{r_i},\quad F_i=-\frac{\rho c}{\tau}T_i^n.$$

Прогоночные коэффициенты находятся по формулам (8). Далее неизвестное поле температуры определяется по выражению (7).

Воспользуемся левым граничным условием для определения начальных прогоночных коэффициентов α_1 и β_1 из соотношения $T_1=\alpha_1\cdot T_2+\beta_1$. На левой границе стоит условие симметрии:

$$\left.\frac{\partial T}{\partial r}\right|_{r=0}=0;$$

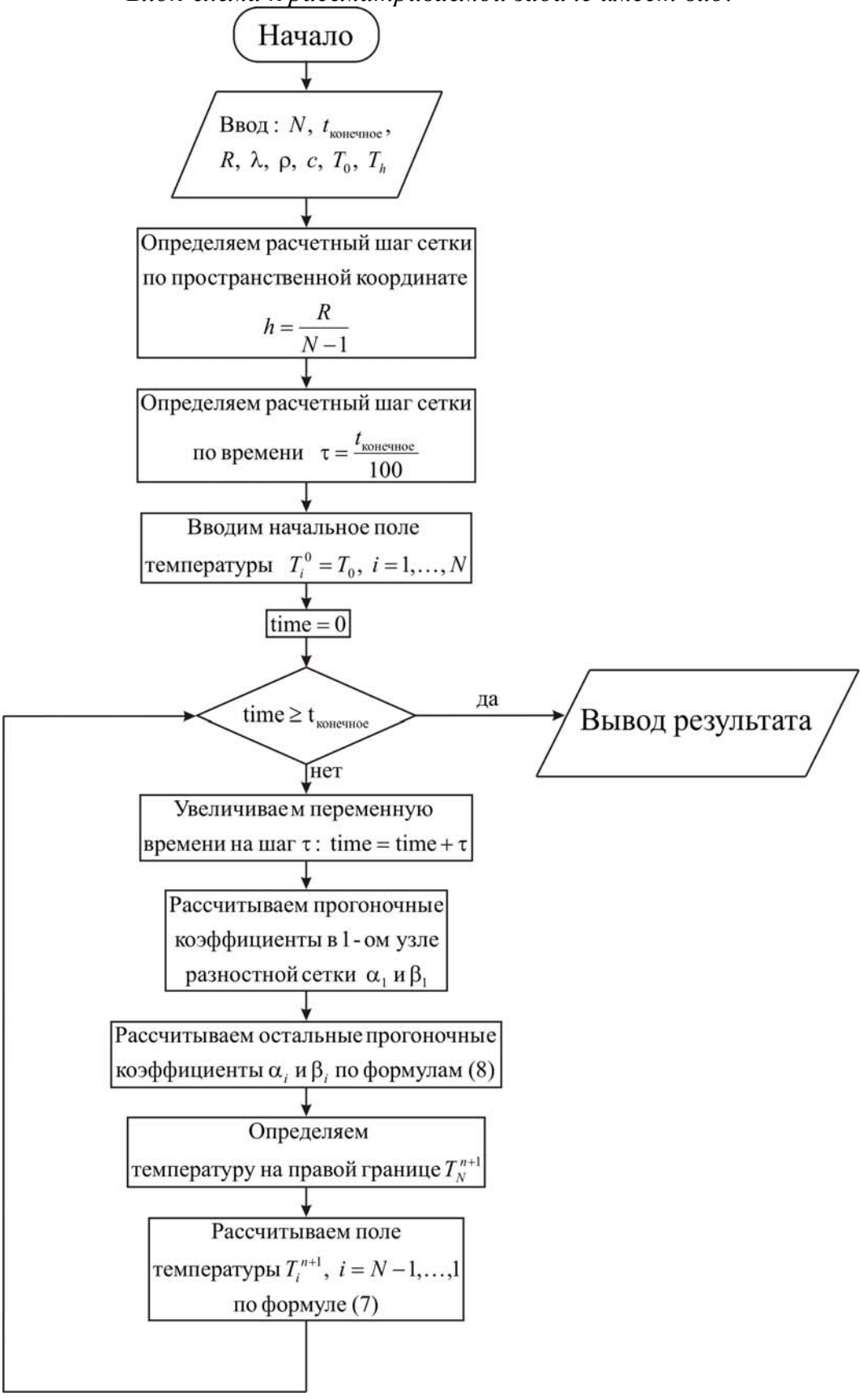
$$\frac{T_2-T_1}{h}=0;$$

$$T_1=T_2;$$

$$\begin{cases}\alpha_1=1;\\ \beta_1=0.\end{cases}$$

А на правой границе температура известна $T\big|_{r=R}=T_h$.

Блок-схема к рассматриваемой задаче имеет вид:



Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
type
    vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N                    : integer;
 T, alfa, beta              : vector;
 ai, bi, ci, fi             : real;
 lamda, ro, c, h, tau       : real;
 T0, Th, R, t_end, time     : real;
 f, g                       : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите радиус цилиндра, R');
 Readln(R);
 Writeln('Введите коэффициент теплопроводности материала пластины, lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
 Writeln('Введите температуру на границе r=R, Th');
 Readln(Th);
{определяем расчетный шаг сетки по пространственной координате}
 h:=R/(N-1);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to N do
```

```
  T[i]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия}
   alfa[1]:=1.0;
   beta[1]:=0.0;
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
   for i:= 2 to N-1 do
    begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
     ai:=0.5*lamda*(2*i-1)/(sqr(h)*(i-1));
     ci:=0.5*lamda*(2*i-3)/(sqr(h)*(i-1));
     bi:=ai+ci+ro*c/tau;
     fi:=-ro*c*T[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
     alfa[i]:=ai/(bi-ci*alfa[i-1]);
     beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
    end;
{определяем значение температуры на правой границе}
   T[N]:=Th;
{используя соотношение (7) определяем неизвестное поле
температуры}
   for i:= N-1 downto 1 do
    T[i]:=alfa[i]*T[i+1]+beta[i];
  end;
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Радиус цилиндра R = ',R:6:4);
 Writeln(f,'Число узлов по координате N = ',N);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
```

```
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура  T0 = ',T0:6:4);
 Writeln(f,'Температура на границе r = R, Th = ',Th:6:4);
 Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):6:3,'   ',T[i]:8:5);
 close(g);
end.
```

В результате были получены следующие распределения температуры

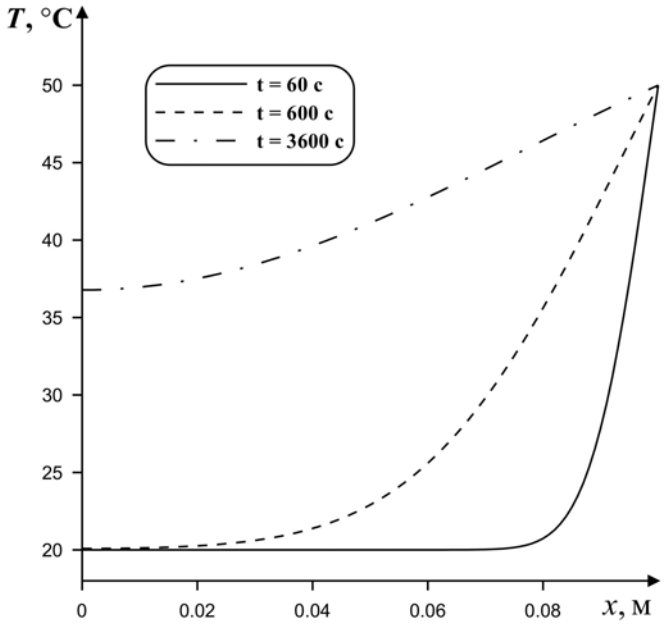


Рис. 9. Распределения температуры по радиусу цилиндра в различные моменты времени

2.2. КОНЕЧНО-РАЗНОСТНАЯ АППРОКСИМАЦИЯ ГРАНИЧНЫХ УСЛОВИЙ ВТОРОГО И ТРЕТЬЕГО РОДА

Если для задачи, изложенной в пункте 2.1, в качестве граничных условий на внешних поверхностях использовать *граничные условия*

второго рода (заданы тепловые потоки q_1 и q_2, рис. 10), тогда математическая формулировка граничных условий будет иметь следующий вид:

$$\begin{aligned} x = 0 :\ & -\lambda \frac{\partial T}{\partial x} = q_1,\ t > 0; \\ x = L :\ & -\lambda \frac{\partial T}{\partial x} = q_2,\ t > 0. \end{aligned} \qquad (16)$$

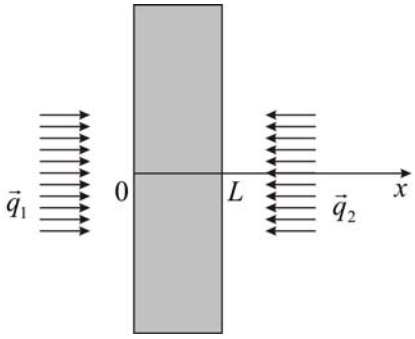


Рис. 10. Геометрия задачи (граничные условия второго рода)

Проанализируем соотношения (16):

$q_1 > 0 \Rightarrow$ на границе $x = 0$ происходит нагрев материала;

$q_1 < 0 \Rightarrow$ на границе $x = 0$ происходит охлаждение материала;

$q_2 > 0 \Rightarrow$ на границе $x = L$ происходит охлаждение материала;

$q_2 < 0 \Rightarrow$ на границе $x = L$ происходит нагрев материала.

Проведем дискретизацию граничных условий II рода с погрешностью $O(h)$. Погрешность аппроксимации вида $O(h)$ означает, что различия между точными значениями и полученными (приближенными) будут одного порядка с шагом по пространству h. Поскольку обычно $h < 1$, то погрешность будет тем меньше, чем выше порядок аппроксимации. Например, погрешность аппроксимации вида $O(h^2)$ дает результаты более точные, т.к. $h^2 < h < 1$.

Поскольку мы будем использовать неявную разностную схему, то левое граничное условие необходимо для определения первых прогоночных коэффициентов α_1 и β_1 из соотношения $T_1 = \alpha_1 \cdot T_2 + \beta_1$.

Итак,

$$-\lambda \left. \frac{\partial T}{\partial x} \right|_{x=0} = q_1;$$

$$-\lambda\frac{T_2-T_1}{h}=q_1;$$

$$T_1=T_2+\frac{h\cdot q_1}{\lambda}.$$

Отсюда следует, что

$$\begin{cases}\alpha_1=1;\\ \beta_1=\dfrac{h\cdot q_1}{\lambda}.\end{cases} \tag{17}$$

А правое граничное условие используют для определения температуры T_N.

Итак,

$$-\lambda\left.\frac{\partial T}{\partial x}\right|_{x=L}=q_2;$$

$$-\lambda\frac{T_N-T_{N-1}}{h}=q_2;$$

$$T_N=T_{N-1}-\frac{h\cdot q_2}{\lambda};$$

т.к. $T_{N-1}=\alpha_{N-1}\cdot T_N+\beta_{N-1}$, то

$$T_N=\alpha_{N-1}\cdot T_N+\beta_{N-1}-\frac{h\cdot q_2}{\lambda};$$

$$T_N=\frac{\lambda\cdot\beta_{N-1}-h\cdot q_2}{\lambda\cdot(1-\alpha_{N-1})}. \tag{18}$$

Проведем дискретизацию граничных условий II рода с погрешностью $O(h^2)$. Предположим, что на границе выполняется уравнение теплопроводности (3):

$$\rho\cdot c\cdot\frac{\partial T}{\partial t}=\lambda\cdot\frac{\partial^2 T}{\partial x^2}\ \text{ или }\ \frac{\partial T}{\partial t}=a\cdot\frac{\partial^2 T}{\partial x^2}, \tag{19}$$

где a – коэффициент температуропроводности материала.

Разложим функцию $T(x)$ в ряд Тейлора в окрестности точки $x=0$ до членов второго порядка относительно h: $T_2^{n+1}=T_1^{n+1}+h\cdot\left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1}+\frac{h^2}{2}\cdot\left.\frac{\partial^2 T}{\partial x^2}\right|_{x=0}^{n+1}$. Используя соотношение (19) получим:

$$T_2^{n+1}=T_1^{n+1}+h\cdot\left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1}+\frac{h^2}{2\cdot a}\cdot\left.\frac{\partial T}{\partial t}\right|_{x=0}^{n+1};$$

$$\left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1} = \frac{T_2^{n+1} - T_1^{n+1}}{h} - \frac{h}{2\cdot a}\cdot\left.\frac{\partial T}{\partial t}\right|_{x=0}^{n+1} = \frac{T_2^{n+1} - T_1^{n+1}}{h} - \frac{h}{2\cdot a}\cdot\frac{T_1^{n+1} - T_1^{n}}{\tau} = -\frac{q_1}{\lambda}.$$

Тогда $\frac{T_2^{n+1} - T_1^{n+1}}{h} - \frac{h}{2\cdot a\cdot\tau}\cdot T_1^{n+1} + \frac{h}{2\cdot a\cdot\tau}\cdot T_1^{n} = -\frac{q_1}{\lambda}.$

Или $T_1^{n+1} = \frac{2\cdot a\cdot\tau}{h^2 + 2\cdot a\cdot\tau}\cdot T_2^{n+1} + \frac{h^2}{h^2 + 2\cdot a\cdot\tau}\cdot T_1^{n} + \frac{2\cdot a\cdot\tau\cdot h\cdot q_1}{\lambda\cdot\left(h^2 + 2\cdot a\cdot\tau\right)}.$

Таким образом,

$$\begin{cases} \alpha_1 = \dfrac{2\cdot a\cdot\tau}{h^2 + 2\cdot a\cdot\tau}; \\ \beta_1 = \dfrac{h^2}{h^2 + 2\cdot a\cdot\tau}\cdot T_1^{n} + \dfrac{2\cdot a\cdot\tau\cdot h\cdot q_1}{\lambda\cdot\left(h^2 + 2\cdot a\cdot\tau\right)}. \end{cases} \qquad (20)$$

Определим T_N используя правое граничное условие.

$$T_{N-1}^{n+1} = T_N^{n+1} - h\cdot\left.\frac{\partial T}{\partial x}\right|_{x=L}^{n+1} + \frac{h^2}{2}\cdot\left.\frac{\partial^2 T}{\partial x^2}\right|_{x=L}^{n+1} = T_N^{n+1} - h\cdot\left.\frac{\partial T}{\partial x}\right|_{x=L}^{n+1} + \frac{h^2}{2\cdot a}\cdot\left.\frac{\partial T}{\partial t}\right|_{x=L}^{n+1};$$

$$-\frac{q_2}{\lambda} = \left.\frac{\partial T}{\partial x}\right|_{x=L}^{n+1} = \frac{T_N^{n+1} - T_{N-1}^{n+1}}{h} + \frac{h}{2\cdot a}\cdot\frac{T_N^{n+1} - T_N^{n}}{\tau}.$$

Таким образом,

$2\cdot a\cdot\tau\cdot\lambda\cdot T_N^{n+1} - 2\cdot a\cdot\tau\cdot\lambda\cdot T_{N-1}^{n+1} + h^2\cdot\lambda\cdot T_N^{n+1} - h^2\cdot\lambda\cdot T_N^{n} + 2\cdot a\cdot\tau\cdot h\cdot q_2 = 0,$

т.к. $T_{N-1} = \alpha_{N-1}\cdot T_N + \beta_{N-1}$, то

$$T_N^{n+1} = \frac{2\cdot a\cdot\tau\cdot\lambda\cdot\beta_{N-1} - 2\cdot a\cdot\tau\cdot h\cdot q_2 + h^2\cdot\lambda\cdot T_N^{n}}{\lambda\cdot h^2 + 2\cdot a\cdot\tau\cdot\lambda\cdot\left(1 - \alpha_{N-1}\right)}. \qquad (21)$$

Граничные условия третьего рода для задачи из пункта 2.1 (если температуры окружающей среды T^{e1} и T^{e2} и коэффициенты теплоотдачи κ_1 и κ_2) можно сформулировать следующим образом:

$$x = 0 : -\lambda\frac{\partial T}{\partial x} = \kappa_1\left(T^{e1} - T\right),\ t > 0,\ \kappa_1 > 0;$$

$$x = L : \lambda\frac{\partial T}{\partial x} = \kappa_2\left(T^{e2} - T\right),\ t > 0,\ \kappa_2 > 0.$$

Проведем дискретизацию граничных условий III рода с погрешностью $O(h)$.

Определим первые прогоночные коэффициенты α_1 и β_1 из соотношения $T_1 = \alpha_1\cdot T_2 + \beta_1$.

Итак, $-\lambda\left.\frac{\partial T}{\partial x}\right|_{x=0} = \kappa_1\left(T^{e1} - T\big|_{x=0}\right) \Rightarrow -\lambda\frac{T_2 - T_1}{h} = \kappa_1\left(T^{e1} - T_1\right).$

Введем обозначение $\frac{\kappa \cdot h}{\lambda} \equiv \mathrm{Bi}$, тогда

$$T_1 - T_2 = \mathrm{Bi}_1 \cdot T^{e1} - \mathrm{Bi}_1 \cdot T_1;$$

$$T_1 = \frac{1}{1+\mathrm{Bi}_1} \cdot T_2 + \frac{\mathrm{Bi}_1}{1+\mathrm{Bi}_1} \cdot T^{e1};$$

$$\begin{cases} \alpha_1 = \dfrac{1}{1+\mathrm{Bi}_1} = \dfrac{\lambda}{\lambda + h \cdot \kappa_1}; \\ \beta_1 = \dfrac{\mathrm{Bi}_1}{1+\mathrm{Bi}_1} \cdot T^{e1} = \dfrac{h \cdot \kappa_1}{\lambda + h \cdot \kappa_1} \cdot T^{e1}. \end{cases} \qquad (22)$$

А правое граничное условие используют для определения температуры T_N.

Итак,

$$\lambda\left.\frac{\partial T}{\partial x}\right|_{x=L} = \kappa_2\left(T^{e2} - T\big|_{x=L}\right);$$

$$\lambda\frac{T_N - T_{N-1}}{h} = \kappa_2\left(T^{e2} - T_N\right).$$

$$\lambda\left.\frac{\partial T}{\partial x}\right|_{x=L} = \kappa_2\left(T^{e2} - T\big|_{x=L}\right);$$

$$\lambda\frac{T_N - T_{N-1}}{h} = \kappa_2\left(T^{e2} - T_N\right);$$

$$T_N \cdot \left(1 + \mathrm{Bi}_2\right) = T_{N-1} + \mathrm{Bi}_2 \cdot T^{e2};$$

т.к. $T_{N-1} = \alpha_{N-1} \cdot T_N + \beta_{N-1}$, то

$$T_N \cdot \left(1 + \mathrm{Bi}_2\right) = \alpha_{N-1} \cdot T_N + \beta_{N-1} + \mathrm{Bi}_2 \cdot T^{e2};$$

$$T_N = \frac{\beta_{N-1} + \mathrm{Bi}_2 \cdot T^{e2}}{1 + \mathrm{Bi}_2 - \alpha_{N-1}}.$$

или $T_N = \dfrac{\lambda \cdot \beta_{N-1} + h \cdot \kappa_2 \cdot T^{e2}}{h \cdot \kappa_2 + \lambda \cdot \left(1 - \alpha_{N-1}\right)}.$ (23)

Проведем дискретизацию граничных условий III рода с погрешностью $O(h^2)$. Предположим, что на границе выполняется уравнение теплопроводности (19). Далее по аналогии с аппроксимацией граничного условия II рода получим:

$$\left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1} = \frac{T_2^{n+1} - T_1^{n+1}}{h} - \frac{h}{2\cdot a}\cdot\frac{T_1^{n+1} - T_1^{n}}{\tau} = \frac{\kappa_1}{\lambda}\cdot T_1^{n+1} - \frac{\kappa_1}{\lambda}\cdot T^{e1},$$ т.к.

$$T_1^{n+1} = \alpha_1 \cdot T_2^{n+1} + \beta_1,$$ то

$$\begin{cases} \alpha_1 = \dfrac{2\cdot a\cdot\tau}{h^2 + 2\cdot a\cdot\tau\cdot(1+\mathrm{Bi}_1)}; \\ \beta_1 = \dfrac{h^2}{h^2 + 2\cdot a\cdot\tau\cdot(1+\mathrm{Bi}_1)}\cdot T_1^n + \dfrac{2\cdot a\cdot\tau\cdot\mathrm{Bi}_1\cdot T^{e1}}{h^2 + 2\cdot a\cdot\tau\cdot(1+\mathrm{Bi}_1)}. \end{cases}$$

или

$$\begin{cases} \alpha_1 = \dfrac{2\cdot a\cdot\tau\cdot\lambda}{\lambda\cdot h^2 + 2\cdot a\cdot\tau\cdot(\lambda + h\cdot\kappa_1)}; \\ \beta_1 = \dfrac{\lambda\cdot h^2\cdot T_1^n + 2\cdot a\cdot\tau\cdot h\cdot\kappa_1\cdot T^{e1}}{\lambda\cdot h^2 + 2\cdot a\cdot\tau\cdot(\lambda + h\cdot\kappa_1)}. \end{cases} \qquad (24)$$

Использование же правого граничного условия дает следующее соотношение:

$$T_N^{n+1} = \frac{h^2\cdot T_N^n + 2\cdot a\cdot\tau\cdot(\beta_{N-1} + \mathrm{Bi}_2\cdot T^{e2})}{h^2 + 2\cdot a\cdot\tau\cdot(1 + \mathrm{Bi}_2 - \alpha_{N-1})} = \frac{\lambda\cdot h^2\cdot T_N^n + 2\cdot a\cdot\tau\cdot(\lambda\cdot\beta_{N-1} + h\cdot\kappa_2\cdot T^{e2})}{\lambda\cdot h^2 + 2\cdot a\cdot\tau\cdot(h\cdot\kappa_2 + \lambda\cdot(1-\alpha_{N-1}))}. \qquad (25)$$

2.3. ПРИМЕРЫ КРАЕВЫХ ЗАДАЧ С РАЗЛИЧНЫМИ ГРАНИЧНЫМИ УСЛОВИЯМИ

1. Определим температурное поле в медной пластинке через 5, 10, 30 и 60 секунд. Толщина пластины $L = 0.3$ м. Начальная температура $T_0 = 20^0$C. Медь имеет следующие теплофизические характеристики λ = 384 Вт/(м·ºС), ρ = 8800 кг/м^3, c = 381 Дж/(кг·ºС). На границе $x = 0$ приложен тепловой поток $q = 10^7$ Вт/м^2, а граница $x = L$ подвержена воздействию внешней среды ($\kappa = 100$ Вт/(м$^2\cdot{}^0$C), $T^e = 300\ ^0$C).

Математическая постановка задачи будет иметь вид:

$$\rho c\frac{\partial T}{\partial t} = \lambda\frac{\partial^2 T}{\partial x^2}, \quad 0 < x < L.$$

Начальные и граничные условия запишутся следующим образом:

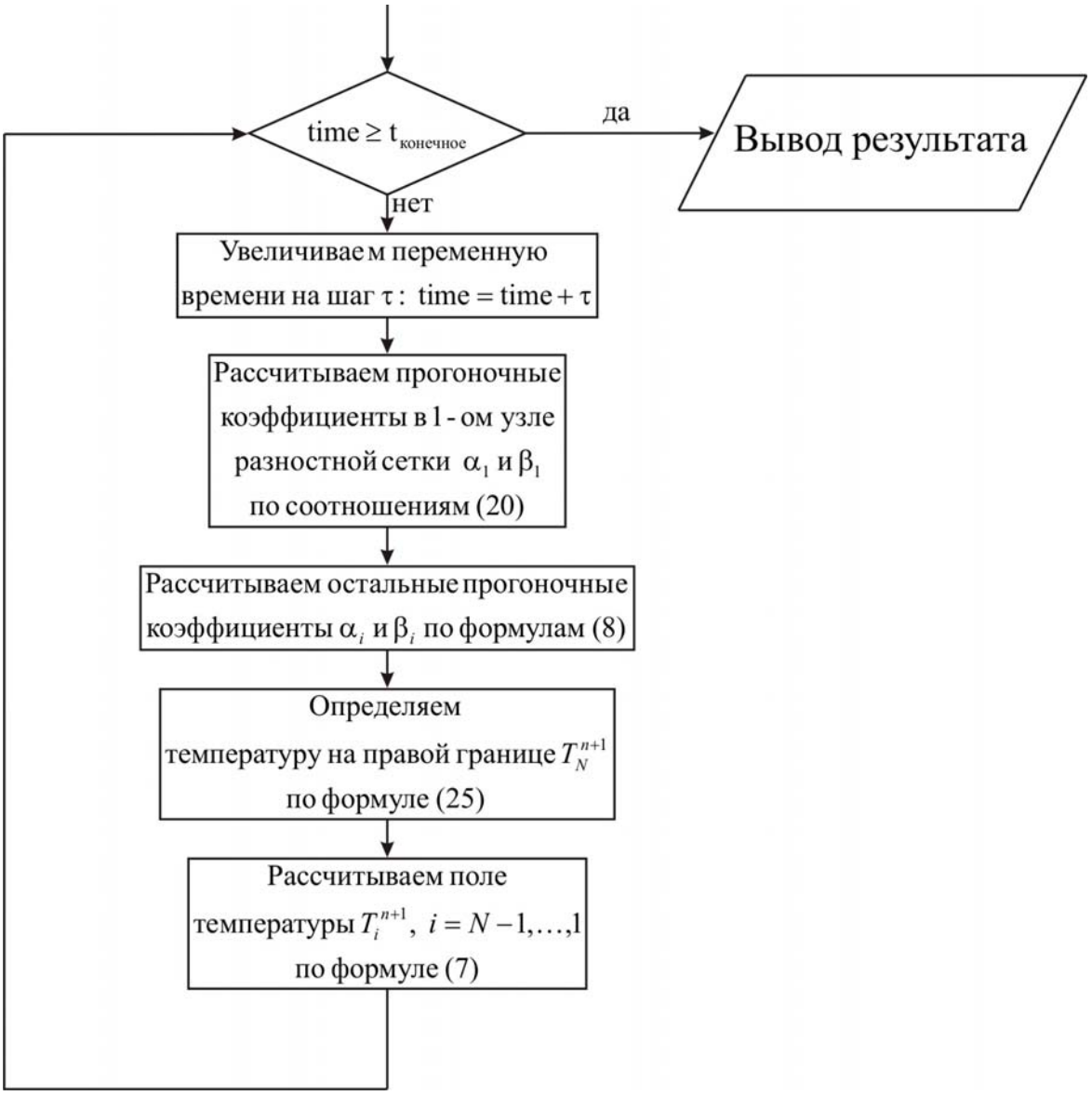
$$t = 0:\ T = T_0,\ 0 \le x \le L;$$

$$x = 0:\ -\lambda \frac{\partial T}{\partial x} = q,\ t > 0;$$

$$x = L:\lambda \frac{\partial T}{\partial x} = \kappa\left(T^{\text{e}} - T\right),\ t > 0.$$

Блок-схема к рассматриваемой задаче имеет вид:




time ≥ t конечное
да
Вывод результата
нет
Увеличиваем переменную времени на шаг τ : time = time + τ
Рассчитываем прогоночные коэффициенты в 1 - ом узле разностной сетки α1 и β1 по соотношениям (20)
Рассчитываем остальные прогоночные коэффициенты αi и βi по формулам (8)
Определяем температуру на правой границе T_N^{n+1} по формуле (25)
Рассчитываем поле температуры T_i^{n+1}, $i = N-1,\ldots,1$ по формуле (7)

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
type
    vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N                    : integer;
 T, alfa, beta              : vector;
 ai, bi, ci, fi             : real;
 a, lamda, ro, c, h, tau    : real;
 q, kapa, Te                : real;
 T0, L, t_end, time         : real;
 f, g                       : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите толщину пластины, L');
 Readln(L);
 Writeln('Введите коэффициент теплопроводности материала пластины, lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите плотность теплового потока, q');
 Readln(q);
 Writeln('Введите коэффициент теплообмена, kapa');
 Readln(kapa);
 Writeln('Введите температуру внешней среды, Te');
 Readln(Te);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
```

```
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
{определяем коэффициент температуропроводности}
 a:=lamda/(ro*c);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to N do
  T[i]:=T0;
{проводим интегрирование нестационарного уравнения теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{определяем начальные прогоночные коэффициенты на основе левого граничного условия, используя соотношения (20)}
   alfa[1]:=2.0*a*tau/(2.0*a*tau+sqr(h));
   beta[1]:=(sqr(h)*T[1]+2.0*a*tau*h*q/lamda)/(2.0*a*tau+sqr(h));
{цикл с параметром для определения прогоночных коэффициентов по формуле (8)}
   for i:= 2 to N-1 do
    begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с трехдиагональной матрицей}
     ai:=lamda/sqr(h);
     bi:=2.0*lamda/sqr(h)+ro*c/tau;
     ci:=lamda/sqr(h);
     fi:=-ro*c*T[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
     alfa[i]:=ai/(bi-ci*alfa[i-1]);
     beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
    end;
{определяем значение температуры на правой границе, используя соотношение (25)}
   T[N]:=(lamda*sqr(h)*T[N]+2.0*a*tau*(lamda*beta[N-1]+kapa*h*Te))
         /(lamda*sqr(h)+2.0*a*tau*(lamda*(1-alfa[N-1])+kapa*h));
{используя соотношение (7) определяем неизвестное поле температуры}
   for i:= N-1 downto 1 do
```

```
    T[i]:=alfa[i]*T[i+1]+beta[i];
  end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Толщина пластины L = ',L:6:4);
 Writeln(f,'Число узлов по координате N = ',N);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура  T0 = ',T0:6:4);
 Writeln(f,'Плотность теплового потока q = ',q:6:4);
 Writeln(f,'Коэффициент теплообмена kapa = ',kapa:6:4);
 Writeln(f,'Температура внешней среды Te = ',Te:6:4);
 Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):10:8,'   ',T[i]:8:5);
 close(g);
end.
```

В результате получены следующие распределения температуры

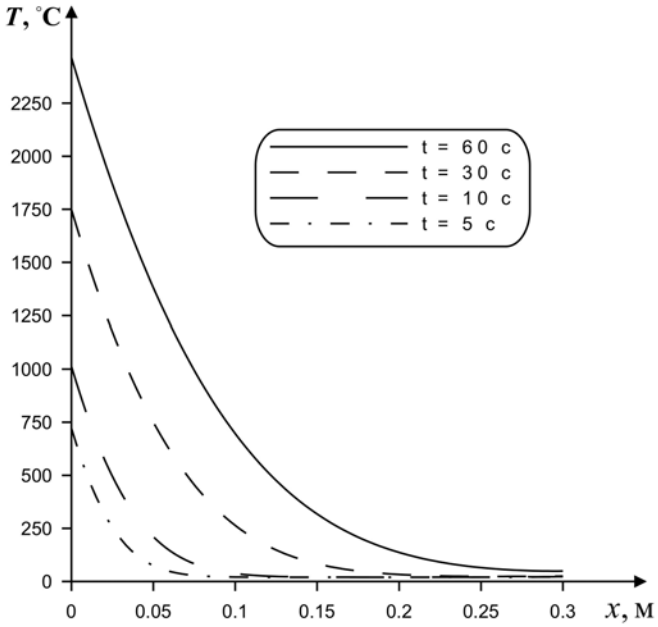


Рис. 11. Распределения температуры по толщине пластины в различные моменты времени

2. Определим температурное поле в медной пластине через 30, 180 и 600 секунд. Толщина пластины $L = 0.3\,\text{м}$. Начальная температура $T_0 = 50^0\,\text{C}$. На границе $x = 0$ и $x = L$ пластина контактирует с внешней средой ($\kappa_1 = 1000\,\text{Вт}/\left(\text{м}^2 \cdot {}^0\,\text{C}\right)$, $T^{e1} = -30\ {}^0\text{C}$ и $\kappa_2 = 500\,\text{Вт}/\left(\text{м}^2 \cdot {}^0\,\text{C}\right)$, $T^{e1} = 10\ {}^0\text{C}$).

Математическая постановка задачи будет иметь вид:

$$\rho c \frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2}, \quad 0 < x < L.$$

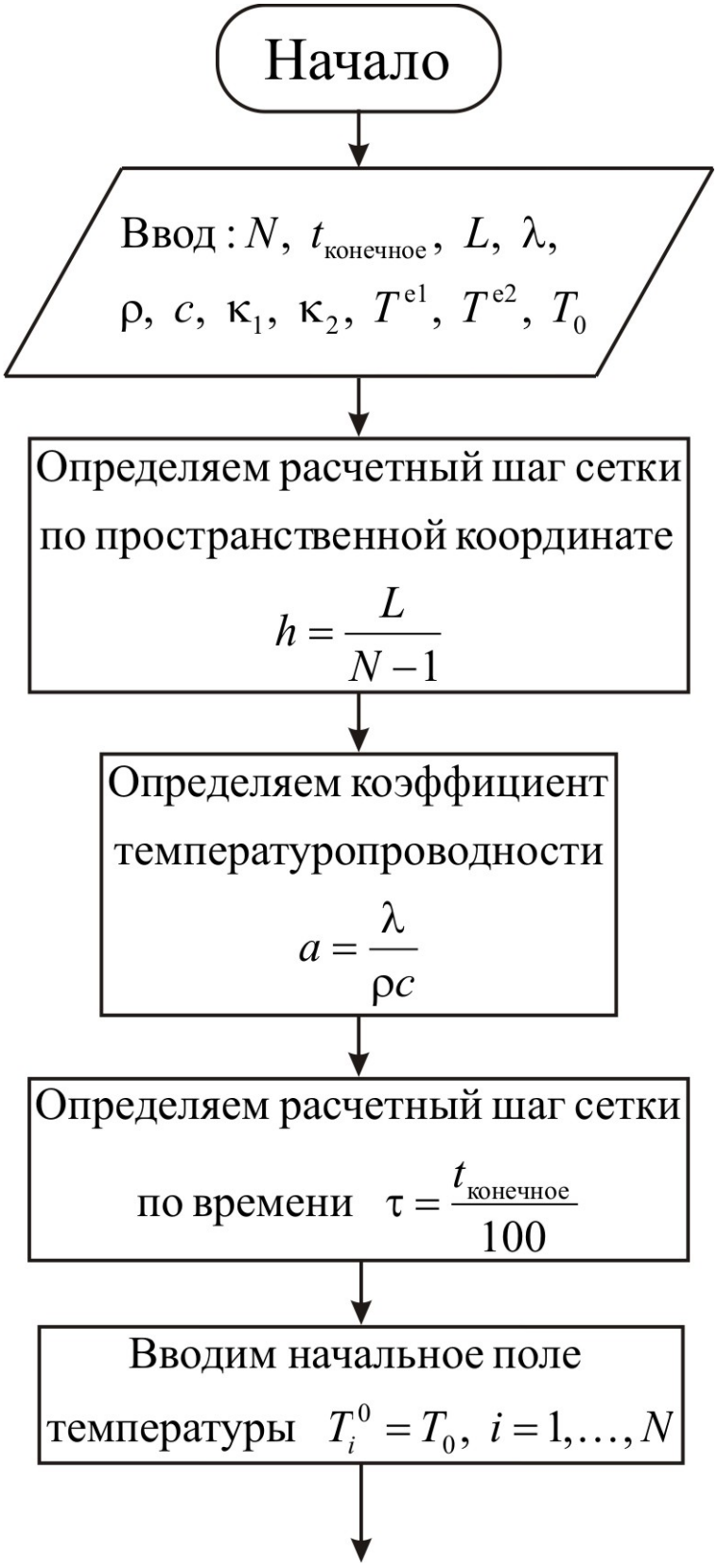
Начальные и граничные условия запишутся следующим образом:

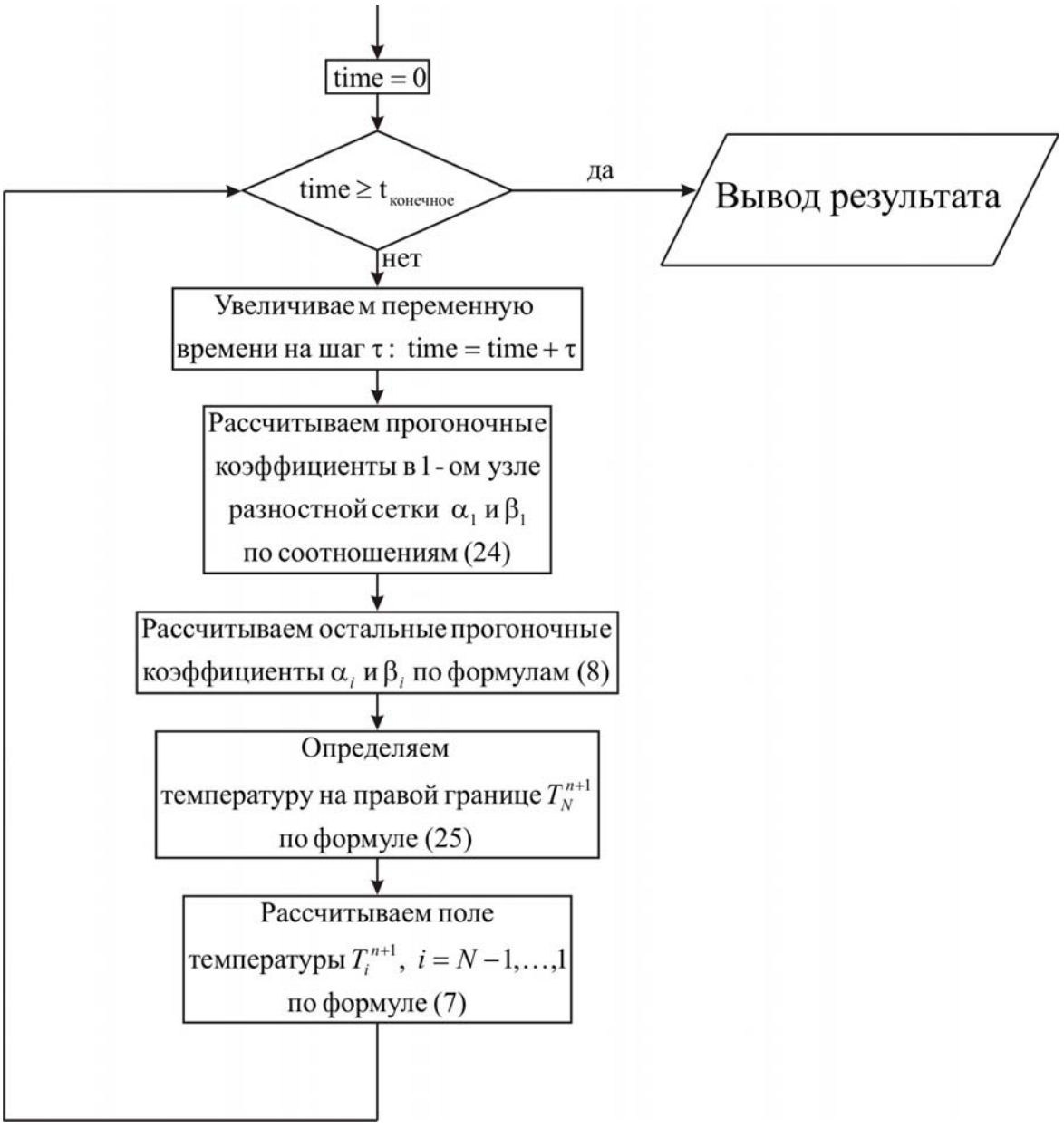
$$t = 0: \ T = T_0, 0 \le x \le L;$$

$$x = 0: \ -\lambda \frac{\partial T}{\partial x} = \kappa_1 \left(T^{e1} - T\right), \ t > 0;$$

$$x = L: \lambda \frac{\partial T}{\partial x} = \kappa_2 \left(T^{e2} - T\right), \ t > 0.$$

Блок-схема к рассматриваемой задаче имеет вид:




time = 0
time ≥ t конечное
да
Вывод результата
нет
Увеличиваем переменную времени на шаг τ : time = time + τ
Рассчитываем прогоночные коэффициенты в 1 - ом узле разностной сетки α1 и β1 по соотношениям (24)
Рассчитываем остальные прогоночные коэффициенты αi и βi по формулам (8)
Определяем температуру на правой границе T_N^{n+1} по формуле (25)
Рассчитываем поле температуры T_i^{n+1}, $i = N-1,\ldots,1$ по формуле (7)

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
type
    vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N                    : integer;
 T, alfa, beta              : vector;
 ai, bi, ci, fi             : real;
 a, lamda, ro, c, h, tau    : real;
 kapa1, kapa2, Te1, Te2     : real;
 T0, L, t_end, time         : real;
 f, g                       : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите толщину пластины, L');
 Readln(L);
 Writeln('Введите коэффициент теплопроводности материала пластины, lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите коэффициент теплообмена на границе x = 0, kapa1');
 Readln(kapa1);
 Writeln('Введите коэффициент теплообмена на границе x = L, kapa2');
 Readln(kapa2);
 Writeln('Введите температуру внешней среды относительно границы x = 0, Te1');
 Readln(Te1);
 Writeln('Введите температуру внешней среды относительно границы x = L, Te2');
```

```
 Readln(Te2);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
{определяем коэффициент температуропроводности}
 a:=lamda/(ro*c);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to N do
  T[i]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия, используя соотношения (24)}
   alfa[1]:=2.0*a*tau*lamda/(2.0*a*tau*(lamda+kapa1*h)+lamda*sqr(h));
   beta[1]:=(lamda*sqr(h)*T[1]+2.0*a*tau*kapa1*h*Te1)
              /(2.0*a*tau*(lamda+kapa1*h)+lamda*sqr(h));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
   for i:= 2 to N-1 do
    begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
     ai:=lamda/sqr(h);
     bi:=2.0*lamda/sqr(h)+ro*c/tau;
     ci:=lamda/sqr(h);
     fi:=-ro*c*T[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
     alfa[i]:=ai/(bi-ci*alfa[i-1]);
     beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
    end;
```

```
{определяем значение температуры на правой границе, используя
соотношение (25)}
   T[N]:=(lamda*sqr(h)*T[N]+2.0*a*tau*(lamda*beta[N-1]+kapa2*h*Te2))
            /(lamda*sqr(h)+2.0*a*tau*(lamda*(1-alfa[N-1])+kapa2*h));
{используя соотношение (7) определяем неизвестное поле
температуры}
   for i:= N-1 downto 1 do
    T[i]:=alfa[i]*T[i+1]+beta[i];
  end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Толщина пластины L = ',L:6:4);
 Writeln(f,'Число узлов по координате N = ',N);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура  T0 = ',T0:6:4);
 Writeln(f,'Коэффициент теплообмена kapa1 = ',kapa1:6:4);
 Writeln(f,'Коэффициент теплообмена kapa2 = ',kapa2:6:4);
 Writeln(f,'Температура внешней среды Te1 = ',Te1:6:4);
 Writeln(f,'Температура внешней среды Te2 = ',Te2:6:4);
 Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):10:8,'   ',T[i]:8:5);
 close(g);
end.
```

Решение задачи имеет вид:

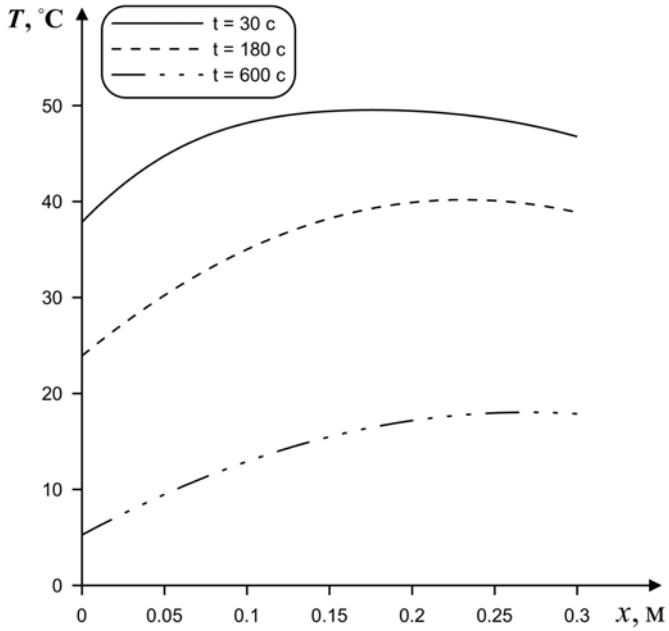


Рис. 12. Распределения температуры по толщине пластины в различные моменты времени

2.4. ДВУХСЛОЙНАЯ ПЛАСТИНА

Во многих важных практических приложениях проводится оценка температурных полей в многослойных деталях конструкций.

Проанализируем процесс теплопереноса в теле, представляющем собой совокупность двух пластин с различными теплофизическими характеристиками (рис. 13).

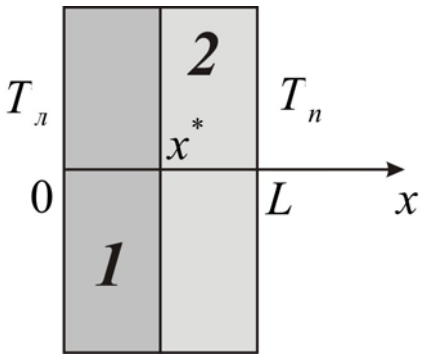


Рис. 13. Геометрия задачи

Математическая постановка задачи будет иметь вид:

$$\begin{cases} \rho_1 c_1 \dfrac{\partial T_1}{\partial t} = \lambda_1 \dfrac{\partial^2 T_1}{\partial x^2}, \quad 0 < x < x^*; \\ \rho_2 c_2 \dfrac{\partial T_2}{\partial t} = \lambda_2 \dfrac{\partial^2 T_2}{\partial x^2}, \quad x^* < x < L; \end{cases}$$

где 1 соответствует левой пластине (***1*** на рис. 13),
2 соответствует левой пластине (***2*** на рис. 13).

Начальные и граничные условия можно записать следующим образом:

$$t = 0: \; T = T_0, 0 \le x \le L;$$

$$x = 0: \; T = T_л, \; t > 0;$$

$$x = L: \; T = T_п, \; t > 0;$$

$$\begin{cases} T_1\left(t, x^*\right) = T_2\left(t, x^*\right), \\ -\lambda_1 \dfrac{\partial T_1}{\partial x}\Big|_{x=x^*} = -\lambda_2 \dfrac{\partial T_2}{\partial x}\Big|_{x=x^*}. \end{cases}$$

На границах $x = 0$ и $x = L$ рассматриваются граничные условия первого рода для простоты дальнейшего изложения. Задание на этих границах условий II или III рода было подробно изложено в пункте 2.2. Принципиальным моментом в настоящем параграфе является исследование граничных условий IV рода в точке контакта двух пластин.

Решение данной задачи проводится также численно на основе неявной разностной схемы. Граничное условие IV рода используется для определения прогоночных коэффициентов в точке x^*.

Алгоритм решения сформулированной краевой задачи можно представить следующим образом.

Сначала проводим аппроксимацию дифференциального уравнения конечными разностями, получаем систему линейных алгебраических уравнений вида (10), которую решаем методом прогонки. При нахождении прогоночных коэффициентов в области $0 \le x < x^*$ используем характеристики среды ***1***, а при $x^* < x \le L$ – среды ***2***. В точке же $x = x^*$ необходимо использовать граничное условие IV рода.

Выведем прогоночные коэффициенты в точке контакта двух сред.

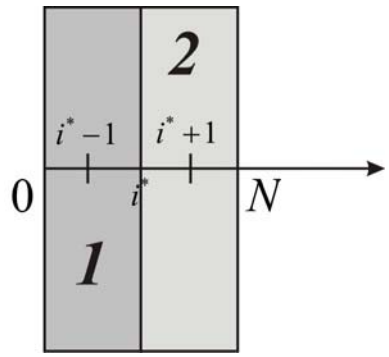


Рис. 14. Шаблон разностной сетки

$$\begin{cases} T_{1,i^*} = T_{2,i^*}, \\ -\lambda_1 \dfrac{\partial T_1}{\partial x}\Big|_{x=x^*} = -\lambda_2 \dfrac{\partial T_2}{\partial x}\Big|_{x=x^*}. \end{cases}$$

Рассмотрим аппроксимацию первого порядка относительно шага по пространственной координате. При этом, принимая во внимание то, что при $i < i^*$ $T = T_1$, а при $i > i^*$ $T = T_2$ исключим в записи индексы, характеризующие среду. Получим следующие соотношения:

$$\begin{cases} T_{1,i^*} = T_{2,i^*}, \\ -\lambda_1 \cdot \dfrac{T_{1,i^*} - T_{i^*-1}}{h} = -\lambda_2 \cdot \dfrac{T_{i^*+1} - T_{2,i^*}}{h}. \end{cases} \tag{26}$$

Введем обозначение $T_{1,i^*} = T_{2,i^*} \equiv T_{i^*}$.

Используя прогоночное соотношение $T_{i^*-1} = \alpha_{i^*-1} \cdot T_{i^*} + \beta_{i^*-1}$ из второго равенства условий (26) получим:

$$T_{i^*} - \alpha_{i^*-1} \cdot T_{i^*} - \beta_{i^*-1} = \frac{\lambda_2}{\lambda_1} \cdot T_{i^*+1} - \frac{\lambda_2}{\lambda_1} \cdot T_{i^*}$$

или

$$T_{i^*} = \frac{\lambda_2}{\lambda_2 + \lambda_1 \cdot \left(1 - \alpha_{i^*-1}\right)} \cdot T_{i^*+1} + \frac{\lambda_1 \cdot \beta_{i^*-1}}{\lambda_2 + \lambda_1 \cdot \left(1 - \alpha_{i^*-1}\right)}.$$

Таким образом,

$$\begin{cases} \alpha_{i^*} = \dfrac{\lambda_2}{\lambda_2 + \lambda_1 \cdot \left(1 - \alpha_{i^*-1}\right)}; \\ \beta_{i^*} = \dfrac{\lambda_1 \cdot \beta_{i^*-1}}{\lambda_2 + \lambda_1 \cdot \left(1 - \alpha_{i^*-1}\right)}. \end{cases} \tag{27}$$

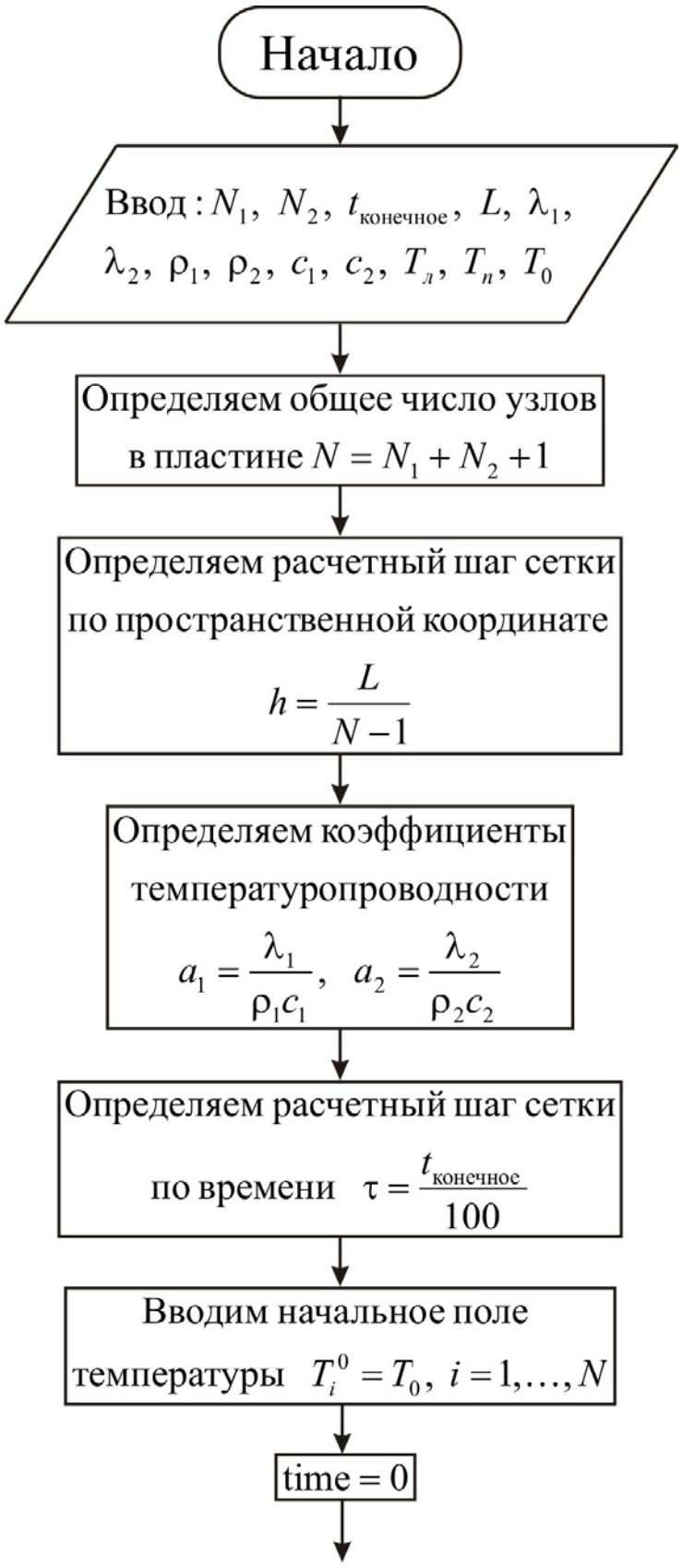
В случае же аппроксимации второго порядка относительно шага по пространственной координате получим

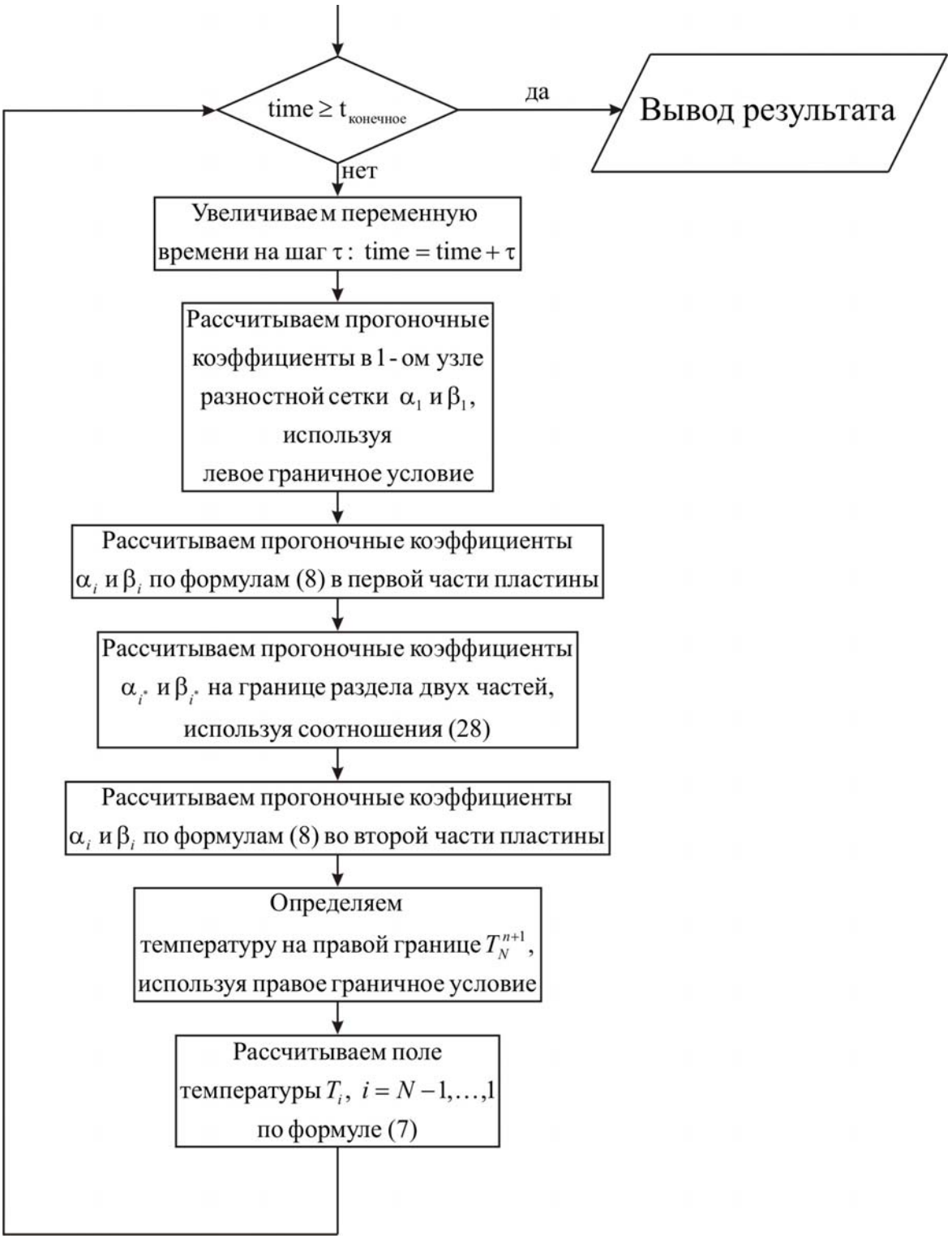
$$\begin{cases} \alpha_{i^*} = \dfrac{2 \cdot a_1 \cdot a_2 \cdot \tau \cdot \lambda_2}{2 \cdot a_1 \cdot a_2 \cdot \tau \cdot \left(\lambda_2 + \lambda_1 \cdot \left(1 - \alpha_{i^*-1}\right)\right) + h^2 \cdot \left(\lambda_1 \cdot a_2 + \lambda_2 \cdot a_1\right)}; \\ \beta_{i^*} = \dfrac{2 \cdot a_1 \cdot a_2 \cdot \tau \cdot \lambda_1 \cdot \beta_{i^*-1} + h^2 \cdot \left(\lambda_1 \cdot a_2 + \lambda_2 \cdot a_1\right) \cdot T_{i^*}^n}{2 \cdot a_1 \cdot a_2 \cdot \tau \cdot \left(\lambda_2 + \lambda_1 \cdot \left(1 - \alpha_{i^*-1}\right)\right) + h^2 \cdot \left(\lambda_1 \cdot a_2 + \lambda_2 \cdot a_1\right)}. \end{cases} \quad (28)$$

Итак, сначала находят прогоночные коэффициенты для первой среды, на границе i^* используют соотношения (27) или (28), а далее определяют прогоночные коэффициенты для второй среды.

В качестве примера определим температурное поле в составной пластинке (рис. 13) через 30, 180 и 600 секунд. Толщина пластины $L = 0.3$ м. Будем полагать, что толщины составных частей одинаковые. Начальная температура $T_0 = 10^0$C. Одна часть пластины (***1***) – сталь (λ = 46 Вт/(м·ºС), ρ = 7800 кг/м^3, c = 460 Дж/(кг·ºС)), а другая часть (***2***) – медь (λ = 384 Вт/(м·ºС), ρ = 8800 кг/м^3, c = 381 Дж/(кг·ºС)). На границе $x = 0$ поддерживается постоянная температура $T = 100\ ^0$C, а на границе $x = L$ $T = 50\ ^0$C.

Блок-схема к рассматриваемой задаче имеет вид:




time ≥ t конечное
да
Вывод результата
нет
Увеличиваем переменную времени на шаг τ : time = time + τ
Рассчитываем прогоночные коэффициенты в 1 - ом узле разностной сетки α1 и β1, используя левое граничное условие
Рассчитываем прогоночные коэффициенты αi и βi по формулам (8) в первой части пластины
Рассчитываем прогоночные коэффициенты αi* и βi* на границе раздела двух частей, используя соотношения (28)
Рассчитываем прогоночные коэффициенты αi и βi по формулам (8) во второй части пластины
Определяем температуру на правой границе TNn+1, используя правое граничное условие
Рассчитываем поле температуры Ti, i = N −1,…,1 по формуле (7)

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
type
    vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N, N1, N2          : integer;
 T, alfa, beta            : vector;
 ai, bi, ci, fi           : real;
 a1, lamda1, ro1, c1      : real;
 a2, lamda2, ro2, c2      : real;
 h, tau, t_end, time      : real;
 T0, Tl, Tr, L            : real;
 f, g                     : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество промежутков в первой части пластины, N1');
 Readln(N1);
 Writeln('Введите количество промежутков во второй части пластины, N2');
 Readln(N2);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите толщину пластины, L');
 Readln(L);
 Writeln('Введите коэффициент теплопроводности материала первой части пластины, lamda1');
 Readln(lamda1);
 Writeln('Введите коэффициент теплопроводности материала второй части пластины, lamda2');
 Readln(lamda2);
 Writeln('Введите плотность материала первой части пластины, ro1');
 Readln(ro1);
 Writeln('Введите плотность материала второй части пластины, ro2');
 Readln(ro2);
 Writeln('Введите теплоемкость материала первой части пластины, c1');
```

```
 Readln(c1);
 Writeln('Введите теплоемкость материала второй части пластины, c2');
 Readln(c2);
 Writeln('Введите температуру на границе x = 0, Tl');
 Readln(Tl);
 Writeln('Введите температуру на границе x = L, Tr');
 Readln(Tr);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
{определяем общее число узлов в пластине}
 N:=N1+N2+1;
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
{определяем коэффициенты температуропроводности}
 a1:=lamda1/(ro1*c1);
 a2:=lamda2/(ro2*c2);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to N do
  T[i]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия}
   alfa[1]:=0.0;
   beta[1]:=Tl;
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8) в первой части пластины}
   for i:= 2 to N1 do
    begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
     ai:=lamda1/sqr(h);
     bi:=2.0*lamda1/sqr(h)+ro1*c1/tau;
     ci:=lamda1/sqr(h);
```

```
    fi:=-ro1*c1*T[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
    alfa[i]:=ai/(bi-ci*alfa[i-1]);
    beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
   end;
{определяем прогоночные коэффициенты на границе раздела двух
частей, используем соотношения (28)}
  alfa[N1+1]:=2.0*a1*a2*tau*lamda2/(2.0*a1*a2*tau*(lamda2+lamda1
                *(1-alfa[N1]))+sqr(h)*(a1*lamda2+a2*lamda1));
  beta[N1+1]:=(2.0*a1*a2*tau*lamda1*beta[N1]+sqr(h)*(a1*lamda2+a2
                *lamda1)*T[N1+1])/(2.0*a1*a2*tau*(lamda2+lamda1
                *(1-alfa[N1]))+sqr(h)*(a1*lamda2+a2*lamda1));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8) во второй части пластины}
  for i:= N1+2 to N-1 do
   begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
    ai:=lamda2/sqr(h);
    bi:=2.0*lamda2/sqr(h)+ro2*c2/tau;
    ci:=lamda2/sqr(h);
    fi:=-ro2*c2*T[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
    alfa[i]:=ai/(bi-ci*alfa[i-1]);
    beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
   end;
{определяем значение температуры на правой границе}
  T[N]:=Tr;
{используя соотношение (7) определяем неизвестное поле
температуры}
  for i:= N-1 downto 1 do
   T[i]:=alfa[i]*T[i+1]+beta[i];
 end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Толщина пластины L = ',L:6:4);
 Writeln(f,'Толщина первой части пластины = ',N1*h:6:4);
 Writeln(f,'Толщина второй части пластины = ',N2*h:6:4);
 Writeln(f,'Число промежутков по координате в первой части пластины
N1 = ',N1);
```

```
 Writeln(f,'Число промежутков по координате во второй части пластины
N2 = ',N2);
 Writeln(f,'Общее число узлов по координате N = ',N);
 Writeln(f,'Коэффициент теплопроводности материала первой части
пластины lamda1 = ', lamda1:6:4);
 Writeln(f,'Коэффициент теплопроводности материала второй части
пластины lamda2 = ', lamda2:6:4);
 Writeln(f,'Плотность материала первой части пластины ro1 = ',ro1:6:4);
 Writeln(f,'Плотность материала второй части пластины ro2 = ',ro2:6:4);
 Writeln(f,'Теплоемкость материала первой части пластины c1 = ',c1:6:4);
 Writeln(f,'Теплоемкость материала второй части пластины c2 = ',c2:6:4);
 Writeln(f,'Начальная температура  T0 = ',T0:6:4);
 Writeln(f,'Температура на границе x = 0, Tl = ',Tl:6:4);
 Writeln(f,'Температура на границе x = L, Tr = ',Tr:6:4);
 Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):10:8,'   ',T[i]:8:5);
 close(g);
end.
```

Получены следующие распределения температуры:

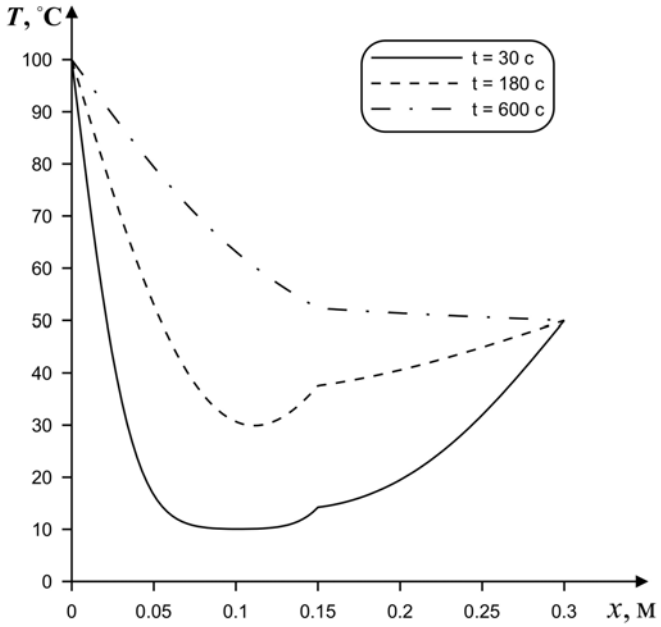


Рис. 15. Распределения температуры по толщине пластины в различные моменты времени

2.5. ЗАДАЧА ТЕПЛОПРОВОДНОСТИ С ВНУТРЕННИМИ ИСТОЧНИКАМИ

Пусть в неограниченной пластине толщины $L = 0.3$ м действуют равномерно распределенные внутренние источники тепла мощностью $Q(x)$. Данные источники находятся в точках $x \in A = \left\{ \frac{L}{4}, \frac{L}{2}, \frac{3 \cdot L}{4} \right\}$. В связи с этим определим функцию $Q(x) = \begin{cases} \frac{L}{4} \cdot q \text{ Вт/м}^3, & x = \frac{L}{4}; \\ \frac{L}{2} \cdot q \text{ Вт/м}^3, & x = \frac{L}{2}; \\ \frac{L}{4} \cdot q \text{ Вт/м}^3, & x = \frac{3 \cdot L}{4}; \\ 0 \text{ Вт/м}^3, & x \notin A, \end{cases}$ где $q = 10^5$ Вт/м^4. Начальная температура $T_0 = 15^0$C. Материал пластины – серебро (λ = 419 Вт/(м·ºС), ρ = 10500 кг/м^3, c = 200 Дж/(кг·ºС)). На границах $x = 0$ и $x = L$ осуществляется теплообмен с окружающей

средой ($\kappa = 50$ Вт/(м$^2 \cdot ^0$ С), $T^e = 60\ ^0$С). Определим температурные поля через 1 с, 5 с и 10 с.

Математическая постановка задачи будет иметь вид:

$$\rho c \frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2} + Q(x), \quad 0 < x < L.$$

Начальные и граничные условия запишутся следующим образом:

$$t = 0: \ T = T_0, 0 \le x \le L;$$

$$x = 0: \ -\lambda \frac{\partial T}{\partial x} = \kappa\left(T^e - T\right), \ t > 0;$$

$$x = L: \lambda \frac{\partial T}{\partial x} = \kappa\left(T^e - T\right), \ t > 0.$$

Для решения сформулированной краевой задачи применим метод конечных разностей на основе неявной четырехточечной схемы. В результате аппроксимации частных производных получаем следующую систему линейных алгебраических уравнений:

$$\rho \cdot c \cdot \frac{T_i^{n+1} - T_i^n}{\tau} = \lambda \cdot \left(\frac{T_{i+1}^{n+1} - 2 \cdot T_i^{n+1} + T_{i-1}^{n+1}}{h^2} \right) + Q_i^n, \qquad (29)$$

$$i = 2, \ldots, N-1, \quad n \ge 0.$$

Полученную систему можно свести к наиболее общему виду:

$$A_i \cdot T_{i+1}^{n+1} - B_i \cdot T_i^{n+1} + C_i \cdot T_{i-1}^{n+1} = F_i,$$

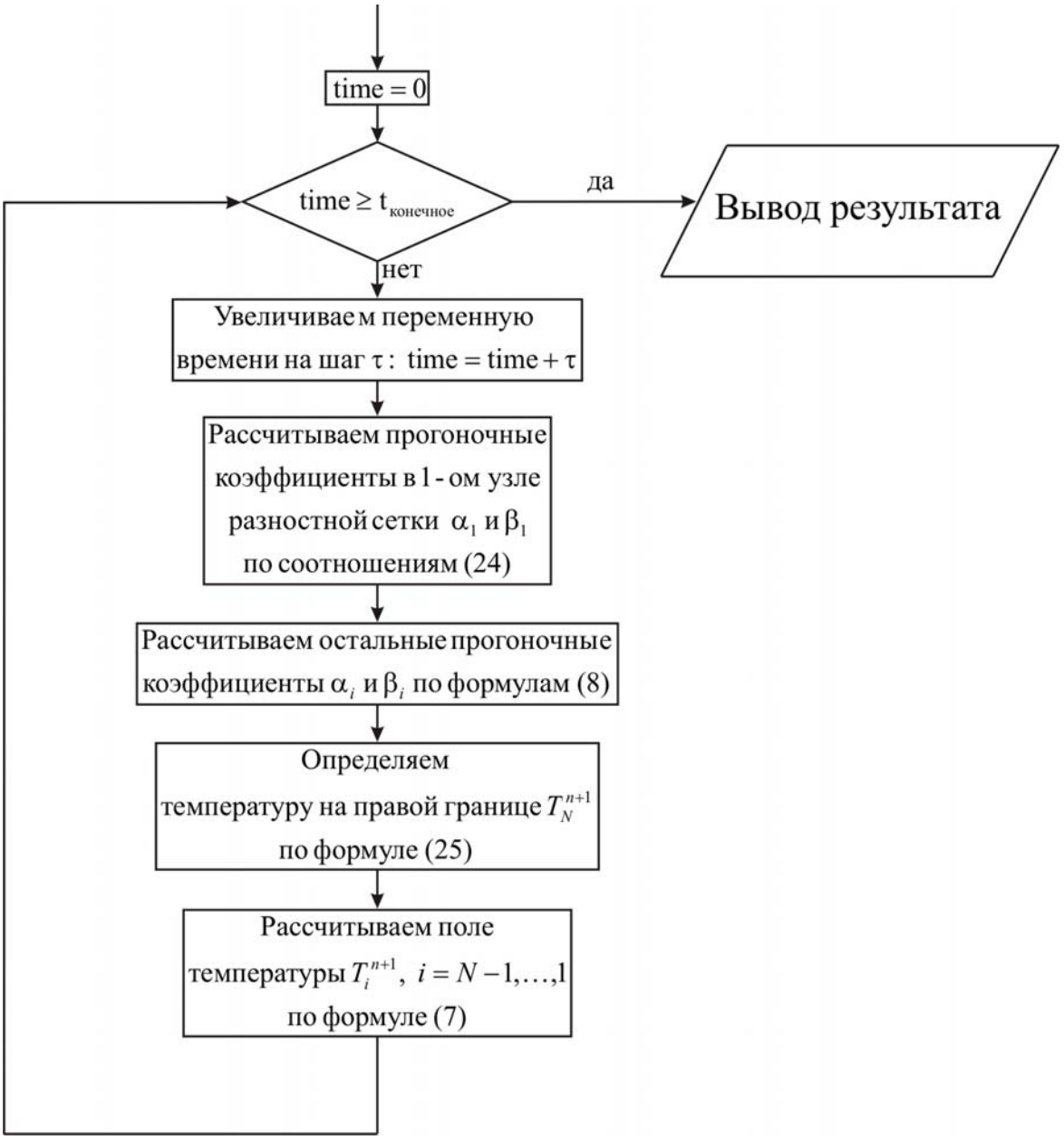
где

$$A_i = C_i = \frac{\lambda}{h^2}, \quad B_i = \frac{2 \cdot \lambda}{h^2} + \frac{\rho c}{\tau}, \quad F_i = -\frac{\rho c}{\tau} T_i^n - Q_i^n.$$

Прогоночные коэффициенты находятся по формулам (8). Далее неизвестное поле температуры определяется по выражению (7).

Блок-схема к рассматриваемой задаче имеет вид:




time = 0
time ≥ t конечное
да
Вывод результата
нет
Увеличиваем переменную времени на шаг τ : time = time + τ
Рассчитываем прогоночные коэффициенты в 1 - ом узле разностной сетки α1 и β1 по соотношениям (24)
Рассчитываем остальные прогоночные коэффициенты αi и βi по формулам (8)
Определяем температуру на правой границе T_N^{n+1} по формуле (25)
Рассчитываем поле температуры T_i^{n+1}, $i = N-1,\ldots,1$ по формуле (7)

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
type
    vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N, N1, N2, N3        : integer;
 T, alfa, beta              : vector;
 ai, bi, ci, fi             : real;
 a, lamda, ro, c            : real;
 h, tau, t_end, time        : real;
 T0, L, kapa, Te, q         : real;
 f, g                       : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов в пластине, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите толщину пластины, L');
 Readln(L);
 Writeln('Введите коэффициент теплопроводности материала пластины,
lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите температуру окружающей среды, Te');
 Readln(Te);
 Writeln('Введите коэффициент теплообмена, kapa');
 Readln(kapa);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
 Writeln('Введите мощность внутренних источников тепла, q');
 Readln(q);
```

```
{определяем номера узлов, в которых расположен источник}
 N1:=N div 4;
 N2:=2*N1;
 N3:=3*N1;
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
{определяем коэффициент температуропроводности}
 a:=lamda/(ro*c);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to N do
  T[i]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия, используя соотношения (24)}
   alfa[1]:=2.0*a*tau*lamda/(lamda*sqr(h)+2.0*a*tau*(lamda+kapa*h));
  beta[1]:=(lamda*sqr(h)*T[1]+2.0*a*tau*kapa*h*Te)/(lamda*sqr(h)+2.0
            *a*tau*(lamda+kapa*h));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
   for i:= 2 to N-1 do
    begin
{ai, bi, ci, fi – коэффициенты канонического представления системы
уравнений с трехдиагональной матрицей}
     ai:=lamda/sqr(h);
     bi:=2.0*lamda/sqr(h)+ro*c/tau;
     ci:=lamda/sqr(h);
     fi:=-ro*c*T[i]/tau;
{определяем fi в зависимости от рассматрвиаемой точки
пространства}
     if (i=N1) or (i=N2) then fi:=-ro*c*T[i]/tau-h*(i-1)*q;
     if i=N3 then fi:=-ro*c*T[i]/tau-h*(N1-1)*q;
{alfa[i], beta[i] – прогоночные коэффициенты}
     alfa[i]:=ai/(bi-ci*alfa[i-1]);
```

```
     beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
    end;
{определяем значение температуры на правой границе, используя
соотношение (25)}
   T[N]:=(lamda*sqr(h)*T[N]+2.0*a*tau*(lamda*beta[N-1]+kapa*h*Te))
          /(lamda*sqr(h)+2.0*a*tau*(lamda*(1-alfa[N-1])+kapa*h));
{используя соотношение (7) определяем неизвестное поле
температуры}
   for i:= N-1 downto 1 do
    T[i]:=alfa[i]*T[i+1]+beta[i];
  end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Толщина пластины L = ',L:6:4);
 Writeln(f,'Число узлов по пространственной координате в пластине N =
',N);
 Writeln(f,'Внутренние источники находятся в точках = ',N1,'  ',N2,'  ',N3);
 Writeln(f,'Внутренние источники находятся в точках = ',(N1-1)*h:6:4,'
',(N2-1)*h:6:4,'  ', (N3-1)*h:6:4);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура T0 = ',T0:6:4);
 Writeln(f,'Коэффициент теплообмена kapa = ',kapa:6:4);
 Writeln(f,'Температура окружающей среды Te = ',Te:6:4);
 Writeln(f,'Составляющая мощности внутренних источников тепла q =
',q:6:4);
 Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):10:8,'   ',T[i]:8:5);
 close(g);
end.
```

Получены следующие распределения температуры:

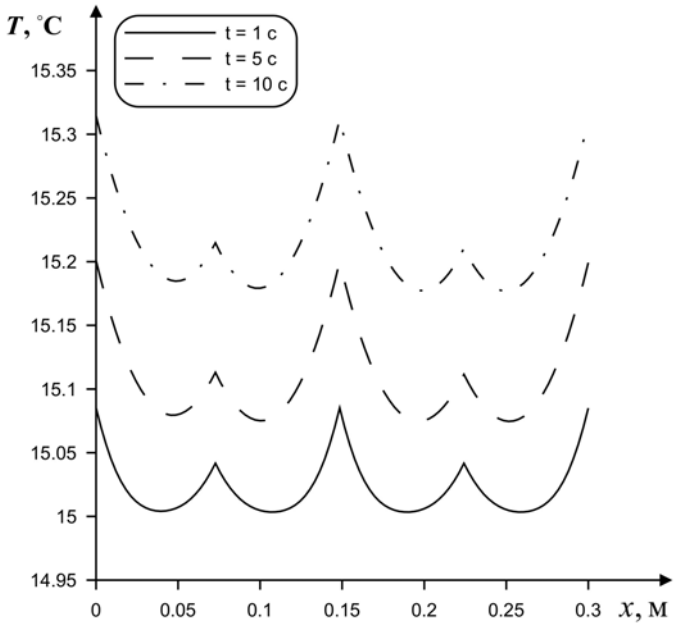


Рис. 16. Распределения температуры по толщине пластины в различные моменты времени

2.6. ДВУМЕРНАЯ ЗАДАЧА ТЕПЛОПРОВОДНОСТИ ДЛЯ ОДНОРОДНОГО ТЕЛА

Проанализируем процесс теплопереноса в пластине (рис. 17).

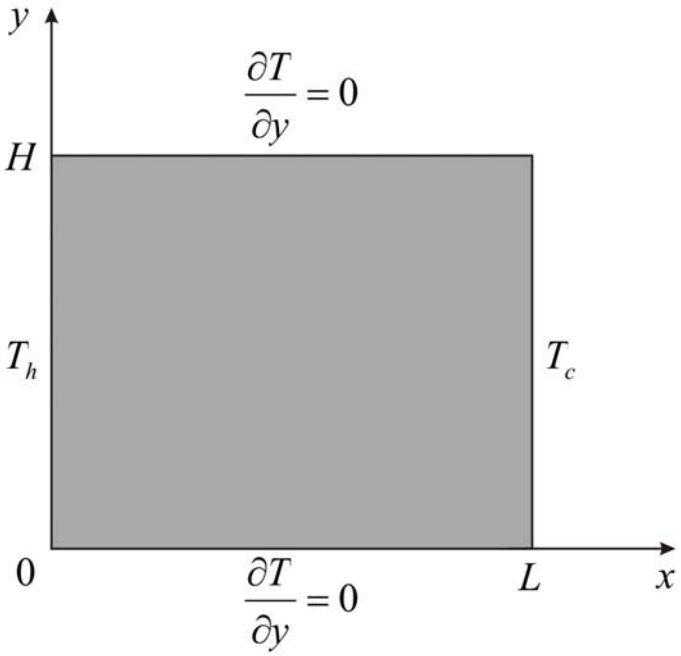


Рис. 17. Область решения

Медная пластина с размерами $L = H = 0.5$ м. Горизонтальные границы являются адиабатическими, а на вертикальных границах поддерживаются постоянные температуры $T_h = 80\ ^0\text{C}$ и $T_c = 30\ ^0\text{C}$. Начальная температура области решения $T_0 = 5\ ^0\text{C}$.

Математическая постановка задачи будет иметь вид:

$$\rho c \frac{\partial T}{\partial t} = \lambda \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right), \quad \left| \begin{array}{l} 0 < x < L; \\ 0 < y < H. \end{array} \right. \tag{30}$$

Начальные и граничные условия запишутся следующим образом:

$$\begin{aligned} & t = 0: \ T = T_0, 0 \le x \le L, \ 0 \le y \le H; \\ & x = 0: \ T = T_h, \ t > 0; \\ & x = L: \ T = T_c, \ t > 0; \\ & y = 0: \ \frac{\partial T}{\partial y} = 0, \ t > 0; \\ & y = H: \ \frac{\partial T}{\partial y} = 0, \ t > 0. \end{aligned} \tag{31}$$

Для аппроксимации дифференциального уравнения (30) разностным введем пространственно-временную сетку с координатами

$$x_i = (i-1) \cdot h_x, \quad y_j = (j-1) \cdot h_y, \quad t_n = n \cdot \tau,$$

где h_x, h_y – шаги сетки по координатам x, y соответственно; τ – шаг по времени; $i = \overline{1, N_x}$; $j = \overline{1, N_y}$; $n = \overline{0, K}$. Т.е. вся расчетная область (рис. 17) покрывается сеткой (рис. 18).

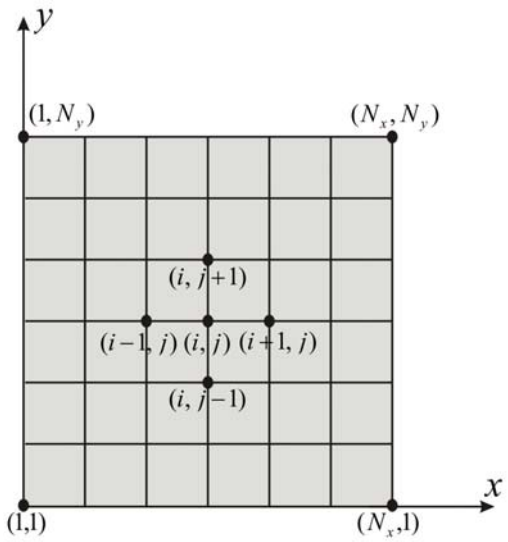


Рис. 18. Разностная сетка области решения

Введем следующее обозначение: $T\left(x_i, y_j, t_n\right) = T_{i,j}^n$.

Дискретизацию уравнения (30) будем проводить на основе локально одномерной схемы А.А. Самарского [2], которая является абсолютно устойчивой и обладает свойством суммарной аппроксимации. Сущность этого подхода состоит в том, что шаг по времени реализуется в два этапа – на промежуточном временном шаге проводим дискретизацию двумерного уравнения (30) только в направлении оси x и получаем одномерное уравнение, после его решения проводим вновь дискретизацию уравнения (30), но уже в направлении оси y и, решая полученное одномерное уравнение, определяем поле температуры на целом шаге по времени.

Итак:

$$\rho \cdot c \cdot \frac{T_{i,j}^{n+\frac{1}{2}} - T_{i,j}^n}{\tau} = \lambda \cdot \left(\frac{T_{i+1,j}^{n+\frac{1}{2}} - 2 \cdot T_{i,j}^{n+\frac{1}{2}} + T_{i-1,j}^{n+\frac{1}{2}}}{h_x^2} \right), \tag{32}$$

$$\rho \cdot c \cdot \frac{T_{i,j}^{n+1} - T_{i,j}^{n+\frac{1}{2}}}{\tau} = \lambda \cdot \left(\frac{T_{i,j+1}^{n+1} - 2 \cdot T_{i,j}^{n+1} + T_{i,j-1}^{n+1}}{h_y^2} \right). \tag{33}$$

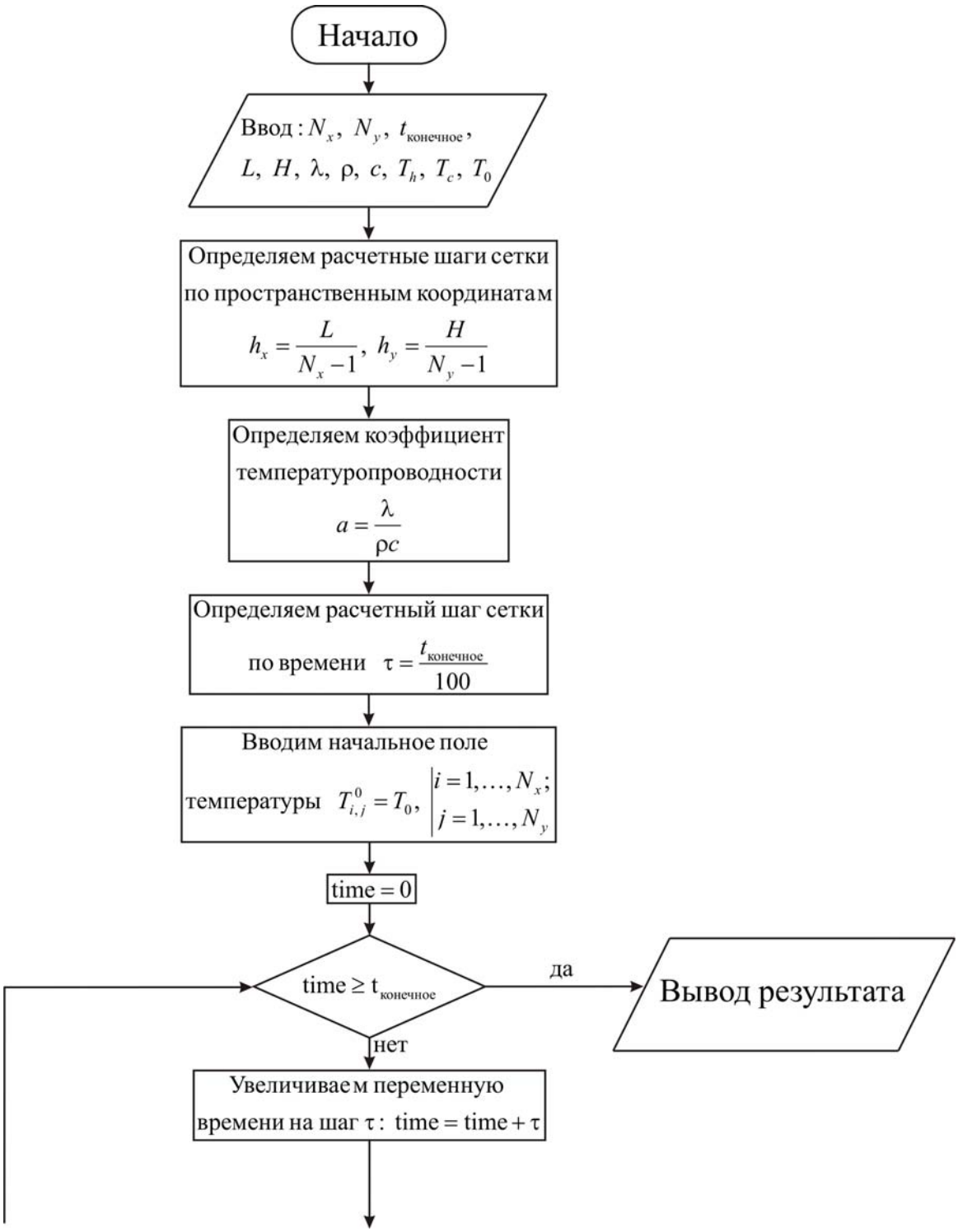
Разностные уравнения (32), (33) сводятся к стандартному трехдиагональному виду и решаются последовательно методом прогонки (пункт 2.1.). Сначала для всей области решается уравнение (32), после того как его решение будет найдено, переходят к решению уравнения (33).

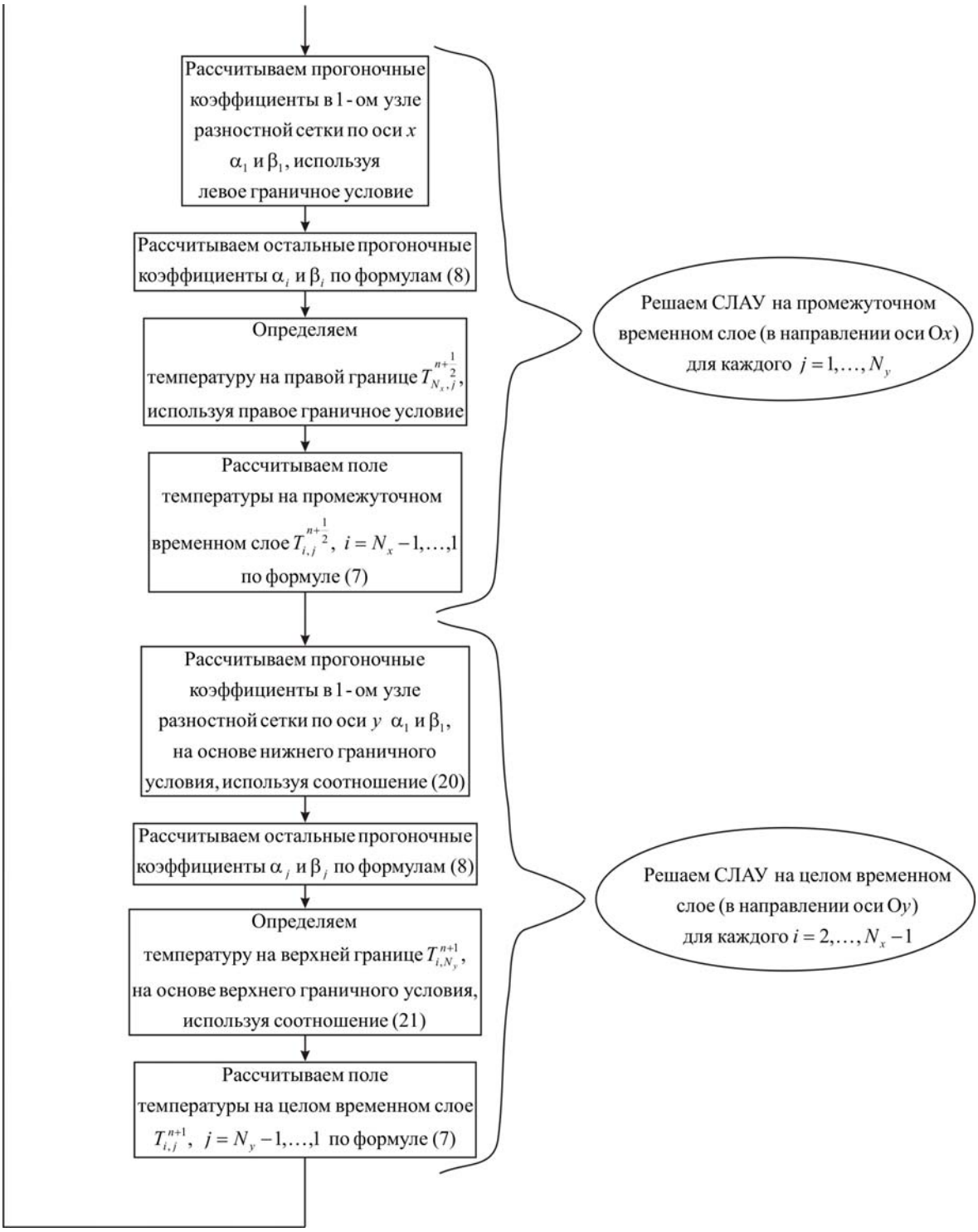
Рассмотрим решение уравнения (32) методом прогонки. Приведем это уравнение к виду $A_i T_{i+1,j}^{n+\frac{1}{2}} - B_i T_{i,j}^{n+\frac{1}{2}} + C_i T_{i-1,j}^{n+\frac{1}{2}} = F_i$. Тогда коэффициенты A_i, B_i, C_i примут вид:

$$A_i = C_i = \frac{\lambda}{h_x^2}, \quad B_i = \frac{2 \cdot \lambda}{h_x^2} + \frac{\rho \cdot c}{\tau}, \quad F_i = -\frac{\rho \cdot c \cdot T_{i,j}^n}{\tau}.$$

Для определения прогоночных коэффициентов по соотношению (8) необходимо найти α_1 и β_1 из левого граничного условия. Далее определяя значение $T_{N_x,j}^{n+\frac{1}{2}}$ из правого граничного условия, находят поле температуры $T_{i,j}^{n+\frac{1}{2}}$ на промежуточном временном слое по формулам (7). После этого приступают к решению уравнения (33). Этапы решения уравнения (33) аналогичны решению уравнения (32).

Блок-схема к рассматриваемой задаче имеет вид:




Рассчитываем прогоночные коэффициенты в 1-ом узле разностной сетки по оси x α_1 и β_1, используя левое граничное условие
Рассчитываем остальные прогоночные коэффициенты α_i и β_i по формулам (8)
Определяем температуру на правой границе $T_{N_x,j}^{n+\frac{1}{2}}$, используя правое граничное условие
Рассчитываем поле температуры на промежуточном временном слое $T_{i,j}^{n+\frac{1}{2}}$, $i = N_x - 1,\ldots,1$ по формуле (7)
Решаем СЛАУ на промежуточном временном слое (в направлении оси Ox) для каждого $j = 1,\ldots,N_y$
Рассчитываем прогоночные коэффициенты в 1-ом узле разностной сетки по оси y α_1 и β_1, на основе нижнего граничного условия, используя соотношение (20)
Рассчитываем остальные прогоночные коэффициенты α_j и β_j по формулам (8)
Определяем температуру на верхней границе T_{i,N_y}^{n+1}, на основе верхнего граничного условия, используя соотношение (21)
Рассчитываем поле температуры на целом временном слое $T_{i,j}^{n+1}$, $j = N_y - 1,\ldots,1$ по формуле (7)
Решаем СЛАУ на целом временном слое (в направлении оси Oy) для каждого $i = 2,\ldots,N_x - 1$

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=102;
type
    vector1=array[1..mf] of real;
    vector2=array[1..mf,1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, Nx, Ny                : integer;
 T                           : vector2;
 alfa, beta                  : vector1;
 ai, bi, ci, fi              : real;
 a, lamda, ro, c             : real;
 hx, hy, tau, t_end, time    : real;
 T0, L, H, Th, Tc            : real;
 f, g                        : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов в пластине по оси x, Nx');
 Readln(Nx);
 Writeln('Введите количество пространственных узлов в пластине по оси y, Ny');
 Readln(Ny);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите длину пластины, L');
 Readln(L);
 Writeln('Введите толщину пластины, H');
 Readln(H);
 Writeln('Введите коэффициент теплопроводности материала пластины, lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите температуру на границе x = 0 области решения, Th');
```

```
 Readln(Th);
 Writeln('Введите температуру на границе x = L области решения, Tc');
 Readln(Tc);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
{определяем расчетные шаги сетки по пространственным
координатам}
 hx:=L/(Nx-1);
 hy:=H/(Ny-1);
{определяем коэффициент температуропроводности}
 a:=lamda/(ro*c);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to Nx do
  for j:= 1 to Ny do
    T[i,j]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{решаем СЛАУ в направлении оси Ox для определения поля
температуры на промежуточном временном слое}
   for j:=1 to Ny do
    begin
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия}
     alfa[1]:=0.0;
     beta[1]:=Th;
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
     for i:= 2 to Nx-1 do
      begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
       ai:=lamda/sqr(hx);
       bi:=2.0*lamda/sqr(hx)+ro*c/tau;
       ci:=lamda/sqr(hx);
```

```
      fi:=-ro*c*T[i,j]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
      alfa[i]:=ai/(bi-ci*alfa[i-1]);
      beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
     end;
{определяем значение температуры на правой границе на основе
правого граничного условия}
    T[Nx,j]:=Tc;
{используя соотношение (7) определяем неизвестное поле температуры
на промежуточном временном слое}
    for i:= Nx-1 downto 1 do
     T[i,j]:=alfa[i]*T[i+1,j]+beta[i];
   end;
{решаем СЛАУ в направлении оси Oy для определения поля
температуры на целом временном слое}
  for i:=2 to Nx-1 do
   begin
{определяем начальные прогоночные коэффициенты на основе нижнего
граничного условия, используя соотношения (20) при условии, что
q1 = 0}
    alfa[1]:=2.0*a*tau/(2.0*a*tau+sqr(hy));
    beta[1]:=sqr(hy)*T[i,1]/(2.0*a*tau+sqr(hy));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
    for j:= 2 to Ny-1 do
     begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
      ai:=lamda/sqr(hy);
      bi:=2.0*lamda/sqr(hy)+ro*c/tau;
      ci:=lamda/sqr(hy);
      fi:=-ro*c*T[i,j]/tau;
{alfa[j], beta[j] – прогоночные коэффициенты}
      alfa[j]:=ai/(bi-ci*alfa[j-1]);
      beta[j]:=(ci*beta[j-1]-fi)/(bi-ci*alfa[j-1]);
     end;
{определяем значение температуры на верхней границе, используя
соотношение (21) при условии, что q2 = 0}
    T[i,Ny]:=(2.0*a*tau*beta[Ny-1]+sqr(hy)*T[i,Ny])/(2.0*a*tau
                *(1.0-alfa[Ny-1])+sqr(hy));
```

```
{используя соотношение (7) определяем неизвестное поле температуры
на промежуточном временном слое}
     for j:= Ny-1 downto 1 do
      T[i,j]:=alfa[j]*T[i,j+1]+beta[j];
    end;
  end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Длина пластины L = ',L:6:4);
 Writeln(f,'Толщина пластины H = ',H:6:4);
 Writeln(f,'Число узлов по пространственной координате x в пластине
Nx = ',Nx);
 Writeln(f,'Число узлов по пространственной координате y в пластине
Ny = ',Ny);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура T0 = ',T0:6:4);
 Writeln(f,'Температура на границе x = 0 области решения Th = ',Th:6:4);
 Writeln(f,'Температура на границе x = L области решения Tc = ',Tc:6:4);
 Writeln(f,'Результат получен с шагом по координате x hx = ',hx:6:4);
 Writeln(f,'Результат получен с шагом по координате y hy = ',hy:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to Nx do
  for j:=1 to Ny do
   writeln(g,'  ',hx*(i-1):10:8,'  ',hy*(j-1):10:8,'   ',T[i,j]:8:5);
 close(g);
end.
```

Получено следующее распределение температуры:

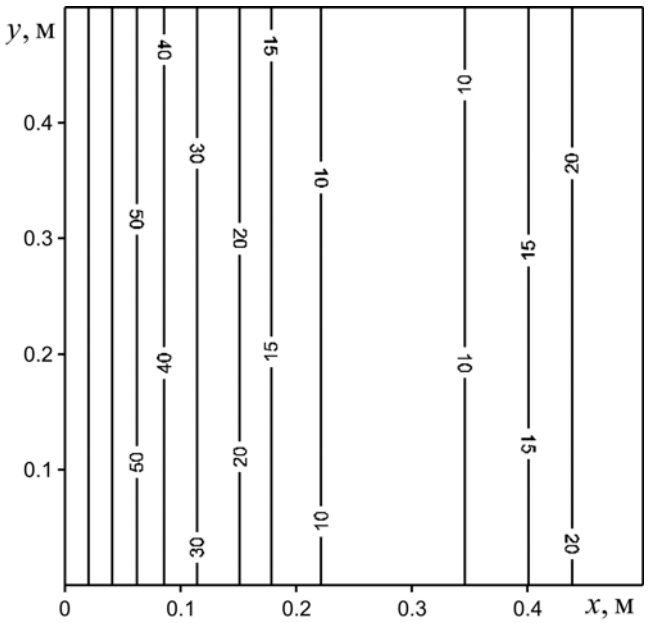


Рис. 19. Изотермы (ºС) в пластине при t = 60 с

2.7. ДВУМЕРНАЯ ЗАДАЧА ТЕПЛОПРОВОДНОСТИ ДЛЯ НЕОДНОРОДНОГО ТЕЛА

Проанализируем процесс теплопереноса в пластине, содержащей два включения (рис. 20). Рассматриваемая задача является учебной, поэтому определяющие размеры l_1, l_2, l_3, l_4, h_1, h_2, h_3, h_4 выбираются таким образом, чтобы используемая разностная сетка была равномерной. Для этой цели в программе в качестве входных параметров будем задавать не линейные размеры, а количество промежутков, характеризующих рассматриваемый отрезок.

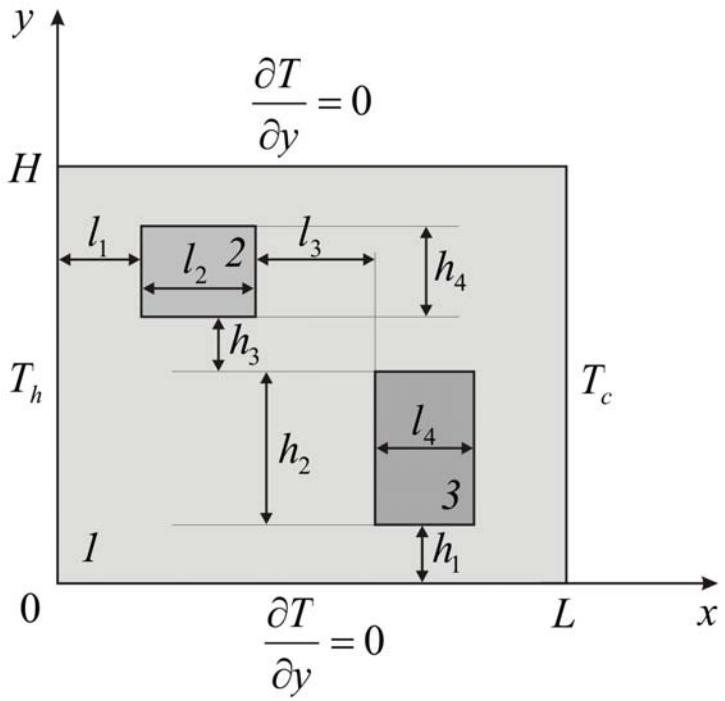


Рис. 20. Область решения

Медная пластина (*1* на рис. 20) с размерами $L = H = 0.5$ м. Материалы включений: сталь (*2* на рис. 20) (λ_2 = 46 Вт/(м·ºС), ρ_2 = 7800 кг/м^3, c_2 = 460 Дж/(кг·ºС)) и железо (*3* на рис. 20) (λ_3 = 71 Вт/(м·ºС), ρ_3 = 7900 кг/м^3, c_3 = 460 Дж/(кг·ºС)). На вертикальных границах области решения поддерживаются постоянные температуры $T_h = 100\ ^0$C при $x = 0$ и $T_c = 0\ ^0$C при $x = L$. Горизонтальные границы являются адиабатическими. Начальная температура области решения $T_0 = 50\ ^0$C.

Математическая постановка задачи будет иметь вид:

$$\begin{cases} \rho_1 c_1 \dfrac{\partial T_1}{\partial t} = \lambda_1 \left(\dfrac{\partial^2 T_1}{\partial x^2} + \dfrac{\partial^2 T_1}{\partial y^2} \right), & \begin{array}{l} 0 < x < l_1, 0 < y < H; \\ l_1 \le x \le l_1 + l_2, 0 < y < h_1 + h_2 + h_3, h_1 + h_2 + h_3 + h_4 < y < H; \\ l_1 + l_2 < x < l_1 + l_2 + l_3, 0 < y < H; \\ l_1 + l_2 + l_3 \le x \le l_1 + l_2 + l_3 + l_4, 0 < y < h_1, h_1 + h_2 < y < H; \\ l_1 + l_2 + l_3 + l_4 < x < L, 0 < y < H; \end{array} \\ \rho_2 c_2 \dfrac{\partial T_2}{\partial t} = \lambda_2 \left(\dfrac{\partial^2 T_2}{\partial x^2} + \dfrac{\partial^2 T_2}{\partial y^2} \right), & l_1 < x < l_1 + l_2, h_1 + h_2 + h_3 < y < h_1 + h_2 + h_3 + h_4; \\ \rho_3 c_3 \dfrac{\partial T_3}{\partial t} = \lambda_3 \left(\dfrac{\partial^2 T_3}{\partial x^2} + \dfrac{\partial^2 T_3}{\partial y^2} \right), & l_1 + l_2 + l_3 < x < l_1 + l_2 + l_3 + l_4, h_1 < y < h_1 + h_2. \end{cases} \tag{34}$$

Начальные и граничные условия запишутся следующим образом:

$$\begin{array}{l} t = 0: \ T = T_0, 0 \le x \le L, \ 0 \le y \le H; \\ x = 0: \ T = T_h, \ t > 0; \\ x = L: \ T = T_c, \ t > 0; \\ y = 0: \ \dfrac{\partial T}{\partial y} = 0, \ t > 0; \\ y = H: \ \dfrac{\partial T}{\partial y} = 0, \ t > 0; \end{array} \tag{35}$$

$$\begin{cases} T_1(t,x,y) = T_2(t,x,y), \\ -\lambda_1 \dfrac{\partial T_1}{\partial x} = -\lambda_2 \dfrac{\partial T_2}{\partial x}, \end{cases} \text{ при } \begin{array}{l} x = l_1, \ h_1 + h_2 + h_3 \le y \le h_1 + h_2 + h_3 + h_4, \\ x = l_1 + l_2, \ h_1 + h_2 + h_3 \le y \le h_1 + h_2 + h_3 + h_4; \end{array}$$

$$\begin{cases} T_1(t,x,y) = T_2(t,x,y), \\ -\lambda_1 \dfrac{\partial T_1}{\partial y} = -\lambda_2 \dfrac{\partial T_2}{\partial y}, \end{cases} \text{ при } \begin{array}{l} y = h_1 + h_2 + h_3, \ l_1 < x < l_1 + l_2, \\ y = h_1 + h_2 + h_3 + h_4, \ l_1 < x < l_1 + l_2; \end{array}$$

$$\begin{cases} T_1(t,x,y) = T_3(t,x,y), \\ -\lambda_1 \dfrac{\partial T_1}{\partial x} = -\lambda_3 \dfrac{\partial T_3}{\partial x}, \end{cases} \text{ при } \begin{array}{l} x = l_1 + l_2 + l_3, \ h_1 \le y \le h_1 + h_2, \\ x = l_1 + l_2 + l_3 + l_4, \ h_1 \le y \le h_1 + h_2; \end{array} \tag{36}$$

$$\begin{cases} T_1(t,x,y) = T_3(t,x,y), \\ -\lambda_1 \dfrac{\partial T_1}{\partial y} = -\lambda_3 \dfrac{\partial T_3}{\partial y}, \end{cases} \text{ при } \begin{array}{l} y = h_1, \ l_1 + l_2 + l_3 < x < l_1 + l_2 + l_3 + l_4, \\ y = h_1 + h_2, \ l_1 + l_2 + l_3 < x < l_1 + l_2 + l_3 + l_4. \end{array}$$

Рассматриваемая задача представляет собой своеобразную комбинацию задач 2.4. и 2.6. Для решения сформулированной задачи (34)–(36) также как и в 2.6. введем равномерную пространственно-временную сетку.

Дискретизацию уравнений (34) будем проводить на основе локально одномерной схемы А.А. Самарского. Решение полученных систем линейных алгебраических уравнений проводится методом прогонки, при этом необходимо учесть, что в пластине *1* присутствуют две неоднородности. Эти включения учитываются в прогоночных коэффициентах на границах сопряжения, а также в коэффициентах канонического уравнения вида (6) в зависимости от материала элемента.

Алгоритм решения рассматриваемой задачи можно представить следующим образом.

Сначала поэтапно решается СЛАУ вида (32), т.е. всю область решения делим на однородные части, например, $0 \le y < h_1,\ h_1 < y < h_1 + h_2,\ h_1 + h_2 < y < h_1 + h_2 + h_3,$ $h_1 + h_2 + h_3 < y < h_1 + h_2 + h_3 + h_4,\ h_1 + h_2 + h_3 + h_4 < y \le H$. И для каждого такого участка решаем систему уравнений вида (32), как для случая одномерной задачи теплопроводности двухслойной пластины (пункт 2.4.). На границах же вида $y = h_1,\ l_1 + l_2 + l_3 < x < l_1 + l_2 + l_3 + l_4$ в качестве теплофизических параметров среды используем среднее арифметическое, в данном случае, $\lambda_{эфф} = \frac{\lambda_1 + \lambda_3}{2},\ \rho_{эфф} = \frac{\rho_1 + \rho_3}{2},\ c_{эфф} = \frac{c_1 + c_3}{2}$. При этом на данной границе в точке $x = l_1 + l_2 + l_3,\ y = h_1$, справа рассматривается среда с параметрами $\lambda_{эфф}, \rho_{эфф}, c_{эфф}$, а слева – λ_1, ρ_1, c_1. После такого решения системы вида (32) переходят к решению системы вида (33), которая разрешается аналогично.

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=101;
type
    vector1=array[1..mf] of real;
    vector2=array[1..mf,1..mf] of real;
```

```
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, Nx, Ny                : integer;
 nx1, nx2, nx3, nx4          : integer;
 ny1, ny2, ny3, ny4          : integer;
 T                           : vector2;
 a1, lamda1, ro1, c1         : real;
 a2, lamda2, ro2, c2         : real;
 a3, lamda3, ro3, c3         : real;
 hx, hy, tau, t_end, time    : real;
 T0, L, H, Th, Tc            : real;
 f, g                        : text;

procedure progonx(j: integer; lamda,ro,c:real; var W:vector2);
{процедура, разрешающая СЛАУ с трехдиагональной матрицей
методом прогонки, в направлении оси Ox
j – номер слоя по оси y вдоль которого происходит решение СЛАУ;
lamda – коэффициент теплопроводности;
ro – плотность;
c – коэффициент теплоемкости;
W – двумерное поле температуры}
var {раздел описания локальных переменных}
  i                 : integer;
  alfa, beta        : vector1;
  ai, bi, ci, fi    : real;
begin
{определяются начальные прогоночные коэффициенты на основе левого
граничного условия}
 alfa[1]:=0.0;
 beta[1]:=Th;
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
 for i:= 2 to Nx-1 do
  begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
   ai:=lamda/sqr(hx);
   bi:=2.0*lamda/sqr(hx)+ro*c/tau;
   ci:=lamda/sqr(hx);
   fi:=-ro*c*W[i,j]/tau;
```

```
{alfa[i], beta[i] – прогоночные коэффициенты}
   alfa[i]:=ai/(bi-ci*alfa[i-1]);
   beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
  end;
{определяем значение температуры на правой границе на основе
правого граничного условия}
 W[Nx,j]:=Tc;
{используя соотношение (7) определяем неизвестное поле
температуры}
 for i:= Nx-1 downto 1 do
  W[i,j]:=alfa[i]*W[i+1,j]+beta[i];
end; {окончание процедуры progonx}

procedure progony(i:integer; lamda,ro,c:real; var W:vector2);
{процедура, разрешающая СЛАУ с трехдиагональной матрицей
методом прогонки, в направлении оси Oy
i – номер слоя по оси x вдоль которого происходит решение СЛАУ;
lamda – коэффициент теплопроводности;
ro – плотность;
c – коэффициент теплоемкости;
W – двумерное поле температуры}
var {раздел описания локальных переменных}
  j                  : integer;
  alfa, beta         : vector1;
  a, ai, bi, ci, fi  : real;
begin
{определяется коэффициент температуропроводности}
 a:=lamda/(ro*c);
{определяются начальные прогоночные коэффициенты на основе
нижнего граничного условия, в соответствии с рассматриваемой
постановкой используется соотношение (20)}
 alfa[1]:=2.0*a*tau/(2.0*a*tau+sqr(hy));
 beta[1]:=sqr(hy)*W[i,1]/(2.0*a*tau+sqr(hy));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
 for j:= 2 to Ny-1 do
  begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
   ai:=lamda/sqr(hy);
   bi:=2.0*lamda/sqr(hy)+ro*c/tau;
```

```
   ci:=lamda/sqr(hy);
   fi:=-ro*c*W[i,j]/tau;
{alfa[j], beta[j] – прогоночные коэффициенты}
   alfa[j]:=ai/(bi-ci*alfa[j-1]);
   beta[j]:=(ci*beta[j-1]-fi)/(bi-ci*alfa[j-1]);
  end;
{определяем значение температуры на верхней границе на основе
верхнего граничного условия, в нашем случае используется
соотношение (21)}
 W[i,Ny]:=(2.0*a*tau*lamda*beta[Ny-1]+lamda*sqr(hy)*W[i,Ny])
              /(lamda*sqr(hy)+2.0*a*tau*lamda*(1.0-alfa[Ny-1]));
{используя соотношение (7) определяем неизвестное поле
температуры}
 for j:= Ny-1 downto 1 do
  W[i,j]:=alfa[j]*W[i,j+1]+beta[j];
end; {окончание процедуры progony}

procedure progonxIV(j,n1,n2:integer; lamda,lamdan,ro,ron,c,cn:real; var
W:vector2);
{процедура, разрешающая СЛАУ с трехдиагональной матрицей
методом прогонки, в направлении оси Ox при наличии включения с
отличающимися теплофизическими характеристиками
j – номер слоя по оси y вдоль которого происходит решение СЛАУ;
n1 – номер узла с которого начинается включение;
n2 – номер узла которым заканчивается включение;
lamda – коэффициент теплопроводности основного материала;
lamdan – коэффициент теплопроводности материала включения;
ro – плотность основного материала;
ron – плотность материала включения;
c – коэффициент теплоемкости основного материала;
cn – коэффициент теплоемкости материала включения;
W – двумерное поле температуры}
var {раздел описания локальных переменных}
  i              : integer;
  alfa, beta     : vector1;
  ai, bi, ci, fi : real;
  a, an          : real;
begin
{определяются коэффициенты температуропроводности}
 a:=lamda/(ro*c);
 an:=lamdan/(ron*cn);
```

```
{определяются начальные прогоночные коэффициенты на основе левого
граничного условия}
 alfa[1]:=0.0;
 beta[1]:=Th;
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8) до включения}
 for i:= 2 to n1 do
  begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
   ai:=lamda/sqr(hx);
   bi:=2.0*lamda/sqr(hx)+ro*c/tau;
   ci:=lamda/sqr(hx);
   fi:=-ro*c*W[i,j]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
   alfa[i]:=ai/(bi-ci*alfa[i-1]);
   beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
  end;
{определяются прогоночные коэффициенты на границе основного
материала и включения}
 alfa[n1+1]:=2.0*a*an*tau*lamdan/(2.0*a*an*tau*(lamdan+lamda
                *(1-alfa[n1]))+sqr(hx)*(a*lamdan+an*lamda));
beta[n1+1]:=(2.0*a*an*tau*lamda*beta[n1]+sqr(hx)*(a*lamdan+an*lamda)
                *W[n1+1,j])/(2.0*a*an*tau*(lamdan+lamda*(1-alfa[n1]))
                +sqr(hx)*(a*lamdan+an*lamda));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8) во включении}
 for i:= n1+2 to n1+n2 do
  begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
   ai:=lamdan/sqr(hx);
   bi:=2.0*lamdan/sqr(hx)+ron*cn/tau;
   ci:=lamdan/sqr(hx);
   fi:=-ron*cn*W[i,j]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
   alfa[i]:=ai/(bi-ci*alfa[i-1]);
   beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
  end;
```

```
{определяются прогоночные коэффициенты на границе включения и
основного материала}
  alfa[n1+n2+1]:=2.0*an*a*tau*lamda/(2.0*an*a*tau*(lamda+lamdan*
                    (1-alfa[n1+n2]))+sqr(hx)*(an*lamda+a*lamdan));
  beta[n1+n2+1]:=(2.0*an*a*tau*lamdan*beta[n1+n2]+sqr(hx)*(a*lamdan
                    +an*lamda)*W[n1+n2+1,j])/(2.0*an*a*tau*(lamda
                    +lamdan*(1-alfa[n1+n2]))+sqr(hx)*(an*lamda+a*lamdan));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8) в основном материале}
 for i:= n1+n2+2 to Nx-1 do
  begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
   ai:=lamda/sqr(hx);
   bi:=2.0*lamda/sqr(hx)+ro*c/tau;
   ci:=lamda/sqr(hx);
   fi:=-ro*c*W[i,j]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
   alfa[i]:=ai/(bi-ci*alfa[i-1]);
   beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
  end;
{определяем значение температуры на правой границе на основе
правого граничного условия}
 W[Nx,j]:=Tc;
{используя соотношение (7) определяем неизвестное поле
температуры}
 for i:= Nx-1 downto 1 do
  W[i,j]:=alfa[i]*W[i+1,j]+beta[i];
end; {окончание процедуры progonxIV}

procedure progonyIV(i,n1,n2:integer; lamda,lamdan,ro,ron,c,cn:real; var
W:vector2);
{процедура, разрешающая СЛАУ с трехдиагональной матрицей
методом прогонки, в направлении оси Oy при наличии включения с
отличающимися теплофизическими характеристиками
i – номер слоя по оси x вдоль которого происходит решение СЛАУ;
n1 – номер узла с которого начинается включение;
n2 – номер узла которым заканчивается включение;
lamda – коэффициент теплопроводности основного материала;
lamdan – коэффициент теплопроводности материала включения;
ro – плотность основного материала;
```

```
ron – плотность материала включения;
c – коэффициент теплоемкости основного материала;
cn – коэффициент теплоемкости материала включения;
W – двумерное поле температуры}
Var {раздел описания локальных переменных}
  j               : integer;
  alfa, beta      : vector1;
  ai, bi, ci, fi  : real;
  a, an           : real;
begin
{определяются коэффициенты температуропроводности}
 a:=lamda/(ro*c);
 an:=lamdan/(ron*cn);
{определяются начальные прогоночные коэффициенты на основе
нижнего граничного условия, в данном случае используем соотношение
(20)}
 alfa[1]:=2.0*a*tau/(2.0*a*tau+sqr(hy));
 beta[1]:=sqr(hy)*W[i,1]/(2.0*a*tau+sqr(hy));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8) до включения}
 for j:= 2 to n1 do
  begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
   ai:=lamda/sqr(hy);
   bi:=2.0*lamda/sqr(hy)+ro*c/tau;
   ci:=lamda/sqr(hy);
   fi:=-ro*c*W[i,j]/tau;
{alfa[j], beta[j] – прогоночные коэффициенты}
   alfa[j]:=ai/(bi-ci*alfa[j-1]);
   beta[j]:=(ci*beta[j-1]-fi)/(bi-ci*alfa[j-1]);
  end;
{определяются прогоночные коэффициенты на границе основного
материала и включения}
 alfa[n1+1]:=2.0*a*an*tau*lamdan/(2.0*a*an*tau*(lamdan+lamda
                *(1-alfa[n1]))+sqr(hy)*(a*lamdan+an*lamda));
 beta[n1+1]:=(2.0*a*an*tau*lamda*beta[n1]+sqr(hy)*(a*lamdan
                +an*lamda)*W[i,n1+1])/(2.0*a*an*tau*(lamdan+lamda
                *(1-alfa[n1]))+sqr(hy)*(a*lamdan+an*lamda));
```

```
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8) во включении}
 for j:= n1+2 to n1+n2 do
  begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
   ai:=lamdan/sqr(hy);
   bi:=2.0*lamdan/sqr(hy)+ron*cn/tau;
   ci:=lamdan/sqr(hy);
   fi:=-ron*cn*W[i,j]/tau;
{alfa[j], beta[j] – прогоночные коэффициенты}
   alfa[j]:=ai/(bi-ci*alfa[j-1]);
   beta[j]:=(ci*beta[j-1]-fi)/(bi-ci*alfa[j-1]);
  end;
{определяются прогоночные коэффициенты на границе включения и
основного материала}
 alfa[n1+n2+1]:=2.0*an*a*tau*lamda/(2.0*an*a*tau*(lamda+lamdan
                    *(1-alfa[n1+n2]))+sqr(hy)*(an*lamda+a*lamdan));
 beta[n1+n2+1]:=(2.0*an*a1*tau*lamdan*beta[n1+n2]+sqr(hy)*(a*lamdan
                    +an*lamda)*W[i,n1+n2+1])/(2.0*an*a*tau*(lamda+lamdan
                    *(1-alfa[n1+n2]))+sqr(hy)*(an*lamda+a*lamdan));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8) в основном материале}
 for j:= n1+n2+2 to Ny-1 do
  begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
   ai:=lamda/sqr(hy);
   bi:=2.0*lamda/sqr(hy)+ro*c/tau;
   ci:=lamda/sqr(hy);
   fi:=-ro*c*W[i,j]/tau;
{alfa[j], beta[j] – прогоночные коэффициенты}
   alfa[j]:=ai/(bi-ci*alfa[j-1]);
   beta[j]:=(ci*beta[j-1]-fi)/(bi-ci*alfa[j-1]);
  end;
{определяем значение температуры на верхней границе на основе
верхнего граничного условия, в данном случае используем соотношение
(21)}
 W[i,Ny]:=(2.0*a*tau*lamda*beta[Ny-1]+lamda*sqr(hy)*W[i,Ny])
            /(lamda*sqr(hy)+2.0*a*tau*lamda*(1.0-alfa[Ny-1]));
```

```
{используя соотношение (7) определяем неизвестное поле
температуры}
 for j:= Ny-1 downto 1 do
  W[i,j]:=alfa[j]*W[i,j+1]+beta[j];
end; {окончание процедуры progonyIV}
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите длину пластины, L');
 Readln(L);
 Writeln('Введите толщину пластины, H');
 Readln(H);
 Writeln('Введите количество пространственных узлов в пластине по оси
x, Nx');
 Readln(Nx);
 Writeln('Введите количество промежутков между границей x=0 и
включением 2 по оси x, nx1');
 Readln(nx1);
 Writeln('Введите количество промежутков во включении 2 по оси x,
nx2');
 Readln(nx2);
 Writeln('Введите количество промежутков между включением 2 и
включением 3 по оси x, nx3');
 Readln(nx3);
 Writeln('Введите количество промежутков во включении 3 по оси x,
nx4');
 Readln(nx4);
 Writeln('Введите количество пространственных узлов в пластине по оси
y, Ny');
 Readln(Ny);
 Writeln('Введите количество промежутков между границей y=0 и
включением 3 по оси y, ny1');
 Readln(ny1);
 Writeln('Введите количество промежутков во включении 3 по оси y,
ny2');
 Readln(ny2);
 Writeln('Введите количество промежутков между включением 3 и
включением 2 по оси y, ny3');
 Readln(ny3);
 Writeln('Введите количество промежутков во включении 2 по оси y,
ny4');
```

```
 Readln(ny4);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите коэффициент теплопроводности материала пластины,
lamda1');
 Readln(lamda1);
 Writeln('Введите плотность материала пластины, ro1');
 Readln(ro1);
 Writeln('Введите теплоемкость материала пластины, c1');
 Readln(c1);
 Writeln('Введите коэффициент теплопроводности материала включения
2, lamda2');
 Readln(lamda2);
 Writeln('Введите плотность материала включения 2, ro2');
 Readln(ro2);
 Writeln('Введите теплоемкость материала включения 2, c2');
 Readln(c2);
 Writeln('Введите коэффициент теплопроводности материала включения
3, lamda3');
 Readln(lamda3);
 Writeln('Введите плотность материала включения 3, ro3');
 Readln(ro3);
 Writeln('Введите теплоемкость материала включения 3, c3');
 Readln(c3);
 Writeln('Введите температуру на границе x=0, Th');
 Readln(Th);
 Writeln('Введите температуру на границе x=L, Tc');
 Readln(Tc);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
{определяем расчетные шаги сетки по пространственным
координатам}
 hx:=L/(Nx-1);
 hy:=H/(Ny-1);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{записываем все входные параметры}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Длина пластины L = ',L:6:4);
```

```
 Writeln(f,'Расстояние между границей x=0 и включением 2 =
',nx1*hx:6:4);
 Writeln(f,'Длина включения 2 = ',nx2*hx:6:4);
 Writeln(f,'Расстояние между включением 2 и включением 3 =
',nx3*hx:6:4);
 Writeln(f,'Длина включения 3 = ',nx4*hx:6:4);
 Writeln(f,'Толщина пластины H = ',H:6:4);
 Writeln(f,'Расстояние между границей y=0 и включением 3 =
',ny1*hy:6:4);
 Writeln(f,'Толщина включения 3 = ',ny2*hy:6:4);
 Writeln(f,'Расстояние между включением 3 и включением 2 =
',ny3*hy:6:4);
 Writeln(f,'Толщина включения 2 = ',ny4*hy:6:4);
 Writeln(f,'Число узлов по пространственной координате x в пластине
Nx = ',Nx);
 Writeln(f,'Число промежутков между границей x=0 и включением 2 =
',nx1);
 Writeln(f,'Число промежутков во включении 2 = ',nx2);
 Writeln(f,'Число промежутков между включением 2 и включением 3 =
',nx3);
 Writeln(f,'Число промежутков во включении 3 = ',nx4);
 Writeln(f,'Число узлов по пространственной координате y в пластине
Ny = ',Ny);
 Writeln(f,'Число промежутков между границей y=0 и включением 3 =
',ny1);
 Writeln(f,'Число промежутков во включении 3 = ',ny2);
 Writeln(f,'Число промежутков между включением 3 и включением 2 =
',ny3);
 Writeln(f,'Число промежутков во включении 2 = ',ny4);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda1
= ',lamda1:6:4);
 Writeln(f,'Плотность материала пластины ro1 = ',ro1:6:4);
 Writeln(f,'Теплоемкость материала пластины c1 = ',c1:6:4);
 Writeln(f,'Коэффициент теплопроводности материала включения 2
lamda2 = ',lamda2:6:4);
 Writeln(f,'Плотность материала включения 2 ro2 = ',ro2:6:4);
 Writeln(f,'Теплоемкость материала включения 2 c2 = ',c2:6:4);
 Writeln(f,'Коэффициент теплопроводности материала включения 3
lamda3 = ',lamda3:6:4);
 Writeln(f,'Плотность материала включения 3 ro3 = ',ro3:6:4);
 Writeln(f,'Теплоемкость материала включения 3 c3 = ',c3:6:4);
```

```
 Writeln(f,'Начальная температура T0 = ',T0:6:4);
 Writeln(f,'Температура на границе x=0 области решения Th = ',Th:6:4);
 Writeln(f,'Температура на границе x=L области решения Tc = ',Tc:6:4);
 Writeln(f,'Результат получен с шагом по координате x hx = ',hx:6:4);
 Writeln(f,'Результат получен с шагом по координате y hy = ',hy:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
{определяем коэффициент температуропроводности}
 a1:=lamda1/(ro1*c1);
 a2:=lamda2/(ro2*c2);
 a3:=lamda3/(ro3*c3);
{определяем поле температуры в начальный момент времени}
 for i:= 1 to Nx do
  for j:= 1 to Ny do
    T[i,j]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{           СЛАУ в направлении оси Ox
   в промежутке между границей y=0 и включением 3}
   for j:=1 to ny1 do
    progonx(j,lamda1,ro1,c1,T);
{на нижней границе включения 3}
progonxIV(ny1+1,nx1+nx2+nx3,nx4,lamda1,0.5*(lamda1+lamda3),ro1,0.5*(
ro1+ro3),c1,0.5*(c1+c3),T);
{во включении 3}
   for j:=ny1+2 to ny1+ny2 do
    progonxIV(j,nx1+nx2+nx3,nx4,lamda1,lamda3,ro1,ro3,c1,c3,T);
{на верхней границе включения 3}
progonxIV(ny1+ny2+1,nx1+nx2+nx3,nx4,lamda1,0.5*(lamda1+lamda3),ro1,
0.5*(ro1+ro3),c1,0.5*(c1+c3),T);
{между включением 3 и включением 2}
   for j:=ny1+ny2+2 to ny1+ny2+ny3 do
    progonx(j,lamda1,ro1,c1,T);
```

```
{на нижней границе включения 2}
progonxIV(ny1+ny2+ny3+1,nx1,nx2,lamda1,0.5*(lamda1+lamda2),ro1,0.5*(
ro1+ro2),c1,0.5*(c1+c2),T);
{во включении 2}
  for j:=ny1+ny2+ny3+2 to ny1+ny2+ny3+ny4 do
   progonxIV(j,nx1,nx2,lamda1,lamda2,ro1,ro2,c1,c2,T);
{на верхней границе включения 2}
progonxIV(ny1+ny2+ny3+ny4+1,nx1,nx2,lamda1,0.5*(lamda1+lamda2),ro1,
0.5*(ro1+ro2),c1,0.5*(c1+c2),T);
{между включением 2 и границей y = H}
  for j:=ny1+ny2+ny3+ny4+2 to Ny do
   progonx(j,lamda1,ro1,c1,T);
{              СЛАУ в направлении оси Oy
      в промежутке между границей x = 0 и включением 2}
  for i:=2 to nx1 do
   progony(i,lamda1,ro1,c1,T);
{на левой границе включения 2}
progonyIV(nx1+1,ny1+ny2+ny3,ny4,lamda1,0.5*(lamda1+lamda2),ro1,0.5*(
ro1+ro2),c1,0.5*(c1+c2),T);
{во включении 2}
  for i:=nx1+2 to nx1+nx2 do
   progonyIV(i,ny1+ny2+ny3,ny4,lamda1,lamda2,ro1,ro2,c1,c2,T);
{на правой границе включения 2}
progonyIV(nx1+nx2+1,ny1+ny2+ny3,ny4,lamda1,0.5*(lamda1+lamda2),ro1,
0.5*(ro1+ro2),c1,0.5*(c1+c2),T);
{в промежутке между включением 2 и включением 3}
  for i:=nx1+nx2+2 to nx1+nx2+nx3 do
   progony(i,lamda1,ro1,c1,T);
{на левой границе включения 3}
progonyIV(nx1+nx2+nx3+1,ny1,ny2,lamda1,0.5*(lamda1+lamda3),ro1,0.5*(
ro1+ro3),c1,0.5*(c1+c3),T);
{во включении 3}
  for i:=nx1+nx2+nx3+2 to nx1+nx2+nx3+nx4 do
   progonyIV(i,ny1,ny2,lamda1,lamda3,ro1,ro3,c1,c3,T);
{на правой границе включения 3}
progonyIV(nx1+nx2+nx3+nx4+1,ny1,ny2,lamda1,0.5*(lamda1+lamda3),ro1,
0.5*(ro1+ro3),c1,0.5*(c1+c3),T);
{в промежутке между включением 3 и границей x = L}
  for i:=nx1+nx2+nx3+nx4+2 to Nx-1 do
   progony(i,lamda1,ro1,c1,T);
 end; {цикл с предусловием окончен}
```

```
{выводим результат в файл}
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to Nx do
  for j:=1 to Ny do
   writeln(g,'  ',hx*(i-1):10:8,'  ',hy*(j-1):10:8,'   ',T[i,j]:8:5);
 close(g);
end.
```

Получено следующее распределение температуры:

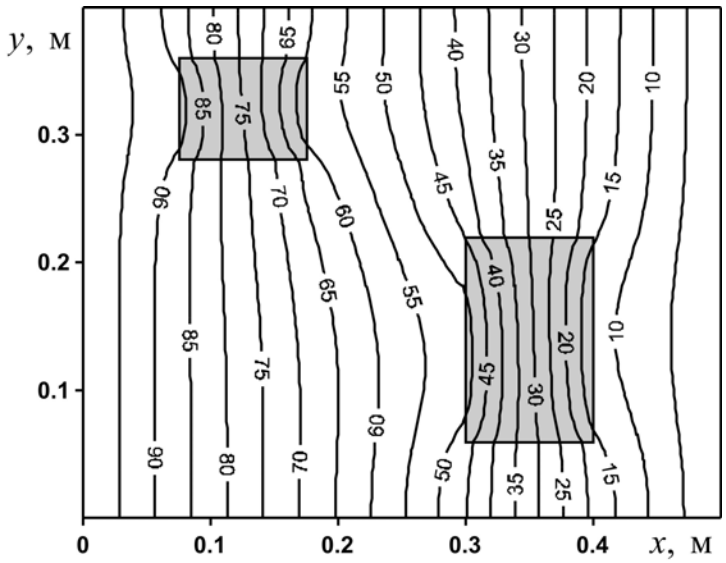


Рис. 21. Изотермы (°С) в пластине при t = 600 с

3. НЕЛИНЕЙНЫЕ ЗАДАЧИ ТЕПЛОПРОВОДНОСТИ
3.1. ОДНОМЕРНОЕ УРАВНЕНИЕ ТЕПЛОПРОВОДНОСТИ С ЗАВИСЯЩИМ ОТ ТЕМПЕРАТУРЫ КОЭФФИЦИЕНТОМ ТЕПЛОПРОВОДНОСТИ

Рассмотренные выше решения для температурных полей в одномерной и двумерной постановке получены при условии постоянства коэффициента теплопроводности. В действительности коэффициент теплопроводности достаточно часто зависит от температуры. Например, для расчета коэффициента теплопроводности диоксида урана UO_2 используется следующая зависимость:

$$\lambda(T) = \frac{5500}{560 + T} + 0.942 \cdot 10^{-10} T^3 . \qquad (37)$$

Эта формула эмпирическая, коэффициенты размерные T, К; λ, Вт/(м · К).

Нелинейное одномерное уравнение теплопроводности в этом случае будет иметь вид:

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\lambda(T) \frac{\partial T}{\partial x} \right), \; 0 < x < L . \tag{38}$$

Для уравнения (38) рассмотрим краевую задачу:

$$\begin{gathered} t = 0 : \; T = T_0 , 0 \le x \le L; \\ x = 0 : \; T = T_h , t > 0; \\ x = L : \; T = T_c , t > 0. \end{gathered} \tag{39}$$

Для того чтобы дать полное математическое описание рассматриваемой задачи, необходимо еще задать физические условия однозначности. Если пластина изготовлена из диоксида урана, то ρ = 10950 кг/м^3, c = 236 Дж/(кг·К). Пластина с размером $L = 0.5$ м. На границах поддерживаются постоянные температуры $T_h = 373$ К при $x = 0$ и $T_c = 363$ К при $x = L$. Начальная температура области решения $T_0 = 323$ К.

Эту задачу также будем решать на равномерной сетке. Аналогично пункту 2.1 пластину разбиваем по толщине на N–1 равных промежутков.

Далее заменим дифференциальные операторы в (38), (39) на их конечно-разностные аналоги. Поскольку отличие заключается в появлении коэффициента теплопроводности, зависящего от температуры, то основной акцент сделаем на аппроксимации диффузионного члена. Рассмотрим сначала явно-неявную схему.

$$\frac{\partial}{\partial x}\left(\lambda(T) \frac{\partial T}{\partial x} \right) = \frac{1}{h}\left(\lambda^n_{i+1/2} \cdot \frac{T^{n+1}_{i+1} - T^{n+1}_i}{h} - \lambda^n_{i-1/2} \cdot \frac{T^{n+1}_i - T^{n+1}_{i-1}}{h} \right),$$

где $\lambda^n_{i+1/2} = \dfrac{\lambda^n_i + \lambda^n_{i+1}}{2}$, $\lambda^n_{i-1/2} = \dfrac{\lambda^n_{i-1} + \lambda^n_i}{2}$.

Таким образом, в результате аппроксимации частных производных соответствующими конечными разностями получаем следующую систему линейных алгебраических уравнений:

$$\rho \cdot c \cdot \frac{T^{n+1}_i - T^n_i}{\tau} = \frac{1}{h}\left(\lambda^n_{i+1/2} \cdot \frac{T^{n+1}_{i+1} - T^{n+1}_i}{h} - \lambda^n_{i-1/2} \cdot \frac{T^{n+1}_i - T^{n+1}_{i-1}}{h} \right), \tag{40}$$

$$i = 2, \ldots, N-1, \;\; n \ge 0,$$

где $\lambda^n_{i+1/2} = \frac{\lambda^n_i + \lambda^n_{i+1}}{2}$, $\lambda^n_{i-1/2} = \frac{\lambda^n_{i-1} + \lambda^n_i}{2}$, при этом λ^n_i вычисляются по формуле (37), например, $\lambda^n_i = \frac{5500}{560 + T^n_i} + 0.942 \cdot 10^{-10} \left(T^n_i\right)^3$. Добавляя к системе (40) конечно-разностные аналоги краевых условий:

$$T^0_i = T_0,\ i = \overline{2, N-1};$$

$$T^n_1 = T_h,\ n > 0;$$

$$T^n_N = T_c,\ n > 0.$$

получим замкнутую разностную задачу.

Полученную систему можно свести к наиболее общему виду:

$$A_i \cdot T^{n+1}_{i+1} - B_i \cdot T^{n+1}_i + C_i \cdot T^{n+1}_{i-1} = F_i,$$

где

$$A_i = C_i = \frac{\lambda^n_{i+1/2}}{h^2},\quad B_i = \frac{\lambda^n_{i+1/2} + \lambda^n_{i-1/2}}{h^2} + \frac{\rho c}{\tau},\quad F_i = -\frac{\rho c}{\tau} T^n_i,$$

$$\lambda^n_{i+1/2} = \frac{\lambda^n_i + \lambda^n_{i+1}}{2},\quad \lambda^n_{i-1/2} = \frac{\lambda^n_{i-1} + \lambda^n_i}{2},\quad \lambda^n_i = \frac{5500}{560 + T^n_i} + 0.942 \cdot 10^{-10} \left(T^n_i\right)^3.$$

Прогоночные коэффициенты находятся по формулам (8). Далее неизвестное поле температуры определяется по выражению (7).

Блок-схема аналогична представленной в пункте 2.1.

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
type
   vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N                          : integer;
 T, alfa, beta                    : vector;
 ai, bi, ci, fi                   : real;
 ro, c, h, tau                    : real;
 Th, T0, Tc, L, t_end, time       : real;
 f, g                             : text;
```

```
function lamda(x:real):real;
{функция вычисления коэффициента теплопроводности по формуле
(31)}
begin
 lamda:=5500/(560+x)+0.942*(1e-10)*x*sqr(x);
end;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите толщину пластины, L');
 Readln(L);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите начальную температуру в К, T0');
 Readln(T0);
 Writeln('Введите температуру в К на границе x=0, Th');
 Readln(Th);
 Writeln('Введите температуру в К на границе x=L, Tc');
 Readln(Tc);
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to N do
  T[i]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
```

```
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия}
  alfa[1]:=0.0;
  beta[1]:=Th;
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
  for i:= 2 to N-1 do
   begin
{ai, bi, ci, fi – коэффициенты канонического представления системы
уравнений с трехдиагональной матрицей}
    ai:=0.5*(lamda(T[i])+lamda(T[i+1]))/sqr(h);
    ci:=0.5*(lamda(T[i-1])+lamda(T[i]))/sqr(h);
    bi:=ai+ci+ro*c/tau;
    fi:=-ro*c*T[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
    alfa[i]:=ai/(bi-ci*alfa[i-1]);
    beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
   end;
{определяем значение температуры на правой границе на основе
правого граничного условия}
  T[N]:=Tc;
{используя соотношение (7) определяем неизвестное поле
температуры}
  for i:= N-1 downto 1 do
   T[i]:=alfa[i]*T[i+1]+beta[i];
 end; {цикл с предусловием окончен}
{Выводим результат в файл}
Assign(f,'res.txt');
Rewrite(f);
Writeln(f,'Толщина пластины L = ',L:6:4);
Writeln(f,'Число узлов по координате N = ',N);
Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
Writeln(f,'Начальная температура  T0 = ',T0:6:4);
Writeln(f,'Температура на границе x = 0, Th = ',Th:6:4);
Writeln(f,'Температура на границе x = L, Tc = ',Tc:6:4);
Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
close(f);
Assign(g,'tempr.txt');
```

```
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):6:3,'   ',T[i]-273:8:5);
 close(g);
end.
```

Получены следующие распределения температуры:

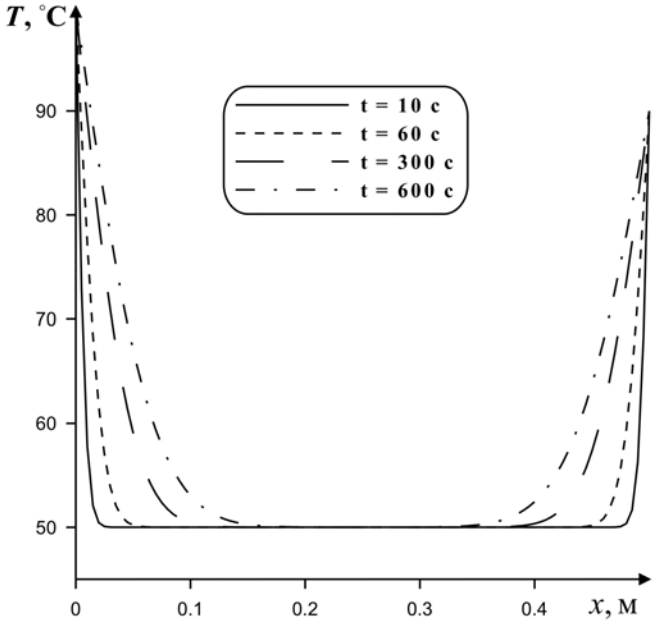


Рис. 22. Распределения температуры по толщине пластины в различные моменты времени

Рассмотрим чисто неявную схему.

$$\frac{\partial}{\partial x}\left(\lambda(T)\frac{\partial T}{\partial x}\right)=\frac{1}{h}\left(\lambda_{i+1/2}^{n+1}\cdot\frac{T_{i+1}^{n+1}-T_i^{n+1}}{h}-\lambda_{i-1/2}^{n+1}\cdot\frac{T_i^{n+1}-T_{i-1}^{n+1}}{h}\right),$$

где $\lambda_{i+1/2}^{n+1}=\dfrac{\lambda_i^{n+1}+\lambda_{i+1}^{n+1}}{2}$, $\lambda_{i-1/2}^{n+1}=\dfrac{\lambda_{i-1}^{n+1}+\lambda_i^{n+1}}{2}$.

Таким образом, в результате аппроксимации частных производных соответствующими конечными разностями получаем следующую систему линейных алгебраических уравнений:

$$\rho\cdot c\cdot\frac{T_i^{n+1}-T_i^n}{\tau}=\frac{1}{h}\left(\lambda_{i+1/2}^{n+1}\cdot\frac{T_{i+1}^{n+1}-T_i^{n+1}}{h}-\lambda_{i-1/2}^{n+1}\cdot\frac{T_i^{n+1}-T_{i-1}^{n+1}}{h}\right),\qquad(41)$$

$$i=2,\ldots,N-1,\quad n\geq 0,$$

где $\lambda_{i+1/2}^{n+1} = \dfrac{\lambda_i^{n+1} + \lambda_{i+1}^{n+1}}{2}$, $\lambda_{i-1/2}^{n+1} = \dfrac{\lambda_{i-1}^{n+1} + \lambda_i^{n+1}}{2}$. при этом λ_i^{n+1} вычисляются по формуле (37), например, $\lambda_i^{n+1} = \dfrac{5500}{560 + T_i^{n+1}} + 0.942 \cdot 10^{-10} \left(T_i^{n+1}\right)^3$. Добавляя к системе (41) конечно-разностные аналоги краевых условий:

$$T_i^0 = T_0,\ i = \overline{2, N-1};$$

$$T_1^n = T_h,\ n > 0;$$

$$T_N^n = T_c,\ n > 0.$$

получим замкнутую разностную задачу.

При этом видно, что полученная система уравнений является нелинейной, поэтому для решения этой системы воспользуемся методом простой итерации. Этот метод заключается в следующем – на каждом шаге по времени мы будем определять поле температуры, до тех пор пока оно не прекратит изменяться с изменением λ, т.е.

$$\rho \cdot c \cdot \frac{T_i^{s+1} - T_i^n}{\tau} = \frac{1}{h}\left(\lambda_{i+1/2}^s \cdot \frac{T_{i+1}^{s+1} - T_i^{s+1}}{h} - \lambda_{i-1/2}^s \cdot \frac{T_i^{s+1} - T_{i-1}^{s+1}}{h}\right), \qquad (42)$$

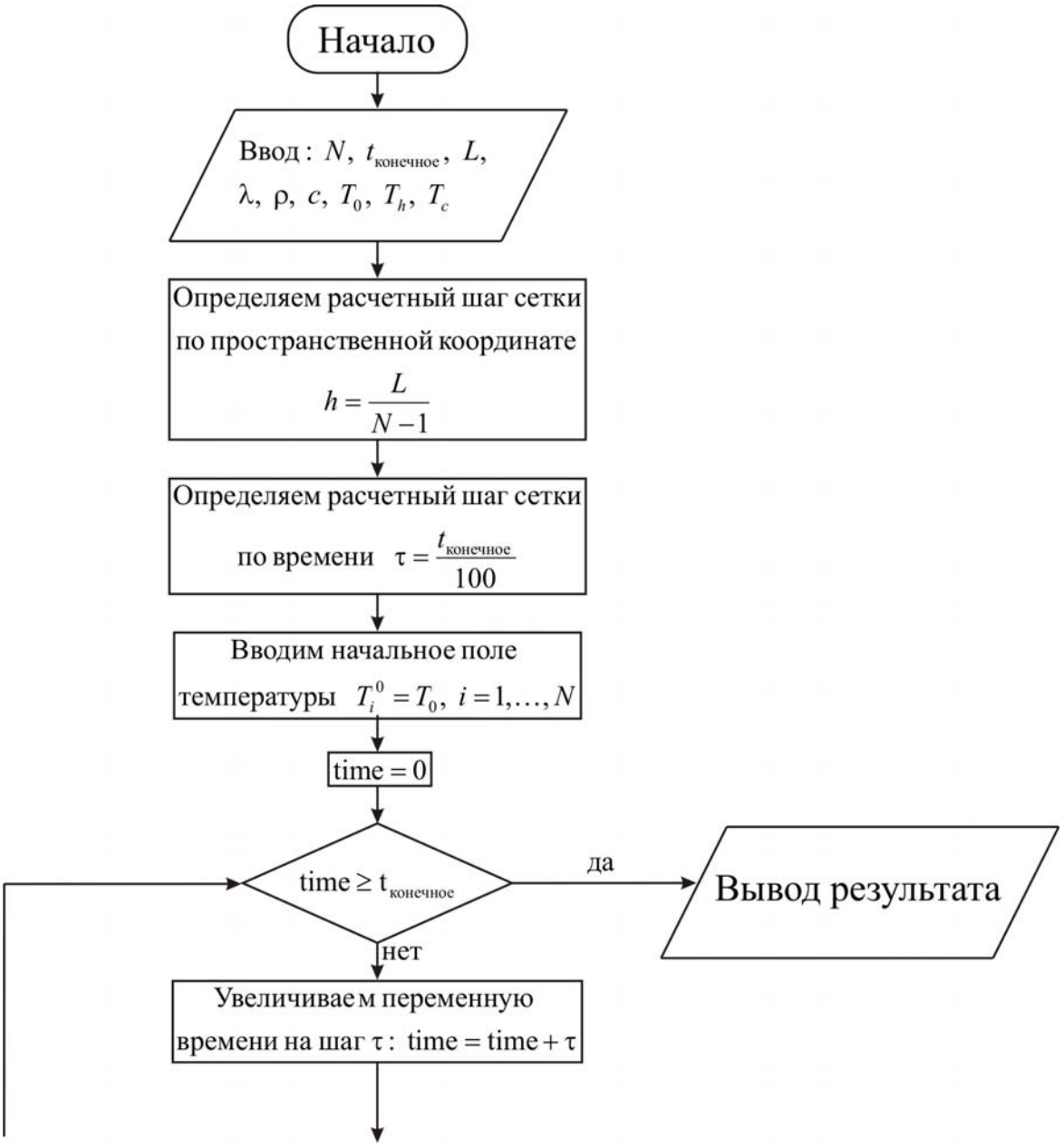
$$i = 2, \ldots, N-1,\quad s, n \geq 0,$$

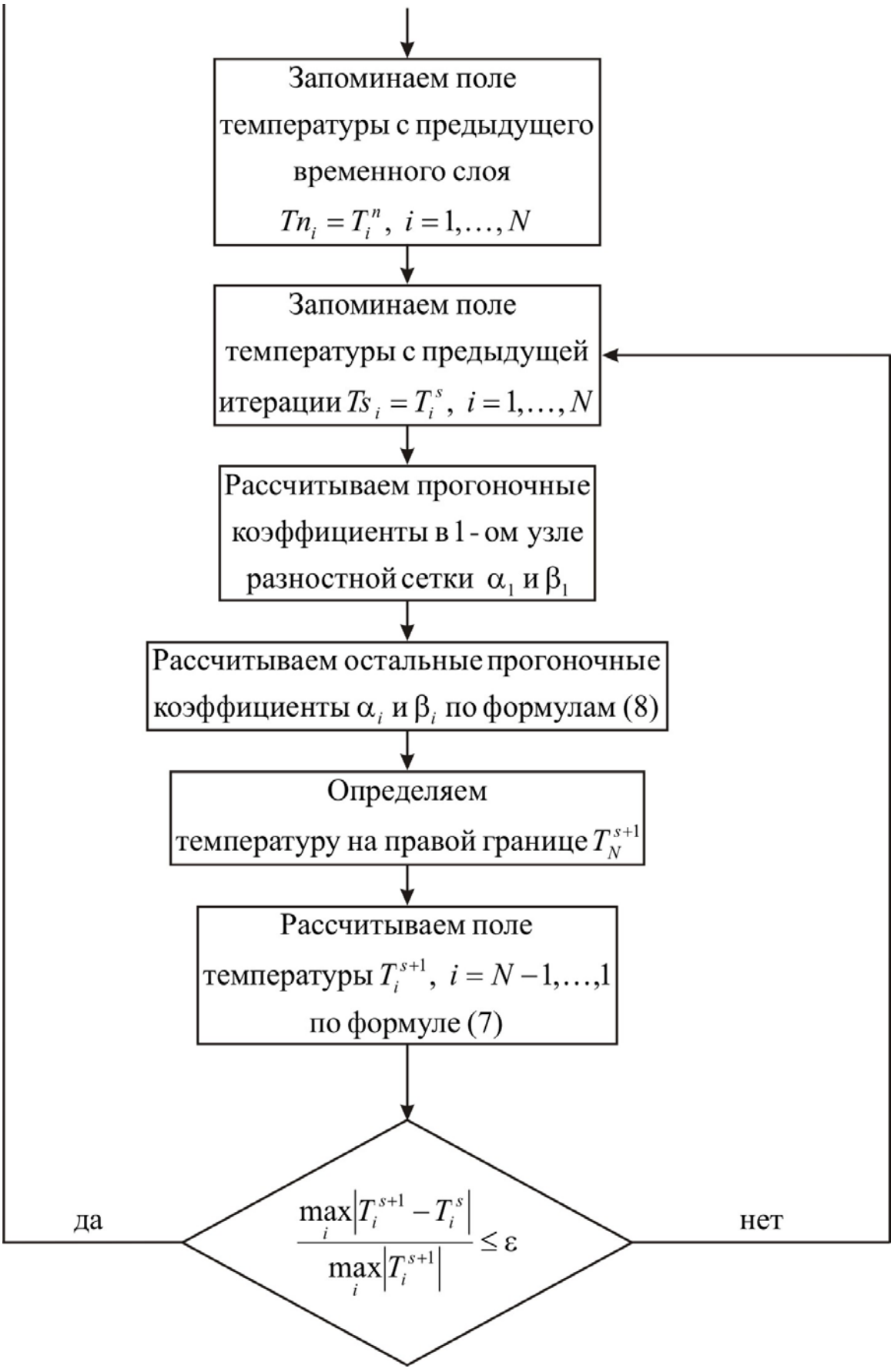
где s – номер итерации. Видно, что система (42) уже является линейной относительно T_i^{s+1}, что позволяет воспользоваться методом прогонки и определить неизвестное поле температуры. Но при этом система (42) решается до тех пор пока поле температуры не перестанет отличаться от предыдущего приближения, т.е. в качестве условия остановки счета на данном временном слое можно использовать следующее соотношение:

$$\frac{\max\limits_i \left|T_i^{s+1} - T_i^s\right|}{\max\limits_i \left|T_i^{s+1}\right|} \leq \varepsilon, \qquad (43)$$

где ε – точность вычислений. Когда условие (43) выполняется, то $T_i^{s+1} = T_i^{n+1}$. В качестве начального приближения можно рассматривать следующее: $T_i^{s=0} = T_i^n$.

Блок-схема к рассматриваемой задаче имеет вид:




Запоминаем поле температуры с предыдущего временного слоя $Tn_i = T_i^n,\ i = 1,\ldots,N$
Запоминаем поле температуры с предыдущей итерации $Ts_i = T_i^s,\ i = 1,\ldots,N$
Рассчитываем прогоночные коэффициенты в 1 - ом узле разностной сетки α_1 и β_1
Рассчитываем остальные прогоночные коэффициенты α_i и β_i по формулам (8)
Определяем температуру на правой границе T_N^{s+1}
Рассчитываем поле температуры $T_i^{s+1},\ i = N-1,\ldots,1$ по формуле (7)
$\frac{\max_i \left| T_i^{s+1} - T_i^s \right|}{\max_i \left| T_i^{s+1} \right|} \le \varepsilon$
да
нет

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
        eps=1e-5;
type
    vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N, s                    : integer;
 T, Ts, Tn, alfa, beta         : vector;
 ai, bi, ci, fi, max1, max2    : real;
 ro, c, h, tau                 : real;
 Th, T0, Tc, L, t_end, time    : real;
 f, g, f1, f2                  : text;
function lamda(x:real):real;
{функция вычисления коэффициента теплопроводности по формуле (31)}
begin
 lamda:=5500/(560+x)+0.942*(1e-10)*x*sqr(x);
end;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите толщину пластины, L');
 Readln(L);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите начальную температуру в К, T0');
 Readln(T0);
 Writeln('Введите температуру в К на границе x=0, Th');
 Readln(Th);
 Writeln('Введите температуру в К на границе x=L, Tc');
 Readln(Tc);
```

```
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{определяем поле температуры в начальный момент времени}
 for i:= 2 to N-1 do
  T[i]:=T0;
{заводим файл содержащий количество итераций на каждом шаге по
времени}
 Assign(f1,'iter.txt');
 Rewrite(f1);
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{запоминаем поле температуры на предыдущем временном слое}
   for i:= 1 to N do
    Tn[i]:=T[i];
   s:=0; {будем считать число итераций}
{цикл с постусловием, позволяющий итерационно вычислять поле
температуры}
   repeat
    inc(s);
{запоминаем поле температуры на предыдущей итерации}
    for i:= 1 to N do
     Ts[i]:=T[i];
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия}
    alfa[1]:=0.0;
    beta[1]:=Th;
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
    for i:= 2 to N-1 do
     begin
{ai, bi, ci, fi – коэффициенты канонического представления системы
уравнений с трехдиагональной матрицей}
      ai:=0.5*(lamda(T[i])+lamda(T[i+1]))/sqr(h);
      ci:=0.5*(lamda(T[i-1])+lamda(T[i]))/sqr(h);
```

```
      bi:=ai+ci+ro*c/tau;
      fi:=-ro*c*Tn[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
      alfa[i]:=ai/(bi-ci*alfa[i-1]);
      beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
     end;
{определяем значение температуры на правой границе на основе
правого граничного условия}
    T[N]:=Tc;
{используя соотношение (7) определяем неизвестное поле
температуры}
    for i:= N-1 downto 1 do
     T[i]:=alfa[i]*T[i+1]+beta[i];
{определяем максимум модуля разности температур на данной и
предыдущей итерации}
    max1:=abs(T[1]-Ts[1]);
    for i:= 2 to N do
     if max1 < abs(T[i]-Ts[i]) then max1:=abs(T[i]-Ts[i]);
{определяем максимальное по модулю значение температуры на данной
итерации}
    max2:=abs(T[1]);
    for i:= 2 to N do
     if max2 < abs(T[i]) then max2:=abs(T[i]);
   until max1/max2<=eps; {цикл с постусловием окончен}
   Writeln(f1,'В момент времени  ',time:6:4,'  проведено ',s,' итераций');
  end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Толщина пластины L = ',L:6:4);
 Writeln(f,'Число узлов по координате N = ',N);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура  T0 = ',T0:6:4);
 Writeln(f,'Температура на границе x = 0, Th = ',Th:6:4);
 Writeln(f,'Температура на границе x = L, Tc = ',Tc:6:4);
 Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
```

```
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):6:3,'   ',T[i]-273:8:5);
 close(g);
 close(f1);
end.
```

Результаты, полученные на основе чисто неявной схемы, полностью совпадают с результатами, полученными по явно-неявной схеме, число итераций при этом не превышает 3.

3.2. ОДНОМЕРНОЕ УРАВНЕНИЕ ТЕПЛОПРОВОДНОСТИ С НЕЛИНЕЙНЫМИ ГРАНИЧНЫМИ УСЛОВИЯМИ (ИЗЛУЧЕНИЕ НА ГРАНИЦЕ)

В качестве примера рассмотрим теплоперенос в бесконечной пластине. Тем самым пренебрегаются два направления переноса тепла, и анализируется одномерное уравнение теплопроводности. На границах области решения будет моделироваться теплообмен за счет конвекции и излучения. Теплоперенос излучением будем рассматривать на основе закона Стефана-Больцмана. Таким образом, сформулированная физическая постановка математически будет выглядеть так:

$$\rho c \frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2}, \quad 0 < x < L; \tag{44}$$

$$t = 0: \; T = T_0, 0 \le x \le L;$$

$$x = 0: -\lambda \frac{\partial T}{\partial x} = \kappa_1 \left(T^{e1} - T\right) + \varepsilon_1 \sigma \left(\left(T^{e1}\right)^4 - T^4\right), \; t > 0, \kappa_1 > 0; \tag{45}$$

$$x = L: \lambda \frac{\partial T}{\partial x} = \kappa_2 \left(T^{e2} - T\right) + \varepsilon_2 \sigma \left(\left(T^{e2}\right)^4 - T^4\right), \; t > 0, \kappa_2 > 0;$$

где $\varepsilon_1, \varepsilon_2$ – приведенная степень черноты, $\sigma = 5.669 \cdot 10^{-8}$ Вт/$\left(\text{м}^2 \cdot \text{К}^4\right)$ – постоянная Стефана-Больцмана.

Основной интерес в сформулированной краевой задаче представляют нелинейные граничные условия.

Проведем дискретизацию нелинейных граничных условий III рода с погрешностью $O(h)$.

Определим первые прогоночные коэффициенты α_1 и β_1 из соотношения $T_1 = \alpha_1 \cdot T_2 + \beta_1$.

Итак, из второго соотношения (45) следует:

$$-\lambda\left.\frac{\partial T}{\partial x}\right|_{x=0}=\kappa_1\left(T^{e1}-T\big|_{x=0}\right)+\varepsilon_1\sigma\left(\left(T^{e1}\right)^4-\left(T\big|_{x=0}\right)^4\right);$$

$$-\lambda\frac{T_2-T_1}{h}=\kappa_1\left(T^{e1}-T_1\right)+\varepsilon_1\sigma\left(\left(T^{e1}\right)^4-\left(T_1\right)^4\right).$$

Введем обозначение $\frac{\kappa_1\cdot h}{\lambda}\equiv \mathrm{Bi}_1$, тогда

$$T_1-T_2=\mathrm{Bi}_1\cdot T^{e1}-\mathrm{Bi}_1\cdot T_1+\frac{\varepsilon_1\sigma h}{\lambda}\left(T^{e1}\right)^4-\frac{\varepsilon_1\delta h}{\lambda}\left(T_1\right)^4;$$

$$T_1=\frac{1}{1+\mathrm{Bi}_1}\cdot T_2+\frac{\mathrm{Bi}_1}{1+\mathrm{Bi}_1}\cdot T^{e1}+\frac{\varepsilon_1\sigma h}{\lambda\left(1+\mathrm{Bi}_1\right)}\left(\left(T^{e1}\right)^4-\left(T_1\right)^4\right);$$

$$\begin{cases}\alpha_1=\dfrac{1}{1+\mathrm{Bi}_1};\\ \beta_1=\dfrac{\mathrm{Bi}_1}{1+\mathrm{Bi}_1}\cdot T^{e1}+\dfrac{\varepsilon_1\sigma h}{\lambda\left(1+\mathrm{Bi}_1\right)}\left(\left(T^{e1}\right)^4-\left(T_1\right)^4\right)\end{cases}$$

или

$$\begin{cases}\alpha_1=\dfrac{\lambda}{\lambda+h\cdot\kappa_1};\\ \beta_1=\dfrac{h\cdot\kappa_1}{\lambda+h\cdot\kappa_1}\cdot T^{e1}+\dfrac{\varepsilon_1\sigma h}{\lambda+h\cdot\kappa_1}\left(\left(T^{e1}\right)^4-\left(T_1\right)^4\right).\end{cases}\qquad(46)$$

Видим, что прогоночный коэффициент β_1 нелинейным образом зависит от температуры на левой границе. Тогда для определения поля температуры необходимо воспользоваться, например, методом простой итерации. Основная идея, которого, заключается в том, чтобы на каждом временном слое расчет поля температуры вести до тех пор, пока не будет выполняться условие, вида: $\left|T_1^{s+1}-T_1^s\right|\le\widetilde{\varepsilon}$, где s – номер итерации, $\widetilde{\varepsilon}$ – точность вычислений.

Правое граничное условие используют для определения температуры T_N.

$$\lambda\left.\frac{\partial T}{\partial x}\right|_{x=L}=\kappa_2\left(T^{e2}-T\big|_{x=L}\right)+\varepsilon_2\sigma\left(\left(T^{e2}\right)^4-\left(T\big|_{x=L}\right)^4\right);$$

$$\lambda\frac{T_N-T_{N-1}}{h}=\kappa_2\left(T^{e2}-T_N\right)+\varepsilon_2\sigma\left(\left(T^{e2}\right)^4-\left(T_N\right)^4\right);$$

т.к. $T_{N-1}=\alpha_{N-1}\cdot T_N+\beta_{N-1}$, то

$$T_N-\alpha_{N-1}\cdot T_N-\beta_{N-1}=\mathrm{Bi}_2\left(T^{e2}-T_N\right)+\frac{\varepsilon_2\sigma h}{\lambda}\left(\left(T^{e2}\right)^4-\left(T_N\right)^4\right);$$

$$T_N = \frac{\beta_{N-1} + \mathrm{Bi}_2 \cdot T^{e2} + \frac{\varepsilon_2 \sigma h}{\lambda}\left(\left(T^{e2}\right)^4 - \left(T_N\right)^4\right)}{1 + \mathrm{Bi}_2 - \alpha_{N-1}}$$

или

$$T_N = \frac{\lambda \cdot \beta_{N-1} + h \cdot \kappa_2 \cdot T^{e2} + \varepsilon_2 \sigma h\left(\left(T^{e2}\right)^4 - \left(T_N\right)^4\right)}{h \cdot \kappa_2 + \lambda \cdot \left(1 - \alpha_{N-1}\right)}. \qquad (47)$$

В результате получили нелинейное уравнение (47) для определения температуры на правой границе. Это уравнение также можно решить наиболее простым методом – методом простых итераций.

Проведем дискретизацию нелинейных граничных условий (45) с погрешностью $O(h^2)$. Предположим, что на границе выполняется уравнение теплопроводности (44). Разложим функцию $T(x)$ в ряд Тейлора в окрестности точки $x = 0$ до членов второго порядка относительно h: $T_2^{n+1} = T_1^{n+1} + h \cdot \left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1} + \frac{h^2}{2} \cdot \left.\frac{\partial^2 T}{\partial x^2}\right|_{x=0}^{n+1}$. Используя соотношение (44) получим:

$$T_2^{n+1} = T_1^{n+1} + h \cdot \left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1} + \frac{\rho c h^2}{2 \cdot \lambda} \cdot \left.\frac{\partial T}{\partial t}\right|_{x=0}^{n+1};$$

$$\left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1} = \frac{T_2^{n+1} - T_1^{n+1}}{h} - \frac{\rho c h}{2 \cdot \lambda} \cdot \left.\frac{\partial T}{\partial t}\right|_{x=0}^{n+1} = \frac{T_2^{n+1} - T_1^{n+1}}{h} - \frac{\rho c h}{2 \cdot \lambda} \cdot \frac{T_1^{n+1} - T_1^n}{\tau}.$$

Из второго соотношения (45):

$$\left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1} = \frac{\kappa_1}{\lambda}\left(T_1^{n+1} - T^{e1}\right) - \frac{\varepsilon_1 \sigma}{\lambda}\left(\left(T^{e1}\right)^4 - \left(T_1^{n+1}\right)^4\right).$$

Приравнивая последние два соотношения, получим:

$$\frac{T_2^{n+1} - T_1^{n+1}}{h} - \frac{\rho c h}{2 \cdot \lambda} \cdot \frac{T_1^{n+1} - T_1^n}{\tau} = \frac{\kappa_1}{\lambda}\left(T_1^{n+1} - T^{e1}\right) - \frac{\varepsilon_1 \sigma}{\lambda}\left(\left(T^{e1}\right)^4 - \left(T_1^{n+1}\right)^4\right);$$

$$T_2^{n+1} - T_1^{n+1} - \frac{\rho c h^2}{2 \cdot \lambda \cdot \tau} \cdot T_1^{n+1} + \frac{\rho c h^2}{2 \cdot \lambda \cdot \tau} \cdot T_1^n = \frac{\kappa_1 h}{\lambda} T_1^{n+1} - \frac{\kappa_1 h}{\lambda} T^{e1} - \frac{\varepsilon_1 \sigma h}{\lambda}\left(\left(T^{e1}\right)^4 - \left(T_1^{n+1}\right)^4\right);$$

$$T_1^{n+1}\left(1 + \frac{\rho c h^2}{2 \cdot \lambda \cdot \tau} + \frac{\kappa_1 h}{\lambda}\right) = T_2^{n+1} + \frac{\rho c h^2}{2 \cdot \lambda \cdot \tau} \cdot T_1^n + \frac{\kappa_1 h}{\lambda} T^{e1} + \frac{\varepsilon_1 \sigma h}{\lambda}\left(\left(T^{e1}\right)^4 - \left(T_1^{n+1}\right)^4\right);$$

$$T_1^{n+1}=\frac{2\cdot\lambda\cdot\tau}{\rho ch^2+2\cdot\tau(\lambda+\kappa_1 h)}T_2^{n+1}+\frac{\rho ch^2}{\rho ch^2+2\cdot\tau(\lambda+\kappa_1 h)}\cdot T_1^n$$
$$+\frac{2\tau\kappa_1 h}{\rho ch^2+2\cdot\tau(\lambda+\kappa_1 h)}T^{e1}+\frac{2\tau\varepsilon_1\sigma h}{\rho ch^2+2\cdot\tau(\lambda+\kappa_1 h)}\left(\left(T^{e1}\right)^4-\left(T_1^{n+1}\right)^4\right).$$

Таким образом,

$$\begin{cases}\alpha_1=\dfrac{2\cdot\lambda\cdot\tau}{\rho ch^2+2\cdot\tau(\lambda+\kappa_1 h)};\\ \beta_1=\dfrac{\rho ch^2}{\rho ch^2+2\cdot\tau(\lambda+\kappa_1 h)}\cdot T_1^n+\dfrac{2\tau\kappa_1 h}{\rho ch^2+2\cdot\tau(\lambda+\kappa_1 h)}T^{e1}\\ +\dfrac{2\tau\varepsilon_1\sigma h}{\rho ch^2+2\cdot\tau(\lambda+\kappa_1 h)}\left(\left(T^{e1}\right)^4-\left(T_1^{n+1}\right)^4\right).\end{cases}\qquad(48)$$

Определим T_N, используя правое граничное условие.

$$T_{N-1}^{n+1}=T_N^{n+1}-h\cdot\left.\frac{\partial T}{\partial x}\right|_{x=L}^{n+1}+\frac{h^2}{2}\cdot\left.\frac{\partial^2 T}{\partial x^2}\right|_{x=L}^{n+1}=T_N^{n+1}-h\cdot\left.\frac{\partial T}{\partial x}\right|_{x=L}^{n+1}+\frac{\rho ch^2}{2\cdot\lambda}\cdot\left.\frac{\partial T}{\partial t}\right|_{x=L}^{n+1};$$

$$T_{N-1}^{n+1}=T_N^{n+1}-\frac{\kappa_2 h}{\lambda}\left(T^{e2}-T_N^{n+1}\right)-\frac{\varepsilon_2\sigma h}{\lambda}\left(\left(T^{e2}\right)^4-\left(T_N^{n+1}\right)^4\right)+\frac{\rho ch^2}{2\cdot\lambda}\cdot\frac{T_N^{n+1}-T_N^n}{\tau}.$$

Таким образом, т.к. $T_{N-1}=\alpha_{N-1}\cdot T_N+\beta_{N-1}$

$$\alpha_{N-1}\cdot T_N^{n+1}+\beta_{N-1}=T_N^{n+1}-\frac{\kappa_2 h}{\lambda}T^{e2}+\frac{\kappa_2 h}{\lambda}T_N^{n+1}+\frac{\rho ch^2}{2\cdot\lambda\cdot\tau}T_N^{n+1}-\frac{\rho ch^2}{2\cdot\lambda\cdot\tau}T_N^n$$
$$-\frac{\varepsilon_2\sigma h}{\lambda}\left(\left(T^{e2}\right)^4-\left(T_N^{n+1}\right)^4\right);$$

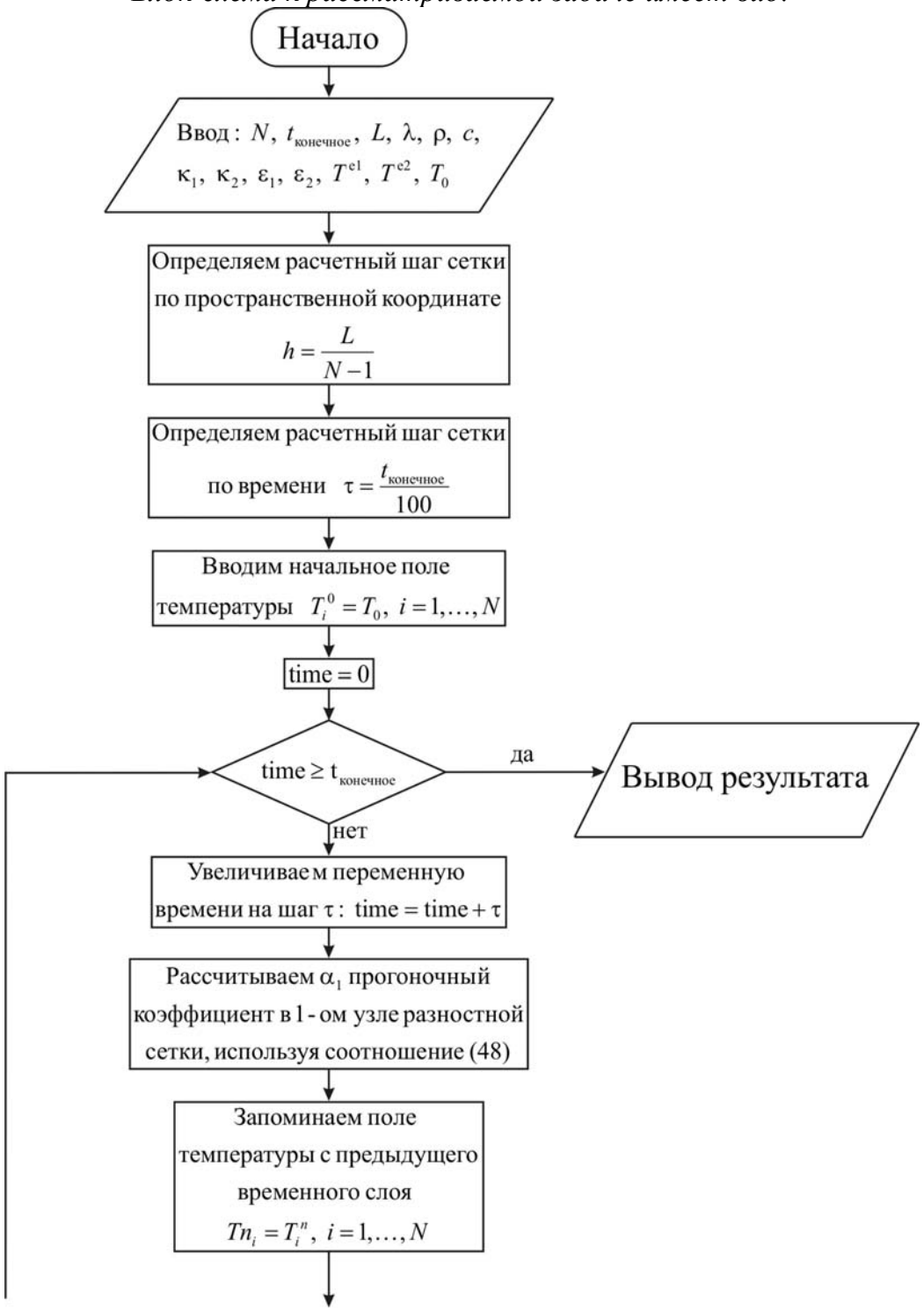
$$T_N^{n+1}\left(1-\alpha_{N-1}+\frac{\kappa_2 h}{\lambda}+\frac{\rho ch^2}{2\cdot\lambda\cdot\tau}\right)=\beta_{N-1}+\frac{\kappa_2 h}{\lambda}T^{e2}+\frac{\rho ch^2}{2\cdot\lambda\cdot\tau}T_N^n$$
$$+\frac{\varepsilon_2\sigma h}{\lambda}\left(\left(T^{e2}\right)^4-\left(T_N^{n+1}\right)^4\right);$$

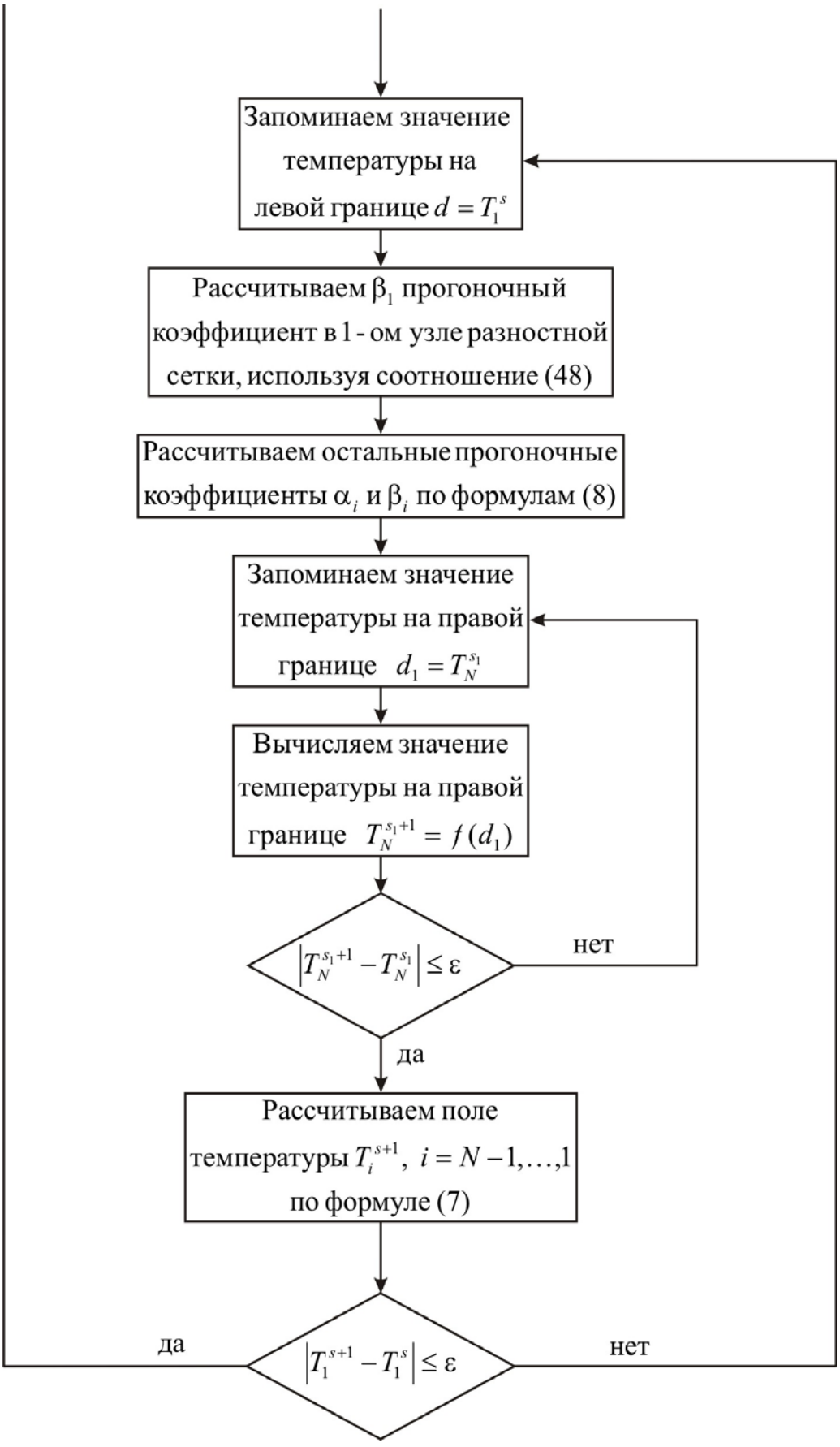
$$T_N^{n+1}=\frac{2\lambda\tau\beta_{N-1}}{2\lambda\tau(1-\alpha_{N-1})+2\tau\kappa_2 h+\rho ch^2}+\frac{2\tau\kappa_2 hT^{e2}}{2\lambda\tau(1-\alpha_{N-1})+2\tau\kappa_2 h+\rho ch^2}$$
$$+\frac{\rho ch^2}{2\lambda\tau(1-\alpha_{N-1})+2\tau\kappa_2 h+\rho ch^2}T_N^n+\frac{2\tau\varepsilon_2\sigma h}{2\lambda\tau(1-\alpha_{N-1})+2\tau\kappa_2 h+\rho ch^2}\left(\left(T^{e2}\right)^4-\left(T_N^{n+1}\right)^4\right).\qquad(49)$$

Определим температурное поле в бетонной пластине через 600, 1800, 3600 и 7200 секунд. Толщина пластины $L=0.3$ м. Начальная температура $T_0=50^0$C. Бетон имеет следующие теплофизические

характеристики λ = 0,9 Вт/(м·ºС), ρ = 2000 кг/м^3, c = 920 Дж/(кг·ºС). На границе $x = 0$ и $x = L$ пластина контактирует с внешней средой ($\kappa_1 = 1000$ Вт/(м$^2 \cdot ^0$ С), $T^{e1} = 30\ ^0$С, $\varepsilon_1 = 0.5$ и $\kappa_2 = 500$ Вт/(м$^2 \cdot ^0$ С), $T^{e1} = 70\ ^0$С, $\varepsilon_2 = 0.2$).

Блок-схема к рассматриваемой задаче имеет вид:




Запоминаем значение температуры на левой границе $d = T_1^s$
Рассчитываем β_1 прогоночный коэффициент в 1 - ом узле разностной сетки, используя соотношение (48)
Рассчитываем остальные прогоночные коэффициенты α_i и β_i по формулам (8)
Запоминаем значение температуры на правой границе $d_1 = T_N^{s_1}$
Вычисляем значение температуры на правой границе $T_N^{s_1+1} = f(d_1)$
$\left|T_N^{s_1+1} - T_N^{s_1}\right| \le \varepsilon$
нет
да
Рассчитываем поле температуры T_i^{s+1}, $i = N-1,\ldots,1$ по формуле (7)
$\left|T_1^{s+1} - T_1^s\right| \le \varepsilon$
да
нет

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
      sigma=5.669e-8;
      eps=1e-5;
type
     vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N                        : integer;
 T, Tn, alfa, beta              : vector;
 ai, bi, ci, fi, d, d1          : real;
 lamda, ro, c, h, tau           : real;
 kapa1, kapa2, Te1, Te2         : real;
 eps1, eps2                     : real;
 T0, L, t_end, time             : real;
 f, g                           : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите толщину пластины, L');
 Readln(L);
 Writeln('Введите коэффициент теплопроводности материала пластины,
lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите коэффициент теплообмена на границе x = 0, kapa1');
 Readln(kapa1);
 Writeln('Введите коэффициент теплообмена на границе x = L, kapa2');
 Readln(kapa2);
 Writeln('Введите температуру внешней среды относительно границы x
= 0, Te1');
 Readln(Te1);
```

```
 Writeln('Введите температуру внешней среды относительно границы x
= L, Te2');
 Readln(Te2);
 Writeln('Введите приведенную степень черноты на границе x = 0, eps1');
 Readln(eps1);
 Writeln('Введите приведенную степень черноты на границе x = L, eps2');
 Readln(eps2);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to N do
  T[i]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{определяем alfa начальный прогоночный коэффициент на основе левого
граничного условия, используя соотношение (48)}
   alfa[1]:=2.0*tau*lamda/(2.0*tau*(lamda+kapa1*h)+ro*c*sqr(h));
{запоминаем поле температуры на предыдущем временном слое}
   for i:=1 to N do
    Tn[i]:=T[i];
{цикл с постусловием, позволяющий итерационно вычислять поле
температуры, вследствие наличия нелинейности в левом граничном
условии}
   repeat
{определяем beta начальный прогоночный коэффициент на основе
левого граничного условия, используя соотношение (48), при этом
начинаем итерационный цикл по левому граничному условию}
    d:=T[1];
    beta[1]:=(ro*c*sqr(h)*Tn[1]+2.0*tau*kapa1*h*Te1+2.0*tau*eps1
              *sigma*h*(sqr(sqr(Te1))-sqr(sqr(d))))/(2.0*tau*(lamda
              +kapa1*h)+ro*c*sqr(h));
```

```
{цикл с параметром для определения прогоночных коэффициентов по формуле (8)}
   for i:= 2 to N-1 do
    begin
{ai, bi, ci, fi – коэффициенты канонического представления системы уравнений с трехдиагональной матрицей}
     ai:=lamda/sqr(h);
     bi:=2.0*lamda/sqr(h)+ro*c/tau;
     ci:=lamda/sqr(h);
     fi:=-ro*c*Tn[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
     alfa[i]:=ai/(bi-ci*alfa[i-1]);
     beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
    end;
{цикл с постусловием, позволяющий итерационно вычислить значение температуры на правой границе, вследствие наличия нелинейности в этом граничном условии}
   repeat
    d1:=T[N];
{определяем значение температуры на правой границе на основе правого граничного условия, используя соотношение (49)}
    T[N]:=(ro*c*sqr(h)*Tn[N]+2.0*tau*(lamda*beta[N-1]+kapa2*h*Te2
            +eps2*sigma*h*(sqr(sqr(Te2))-sqr(sqr(d1)))))
            /(ro*c*sqr(h)+2.0*tau*(lamda*(1-alfa[N-1])+kapa2*h));
   until abs(d1-T[N])<=eps; {значение температуры справа определили}
{используя соотношение (7) определяем неизвестное поле температуры}
   for i:= N-1 downto 1 do
    T[i]:=alfa[i]*T[i+1]+beta[i];
  until abs(d-T[1])<=eps; {значение температуры справа определили}
 end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Толщина пластины L = ',L:6:4);
 Writeln(f,'Число узлов по координате N = ',N);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda = ',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура  T0 = ',T0:6:4);
```

```
 Writeln(f,'Коэффициент теплообмена kapa1 = ',kapa1:6:4);
 Writeln(f,'Коэффициент теплообмена kapa2 = ',kapa2:6:4);
 Writeln(f,'Температура внешней среды Te1 = ',Te1:6:4);
 Writeln(f,'Температура внешней среды Te2 = ',Te2:6:4);
 Writeln(f,'Приведенная степень черноты eps1 = ',eps1:6:4);
 Writeln(f,'Приведенная степень черноты eps2 = ',eps2:6:4);
 Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):10:8,'   ',T[i]:8:5);
 close(g);
end.
```

Получены следующие распределения температуры:

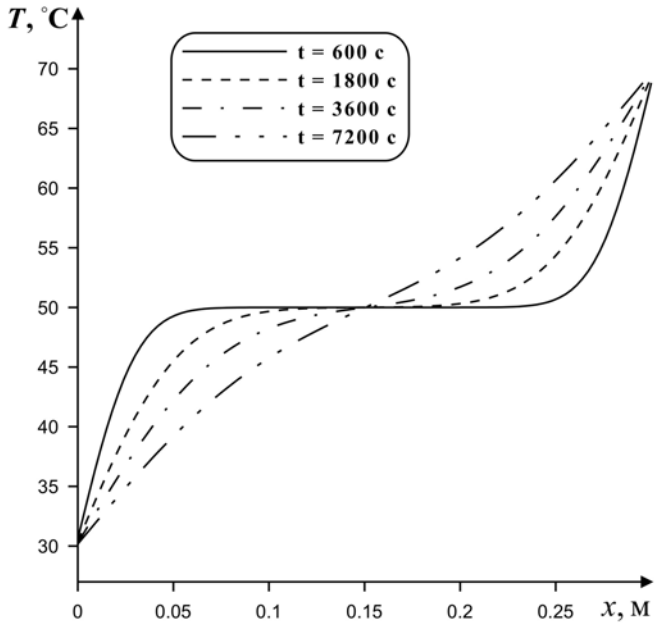


Рис. 23. Распределения температуры по толщине пластины в различные моменты времени

3.3. ОДНОМЕРНОЕ УРАВНЕНИЕ ТЕПЛОПРОВОДНОСТИ С ФАЗОВЫМ ПЕРЕХОДОМ НА ГРАНИЦЕ (ИСПАРЕНИЕ МАТЕРИАЛА)

Рассмотрим процесс теплопроводности в пластине с испарением на двух границах. Математическая постановка задачи:

$$\rho c \frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2},\ 0 < x < L; \tag{50}$$

$$t = 0:\ T = T_0,\ 0 \le x \le L;$$

$$x = 0: -\lambda \frac{\partial T}{\partial x} = q - w_{\text{исп}} \cdot Q_{\text{исп}},\ t > 0; \tag{51}$$

$$x = L:\ \lambda \frac{\partial T}{\partial x} = q - w_{\text{исп}} \cdot Q_{\text{исп}},\ t > 0;$$

где $w_{\text{исп}} = \dfrac{A \cdot \left(P^{н} - P^{*}\right)}{\sqrt{\dfrac{2\pi RT}{M}}}$ – скорость испарения, $P^{н} = P_0 \cdot \exp\left(-\dfrac{Q_{\text{исп}}}{RT}\right)$ – давление насыщенного пара, A – коэффициент аккомодации, P_0 – предэкспонент фазового перехода, $Q_{\text{исп}}$ – энергия активации процесса испарения, $R = 8.31\,\text{Дж}/\left(\text{моль} \cdot \text{К}\right)$ – универсальная газовая постоянная, M – молекулярный вес. В качестве P^{*} рассмотрим атмосферное давление, т.е. $P^{*} = P_{\text{атм}}$.

Основной интерес в сформулированной краевой задаче представляют нелинейные граничные условия.

Проведем дискретизацию нелинейного граничного условия II рода с погрешностью $O(h)$.

Определим первые прогоночные коэффициенты α_1 и β_1 из соотношения $T_1 = \alpha_1 \cdot T_2 + \beta_1$.

Итак, из второго соотношения (51):

$$-\lambda \left.\frac{\partial T}{\partial x}\right|_{x=0} = q - \frac{A \cdot \left(P_0 \cdot \exp\left(-\dfrac{Q_{\text{исп}}}{RT\big|_{x=0}} \right) - P_{\text{атм}} \right) \cdot Q_{\text{исп}}}{\sqrt{\dfrac{2\pi RT\big|_{x=0}}{M}}};$$

$$-\lambda\frac{T_2-T_1}{h}=q-\frac{A\cdot\left(P_0\cdot\exp\left(-\frac{Q_{\text{исп}}}{RT_1}\right)-P_{\text{атм}}\right)\cdot Q_{\text{исп}}}{\sqrt{\frac{2\pi RT_1}{M}}};$$

$$T_1-T_2=\frac{qh}{\lambda}-\frac{A\cdot h\cdot\left(P_0\cdot\exp\left(-\frac{Q_{\text{исп}}}{RT_1}\right)-P_{\text{атм}}\right)\cdot Q_{\text{исп}}}{\lambda\cdot\sqrt{\frac{2\pi RT_1}{M}}};$$

$$T_1=T_2+\frac{qh}{\lambda}-\frac{A\cdot h\cdot\left(P_0\cdot\exp\left(-\frac{Q_{\text{исп}}}{RT_1}\right)-P_{\text{атм}}\right)\cdot Q_{\text{исп}}}{\lambda\cdot\sqrt{\frac{2\pi RT_1}{M}}}.$$

Таким образом,

$$\begin{cases}\alpha_1=1;\\ \beta_1=\dfrac{qh}{\lambda}-\dfrac{A\cdot h\cdot\left(P_0\cdot\exp\left(-\dfrac{Q_{\text{исп}}}{RT_1}\right)-P_{\text{атм}}\right)\cdot Q_{\text{исп}}}{\lambda\cdot\sqrt{\dfrac{2\pi RT_1}{M}}}.\end{cases}\qquad(52)$$

Видим, что прогоночный коэффициент β_1 нелинейным образом зависит от температуры на левой границе. Для определения поля температуры необходимо воспользоваться методом простой итерации, основная идея которого изложена в пункте 3.2.

Правое граничное условие используют для определения температуры T_N.

$$\lambda\left.\frac{\partial T}{\partial x}\right|_{x=L}=q-\frac{A\cdot\left(P_0\cdot\exp\left(-\frac{Q_{\text{исп}}}{RT\big|_{x=L}}\right)-P_{\text{атм}}\right)\cdot Q_{\text{исп}}}{\sqrt{\frac{2\pi RT\big|_{x=L}}{M}}};$$

$$\lambda\frac{T_N-T_{N-1}}{h}=q-\frac{A\cdot\left(P_0\cdot\exp\left(-\frac{Q_{\text{исп}}}{RT\big|_{x=L}}\right)-P_{\text{атм}}\right)\cdot Q_{\text{исп}}}{\sqrt{\frac{2\pi RT\big|_{x=L}}{M}}};$$

т.к. $T_{N-1} = \alpha_{N-1} \cdot T_N + \beta_{N-1}$, то

$$T_N - \alpha_{N-1} \cdot T_N - \beta_{N-1} = \frac{qh}{\lambda} - \frac{A \cdot h \cdot \left(P_0 \cdot \exp\left(-\frac{Q_{\text{исп}}}{RT_N} \right) - P_{\text{атм}} \right) \cdot Q_{\text{исп}}}{\lambda \cdot \sqrt{\frac{2\pi RT_N}{M}}};$$

$$T_N = \frac{\lambda \cdot \beta_{N-1} + h \cdot q}{\lambda \cdot (1 - \alpha_{N-1})} - \frac{A \cdot h \cdot \left(P_0 \cdot \exp\left(-\frac{Q_{\text{исп}}}{RT_N} \right) - P_{\text{атм}} \right) \cdot Q_{\text{исп}}}{\lambda \cdot (1 - \alpha_{N-1}) \cdot \sqrt{\frac{2\pi RT_N}{M}}}. \qquad (53)$$

В результате получили нелинейное уравнение (53) для определения температуры на правой границе. Это уравнение также можно решить методом простой итерации.

Проведем дискретизацию нелинейных граничных условий (51) с погрешностью $O(h^2)$. Предположим, что на границе выполняется уравнение теплопроводности (50). Разложим функцию $T(x)$ в ряд Тейлора в окрестности точки $x = 0$ до членов второго порядка относительно h: $T_2^{n+1} = T_1^{n+1} + h \cdot \left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1} + \frac{h^2}{2} \cdot \left.\frac{\partial^2 T}{\partial x^2}\right|_{x=0}^{n+1}$. Используя соотношение (50) получим:

$$T_2^{n+1} = T_1^{n+1} + h \cdot \left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1} + \frac{\rho c h^2}{2 \cdot \lambda} \cdot \left.\frac{\partial T}{\partial t}\right|_{x=0}^{n+1};$$

$$\left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1} = \frac{T_2^{n+1} - T_1^{n+1}}{h} - \frac{\rho c h}{2 \cdot \lambda} \cdot \left.\frac{\partial T}{\partial t}\right|_{x=0}^{n+1} = \frac{T_2^{n+1} - T_1^{n+1}}{h} - \frac{\rho c h}{2 \cdot \lambda} \cdot \frac{T_1^{n+1} - T_1^n}{\tau}.$$

Из соотношения (51):

$$\left.\frac{\partial T}{\partial x}\right|_{x=0}^{n+1} = -\frac{q}{\lambda} + \frac{A \cdot \left(P_0 \cdot \exp\left(-\frac{Q_{\text{исп}}}{RT_1^{n+1}} \right) - P_{\text{атм}} \right) \cdot Q_{\text{исп}}}{\lambda \cdot \sqrt{\frac{2\pi RT_1^{n+1}}{M}}}.$$

Приравнивая последние два соотношения, получим:

$$\frac{T_2^{n+1}-T_1^{n+1}}{h}-\frac{\rho ch}{2\cdot\lambda}\cdot\frac{T_1^{n+1}-T_1^{n}}{\tau}=-\frac{q}{\lambda}+\frac{A\cdot\left(P_0\cdot\exp\left(-\frac{Q_{\text{исп}}}{RT_1^{n+1}}\right)-P_{\text{атм}}\right)\cdot Q_{\text{исп}}}{\lambda\cdot\sqrt{\frac{2\pi RT_1^{n+1}}{M}}};$$

$$T_2^{n+1}-T_1^{n+1}-\frac{\rho ch^2}{2\cdot\lambda\cdot\tau}\cdot T_1^{n+1}+\frac{\rho ch^2}{2\cdot\lambda\cdot\tau}\cdot T_1^{n}=-\frac{qh}{\lambda}+$$

$$\frac{A\cdot h\cdot\left(P_0\cdot\exp\left(-\frac{Q_{\text{исп}}}{RT_1^{n+1}}\right)-P_{\text{атм}}\right)\cdot Q_{\text{исп}}}{\lambda\cdot\sqrt{\frac{2\pi RT_1^{n+1}}{M}}};$$

$$T_1^{n+1}\left(1+\frac{\rho ch^2}{2\cdot\lambda\cdot\tau}\right)=T_2^{n+1}+\frac{\rho ch^2}{2\cdot\lambda\cdot\tau}\cdot T_1^{n}+\frac{qh}{\lambda}-$$

$$\frac{A\cdot h\cdot\left(P_0\cdot\exp\left(-\frac{Q_{\text{исп}}}{RT_1^{n+1}}\right)-P_{\text{атм}}\right)\cdot Q_{\text{исп}}}{\lambda\cdot\sqrt{\frac{2\pi RT_1^{n+1}}{M}}};$$

$$T_1^{n+1}=\frac{2\lambda\tau}{\rho ch^2+2\tau\lambda}T_2^{n+1}+\frac{\rho ch^2T_1^{n}+2\tau qh}{\rho ch^2+2\tau\lambda}-$$

$$\frac{2\tau Ah\cdot\left(P_0\cdot\exp\left(-\frac{Q_{\text{исп}}}{RT_1^{n+1}}\right)-P_{\text{атм}}\right)\cdot Q_{\text{исп}}}{\left(\rho ch^2+2\tau\lambda\right)\cdot\sqrt{\frac{2\pi RT_1^{n+1}}{M}}}.$$

Таким образом,

$$\begin{cases}\alpha_1=\dfrac{2\lambda\tau}{\rho ch^2+2\tau\lambda};\\[2ex]\beta_1=\dfrac{\rho ch^2T_1^{n}+2\tau qh}{\rho ch^2+2\tau\lambda}-\dfrac{2\tau Ah\cdot\left(P_0\cdot\exp\left(-\frac{Q_{\text{исп}}}{RT_1^{n+1}}\right)-P_{\text{атм}}\right)\cdot Q_{\text{исп}}}{\left(\rho ch^2+2\tau\lambda\right)\cdot\sqrt{\frac{2\pi RT_1^{n+1}}{M}}}.\end{cases}\qquad(54)$$

Определим T_N, используя правое граничное условие.

$$T_{N-1}^{n+1} = T_N^{n+1} - h \cdot \left.\frac{\partial T}{\partial x}\right|_{x=L}^{n+1} + \frac{h^2}{2} \cdot \left.\frac{\partial^2 T}{\partial x^2}\right|_{x=L}^{n+1} = T_N^{n+1} - h \cdot \left.\frac{\partial T}{\partial x}\right|_{x=L}^{n+1} + \frac{\rho c h^2}{2 \cdot \lambda} \cdot \left.\frac{\partial T}{\partial t}\right|_{x=L}^{n+1};$$

$$T_{N-1}^{n+1} = T_N^{n+1} - \frac{qh}{\lambda} + \frac{A \cdot h \cdot \left(P_0 \cdot \exp\left(-\frac{Q_{\text{исп}}}{R T_N^{n+1}} \right) - P_{\text{атм}} \right) \cdot Q_{\text{исп}}}{\lambda \cdot \sqrt{\frac{2\pi R T_N^{n+1}}{M}}} + \frac{\rho c h^2}{2 \cdot \lambda} \cdot \frac{T_N^{n+1} - T_N^n}{\tau}.$$

Таким образом, т.к. $T_{N-1} = \alpha_{N-1} \cdot T_N + \beta_{N-1}$

$$\alpha_{N-1} \cdot T_N^{n+1} + \beta_{N-1} = T_N^{n+1} - \frac{qh}{\lambda} + \frac{A \cdot h \cdot \left(P_0 \cdot \exp\left(-\frac{Q_{\text{исп}}}{R T_N^{n+1}} \right) - P_{\text{атм}} \right) \cdot Q_{\text{исп}}}{\lambda \cdot \sqrt{\frac{2\pi R T_N^{n+1}}{M}}} +$$

$$\frac{\rho c h^2}{2 \cdot \lambda \cdot \tau} T_N^{n+1} - \frac{\rho c h^2}{2 \cdot \lambda \cdot \tau} T_N^n;$$

$$T_N^{n+1} \left(1 - \alpha_{N-1} + \frac{\rho c h^2}{2 \cdot \lambda \cdot \tau} \right) = \beta_{N-1} + \frac{qh}{\lambda} + \frac{\rho c h^2}{2 \cdot \lambda \cdot \tau} T_N^n -$$

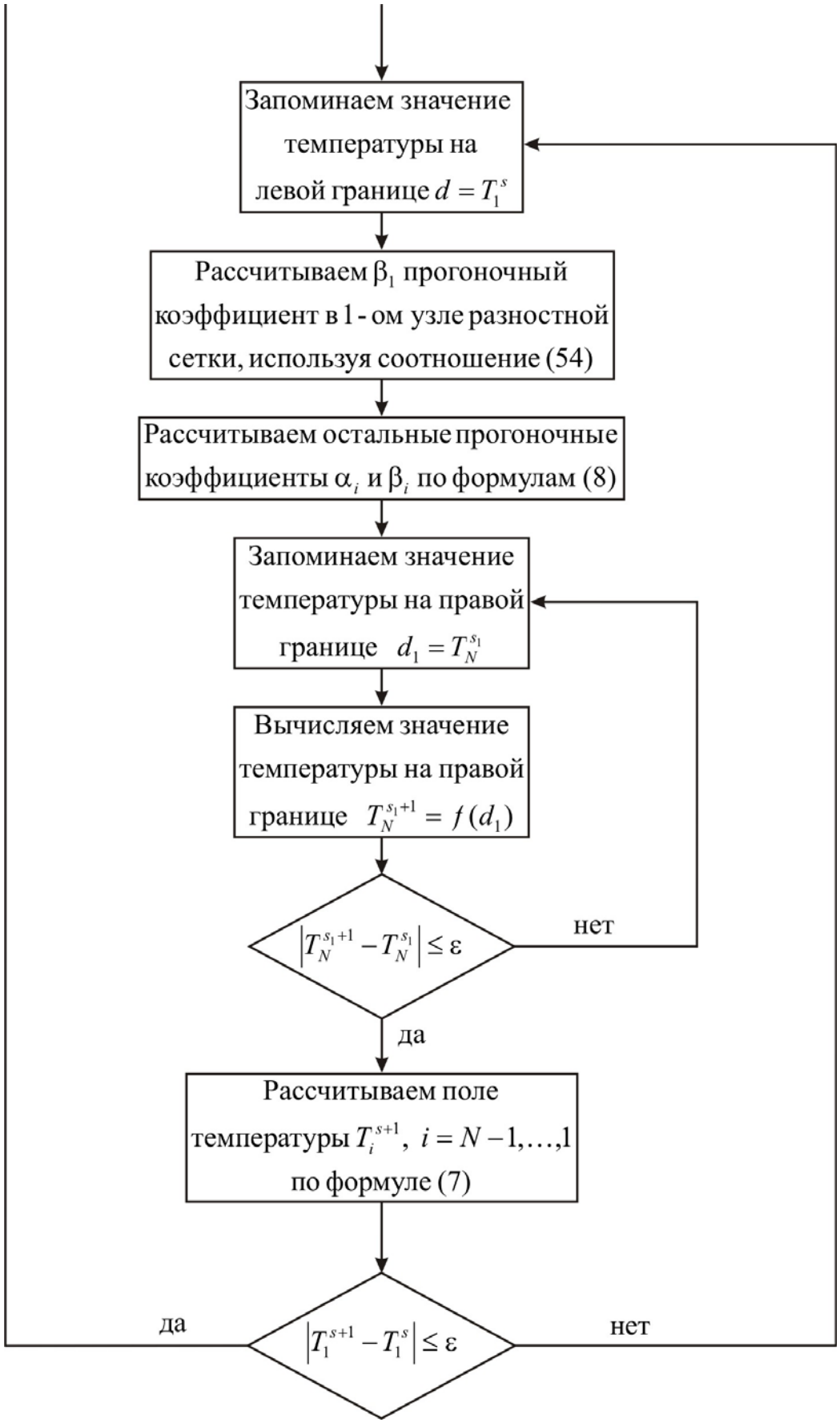
$$\frac{A \cdot h \cdot \left(P_0 \cdot \exp\left(-\frac{Q_{\text{исп}}}{R T_N^{n+1}} \right) - P_{\text{атм}} \right) \cdot Q_{\text{исп}}}{\lambda \cdot \sqrt{\frac{2\pi R T_N^{n+1}}{M}}};$$

$$T_N^{n+1} = \frac{2\lambda\tau\beta_{N-1} + 2\tau q h + \rho c h^2 T_N^n}{2\lambda\tau(1 - \alpha_{N-1}) + \rho c h^2} - \frac{2\tau A h \cdot \left(P_0 \cdot \exp\left(-\frac{Q_{\text{исп}}}{R T_N^{n+1}} \right) - P_{\text{атм}} \right) \cdot Q_{\text{исп}}}{\left(2\lambda\tau(1 - \alpha_{N-1}) + \rho c h^2 \right) \cdot \sqrt{\frac{2\pi R T_N^{n+1}}{M}}}. \quad (55)$$

Проанализируем процесс сублимации трифторид алюминия AlF_3. Определим температурное поле в бетонной пластине через 60, 120, 300 секунд. Толщина образца $L = 0.3$ м. Начальная температура $T_0 = 1273$ K. AlF_3 имеет следующие теплофизические характеристики λ = 60 Вт/(м·K), ρ = 3070 кг/м^3, c = 1260 Дж/(кг·K), M = 0.084 кг/моль. $A = 0.1, P_0 = 10^5, q = 10^6$ Вт/м^2, $Q_{\text{исп}} = 3.8 \cdot 10^4$ Дж/кг.

Блок-схема к рассматриваемой задаче имеет вид:




Запоминаем значение температуры на левой границе $d = T_1^s$
Рассчитываем β_1 прогоночный коэффициент в 1 - ом узле разностной сетки, используя соотношение (54)
Рассчитываем остальные прогоночные коэффициенты α_i и β_i по формулам (8)
Запоминаем значение температуры на правой границе $d_1 = T_N^{s_1}$
Вычисляем значение температуры на правой границе $T_N^{s_1+1} = f(d_1)$
$\left|T_N^{s_1+1} - T_N^{s_1}\right| \le \varepsilon$
нет
да
Рассчитываем поле температуры T_i^{s+1}, $i = N-1,\ldots,1$ по формуле (7)
$\left|T_1^{s+1} - T_1^{s}\right| \le \varepsilon$
да
нет

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
       R=8.31;
      eps=1e-5;
      Patm=101325;
type
    vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N                    : integer;
 T, Tn, alfa, beta          : vector;
 ai, bi, ci, fi, d, d1      : real;
 lamda, ro, c, h, tau       : real;
 q, A, qev, P0, M           : real;
 T0, L, t_end, time         : real;
 f, g                       : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите толщину пластины, L');
 Readln(L);
 Writeln('Введите коэффициент теплопроводности материала пластины, lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите коэффициент аккомодации, A');
 Readln(A);
 Writeln('Введите плотность теплового потока на границе, q');
 Readln(q);
 Writeln('Введите тепловой эффект фазового перехода, qev');
 Readln(qev);
```

```
 Writeln('Введите молярную массу, M');
 Readln(M);
 Writeln('Введите предэкспонент, P0');
 Readln(P0);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/1000.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to N do
  T[i]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{определяем alfa начальный прогоночный коэффициент на основе левого
граничного условия, используя соотношение (54)}
   alfa[1]:=2.0*tau*lamda/(2.0*tau*lamda+ro*c*sqr(h));
{запоминаем поле температуры на предыдущем временном слое}
   for i:=1 to N do
    Tn[i]:=T[i];
{цикл с постусловием, позволяющий итерационно вычислять поле
температуры, вследствие наличия нелинейности в левом граничном
условии}
   repeat
{определяем beta начальный прогоночный коэффициент на основе
левого граничного условия, используя соотношение (54), при этом
начинаем итерационный цикл по левому граничному условию}
    d:=T[1];
    beta[1]:=(ro*c*sqr(h)*Tn[1]+2.0*tau*h*q)/(ro*c*sqr(h)+2.0*tau
                *lamda)-2.0*tau*A*h*qev*(P0*exp(-qev/(R*d))-Patm)
                /((2.0*tau*lamda+ro*c*sqr(h))*sqrt(2.0*pi*R*d/M));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
    for i:= 2 to N-1 do
     begin
```

```
{ai, bi, ci, fi – коэффициенты канонического представления системы
уравнений с трехдиагональной матрицей}
      ai:=lamda/sqr(h);
      bi:=2.0*lamda/sqr(h)+ro*c/tau;
      ci:=lamda/sqr(h);
      fi:=-ro*c*Tn[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
      alfa[i]:=ai/(bi-ci*alfa[i-1]);
      beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
     end;
{цикл с постусловием, позволяющий итерационно вычислить значение
температуры на правой границе, вследствие наличия нелинейности в
этом граничном условии}
    repeat
     d1:=T[N];
{определяем значение температуры на правой границе на основе
правого граничного условия, используя соотношение (55)}
     T[N]:=(ro*c*sqr(h)*Tn[N]+2.0*tau*(lamda*beta[N-1]+h*q))/(ro
               *c*sqr(h)+2.0*tau*lamda*(1-alfa[N-1]))-2.0*tau*A*h*qev
               *(P0*exp(-qev/(R*d1))-Patm)/((ro*c*sqr(h)+2.0*tau*lamda
               *(1-alfa[N-1]))*sqrt(2.0*pi*R*d1/M));
    until abs(d1-T[N])<=eps; {значение температуры справа определили}
{используя соотношение (7) определяем неизвестное поле
температуры}
    for i:= N-1 downto 1 do
     T[i]:=alfa[i]*T[i+1]+beta[i];
   until abs(d-T[1])<=eps; {значение температуры слева определили}
  end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Толщина пластины L = ',L:6:4);
 Writeln(f,'Число узлов по координате N = ',N);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Коэффициент аккомодации A = ',A:6:4);
 Writeln(f,'Плотность теплового потока на границе q = ',q:6:4);
 Writeln(f,'Тепловой эффект фазового перехода qev = ',qev:6:4);
 Writeln(f,'Молярная масса M = ',M:6:4);
```

```
 Writeln(f,'Предэкспонент P0 = ',P0:6:4);
 Writeln(f,'Начальная температура  T0 = ',T0:6:4);
 Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):10:8,'   ',(T[i]-273):8:5);
 close(g);
end.
```

Получены следующие распределения температуры:

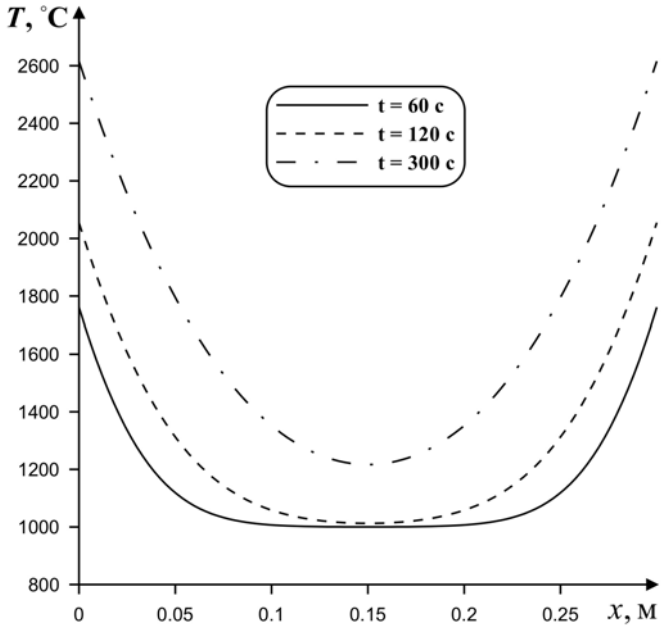


Рис. 24. Распределения температуры по толщине пластины в различные моменты времени

3.4. ОДНОМЕРНОЕ УРАВНЕНИЕ ТЕПЛОПРОВОДНОСТИ С ХИМИЧЕСКОЙ РЕАКЦИЕЙ В МАТЕРИАЛЕ (ТЕРМИЧЕСКОЕ РАЗЛОЖЕНИЕ)

Рассмотрим теплоперенос в бесконечной пластине, подверженной термическому разложению (например, полимерный материал). На границах осуществляется теплообмен с окружающей средой. Математическая постановка сформулированной задачи:

$$\rho c\frac{\partial T}{\partial t}=\lambda\frac{\partial^2 T}{\partial x^2}+q_{\text{хим}}k_0\rho\exp\left(-\frac{E}{RT}\right),\ 0<x<L; \tag{56}$$

$$t=0:\ T=T_0,\ 0\le x\le L;$$

$$x=0:-\lambda\frac{\partial T}{\partial x}=\kappa\left(T^{\text{e}}-T\right),\ t>0; \tag{57}$$

$$x=L:\lambda\frac{\partial T}{\partial x}=\kappa\left(T^{\text{e}}-T\right),\ t>0;$$

где $q_{\text{хим}}$ – тепловой эффект химической реакции, k_0 – предэкспонент химической реакции, E – энергия активации химической реакции, $R=8.31\ \text{Дж}/\left(\text{моль}\cdot\text{К}\right)$ – универсальная газовая постоянная.

Для решения сформулированной краевой задачи применим метод конечных разностей на основе неявной четырехточечной схемы совместно с методом простой итерации. В результате аппроксимации частных производных получаем следующую систему уравнений:

$$\rho\cdot c\cdot\frac{T_i^{n+1}-T_i^n}{\tau}=\lambda\cdot\left(\frac{T_{i+1}^{n+1}-2\cdot T_i^{n+1}+T_{i-1}^{n+1}}{h^2}\right)+q_{\text{хим}}k_0\rho\exp\left(-\frac{E}{RT_i^{n+1}}\right), \tag{58}$$

$$i=2,\ldots,N-1,\quad n\ge 0.$$

Полученную систему можно свести к наиболее общему виду:

$$A_i\cdot T_{i+1}^{n+1}-B_i\cdot T_i^{n+1}+C_i\cdot T_{i-1}^{n+1}=F_i,$$

где

$$A_i=C_i=\frac{\lambda}{h^2},\quad B_i=\frac{2\cdot\lambda}{h^2}+\frac{\rho c}{\tau},\quad F_i=-\frac{\rho c}{\tau}T_i^n-q_{\text{хим}}k_0\rho\exp\left(-\frac{E}{RT_i^{n+1}}\right).$$

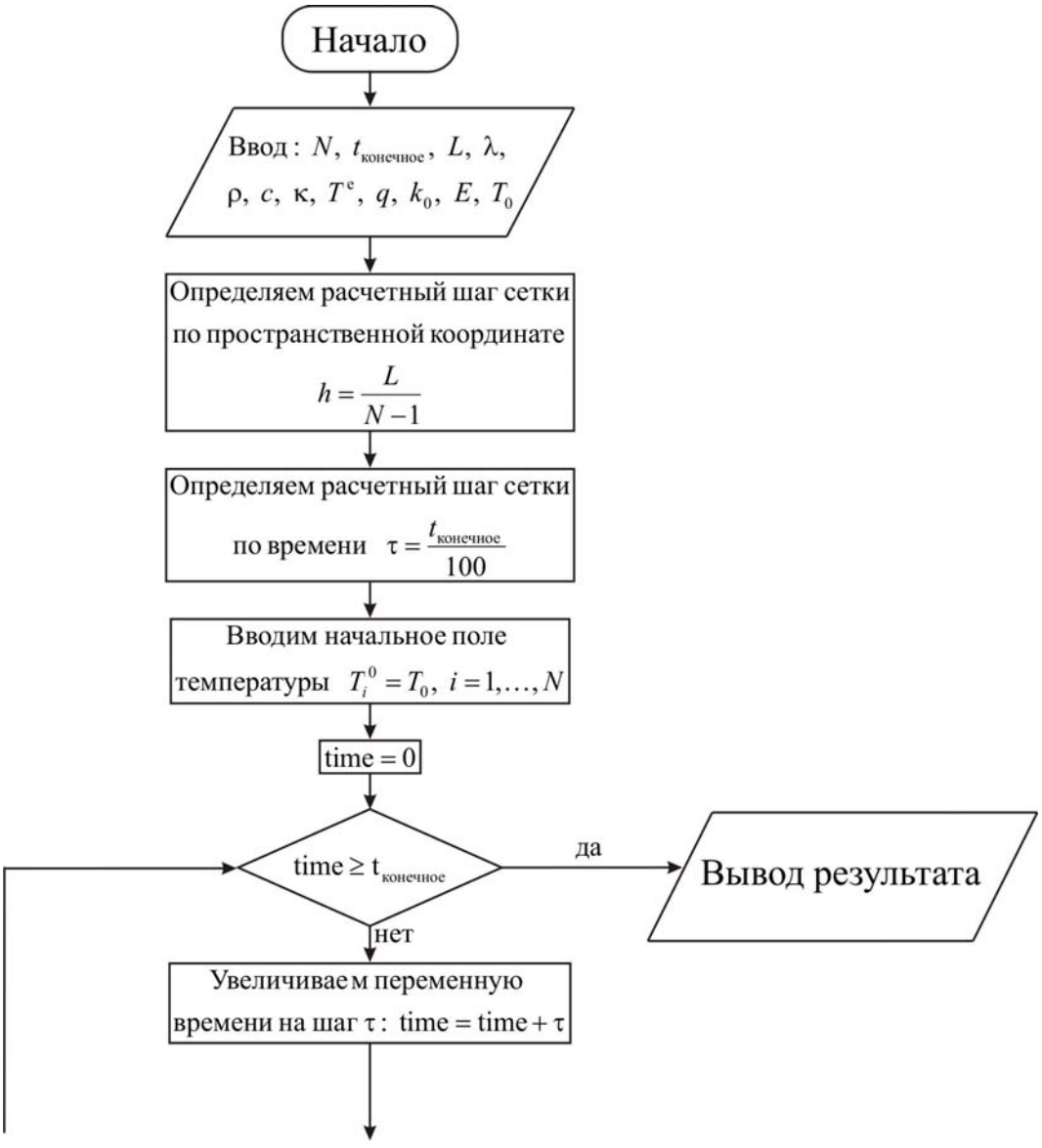
Коэффициент F_i зависит от температуры на новом временном слое, поэтому необходимо воспользоваться методом простой итерации. Основная идея, которого будет заключаться в определении поля температуры на каждом временном слое до тех пор, пока максимальная разность между локальными значениями температуры на данной и на предыдущей итерации не будет минимальна или:

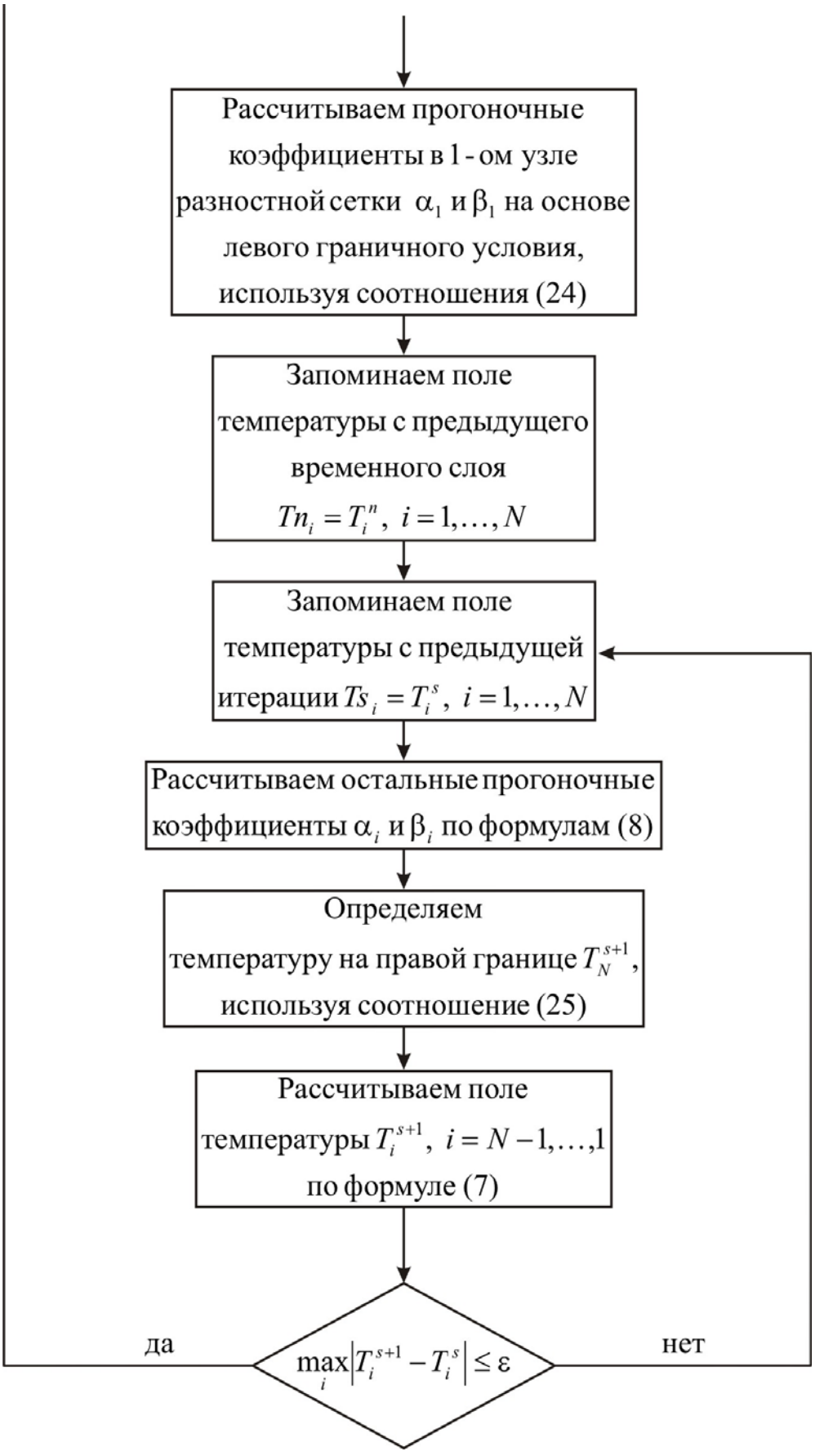
$$\max_i \left| T_i^{s+1} - T_i^s \right| \le \varepsilon ,$$

где ε – точность вычислений.

Определим температурное поле в пластине через 600, 1800 и 3600 секунд. Толщина пластины $L = 0.2$ м. Начальная температура $T_0 = 298$ K. Материал пластины – полимер со следующими теплофизическими характеристиками λ = 0,7 Вт/(м·К), ρ = 1500 кг/м^3, c = 750 Дж/(кг·К). На границах $x = 0$ и $x = L$ пластина контактирует с окружающей средой ($\text{к} = 40$ Вт/(м^2 · К), $T^{\text{e}} = 243$ K). $q_{\text{хим}} = 10^3$ Вт/кг, $k_0 = 3 \cdot 10^4$, $E = 8 \cdot 10^4$ Дж/моль.

Блок-схема к рассматриваемой задаче имеет вид:




Рассчитываем прогоночные коэффициенты в 1 - ом узле разностной сетки α_1 и β_1 на основе левого граничного условия, используя соотношения (24)
Запоминаем поле температуры с предыдущего временного слоя $Tn_i = T_i^n,\ i = 1,\ldots,N$
Запоминаем поле температуры с предыдущей итерации $Ts_i = T_i^s,\ i = 1,\ldots,N$
Рассчитываем остальные прогоночные коэффициенты α_i и β_i по формулам (8)
Определяем температуру на правой границе T_N^{s+1}, используя соотношение (25)
Рассчитываем поле температуры $T_i^{s+1},\ i = N-1,\ldots,1$ по формуле (7)
$\max_i \left|T_i^{s+1} - T_i^s\right| \le \varepsilon$
да
нет

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
        eps=1e-5;
        R=8.31;
type
    vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, N                    : integer;
 T, Ts, Tn, alfa, beta      : vector;
 ai, bi, ci, fi, max        : real;
 lamda, ro, c, h, tau       : real;
 kapa, Te, q, k0, E         : real;
 T0, L, t_end, time         : real;
 f, g                       : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите толщину пластины, L');
 Readln(L);
 Writeln('Введите коэффициент теплопроводности материала пластины,
lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите коэффициент теплообмена с окружающей средой,
kapa');
 Readln(kapa);
 Writeln('Введите температуру окружающей среды, Te');
 Readln(Te);
 Writeln('Введите тепловой эффект химической реакции, q');
 Readln(q);
 Writeln('Введите предэкспонент, k0');
```

```
 Readln(k0);
 Writeln('Введите энергию активации химической реакции, E');
 Readln(E);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to N do
  T[i]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия, используя соотношения (24)}
   alfa[1]:=2.0*tau*lamda/(2.0*tau*(lamda+kapa*h)+ro*c*sqr(h));
   beta[1]:=(ro*c*sqr(h)*T[1]+2.0*tau*kapa*h*Te)/(2.0*tau
              *(lamda+kapa*h)+ro*c*sqr(h));
{запоминаем поле температуры на предыдущем временном слое}
   for i:=1 to N do
    Tn[i]:=T[i];
{цикл с постусловием, позволяющий итерационно вычислять поле
температуры, вследствие наличия нелинейности в самом уравнении
теплопроводности (56)}
   repeat
{запоминаем поле температуры на предыдущей итерации}
    for i:=1 to N do
     Ts[i]:=T[i];
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
    for i:= 2 to N-1 do
     begin
{ai, bi, ci, fi – коэффициенты канонического представления системы
уравнений с трехдиагональной матрицей}
      ai:=lamda/sqr(h);
```

```
      bi:=2.0*lamda/sqr(h)+ro*c/tau;
      ci:=lamda/sqr(h);
      fi:=-ro*c*Tn[i]/tau-q*k0*ro*exp(-E/(R*T[i]));
{alfa[i], beta[i] – прогоночные коэффициенты}
      alfa[i]:=ai/(bi-ci*alfa[i-1]);
      beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
     end;
{определяем значение температуры на правой границе на основе
правого граничного условия, используя соотношение (25)}
     T[N]:=(ro*c*sqr(h)*Tn[N]+2.0*tau*(lamda*beta[N-1]+kapa*h*Te))
              /(ro*c*sqr(h)+2.0*tau*(lamda*(1-alfa[N-1])+kapa*h));
{используя соотношение (7) определяем неизвестное поле
температуры}
    for i:= N-1 downto 1 do
     T[i]:=alfa[i]*T[i+1]+beta[i];
{определяем максимум модуля разности температур на данной и
предыдущей итерации}
    max:=abs(T[1]-Ts[1]);
    for i:=2 to N do
     if max < abs(T[i]-Ts[i]) then max:=abs(T[i]-Ts[i]);
   until max<=eps; {поле температуры на данном временном слое
определили}
  end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Толщина пластины L = ',L:6:4);
 Writeln(f,'Число узлов по координате N = ',N);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура  T0 = ',T0:6:4);
 Writeln(f,'Коэффициент теплообмена kapa = ',kapa:6:4);
 Writeln(f,'Температура внешней среды Te = ',Te:6:4);
 Writeln(f,'Тепловой эффект химической реакции q = ',q:6:4);
 Writeln(f,'Предэкспонент k0 = ',k0:6:4);
 Writeln(f,'Энергия активации химической реакции E = ',E:6:4);
 Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
```

```
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):10:8,'   ',(T[i]-273):8:5);
 close(g);
end.
```

Получены следующие распределения температуры:

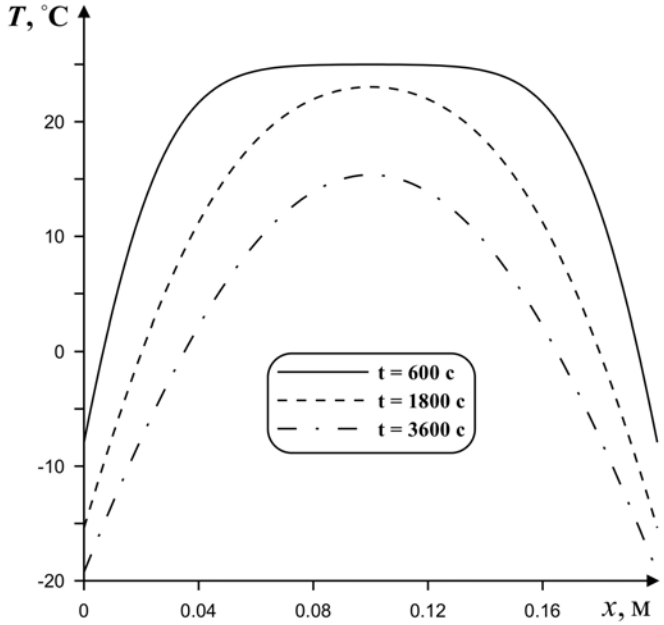


Рис. 25. Распределения температуры по толщине пластины в различные моменты времени

3.5. ОДНОМЕРНОЕ УРАВНЕНИЕ ТЕПЛОПРОВОДНОСТИ С ПОДВИЖНОЙ ГРАНИЦЕЙ (ПРОМЕРЗАНИЕ ВЛАЖНОГО ГРУНТА)

Постановка задачи. Влажный грунт (рис. 26) находится в талом состоянии и имеет начальную постоянную температуру T_0. В начальный момент времени на поверхности грунта внезапно устанавливается некоторая температура T_c, которая ниже температуры замерзания $T_з$. В результате образуется промерзший слой переменной толщины $\xi = f(t)$. Нижняя подвижная граница его всегда имеет температуру замерзания $T_з$. На этой границе происходит переход из одного агрегатного состояния в другое, на что требуется теплота

перехода $Q_{\text{ф}}, (\text{Дж}/\text{кг})$. Таким образом, верхняя граница $(x = \xi)$ талой зоны имеет постоянную температуру замерзания, а нижняя граница $(x = L)$ – некоторую постоянную температур грунта на большой его глубине. Коэффициенты переноса промерзшей и талой зон различны. Предполагается, что перенос тепла в грунте происходит только вследствие теплопроводности.

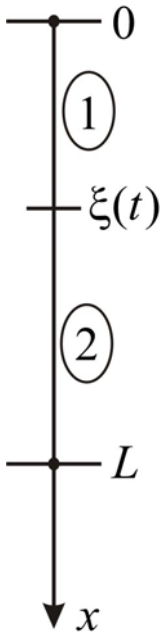


Рис. 26. Область решения

На основе всего вышесказанного математическая постановка задачи примет вид:

$$\begin{cases} \dfrac{\partial T_1}{\partial t} = a_1 \dfrac{\partial^2 T_1}{\partial x^2}, \;\; 0 < x < \xi(t), \; t > 0; \\ \dfrac{\partial T_2}{\partial t} = a_2 \dfrac{\partial^2 T_2}{\partial x^2}, \;\; \xi(t) < x < L, \; t > 0; \end{cases} \tag{59}$$

$$t = 0: \; T(x) = T_0, \; 0 \le x \le L;$$

$$x = 0: \; T(t) = T_c, \; t > 0;$$

$$x = L: \; \frac{\partial T}{\partial x} = 0, \; t > 0; \tag{60}$$

$$x = \xi(t): \begin{cases} T_1 = T_2 = T_з; \\ \lambda_1 \dfrac{\partial T_1}{\partial x} - \lambda_2 \dfrac{\partial T_2}{\partial x} = Q_{\text{ф}} \rho w \dfrac{d\xi}{dt}; \end{cases}$$

где ρ – плотность грунта, $\left(\text{кг}/\text{м}^3\right)$; w – влажность грунта (масса влаги в единице массы абсолютно сухого грунта), $\left(\text{кг}/\text{кг}\right)$.

Постоянные температуры, участвующие в постановке задачи: $T_c < T_з < T_0$.

Сформулированную задачу будем решать методом ловли фронта в узел пространственной сетки. Для этого вводится равномерная пространственная сетка:

$$x_i = (i-1)\cdot h, \quad i = 1,\ldots,N;$$

$$x_1 = 0,\ldots,x_N = L;$$

$$h = \frac{L}{N-1}.$$

Также вводится неравномерная временная сетка:

$$t_{n+1} = t_n + \tau_{n+1}, \quad n = 0,1,\ldots,M-1;$$

$$t_0 = 0,\ldots,t_M = t_{\text{конечное}};$$

$$\tau_{n+1} > 0.$$

Следует выбирать шаг по времени τ_{n+1}, $n = 0,1,\ldots,M-1$ таким, чтобы за этот временной промежуток (от t_n до t_{n+1}) граница фазового перехода сдвинулась ровно на один шаг пространственной сетки, тогда можно записать:

$$\frac{d\xi}{dt} \approx \frac{h}{\tau_{n+1}}.$$

Рассмотрим разностную схему в первой части грунта. Для дискретизации первого уравнения системы (59) воспользуемся неявной четырехточечной схемой.

$$\frac{\partial T_1}{\partial t} = a_1 \frac{\partial^2 T_1}{\partial x^2}, \quad 0 < x < \xi(t), \; t > 0;$$

$$\frac{T_{1,i}^{n+1} - T_{1,i}^{n}}{\tau_{n+1}} = a_1 \frac{T_{1,i+1}^{n+1} - 2T_{1,i}^{n+1} + T_{1,i-1}^{n+1}}{h^2}, \quad i = 2,\ldots,i^* - 1;$$

$$T_1\big|_{i=1} = T_c;$$

$$T_1\big|_{i=i^*} = T_з;$$

где $i = i^*$ – граница фазового перехода.

Полученную систему можно свести к наиболее общему виду:

$$A_i \cdot T_{1,i+1}^{n+1} - B_i \cdot T_{1,i}^{n+1} + C_i \cdot T_{1,i-1}^{n+1} = F_i,$$

где

$$A_i = C_i = \frac{a_1}{h^2}, \quad B_i = \frac{2 \cdot a_1}{h^2} + \frac{1}{\tau_{n+1}}, \quad F_i = -\frac{T_{1,i}^n}{\tau_{n+1}}.$$

Прогоночные коэффициенты находятся по формулам (8). Далее неизвестное поле температуры определяется по выражению (7).

Рассмотрим разностную схему во второй части грунта. Для дискретизации второго уравнения системы (59) также воспользуемся неявной четырехточечной схемой.

$$\frac{\partial T_2}{\partial t} = a_2 \frac{\partial^2 T_2}{\partial x^2}, \quad \xi(t) < x < L, \; t > 0;$$

$$\frac{T_{2,i}^{n+1} - T_{2,i}^n}{\tau_{n+1}} = a_2 \frac{T_{2,i+1}^{n+1} - 2T_{2,i}^{n+1} + T_{2,i-1}^{n+1}}{h^2}, \quad i = i^* + 1, \ldots, N-1;$$

$$T_2\big|_{i=i^*} = T_з;$$

$$\left.\frac{\partial T_2}{\partial x}\right|_{i=N} = 0.$$

Полученную систему можно свести к наиболее общему виду:

$$A_i \cdot T_{2,i+1}^{n+1} - B_i \cdot T_{2,i}^{n+1} + C_i \cdot T_{2,i-1}^{n+1} = F_i,$$

где

$$A_i = C_i = \frac{a_2}{h^2}, \quad B_i = \frac{2 \cdot a_2}{h^2} + \frac{1}{\tau_{n+1}}, \quad F_i = -\frac{T_{2,i}^n}{\tau_{n+1}}.$$

Прогоночные коэффициенты находятся по формулам (8). Далее неизвестное поле температуры определяется по выражению (7).

Проведем дискретизацию граничного условия при $x = \xi(t)$:

$$\lambda_1 \frac{\partial T_1}{\partial x} - \lambda_2 \frac{\partial T_2}{\partial x} = Q_ф \rho w \frac{d\xi}{dt}. \qquad (61)$$

Это условие необходимо для определения шага по времени на каждом временном слое.

Проведем дискретизацию с погрешностью $O(h)$.

$$\lambda_1 \left.\frac{\partial T_1}{\partial x}\right|_{x=\xi} - \lambda_2 \left.\frac{\partial T_2}{\partial x}\right|_{x=\xi} = Q_ф \rho w \frac{d\xi}{dt};$$

$$\lambda_1 \frac{T_{1,i^*} - T_{1,i^*-1}}{h} - \lambda_2 \frac{T_{2,i^*+1} - T_{2,i^*}}{h} = Q_ф \rho w \frac{h}{\tau_{n+1}}.$$

Далее будем опускать индексы, характеризующие рассматриваемую часть грунта, исходя из того, что при $i < i^*$ – часть *1*, а при $i > i^*$ – часть *2*.

$$\lambda_1 \frac{T_э - T_{i^*-1}}{h} - \lambda_2 \frac{T_{i^*+1} - T_э}{h} = Q_ф \rho w \frac{h}{\tau_{n+1}}.$$

И следовательно,

$$\tau_{n+1} = \frac{Q_ф \rho w h^2}{\lambda_1 \left(T_э - T_{i^*-1}\right) - \lambda_2 \left(T_{i^*+1} - T_э\right)}. \qquad (62)$$

Видим, что шаг по времени зависит от температуры. Тогда для определения поля температуры необходимо воспользоваться методом простой итерации. Основная идея, которого была изложена ранее.

Проведем дискретизацию нелинейного граничного условия (61) с погрешностью $O(h^2)$. Разложим функцию $T(x)$ в ряд Тейлора в окрестности точки $x = \xi$ до членов второго порядка относительно h:

$$T_{i^*+1}^{n+1} = T_{i^*}^{n+1} + h \cdot \left.\frac{\partial T_2}{\partial x}\right|_{x=\xi}^{n+1} + \frac{h^2}{2} \cdot \left.\frac{\partial^2 T_2}{\partial x^2}\right|_{x=\xi}^{n+1};$$

$$T_{i^*-1}^{n+1} = T_{i^*}^{n+1} - h \cdot \left.\frac{\partial T_1}{\partial x}\right|_{x=\xi}^{n+1} + \frac{h^2}{2} \cdot \left.\frac{\partial^2 T_1}{\partial x^2}\right|_{x=\xi}^{n+1}.$$

Используя соотношения (59) получим:

$$\left.\frac{\partial^2 T_1}{\partial x^2}\right|_{x=\xi}^{n+1} = \frac{1}{a_1} \left.\frac{\partial T_1}{\partial t}\right|_{x=\xi}^{n+1} = \frac{T_{i^*}^{n+1} - T_{i^*}^{n}}{a_1 \tau_{n+1}};$$

$$\left.\frac{\partial^2 T_2}{\partial x^2}\right|_{x=\xi}^{n+1} = \frac{1}{a_2} \left.\frac{\partial T_2}{\partial t}\right|_{x=\xi}^{n+1} = \frac{T_{i^*}^{n+1} - T_{i^*}^{n}}{a_2 \tau_{n+1}}.$$

Тогда

$$\lambda_1 \left.\frac{\partial T_1}{\partial x}\right|_{x=\xi}^{n+1} = \frac{\lambda_1}{h}\left(T_{i^*}^{n+1} - T_{i^*-1}^{n+1}\right) + \frac{\lambda_1 h}{2 a_1 \tau_{n+1}}\left(T_{i^*}^{n+1} - T_{i^*}^{n}\right);$$

$$\lambda_2 \left.\frac{\partial T_2}{\partial x}\right|_{x=\xi}^{n+1} = \frac{\lambda_2}{h}\left(T_{i^*+1}^{n+1} - T_{i^*}^{n+1}\right) - \frac{\lambda_2 h}{2 a_2 \tau_{n+1}}\left(T_{i^*}^{n+1} - T_{i^*}^{n}\right);$$

Аппроксимация граничного условия (61) с погрешностью $O(h^2)$ примет вид:

$$\frac{\lambda_1}{h}\left(T_{i^*}^{n+1} - T_{i^*-1}^{n+1}\right) + \frac{\lambda_1 h}{2 a_1 \tau_{n+1}}\left(T_{i^*}^{n+1} - T_{i^*}^{n}\right) - \frac{\lambda_2}{h}\left(T_{i^*+1}^{n+1} - T_{i^*}^{n+1}\right) +$$

$$\frac{\lambda_2 h}{2 a_2 \tau_{n+1}}\left(T_{i^*}^{n+1} - T_{i^*}^{n}\right) = Q_ф \rho w \frac{h}{\tau_{n+1}};$$

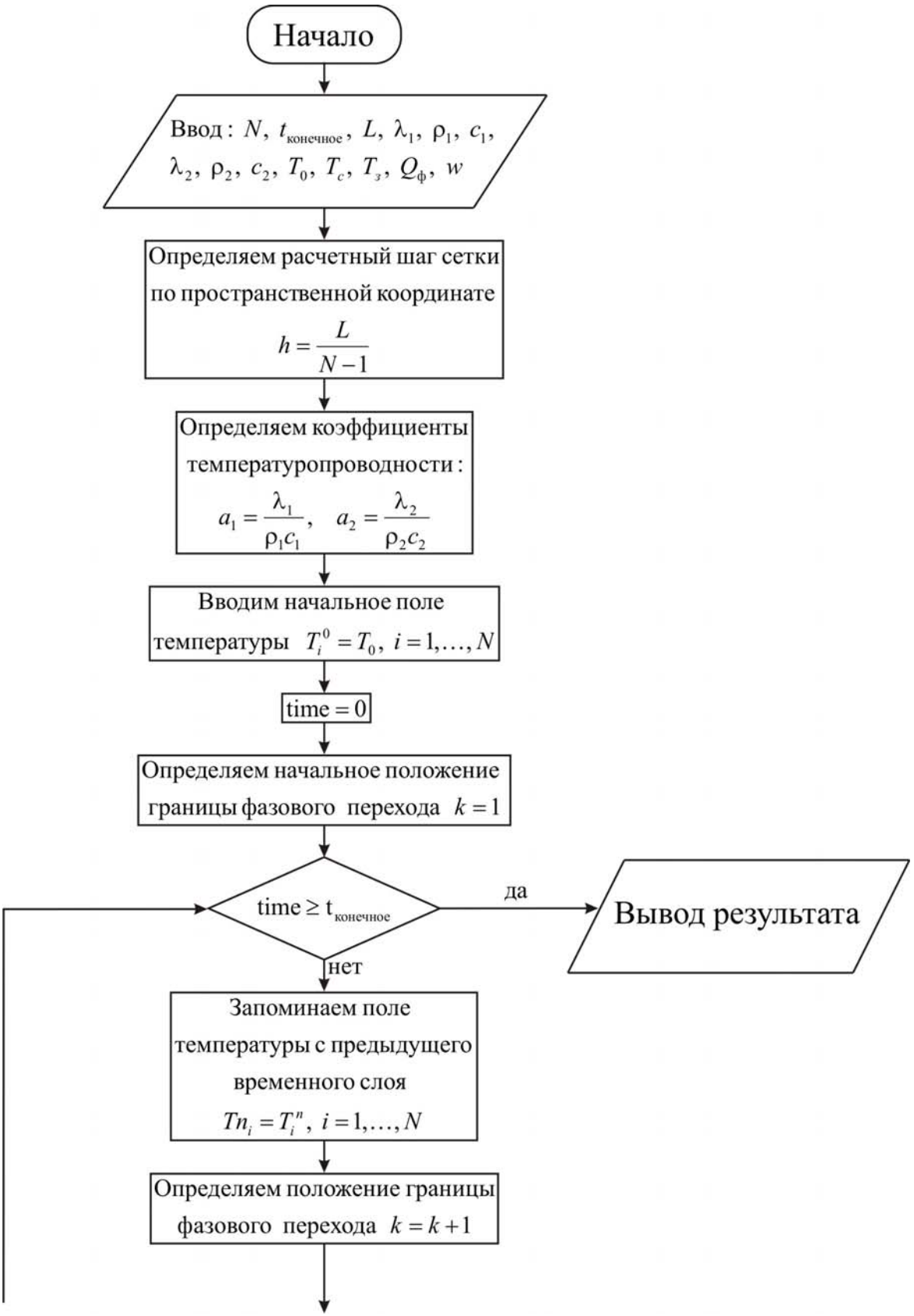
$$\tau_{n+1} = \frac{2a_1a_2h^2Q_{\text{ф}}\rho w - h^2\left(\lambda_1a_2 + \lambda_2a_1\right)\left(T_{i^*}^{n+1} - T_{i^*}^{n}\right)}{2a_1a_2\left[\lambda_1\left(T_{i^*}^{n+1} - T_{i^*-1}^{n+1}\right) - \lambda_2\left(T_{i^*+1}^{n+1} - T_{i^*}^{n+1}\right)\right]};$$

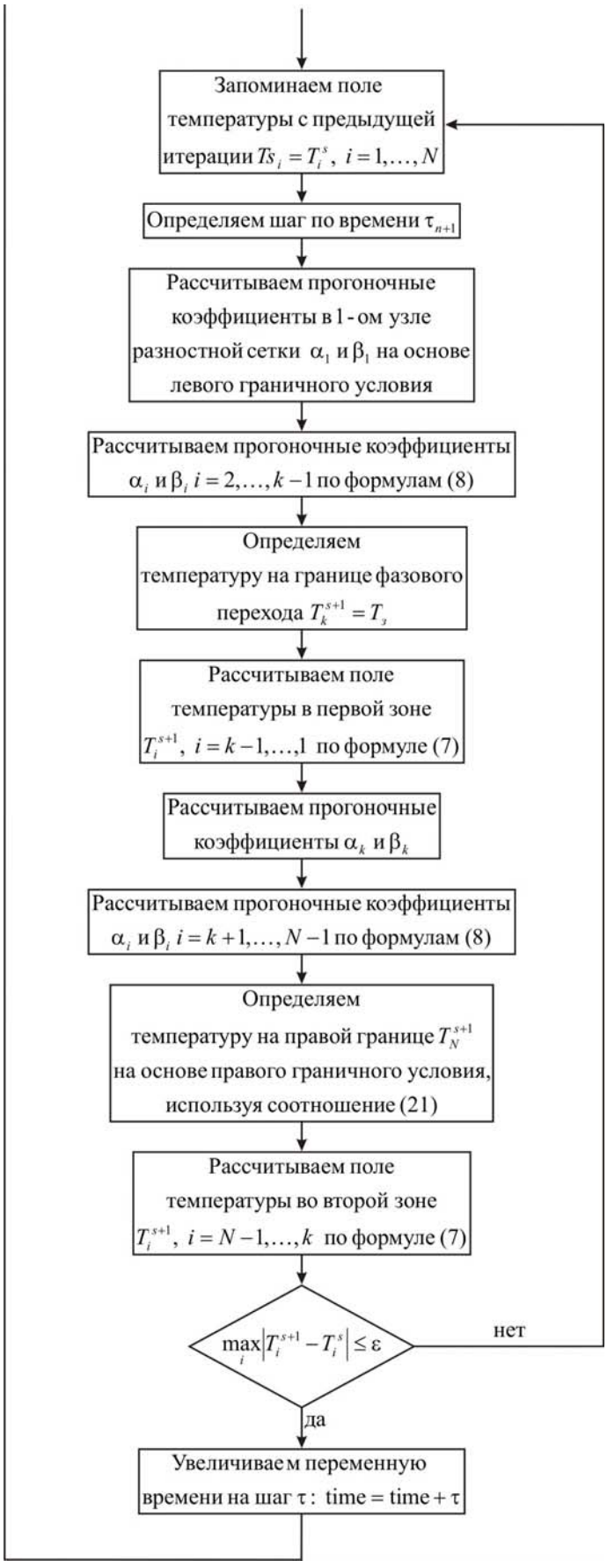
или

$$\tau_{n+1} = \frac{2a_1a_2h^2Q_{\text{ф}}\rho w - h^2\left(\lambda_1a_2 + \lambda_2a_1\right)\left(T_{з} - T_{i^*}^{n}\right)}{2a_1a_2\left[\lambda_1\left(T_{з} - T_{i^*-1}^{n+1}\right) - \lambda_2\left(T_{i^*+1}^{n+1} - T_{з}\right)\right]}. \qquad (63)$$

Слой грунта с глубиной $L = 0.3\,\text{м}$. Теплофизические характеристики промерзшей зоны грунта – λ_1 = 2.3 Вт/(м·К), ρ_1 = 917 кг/м^3, c_1 = 2090 Дж/(кг·К)). Теплофизические характеристики талой зоны грунта – λ_2 = 0.6 Вт/(м·К), ρ_2 = 1000 кг/м^3, c_2 = 4220 Дж/(кг·К)). Характерные температуры $T_0 = 293\,\text{К}$, $T_з = 273\,\text{К}$, $T_c = 268\,\text{К}$. Теплота фазового перехода $Q_{\text{ф}} = 3.32 \cdot 10^5$ Дж/кг. Влажность грунта $w = 1\,\text{кг/кг}$.

Блок-схема к рассматриваемой задаче имеет вид:




Запоминаем поле температуры с предыдущей итерации $Ts_i = T_i^s,\ i = 1,\ldots,N$
Определяем шаг по времени τ_{n+1}
Рассчитываем прогоночные коэффициенты в 1-ом узле разностной сетки α_1 и β_1 на основе левого граничного условия
Рассчитываем прогоночные коэффициенты α_i и β_i $i = 2,\ldots,k-1$ по формулам (8)
Определяем температуру на границе фазового перехода $T_k^{s+1} = T_з$
Рассчитываем поле температуры в первой зоне $T_i^{s+1},\ i = k-1,\ldots,1$ по формуле (7)
Рассчитываем прогоночные коэффициенты α_k и β_k
Рассчитываем прогоночные коэффициенты α_i и β_i $i = k+1,\ldots,N-1$ по формулам (8)
Определяем температуру на правой границе T_N^{s+1} на основе правого граничного условия, используя соотношение (21)
Рассчитываем поле температуры во второй зоне $T_i^{s+1},\ i = N-1,\ldots,k$ по формуле (7)
$\max_i \left|T_i^{s+1} - T_i^s\right| \le \varepsilon$
нет
да
Увеличиваем переменную времени на шаг τ: time = time + τ

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=500;
      eps=1e-5;
type
    vector=array[1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, k, N                  : integer;
 T, Tn, Ts, alfa, beta       : vector;
 ai, bi, ci, fi, max         : real;
 a1, lamda1, ro1, c1         : real;
 a2, lamda2, ro2, c2         : real;
 T0, Tfr, Tc, h, tau         : real;
 Qfr, w, L, t_end, time      : real;
 f, f1, g                    : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов, N');
 Readln(N);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите глубину грунта, L');
 Readln(L);
 Writeln('Введите коэффициент теплопроводности промерзшей зоны грунта, lamda1');
 Readln(lamda1);
 Writeln('Введите плотность промерзшей зоны грунта, ro1');
 Readln(ro1);
 Writeln('Введите теплоемкость промерзшей зоны грунта, c1');
 Readln(c1);
 Writeln('Введите коэффициент теплопроводности талой зоны грунта, lamda2');
 Readln(lamda2);
 Writeln('Введите плотность талой зоны грунта, ro2');
 Readln(ro2);
 Writeln('Введите теплоемкость талой зоны грунта, c2');
 Readln(c2);
```

```
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
 Writeln('Введите температуру на границе x = 0, Tc');
 Readln(Tc);
 Writeln('Введите температуру фазового перехода, Tfr');
 Readln(Tfr);
 Writeln('Введите теплоту фазового перехода, Qfr');
 Readln(Qfr);
 Writeln('Введите влагосодержание грунта, w');
 Readln(w);
{определяем расчетный шаг сетки по пространственной координате}
 h:=L/(N-1);
{определяем коэффициенты температуропроводности}
 a1:=lamda1/(ro1*c1);
 a2:=lamda2/(ro2*c2);
{определяем поле температуры в начальный момент времени}
 time:=0;
 for i:= 1 to N do
  T[i]:=T0;
{определяем файл, содержащий шаги по времени}
 Assign(f1,'time.txt');
 Rewrite(f1);
{определяем положение границы фазового перехода}
 k:=1;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 while time<t_end do {используем цикл с предусловием}
  begin
{запоминаем поле температуры на предыдущем временном слое}
   for i:= 1 to N do
    Tn[i]:=T[i];
{граница фазового перехода на каждом соответствующем временном
шаге смещается на пространственный шаг вправо}
   inc(k);
{цикл с постусловием, позволяющий итерационно вычислять поле
температуры, вследствие наличия нелинейности в граничном условии
(61)}
   repeat
{запоминаем поле температуры на предыдущей итерации}
    for i:= 1 to N do
     Ts[i]:=T[i];
```

```
{определяем соответствующий шаг по времени}
   tau:=(2.0*a1*a2*Qfr*w*0.5*(ro1+ro2)*sqr(h)-sqr(h)*(lamda1*a2+lamda2
         *a1)*(Tfr-Tn[k]))/(2.0*a1*a2*(lamda1*(Tfr-T[k-1])-lamda2
         *(T[k+1]-Tfr)));
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия}
   alfa[1]:=0.0;
   beta[1]:=Tc;
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
   for i:= 2 to k-1 do
    begin
{ai, bi, ci, fi – коэффициенты канонического представления системы
уравнений с трехдиагональной матрицей}
     ai:=a1/sqr(h);
     bi:=2.0*a1/sqr(h)+1.0/tau;
     ci:=a1/sqr(h);
     fi:=-Tn[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
     alfa[i]:=ai/(bi-ci*alfa[i-1]);
     beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
    end;
{определяем значение температуры на границе фазового перехода}
   T[k]:=Tfr;
{используя соотношение (7) определяем неизвестное поле температуры
в промерзшей зоне грунта}
   for i:= k-1 downto 1 do
    T[i]:=alfa[i]*T[i+1]+beta[i];
{определяем начальные прогоночные коэффициенты для талой зоны
грунта на основе условия на границе раздела двух сред}
   alfa[k]:=0.0;
   beta[k]:=Tfr;
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
   for i:= k+1 to N-1 do
    begin
{ai, bi, ci, fi – коэффициенты канонического представления системы
уравнений с трехдиагональной матрицей}
     ai:=a2/sqr(h);
     bi:=2.0*a2/sqr(h)+1.0/tau;
     ci:=a2/sqr(h);
```

```
      fi:=-Tn[i]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
      alfa[i]:=ai/(bi-ci*alfa[i-1]);
      beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
     end;
{определяем значение температуры на правой границе на основе
граничного условия, используя соотношение (21)}
    T[N]:=(2.0*a2*tau*beta[N-1]+sqr(h)*Tn[N])/(2.0*a2*tau
            *(1.0-alfa[N-1])+sqr(h));
{используя соотношение (7) определяем неизвестное поле температуры
в талой зоне грунта}
    for i:= N-1 downto k do
     T[i]:=alfa[i]*T[i+1]+beta[i];
{определяем максимум модуля разности между соответствующими
значениями температуры на текущей и предыдущей итерации}
    max:=abs(T[1]-Ts[1]);
    for i:= 2 to N do
     if max < abs(T[i]-Ts[i]) then max:=abs(T[i]-Ts[i]);
   until max<=eps; {поле температуры на данном временном слое
определили}
{увеличиваем переменную времени на шаг τn+1}
   time:=time+tau;
{формируем файл, содержащий шаги по времени τn+1}
    Writeln(f1,'tau',(k-1),' = ',tau:6:4);
  end;
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Глубина грунта L = ',L:6:4);
 Writeln(f,'Число узлов по координате N = ',N);
 Writeln(f,'Коэффициент теплопроводности промерзшей зоны грунта
lamda1 = ',lamda1:6:4);
 Writeln(f,'Плотность промерзшей зоны грунта ro1 = ',ro1:6:4);
 Writeln(f,'Теплоемкость промерзшей зоны грунта c1 = ',c1:6:4);
 Writeln(f,'Коэффициент теплопроводности талой зоны грунта lamda2 =
',lamda2:6:4);
 Writeln(f,'Плотность талой зоны грунта ro2 = ',ro2:6:4);
 Writeln(f,'Теплоемкость талой зоны грунта c2 = ',c2:6:4);
 Writeln(f,'Начальная температура T0 = ',T0:6:4);
 Writeln(f,'Температура на границе x = 0',Tc:6:4);
 Writeln(f,'Температура фазового перехода',Tfr:6:4);
```

```
 Writeln(f,'Теплота фазового перехода',Qfr:6:4);
 Writeln(f,'Влагосодержание грунта',w:6:4);
 Writeln(f,'Результат получен с шагом по координате h = ',h:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',time:6:4);
 close(f);
 close(f1);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to N do
  writeln(g,'  ',h*(i-1):10:8,'   ',(T[i]-273):8:5);
 close(g);
end.
```

Получены следующие распределения температуры:

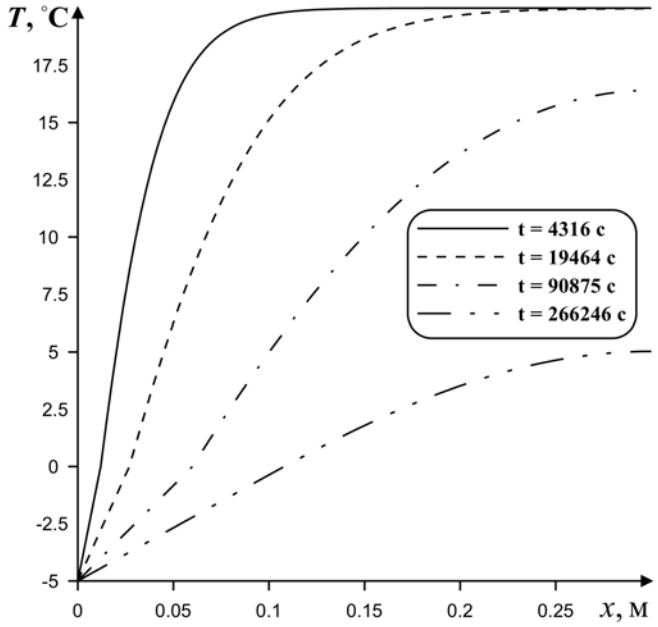


Рис. 27. Распределения температуры по глубине грунта

3.6. ДВУМЕРНОЕ УРАВНЕНИЕ ТЕПЛОПРОВОДНОСТИ С ИЗЛУЧЕНИЕМ НА ГРАНИЦАХ

Проанализируем процесс теплопереноса в пластине, на двух границах которой осуществляется теплообмен с внешней средой за счет излучения и конвекции. Область решения имеет вид аналогичный рис. 17.

Пластина с размерами $L = H = 0.3$ м. Материал пластины – твердая резина (λ = 0.16 Вт/(м·ºС), ρ = 1190 кг/м^3, c = 1900 Дж/(кг·ºС)).

Начальная температура области решения $T_0 = 30\ ^0\text{C}$. $\kappa_1 = 50\ \text{Вт}/\left(\text{м}^2 \cdot^0 \text{C}\right)$, $T^{e1} = 20\ ^0\text{C}$, $\varepsilon = 0.8$, $\kappa_2 = 35\ \text{Вт}/\left(\text{м}^2 \cdot^0 \text{C}\right)$, $T^{e2} = 35\ ^0\text{C}$.

Математическая постановка задачи будет иметь вид:

$$\rho c \frac{\partial T}{\partial t} = \lambda\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right), \quad \left|\begin{array}{l} 0 < x < L; \\ 0 < y < H. \end{array}\right. \tag{64}$$

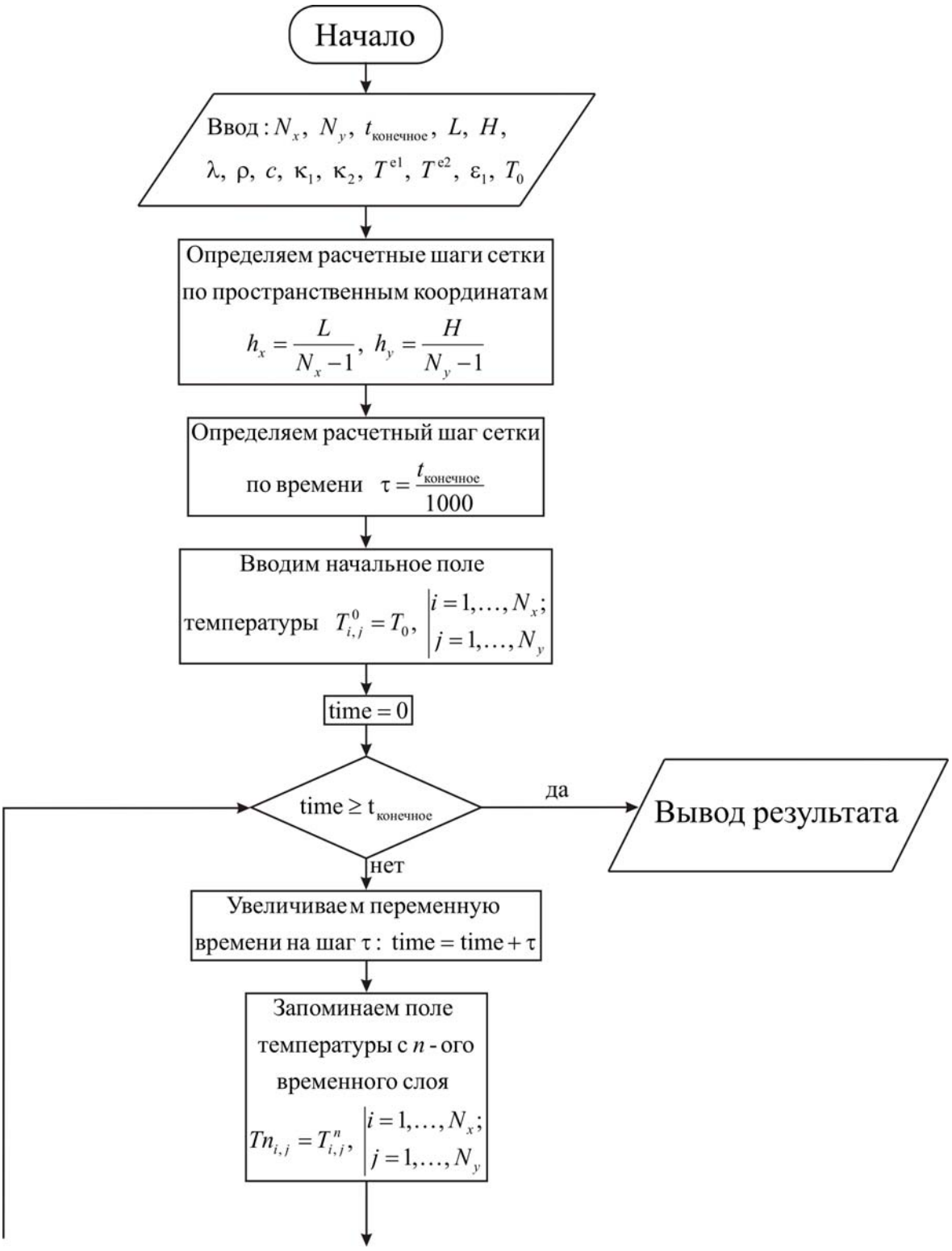
Начальные и граничные условия запишутся следующим образом:

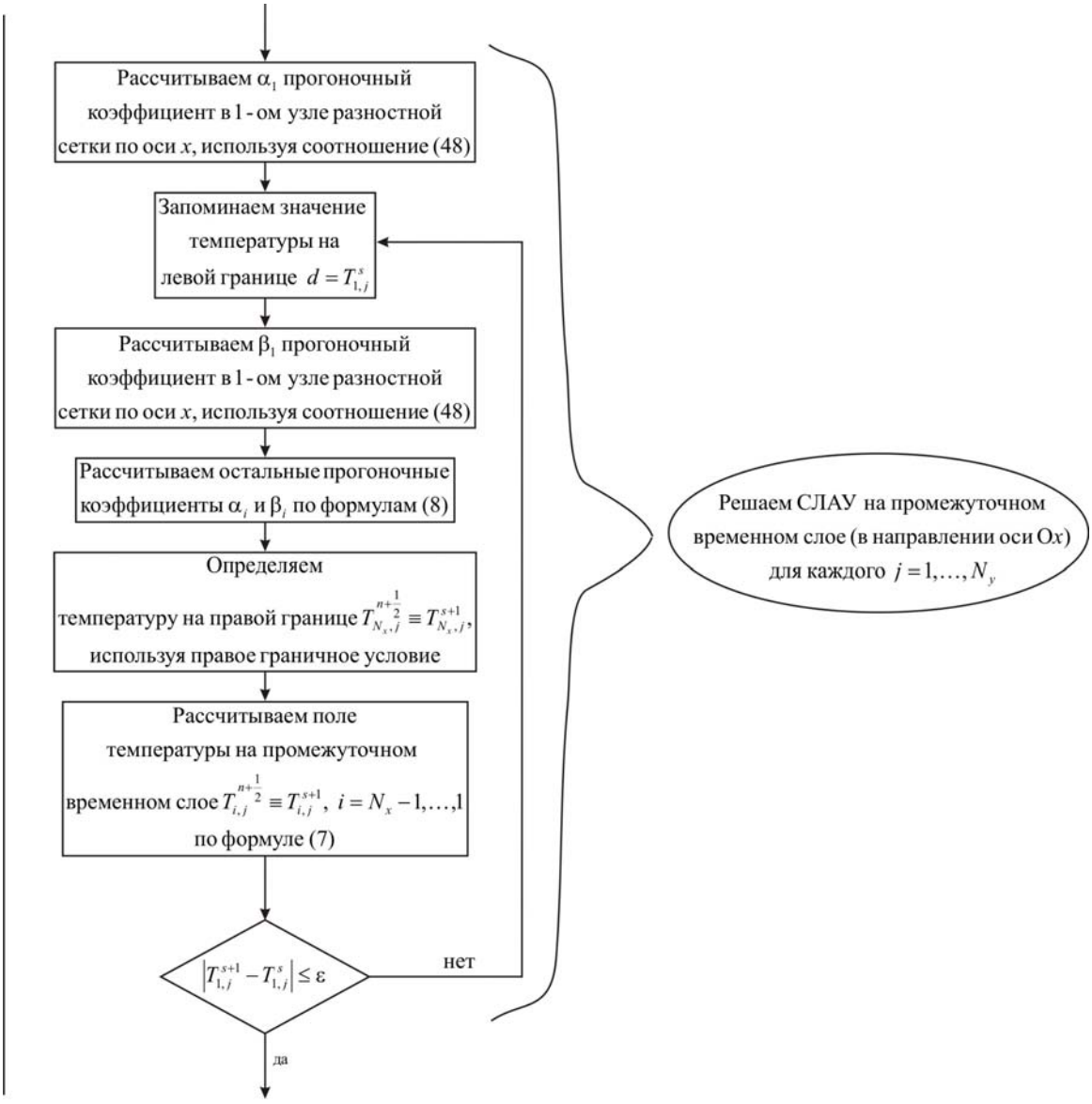
$$\begin{aligned}
&t = 0:\ T = T_0,\ 0 \le x \le L,\ 0 \le y \le H; \\
&x = 0:\ -\lambda \frac{\partial T}{\partial x} = \kappa_1\left(T^{e1} - T\right) + \varepsilon\sigma\left(\left(T^{e1}\right)^4 - T^4\right),\ t > 0,\ \kappa_1 > 0; \\
&x = L:\ \frac{\partial T}{\partial x} = 0,\ t > 0; \\
&y = 0:\ \frac{\partial T}{\partial y} = 0,\ t > 0; \\
&y = H:\ \lambda \frac{\partial T}{\partial y} = \kappa_2\left(T^{e2} - T\right) + \varepsilon\sigma\left(\left(T^{e2}\right)^4 - T^4\right),\ t > 0,\ \kappa_2 > 0;
\end{aligned} \tag{65}$$

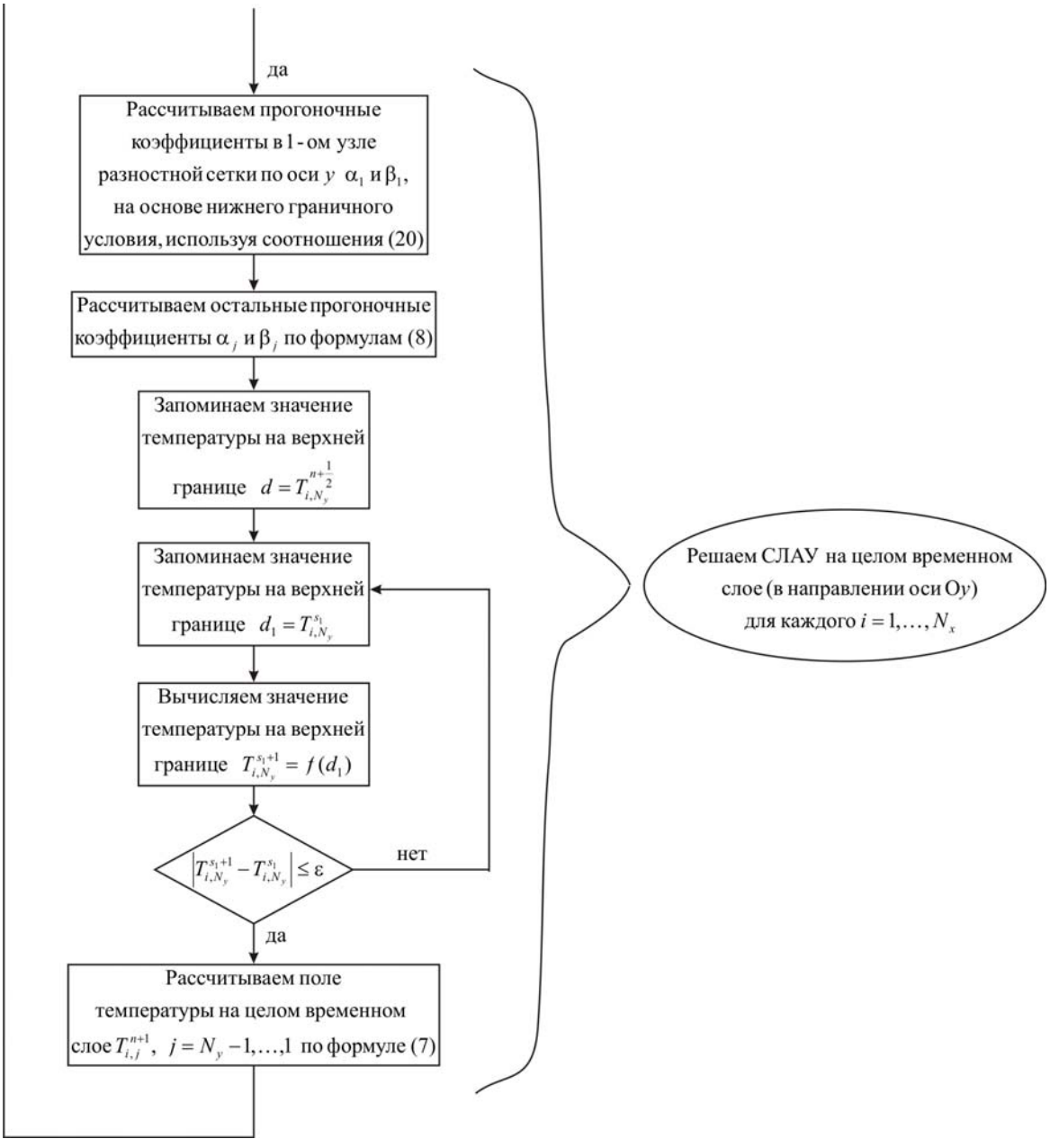
Рассматриваемая задача объединяет в себе постановки 2.6 и 3.2. Для решения сформулированной задачи (64), (65) введем равномерную пространственно-временную сетку.

Дискретизацию уравнения (64) также как и в пункте 2.6 будем проводить на основе локально одномерной схемы А.А. Самарского. Решение полученных систем линейных алгебраических уравнений проводится методом прогонки, при этом необходимо учесть, что на границе присутствует излучение (пункт 3.2), которое моделируется нелинейным соотношением, поэтому необходимо воспользоваться методом простой итерации.

Блок-схема к рассматриваемой задаче имеет вид:




Рассчитываем α_1 прогоночный коэффициент в 1-ом узле разностной сетки по оси x, используя соотношение (48)
Запоминаем значение температуры на левой границе $d = T^s_{1,j}$
Рассчитываем β_1 прогоночный коэффициент в 1-ом узле разностной сетки по оси x, используя соотношение (48)
Рассчитываем остальные прогоночные коэффициенты α_i и β_i по формулам (8)
Определяем температуру на правой границе $T^{n+\frac{1}{2}}_{N_x,j} \equiv T^{s+1}_{N_x,j}$, используя правое граничное условие
Рассчитываем поле температуры на промежуточном временном слое $T^{n+\frac{1}{2}}_{i,j} \equiv T^{s+1}_{i,j}$, $i = N_x - 1,\ldots,1$ по формуле (7)
$\left|T^{s+1}_{1,j} - T^s_{1,j}\right| \le \varepsilon$
нет
да
Решаем СЛАУ на промежуточном временном слое (в направлении оси Ox) для каждого $j = 1,\ldots,N_y$


да
Рассчитываем прогоночные коэффициенты в 1-ом узле разностной сетки по оси y α_1 и β_1, на основе нижнего граничного условия, используя соотношения (20)
Рассчитываем остальные прогоночные коэффициенты α_j и β_j по формулам (8)
Запоминаем значение температуры на верхней границе $d = T_{i,N_y}^{n+\frac{1}{2}}$
Запоминаем значение температуры на верхней границе $d_1 = T_{i,N_y}^{s_1}$
Вычисляем значение температуры на верхней границе $T_{i,N_y}^{s_1+1} = f(d_1)$
$\left|T_{i,N_y}^{s_1+1} - T_{i,N_y}^{s_1}\right| \le \varepsilon$
нет
да
Рассчитываем поле температуры на целом временном слое $T_{i,j}^{n+1}$, $j = N_y - 1,\ldots,1$ по формуле (7)
Решаем СЛАУ на целом временном слое (в направлении оси Oy) для каждого $i = 1,\ldots,N_x$

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=72;
        sigma=5.669e-8;
        eps=1e-5;
type
    vector1=array[1..mf] of real;
    vector2=array[1..mf,1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, Nx, Ny                    : integer;
 T, Tn                           : vector2;
 alfa, beta                      : vector1;
 ai, bi, ci, fi, d, d1           : real;
 lamda, ro, c                    : real;
 kapa1, kapa2, Te1, Te2, eps1    : real;
 hx, hy, tau, t_end, time        : real;
 T0, L, H                        : real;
 f, g                            : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов в пластине по оси x, Nx');
 Readln(Nx);
 Writeln('Введите количество пространственных узлов в пластине по оси y, Ny');
 Readln(Ny);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите длину пластины, L');
 Readln(L);
 Writeln('Введите толщину пластины, H');
 Readln(H);
 Writeln('Введите коэффициент теплопроводности материала пластины, lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
 Readln(ro);
```

```
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите коэффициент теплообмена на границах x,y = 0, kapa1');
 Readln(kapa1);
 Writeln('Введите коэффициент теплообмена на границах x = L, y = H,
kapa2');
 Readln(kapa2);
 Writeln('Введите температуру внешней среды относительно границ x,y
= 0, Te1');
 Readln(Te1);
 Writeln('Введите температуру внешней среды относительно границ x =
L, y = H, Te2');
 Readln(Te2);
 Writeln('Введите приведенную степень черноты, eps1');
 Readln(eps1);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
{определяем расчетные шаги сетки по пространственным
координатам}
 hx:=L/(Nx-1);
 hy:=H/(Ny-1);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/1000.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to Nx do
  for j:= 1 to Ny do
    T[i,j]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{запоминаем поле температуры на n-ом временном слое}
   for i:=1 to Nx do
    for j:=1 to Ny do
     Tn[i,j]:=T[i,j];
{решаем СЛАУ в направлении оси Ox для определения поля
температуры на промежуточном (n+1/2) временном слое}
   for j:=1 to Ny do
```

```
   begin
{определяем alfa начальный прогоночный коэффициент на основе левого
граничного условия, используя соотношение (48)}
    alfa[1]:=2.0*tau*lamda/(2.0*tau*(lamda+kapa1*hx)+ro*c*sqr(hx));
{цикл с постусловием, позволяющий итерационно вычислять поле
температуры, вследствие наличия нелинейности в левом граничном
условии}
    repeat
{определяем beta начальный прогоночный коэффициент на основе
левого граничного условия, используя соотношение (48), при этом
начинаем итерационный цикл по левому граничному условию}
     d:=T[1,j];
     beta[1]:=(ro*c*sqr(hx)*Tn[1,j]+2.0*tau*kapa1*hx*Te1+2.0*tau
                *eps1*sigma*hx*(sqr(sqr(Te1))-sqr(sqr(d))))
                /(2.0*tau*(lamda+kapa1*hx)+ro*c*sqr(hx));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
     for i:= 2 to Nx-1 do
      begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
       ai:=lamda/sqr(hx);
       bi:=2.0*lamda/sqr(hx)+ro*c/tau;
       ci:=lamda/sqr(hx);
       fi:=-ro*c*Tn[i,j]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
       alfa[i]:=ai/(bi-ci*alfa[i-1]);
       beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
      end;
{определяем значение температуры на правой границе, используя
соотношение (21) при условии, что q2 = 0}
     T[Nx,j]:=(ro*c*sqr(hx)*Tn[Nx,j]+2.0*tau*lamda*beta[Nx-1])
                /(ro*c*sqr(hx)+2.0*tau*lamda*(1-alfa[Nx-1]));
{используя соотношение (7) определяем неизвестное поле температуры
на промежуточном (n+1/2) временном слое}
     for i:= Nx-1 downto 1 do
      T[i,j]:=alfa[i]*T[i+1,j]+beta[i];
    until abs(d-T[1,j])<=eps; {значение температуры на левой границе
определили}
   end; {поле температуры на промежуточном (n+1/2) временном слое
определили}
```

```
{решаем СЛАУ в направлении оси Oy для определения поля
температуры на целом (n+1) временном слое}
  for i:=1 to Nx do
   begin
{определяем начальные прогоночные коэффициенты на основе нижнего
граничного условия, используя соотношения (20) при условии, что
q1 = 0}
    alfa[1]:=2.0*tau*lamda/(2.0*tau*lamda+ro*c*sqr(hy));
    beta[1]:=ro*c*sqr(hy)*T[i,1]/(2.0*tau*lamda+ro*c*sqr(hy));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
    for j:= 2 to Ny-1 do
     begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
      ai:=lamda/sqr(hy);
      bi:=2.0*lamda/sqr(hy)+ro*c/tau;
      ci:=lamda/sqr(hy);
      fi:=-ro*c*T[i,j]/tau;
{alfa[j], beta[j] – прогоночные коэффициенты}
      alfa[j]:=ai/(bi-ci*alfa[j-1]);
      beta[j]:=(ci*beta[j-1]-fi)/(bi-ci*alfa[j-1]);
     end;
{запоминаем значение температуры на правой границе с
промежуточного (n+1/2) временного слоя}
    d:=T[i,Ny];
{цикл с постусловием, позволяющий итерационно вычислить значение
температуры на правой границе, вследствие наличия нелинейности в
этом граничном условии}
    repeat
     d1:=T[i,Ny];
{определяем значение температуры на правой границе на основе
правого граничного условия, используя соотношение (49)}
     T[i,Ny]:=(ro*c*sqr(hy)*d+2.0*tau*(lamda*beta[Ny-1]+kapa2*hy*Te2
              +eps1*sigma*hy*(sqr(sqr(Te2))-sqr(sqr(d1)))))/(ro*c*sqr(hy)
              +2.0*tau*(lamda*(1-alfa[Ny-1])+kapa2*hy));
    until abs(d1-T[i,Ny])<=eps; {значение температуры на правой
границе определили}
{используя соотношение (7) определяем неизвестное поле температуры
на целом (n+1) временном слое}
    for j:= Ny-1 downto 1 do
```

```
      T[i,j]:=alfa[j]*T[i,j+1]+beta[j];
    end; {поле температуры на целом (n+1) временном слое определили}
  end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Длина пластины L = ',L:6:4);
 Writeln(f,'Толщина пластины H = ',H:6:4);
 Writeln(f,'Число узлов по пространственной координате x в пластине
Nx = ',Nx);
 Writeln(f,'Число узлов по пространственной координате y в пластине
Ny = ',Ny);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура T0 = ',T0:6:4);
 Writeln(f,'Коэффициент теплообмена kapa1 = ',kapa1:6:4);
 Writeln(f,'Коэффициент теплообмена kapa2 = ',kapa2:6:4);
 Writeln(f,'Температура внешней среды Te1 = ',Te1:6:4);
 Writeln(f,'Температура внешней среды Te2 = ',Te2:6:4);
 Writeln(f,'Приведенная степень черноты eps1 = ',eps1:6:4);
 Writeln(f,'Результат получен с шагом по координате x hx = ',hx:6:4);
 Writeln(f,'Результат получен с шагом по координате y hy = ',hy:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to Nx do
  for j:=1 to Ny do
   writeln(g,'  ',hx*(i-1):10:8,'  ',hy*(j-1):10:8,'   ',T[i,j]:8:5);
 close(g);
end.
```

Получено следующее распределение температуры:

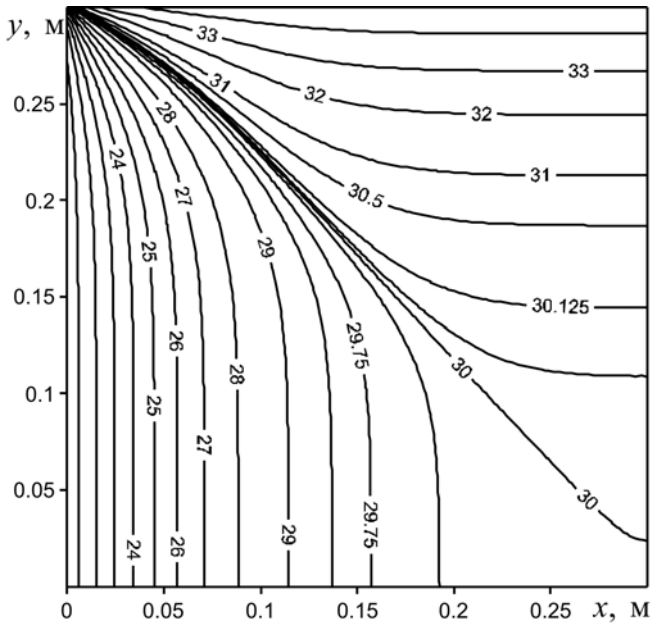


Рис. 28. Распределение температуры (ºС) в пластине при t = 36000 с

3.7. ДВУМЕРНОЕ УРАВНЕНИЕ ТЕПЛОПРОВОДНОСТИ С ФАЗОВЫМ ПЕРЕХОДОМ НА ОДНОЙ ИЗ ГРАНИЦ

Проанализируем процесс теплопереноса в пластине, две границы которой являются адиабатическими, а на одной моделируется процесс испарения. Область решения имеет вид аналогичный рис. 17.

Пластина с размерами $L = H = 0.3$ м. Материал пластины – AlF_3 (λ = 60 Вт/(м·К), ρ = 3070 кг/м^3, c = 1260 Дж/(кг·К), M = 0.084 кг/моль). Начальная температура области решения $T_0 = 1273$ К. $A = 0.1,\ k_0 = 10^5, q = 10^4$ Вт/м^2, $Q_{\text{исп}} = 3.8 \cdot 10^4$ Дж/кг, к = 50 Вт/(м^2 · К), $T^{\text{e}} = 600$ К.

Математическая постановка задачи будет иметь вид:

$$\rho c \frac{\partial T}{\partial t} = \lambda \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right), \quad \left| \begin{array}{l} 0 < x < L; \\ 0 < y < H. \end{array} \right. \tag{66}$$

Начальные и граничные условия запишутся следующим образом:

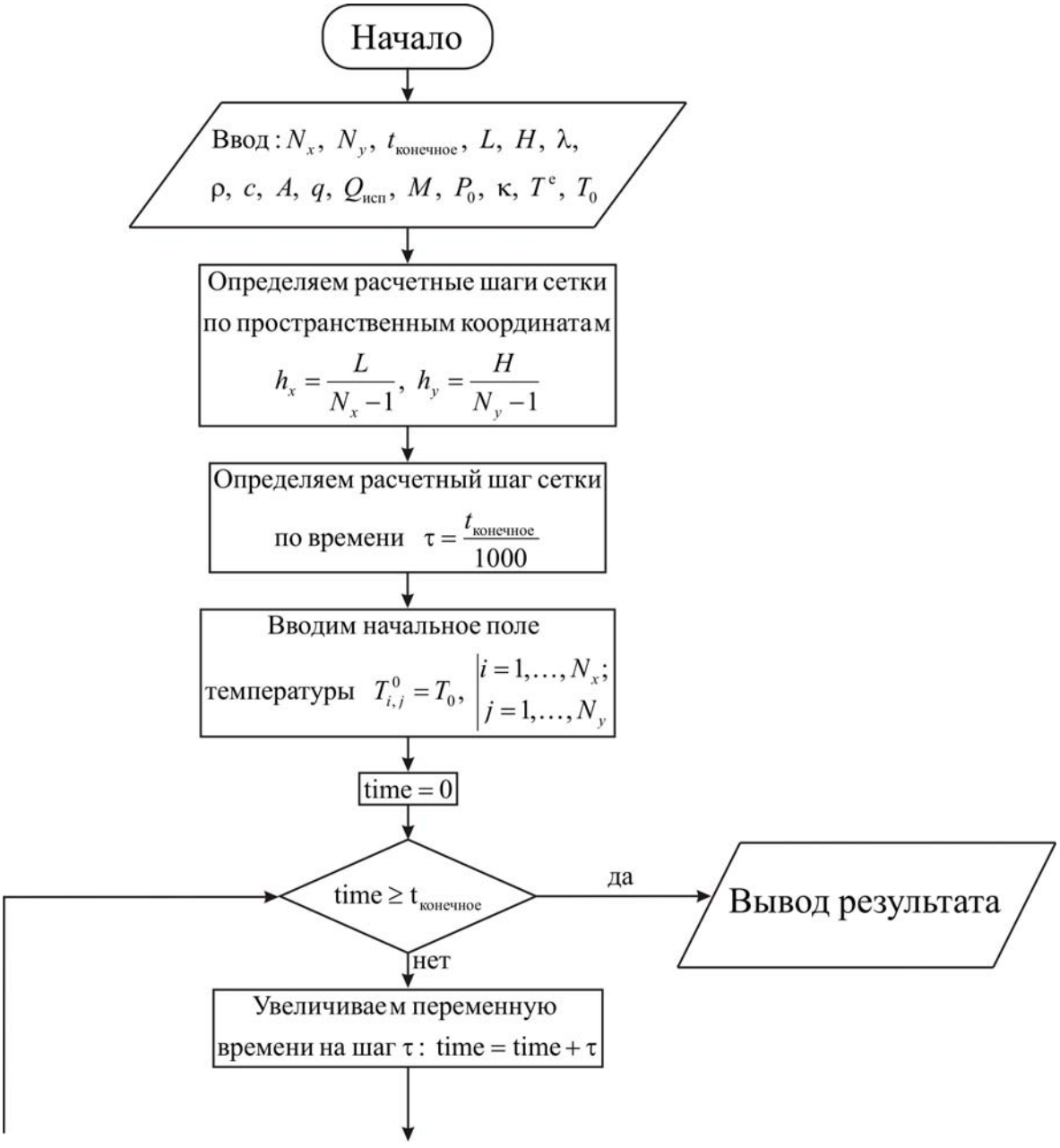
$$
\begin{aligned}
&t = 0:\ T = T_0,\ 0 \le x \le L,\ 0 \le y \le H;\\
&x = 0:\ \frac{\partial T}{\partial x} = 0,\ t > 0;\\
&x = L:\ \frac{\partial T}{\partial x} = 0,\ t > 0;\\
&y = 0:\ -\lambda \frac{\partial T}{\partial y} = \kappa\left(T^{\text{e}} - T\right),\ t > 0,\ \kappa > 0;\\
&y = H:\ \lambda \frac{\partial T}{\partial y} = q - w_{\text{исп}} \cdot Q_{\text{исп}},\ t > 0;
\end{aligned}
\qquad (67)
$$

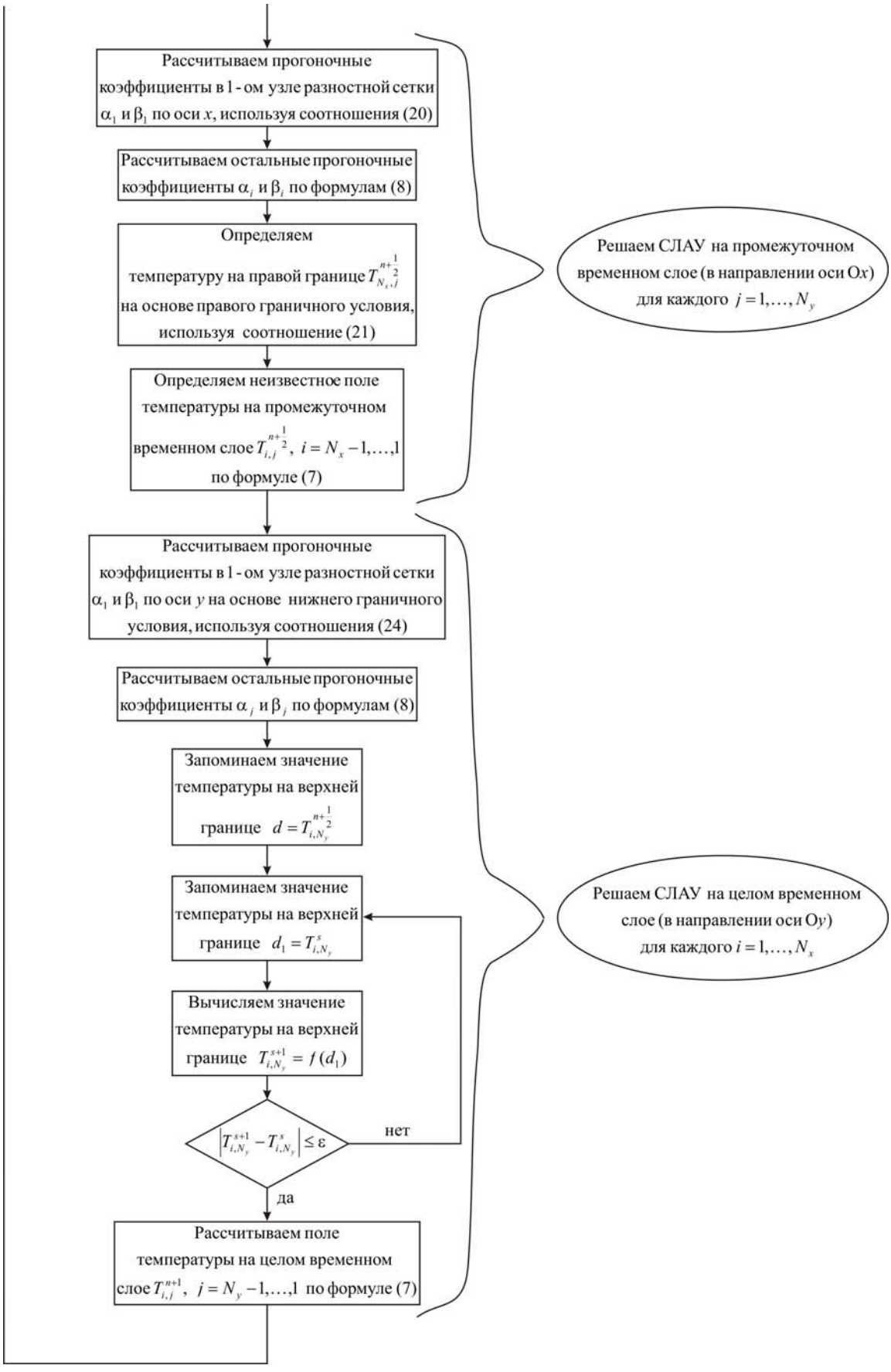
где $w_{\text{исп}} = \dfrac{A \cdot \left(P^{н} - P^{*}\right)}{\sqrt{\dfrac{2\pi RT}{M}}}$ – скорость испарения, $P^{н} = P_0 \cdot \exp\left(-\dfrac{Q_{\text{исп}}}{RT}\right)$ – давление насыщенного пара.

Рассматриваемая задача объединяет в себе постановки 2.6 и 3.3. Для решения сформулированной задачи (66), (67) введем равномерную пространственно-временную сетку.

Дискретизацию уравнения (66) также как и в пункте 2.6 будем проводить на основе локально одномерной схемы А.А. Самарского. Решение полученных систем линейных алгебраических уравнений проводится методом прогонки, при этом необходимо учесть, что на границе присутствует процесс испарения материала (пункт 3.3), который моделируется нелинейным соотношением, поэтому необходимо воспользоваться методом простой итерации.

Блок-схема к рассматриваемой задаче имеет вид:




Рассчитываем прогоночные коэффициенты в 1 - ом узле разностной сетки α_1 и β_1 по оси x, используя соотношения (20)
Рассчитываем остальные прогоночные коэффициенты α_i и β_i по формулам (8)
Определяем температуру на правой границе $T_{N_x,j}^{n+\frac{1}{2}}$ на основе правого граничного условия, используя соотношение (21)
Определяем неизвестное поле температуры на промежуточном временном слое $T_{i,j}^{n+\frac{1}{2}}$, $i = N_x - 1,\ldots,1$ по формуле (7)
Решаем СЛАУ на промежуточном временном слое (в направлении оси Ox) для каждого $j = 1,\ldots,N_y$
Рассчитываем прогоночные коэффициенты в 1 - ом узле разностной сетки α_1 и β_1 по оси y на основе нижнего граничного условия, используя соотношения (24)
Рассчитываем остальные прогоночные коэффициенты α_j и β_j по формулам (8)
Запоминаем значение температуры на верхней границе $d = T_{i,N_y}^{n+\frac{1}{2}}$
Запоминаем значение температуры на верхней границе $d_1 = T_{i,N_y}^{s}$
Вычисляем значение температуры на верхней границе $T_{i,N_y}^{s+1} = f(d_1)$
$\left|T_{i,N_y}^{s+1} - T_{i,N_y}^{s}\right| \le \varepsilon$
нет
да
Рассчитываем поле температуры на целом временном слое $T_{i,j}^{n+1}$, $j = N_y - 1,\ldots,1$ по формуле (7)
Решаем СЛАУ на целом временном слое (в направлении оси Oy) для каждого $i = 1,\ldots,N_x$

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=102;
       R=8.31;
      eps=1e-5;
      Patm=101325;
type
    vector1=array[1..mf] of real;
    vector2=array[1..mf,1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, Nx, Ny                : integer;
 T                           : vector2;
 alfa, beta                  : vector1;
 ai, bi, ci, fi, d, d1       : real;
 lamda, ro, c                : real;
 q, A, qev, P0, M            : real;
 hx, hy, tau, t_end, time    : real;
 T0, Te, kapa, L, H          : real;
 f, g                        : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов в пластине по оси x, Nx');
 Readln(Nx);
 Writeln('Введите количество пространственных узлов в пластине по оси y, Ny');
 Readln(Ny);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите длину пластины, L');
 Readln(L);
 Writeln('Введите толщину пластины, H');
 Readln(H);
 Writeln('Введите коэффициент теплопроводности материала пластины, lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
```

```
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите коэффициент аккомодации, A');
 Readln(A);
 Writeln('Введите плотность теплового потока на границе, q');
 Readln(q);
 Writeln('Введите тепловой эффект фазового перехода, qev');
 Readln(qev);
 Writeln('Введите молярную массу, M');
 Readln(M);
 Writeln('Введите предэкспонент, P0');
 Readln(P0);
 Writeln('Введите коэффициент теплообмена на границе y = 0, kapa');
 Readln(kapa);
 Writeln('Введите температуру внешней среды относительно границы y
= 0, Te');
 Readln(Te);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
{определяем расчетные шаги сетки по пространственным
координатам}
 hx:=L/(Nx-1);
 hy:=H/(Ny-1);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/1000.0;
{определяем поле температуры в начальный момент времени}
 for i:= 1 to Nx do
  for j:= 1 to Ny do
    T[i,j]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{решаем СЛАУ в направлении оси Ox для определения поля
температуры на промежуточном (n+1/2) временном слое}
   for j:=1 to Ny do
    begin
```

```
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия, используя соотношение (20) при условии, что
q1 = 0}
    alfa[1]:=2.0*tau*lamda/(2.0*tau*lamda+ro*c*sqr(hx));
    beta[1]:=ro*c*sqr(hx)*T[1,j]/(2.0*tau*lamda+ro*c*sqr(hx));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
    for i:= 2 to Nx-1 do
      begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
       ai:=lamda/sqr(hx);
       bi:=2.0*lamda/sqr(hx)+ro*c/tau;
       ci:=lamda/sqr(hx);
       fi:=-ro*c*T[i,j]/tau;
{alfa[i], beta[i] – прогоночные коэффициенты}
       alfa[i]:=ai/(bi-ci*alfa[i-1]);
       beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
      end;
{определяем значение температуры на правой границе, используя
соотношение (21) при условии, что q2 = 0}
    T[Nx,j]:=(ro*c*sqr(hx)*T[Nx,j]+2.0*tau*lamda*beta[Nx-1])
             /(ro*c*sqr(hx)+2.0*tau*lamda*(1-alfa[Nx-1]));
{используя соотношение (7) определяем неизвестное поле температуры
на промежуточном (n+1/2) временном слое}
    for i:= Nx-1 downto 1 do
      T[i,j]:=alfa[i]*T[i+1,j]+beta[i];
   end; {поле температуры на промежуточном (n+1/2) временном слое
определили}
{решаем СЛАУ в направлении оси Oy для определения поля
температуры на целом (n+1) временном слое}
  for i:=1 to Nx do
   begin
{определяем начальные прогоночные коэффициенты на основе нижнего
граничного условия, используя соотношения (24)}
    alfa[1]:=2.0*tau*lamda/(2.0*tau*(lamda+kapa*hy)+ro*c*sqr(hy));
    beta[1]:=(ro*c*sqr(hy)*T[i,1]+2.0*tau*kapa*hy*Te)
             /(2.0*tau*(lamda+kapa*hy)+ro*c*sqr(hy));
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
    for j:= 2 to Ny-1 do
```

```
      begin
{ai, bi, ci, fi – коэффициенты канонического представления СЛАУ с
трехдиагональной матрицей}
       ai:=lamda/sqr(hy);
       bi:=2.0*lamda/sqr(hy)+ro*c/tau;
       ci:=lamda/sqr(hy);
       fi:=-ro*c*T[i,j]/tau;
{alfa[j], beta[j] – прогоночные коэффициенты}
       alfa[j]:=ai/(bi-ci*alfa[j-1]);
       beta[j]:=(ci*beta[j-1]-fi)/(bi-ci*alfa[j-1]);
      end;
{запоминаем значение температуры на верхней границе с
промежуточного (n+1/2) временного слоя}
     d:=T[i,Ny];
{цикл с постусловием, позволяющий итерационно вычислить значение
температуры на верхней границе, вследствие наличия нелинейности в
этом граничном условии}
     repeat
      d1:=T[i,Ny];
{определяем значение температуры на верхней границе на основе
верхнего граничного условия, используя соотношение (55)}
      T[i,Ny]:=(ro*c*sqr(hy)*d+2.0*tau*(lamda*beta[Ny-1]+h*q))
                 /(ro*c*sqr(hy)+2.0*tau*lamda*(1-alfa[Ny-1]))-2.0*tau
                *A*hy*qev*(P0*exp(-qev/(R*d1))-Patm)/((ro*c*sqr(hy)
                +2.0*tau*lamda*(1-alfa[Ny-1]))*sqrt(2.0*pi*R*d1/M));
     until abs(d1-T[i,Ny])<=eps; {значение температуры на верхней
границе определили}
{используя соотношение (7) определяем неизвестное поле температуры
на целом (n+1) временном слое}
     for j:= Ny-1 downto 1 do
      T[i,j]:=alfa[j]*T[i,j+1]+beta[j];
    end; {поле температуры на целом (n+1) временном слое определили}
  end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Длина пластины L = ',L:6:4);
 Writeln(f,'Толщина пластины H = ',H:6:4);
 Writeln(f,'Число узлов по пространственной координате x в пластине
Nx = ',Nx);
```

```
 Writeln(f,'Число узлов по пространственной координате y в пластине
Ny = ',Ny);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Начальная температура T0 = ',T0:6:4);
 Writeln(f,'Коэффициент теплообмена kapa = ',kapa:6:4);
 Writeln(f,'Температура внешней среды Te = ',Te:6:4);
 Writeln(f,'Коэффициент аккомодации A = ',A:6:4);
 Writeln(f,'Плотность теплового потока на границе q = ',q:6:4);
 Writeln(f,'Тепловой эффект фазового перехода qev = ',qev:6:4);
 Writeln(f,'Молярная масса M = ',M:6:4);
 Writeln(f,'Предэкспонент P0 = ',P0:6:4);
 Writeln(f,'Результат получен с шагом по координате x hx = ',hx:6:4);
 Writeln(f,'Результат получен с шагом по координате y hy = ',hy:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
 Assign(g,'tempr.txt');
 Rewrite(g);
 for i:=1 to Nx do
  for j:=1 to Ny do
   writeln(g,'  ',hx*(i-1):10:8,'  ',hy*(j-1):10:8,'   ',(T[i,j]-273):8:5);
 close(g);
end.
```

Получено следующее распределение температуры:

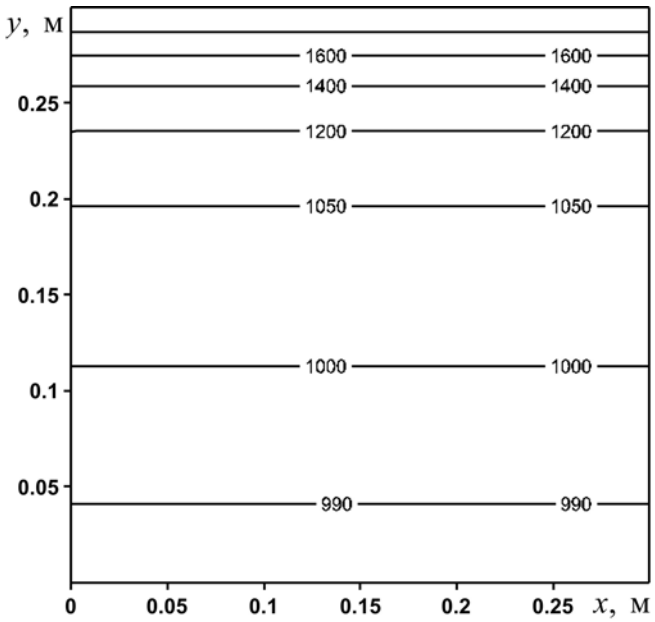


Рис. 29. Распределение температуры (ºС) в пластине при t = 120 с

3.8. КВАЗИТРЕХМЕРНАЯ МОДЕЛЬ (ИЗЛУЧЕНИЕ И КОНВЕКЦИЯ ПО ТРЕТЬЕЙ КООРДИНАТЕ)

Проанализируем теплоперенос в пластине толщины h (рис. 30). Учтем перенос теплоты конвекцией и излучением с двух сторон пластины (по третьей координате). При этом уравнение теплопроводности является двумерным. Такой подход при решении многих пространственных задач позволяет значительно сократить время счета, но при этом получить достаточно точные результаты.

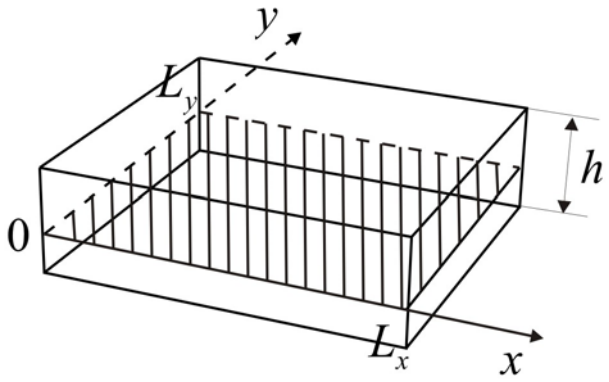


Рис. 30. Область решения

Пластина с размерами $L_x = L_y = 0.3$ м, $h = 0.1$ м. Материал пластины – дерево (дуб) (λ = 0.3 Вт/(м·ºС), ρ = 800 кг/м^3, c = 2400 Дж/(кг·ºС)). Начальная температура области решения $T_0 = 25\ ^0\text{C}$. $\kappa = 50\ \text{Вт}/\left(\text{м}^2 \cdot^0 \text{С}\right)$, $T^e = 40\ ^0\text{C}$, $\varepsilon = 0.3$, $T_h = 50\ ^0\text{C}$, $T_c = 10\ ^0\text{C}$.

Математическая постановка задачи будет иметь вид:

$$\rho c \frac{\partial T}{\partial t} = \lambda\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) + 2\kappa\frac{T^e - T}{h} + 2\varepsilon\sigma\frac{\left(T^e\right)^4 - T^4}{h}, \quad \left|\begin{array}{l} 0 < x < L_x; \\ 0 < y < L_y. \end{array}\right. \tag{68}$$

Начальные и граничные условия запишутся следующим образом:

$$\begin{aligned} &t = 0:\ T = T_0, 0 \le x \le L,\ 0 \le y \le H; \\ &x = 0:\ T = T_h,\ t > 0; \\ &x = L_x:\ T = T_c,\ t > 0; \\ &y = 0:\ \frac{\partial T}{\partial y} = 0,\ t > 0; \\ &y = L_y:\ \frac{\partial T}{\partial y} = 0,\ t > 0; \end{aligned} \tag{69}$$

Дискретизацию уравнения (68) будем проводить на основе локально одномерной схемы А.А. Самарского [2].

В результате получим:

$$\begin{aligned} &\rho \cdot c \cdot \frac{T_{i,j}^{n+\frac{1}{2}} - T_{i,j}^{n}}{\tau} = \lambda \cdot \left(\frac{T_{i+1,j}^{n+\frac{1}{2}} - 2 \cdot T_{i,j}^{n+\frac{1}{2}} + T_{i-1,j}^{n+\frac{1}{2}}}{h_x^2}\right) + \kappa\frac{T^e - T_{i,j}^{n+\frac{1}{2}}}{h} \\ &+ \varepsilon\sigma\frac{\left(T^e\right)^4 - \left(T_{i,j}^{n+\frac{1}{2}}\right)^4}{h}, \end{aligned} \tag{70}$$

$$\begin{aligned} &\rho \cdot c \cdot \frac{T_{i,j}^{n+1} - T_{i,j}^{n+\frac{1}{2}}}{\tau} = \lambda \cdot \left(\frac{T_{i,j+1}^{n+1} - 2 \cdot T_{i,j}^{n+1} + T_{i,j-1}^{n+1}}{h_y^2}\right) + \kappa\frac{T^e - T_{i,j}^{n+1}}{h} \\ &+ \varepsilon\sigma\frac{\left(T^e\right)^4 - \left(T_{i,j}^{n+1}\right)^4}{h}. \end{aligned} \tag{71}$$

Разностные уравнения (70), (71) сводятся к стандартному трехдиагональному виду и решаются последовательно методом прогонки (пункт 2.1). Сначала для всей области решается уравнение

(70), после того как его решение будет найдено, переходят к решению уравнения (71).

Рассмотрим решение уравнения (70) методом прогонки. Приведем это уравнение к виду $A_i T_{i+1,j}^{n+\frac{1}{2}} - B_i T_{i,j}^{n+\frac{1}{2}} + C_i T_{i-1,j}^{n+\frac{1}{2}} = F_i$. Тогда коэффициенты A_i, B_i, C_i примут вид:

$$A_i = C_i = \frac{\lambda}{h_x^2}, \quad B_i = \frac{2 \cdot \lambda}{h_x^2} + \frac{\kappa}{h} + \frac{\rho \cdot c}{\tau}, \quad F_i = -\frac{\rho \cdot c \cdot T_{i,j}^n}{\tau} - \frac{\kappa T^{\mathrm{e}}}{h}$$

$$- \varepsilon\sigma \frac{\left(T^{\mathrm{e}}\right)^4 - \left(T_{i,j}^{n+\frac{1}{2}}\right)^4}{h}.$$

Для определения прогоночных коэффициентов по соотношению (8) необходимо найти α_1 и β_1 из левого граничного условия. Далее определяя значение $T_{N_x,j}^{n+\frac{1}{2}}$ из правого граничного условия, находят поле температуры $T_{i,j}^{n+\frac{1}{2}}$ на промежуточном временном слое по формулам (7). При этом, поскольку F_i является функцией $T_{i,j}^{n+\frac{1}{2}}$, то необходимо воспользоваться методом простой итерации, аналогично пункту 3.4. После этого приступают к решению уравнения (71). Этапы решения уравнения (71) аналогичны решению уравнения (70). А именно, приведем уравнение (71) к виду $A_j T_{i,j+1}^{n+1} - B_j T_{i,j}^{n+1} + C_j T_{i,j-1}^{n+1} = F_j$. Тогда коэффициенты A_j, B_j, C_j примут вид:

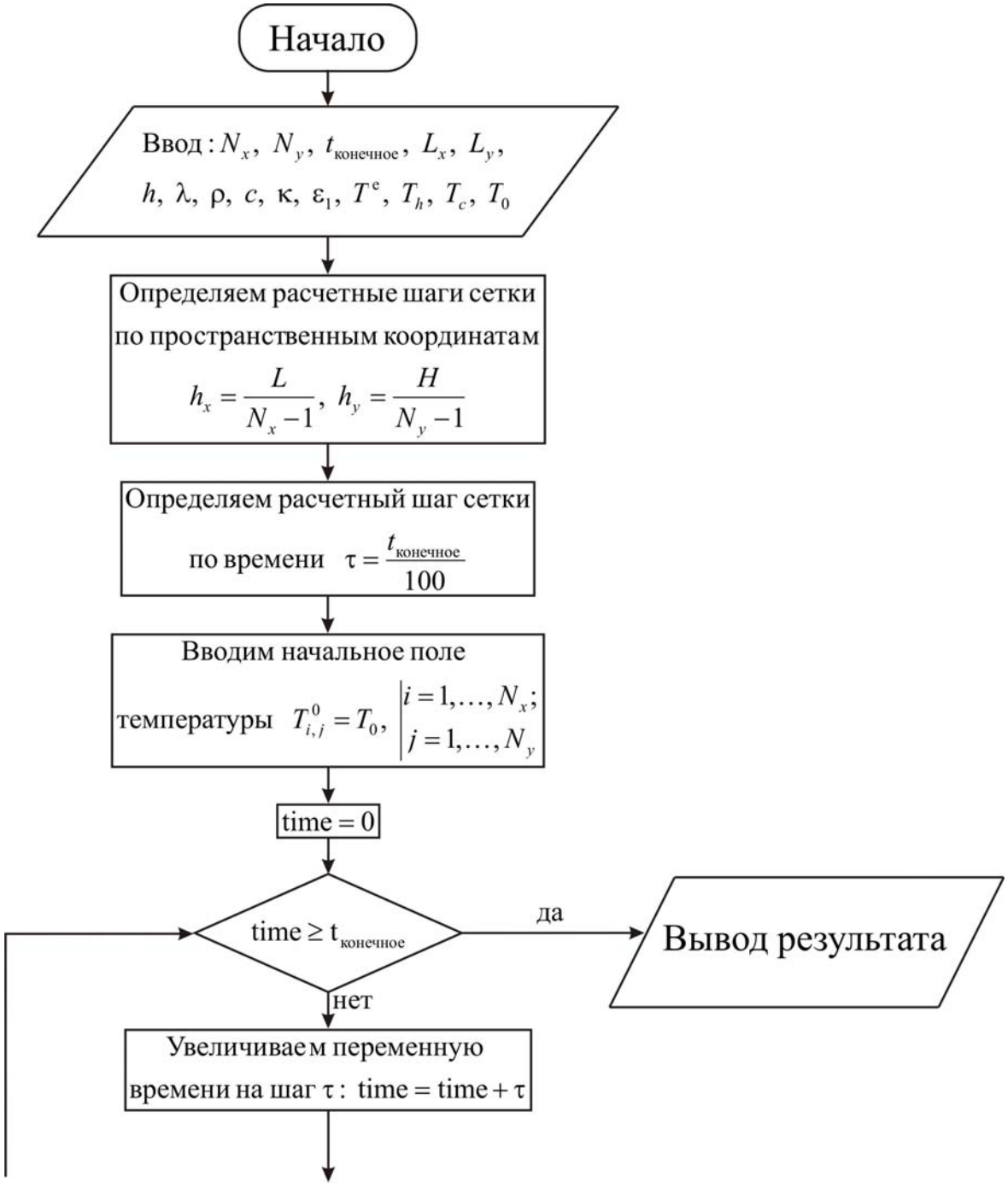
$$A_j = C_j = \frac{\lambda}{h_y^2}, \quad B_j = \frac{2 \cdot \lambda}{h_y^2} + \frac{\kappa}{h} + \frac{\rho \cdot c}{\tau}, \quad F_j = -\frac{\rho \cdot c \cdot T_{i,j}^{n+\frac{1}{2}}}{\tau} - \frac{\kappa T^{\mathrm{e}}}{h}$$

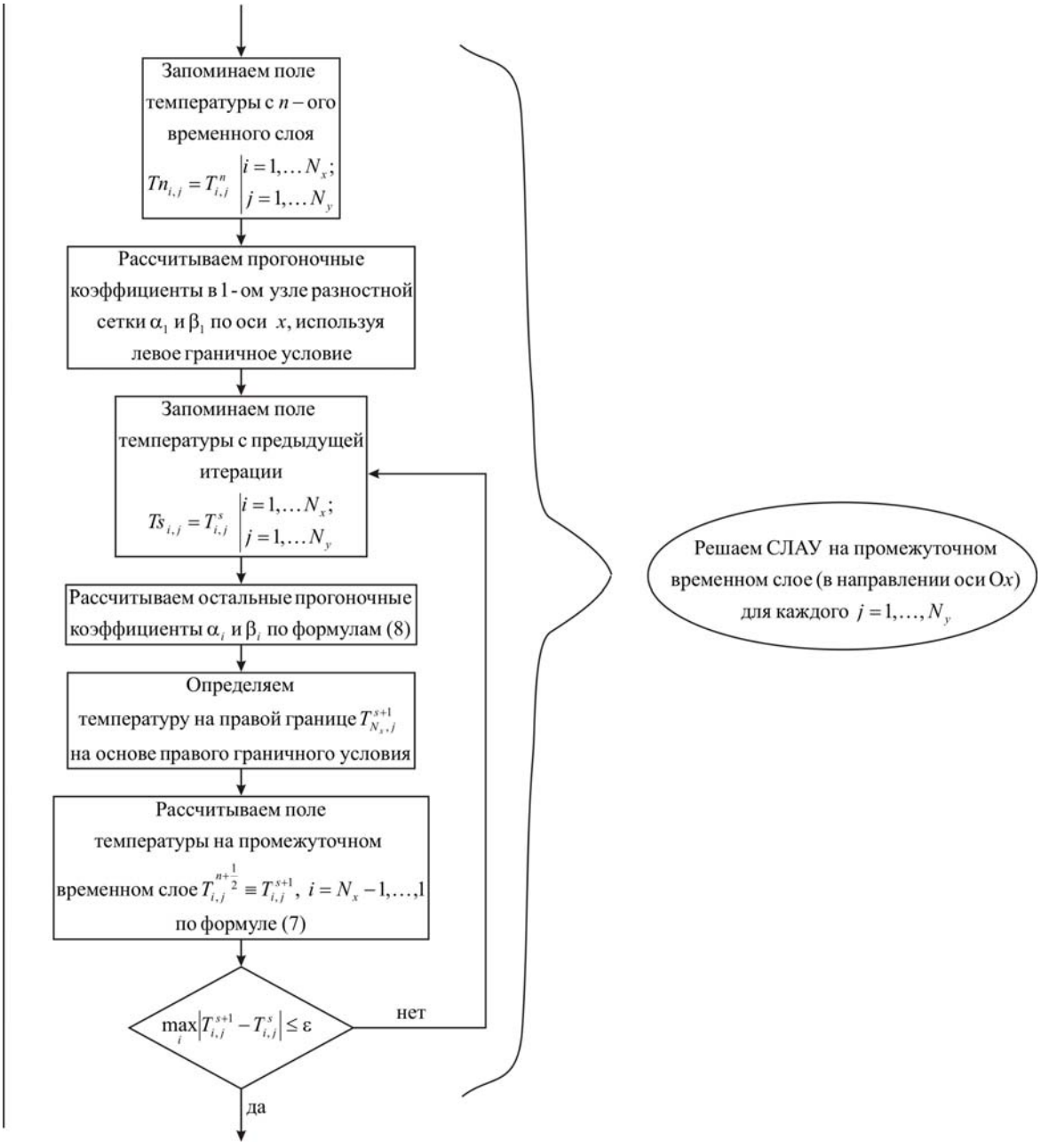
$$- \varepsilon\sigma \frac{\left(T^{\mathrm{e}}\right)^4 - \left(T_{i,j}^{n+1}\right)^4}{h}.$$

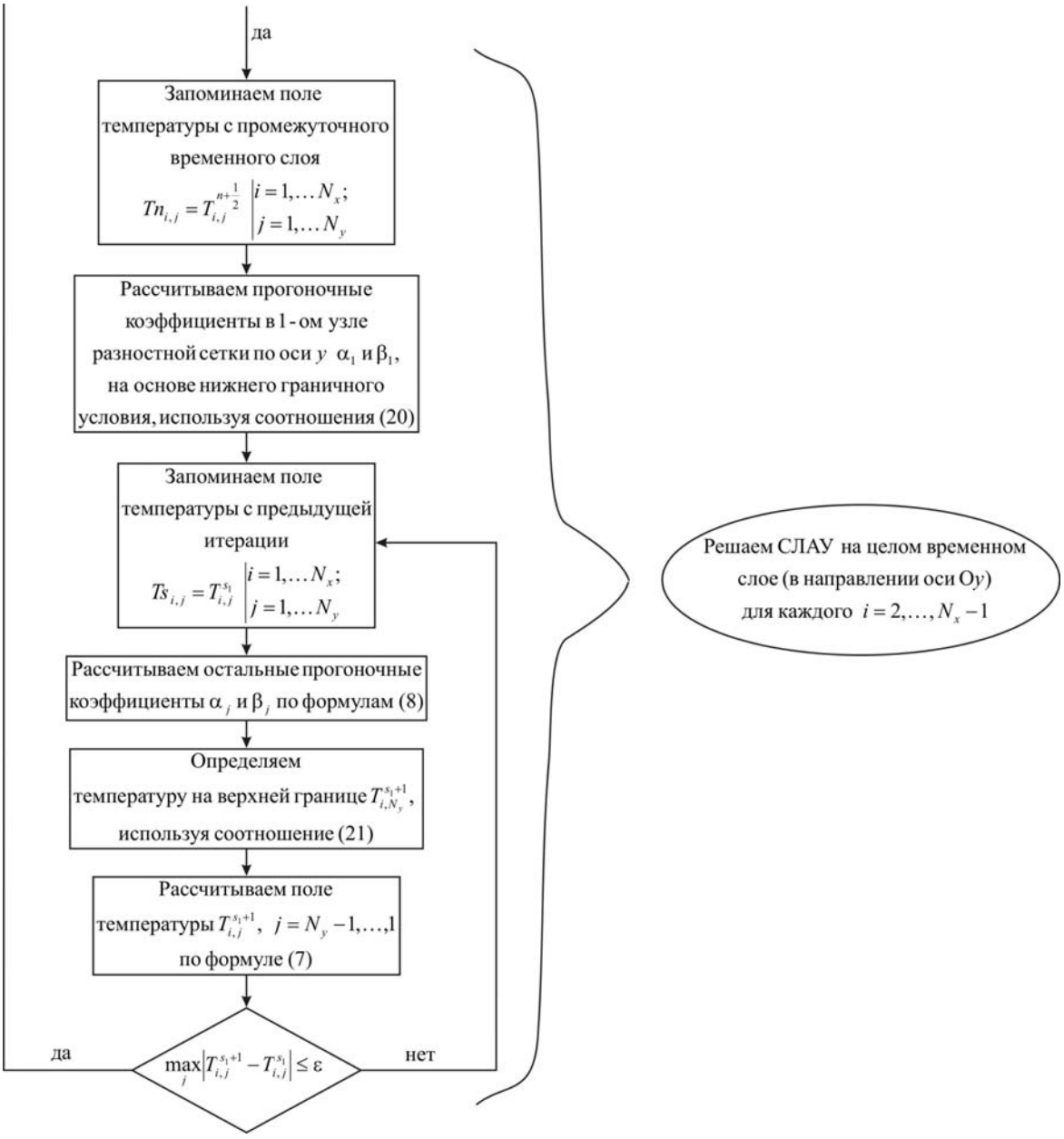
Для определения прогоночных коэффициентов по соотношению (8) необходимо найти α_1 и β_1 из левого граничного условия. Далее определяя значение T_{i,N_y}^{n+1} из правого граничного условия, находят поле температуры $T_{i,j}^{n+1}$ на целом временном слое по формулам (7). При этом,

поскольку F_i является функцией $T_{i,j}^{n+1}$, то также необходимо воспользоваться методом простой итерации.

Блок-схема к рассматриваемой задаче имеет вид:




Запоминаем поле температуры с $n-$ого временного слоя
$Tn_{i,j} = T^n_{i,j} \begin{vmatrix} i = 1,\dots N_x; \\ j = 1,\dots N_y \end{vmatrix}$
Рассчитываем прогоночные коэффициенты в 1 - ом узле разностной сетки α_1 и β_1 по оси x, используя левое граничное условие
Запоминаем поле температуры с предыдущей итерации
$Ts_{i,j} = T^s_{i,j} \begin{vmatrix} i = 1,\dots N_x; \\ j = 1,\dots N_y \end{vmatrix}$
Рассчитываем остальные прогоночные коэффициенты α_i и β_i по формулам (8)
Определяем температуру на правой границе $T^{s+1}_{N_x,j}$ на основе правого граничного условия
Рассчитываем поле температуры на промежуточном временном слое $T^{n+\frac{1}{2}}_{i,j} \equiv T^{s+1}_{i,j}$, $i = N_x - 1,\dots,1$ по формуле (7)
$\max_i \left| T^{s+1}_{i,j} - T^s_{i,j} \right| \le \varepsilon$
нет
да
Решаем СЛАУ на промежуточном временном слое (в направлении оси Ox) для каждого $j = 1,\dots, N_y$


да
Запоминаем поле температуры с промежуточного временного слоя
$Tn_{i,j} = T_{i,j}^{n+\frac{1}{2}} \mid i = 1,\dots N_x;\ j = 1,\dots N_y$
Рассчитываем прогоночные коэффициенты в 1-ом узле разностной сетки по оси y α_1 и β_1, на основе нижнего граничного условия, используя соотношения (20)
Запоминаем поле температуры с предыдущей итерации
$Ts_{i,j} = T_{i,j}^{s_1} \mid i = 1,\dots N_x;\ j = 1,\dots N_y$
Рассчитываем остальные прогоночные коэффициенты α_j и β_j по формулам (8)
Определяем температуру на верхней границе $T_{i,N_y}^{s_1+1}$, используя соотношение (21)
Рассчитываем поле температуры $T_{i,j}^{s_1+1}$, $j = N_y - 1,\dots,1$ по формуле (7)
$\max_j \left|T_{i,j}^{s_1+1} - T_{i,j}^{s_1}\right| \le \varepsilon$
да
нет
Решаем СЛАУ на целом временном слое (в направлении оси Oy) для каждого $i = 2,\dots,N_x - 1$

Ниже приведен листинг программы для решения рассматриваемой задачи *(на языке программирования Pascal)*

```
uses crt;
const mf=51;
        eps=1e-5;
        sigma=5.669e-8;
type
    vector1=array[1..mf] of real;
    vector2=array[1..mf,1..mf] of real;
var {раздел описания переменных, которые мы будем использовать в программе}
 i, j, Nx, Ny, k, l              : integer;
 T, Ts, Tn                       : vector2;
 alfa, beta                      : vector1;
 ai, bi, ci, fi, max             : real;
 lamda, ro, c, eps1, kapa        : real;
 hx, hy, tau, t_end, time        : real;
 T0, Te, Lx, Ly, h, Th, Tc       : real;
 f, g                            : text;
begin
 clrscr;
{с клавиатуры вводим все необходимые входные параметры}
 Writeln('Введите количество пространственных узлов в пластине по оси x, Nx');
 Readln(Nx);
 Writeln('Введите количество пространственных узлов в пластине по оси y, Ny');
 Readln(Ny);
 Writeln('Введите окончание по времени, t_end');
 Readln(t_end);
 Writeln('Введите длину пластины, Lx');
 Readln(Lx);
 Writeln('Введите ширину пластины, Ly');
 Readln(Ly);
 Writeln('Введите толщину пластины, h');
 Readln(h);
 Writeln('Введите коэффициент теплопроводности материала пластины, lamda');
 Readln(lamda);
 Writeln('Введите плотность материала пластины, ro');
```

```
 Readln(ro);
 Writeln('Введите теплоемкость материала пластины, c');
 Readln(c);
 Writeln('Введите коэффициент теплообмена, kapa');
 Readln(kapa);
 Writeln('Введите приведенную степень черноты, eps1');
 Readln(eps1);
 Writeln('Введите температуру окружающей среды, Te');
 Readln(Te);
 Writeln('Введите температуру на границе x = 0, Th');
 Readln(Th);
 Writeln('Введите температуру на границе x = Lx, Tc');
 Readln(Tc);
 Writeln('Введите начальную температуру, T0');
 Readln(T0);
{определяем расчетные шаги сетки по пространственным
координатам}
 hx:=Lx/(Nx-1);
 hy:=Ly/(Ny-1);
{определяем расчетный шаг сетки по времени}
 tau:=t_end/100.0;
{записываем в файл входные параметры для проверки}
 Assign(f,'res.txt');
 Rewrite(f);
 Writeln(f,'Длина пластины Lx = ',Lx:6:4);
 Writeln(f,'Ширина пластины Ly = ',Ly:6:4);
 Writeln(f,'Толщина пластины h = ',h:6:4);
 Writeln(f,'Число узлов по пространственной координате x в пластине
Nx = ',Nx);
 Writeln(f,'Число узлов по пространственной координате y в пластине
Ny = ',Ny);
 Writeln(f,'Коэффициент теплопроводности материала пластины lamda =
',lamda:6:4);
 Writeln(f,'Плотность материала пластины ro = ',ro:6:4);
 Writeln(f,'Теплоемкость материала пластины c = ',c:6:4);
 Writeln(f,'Коэффициент теплообмена kapa = ',kapa:6:4);
 Writeln(f,'Приведенная степень черноты eps1 = ',eps1:6:4);
 Writeln(f,'Начальная температура T0 = ',T0:6:4);
 Writeln(f,'Температура окружающей среды Te = ',Te:6:4);
 Writeln(f,'Температура на границе x = 0 Th = ',Th:6:4);
 Writeln(f,'Температура на границе x = Lx Tc = ',Tc:6:4);
```

```
 Writeln(f,'Результат получен с шагом по координате x hx = ',hx:6:4);
 Writeln(f,'Результат получен с шагом по координате y hy = ',hy:6:4);
 Writeln(f,'Результат получен с шагом по времени tau = ',tau:6:4);
 Writeln(f,'Температурное поле в момент времени t = ',t_end:6:4);
 close(f);
{определяем поле температуры в начальный момент времени}
 for i:= 1 to Nx do
  for j:= 1 to Ny do
    T[i,j]:=T0;
{проводим интегрирование нестационарного уравнения
теплопроводности}
 time:=0;
 while time<t_end do {используем цикл с предусловием}
  begin
{увеличиваем переменную времени на шаг τ}
   time:=time+tau;
{запоминаем поле температуры на предыдущем n временном слое}
   for i:=1 to Nx do
    for j:=1 to Ny do
      Tn[i,j]:=T[i,j];
{решаем СЛАУ в направлении оси Ox для определения поля
температуры на промежуточном (n+1/2) временном слое}
   for j:=1 to Ny do
    begin
{определяем начальные прогоночные коэффициенты на основе левого
граничного условия}
     alfa[1]:=0.0;
     beta[1]:=Th;
{цикл с постусловием, позволяющий итерационно вычислять значения
температуры в области решения, вследствие наличия нелинейности в
самом уравнении теплопроводности}
     repeat
{запоминаем поле температуры на предыдущей s итерации}
      for k:=1 to Nx do
       for l:=1 to Ny do
        Ts[k,l]:=T[k,l];
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
      for i:= 2 to Nx-1 do
       begin
```

```
{ai, bi, ci, fi – коэффициенты канонического представления систему
уравнений с трехдиагональной матрицей}
       ai:=lamda/sqr(hx);
       bi:=2.0*lamda/sqr(hx)+kapa/h+ro*c/tau;
       ci:=lamda/sqr(hx);
       fi:=-ro*c*Tn[i,j]/tau-kapa*Te/h-eps1*sigma
            *(sqr(sqr(Te))-sqr(sqr(T[i,j])))/h;
{alfa[i], beta[i] – прогоночные коэффициенты}
       alfa[i]:=ai/(bi-ci*alfa[i-1]);
       beta[i]:=(ci*beta[i-1]-fi)/(bi-ci*alfa[i-1]);
      end;
{определяем значение температуры на правой границе на основе
правого граничного условия}
     T[Nx,j]:=Tc;
{используя соотношение (7) определяем неизвестное поле температуры
на промежуточном (n+1/2) временном слое}
     for i:= Nx-1 downto 1 do
      T[i,j]:=alfa[i]*T[i+1,j]+beta[i];
{определяем максимум модуля разности локальных значений
температур на текущей и на предыдущей итерации}
     max:=abs(T[1,j]-Ts[1,j]);
     for i:=2 to Nx do
      if max < abs(T[i,j]-Ts[i,j]) then max:=abs(T[i,j]-Ts[i,j]);
    until max<=eps; {окончание цикла с постусловием}
   end; {поле температуры на промежуточном (n+1/2) временном слое
определили}
{запоминаем поле температуры на промежуточном (n+1/2) временном
слое}
  for i:=1 to Nx do
   for j:=1 to Ny do
    Tn[i,j]:=T[i,j];
{решаем СЛАУ в направлении оси Oy для определения поля
температуры на целом (n+1) временном слое}
  for i:=2 to Nx-1 do
   begin
{определяем начальные прогоночные коэффициенты на основе нижнего
граничного условия, используя соотношения (20) при условии, что
q1 = 0}
    alfa[1]:=2.0*lamda*tau/(ro*c*sqr(hy)+2.0*lamda*tau);
    beta[1]:=ro*c*sqr(hy)*Tn[i,1]/(ro*c*sqr(hy)+2.0*lamda*tau);
```

```
{цикл с постусловием, позволяющий итерационно вычислять значения
температуры в области решения, вследствие наличия нелинейности в
самом уравнении теплопроводности}
    repeat
{запоминаем поле температуры на предыдущей s итерации}
     for k:=1 to Nx do
      for l:=1 to Ny do
       Ts[k,l]:=T[k,l];
{цикл с параметром для определения прогоночных коэффициентов по
формуле (8)}
     for j:= 2 to Ny-1 do
      begin
{ai, bi, ci, fi – коэффициенты канонического представления систему
уравнений с трехдиагональной матрицей}
       ai:=lamda/sqr(hy);
       bi:=2.0*lamda/sqr(hy)+kapa/h+ro*c/tau;
       ci:=lamda/sqr(hy);
       fi:=-ro*c*Tn[i,j]/tau-kapa*Te/h-eps1*sigma
           *(sqr(sqr(Te))-sqr(sqr(T[i,j])))/h;
{alfa[j], beta[j] – прогоночные коэффициенты}
       alfa[j]:=ai/(bi-ci*alfa[j-1]);
       beta[j]:=(ci*beta[j-1]-fi)/(bi-ci*alfa[j-1]);
      end;
{определяем значение температуры на верхней границе, используя
соотношение (21) при условии, что q2 = 0}
     T[i,Ny]:=(2.0*lamda*tau*beta[Ny-1]+ro*c*sqr(hy)*Tn[i,Ny])
                /(ro*c*sqr(hy)+2.0*lamda*tau*(1.0-alfa[Ny-1]));
{используя соотношение (7) определяем неизвестное поле температуры
на целом (n+1) временном слое}
     for j:= Ny-1 downto 1 do
      T[i,j]:=alfa[j]*T[i,j+1]+beta[j];
{определяем максимум модуля разности локальных значений
температур между текущей и предыдущей итерациями}
     max:=abs(T[i,1]-Ts[i,1]);
     for j:=2 to Ny do
      if max < abs(T[i,j]-Ts[i,j]) then max:=abs(T[i,j]-Ts[i,j]);
    until max<=eps; {окончание цикла с постусловием}
   end; {поле температуры на целом (n+1) временном слое определили}
 end; {цикл с предусловием окончен}
{выводим результат в файл}
 Assign(g,'tempr.txt');
```

```
 Rewrite(g);
 for i:=1 to Nx do
  for j:=1 to Ny do
   writeln(g,'  ',hx*(i-1):10:8,'  ',hy*(j-1):10:8,'   ',T[i,j]:8:5);
 close(g);
end.
```

Получено следующее распределение температуры:

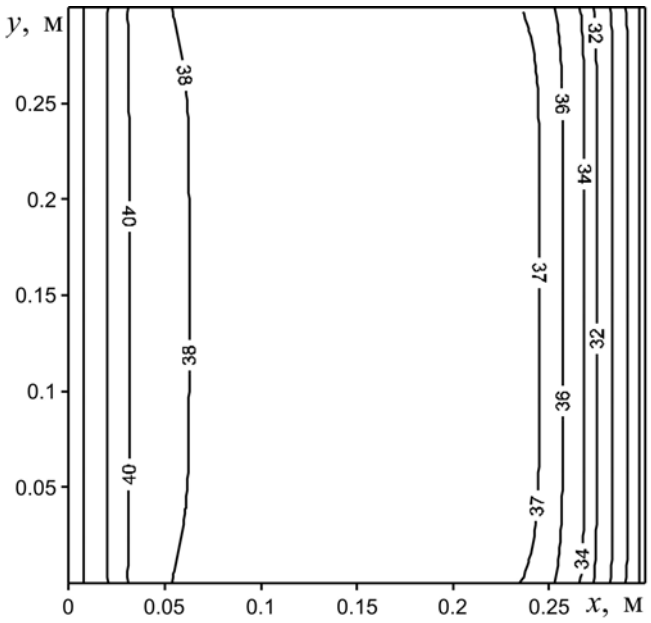


Рис. 31. Распределение температуры (ºС) в пластине при t = 3600 с

Список литературы

1. Вержбицкий В.М. Основы численных методов. – М.: Высшая школа, 2002. – 840 с.
2. Самарский А.А. Теория разностных схем. – М.: Наука, 1977. – 656 с.
3. Березин И.С., Жидков Н.П. Методы вычислений. – М.: Наука, 1962. – Т. 1. – 464 с.
4. Березин И.С., Жидков Н.П. Методы вычислений. – М.: Наука, 1962. – Т. 2. – 639 с.
5. Калиткин Н.Н. Численные методы. – М.: Наука, 1978. – 512 с.
6. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. – М.: Наука, 1987. – 598 с.
7. Волков Е. А. Численные методы. – М.: Наука, 1987. – 248 с.
8. Турчак Л. И. Основы численных методов. – М.: Наука, 1987. – 318 с.
9. Копченова Н. В., Марон И. А. Вычислительная математика в примерах и задачах. – М.: Наука, 1972. – 367 с.
10. Крылов В. И., Бобков В. В., Монастырный П. И. Вычислительные методы. – М.: Наука, 1976. – Т. 1. – 302 с.
11. Демидович Б. П., Марон И. А. Основы вычислительной математики. – М.: Наука, 1966. – 695 с.
12. Петухов Б.С., Генин Л.Г., Ковалев С.А., Соловьев С.Л. Теплообмен в ядерных энергетических установках. – М.: Изд-во МЭИ, 2003. – 548 с.
13. Лыков А.В. Теория теплопроводности. – М.: Высшая школа, 1967. – 600 с.
14. Лыков А.В. Тепломассообмен: (Справочник). – М.: Энергия, 1978. – 480 с.

15. Самарский А.А., Вабищевич П.Н. Вычислительная теплопередача. – М.: Едиториал УРСС, 2003. – 782 с.
16. Берковский Б.М., Ноготов Е.Ф. Разностные методы исследования задач теплообмена. – Минск: Наука и техника, 1976. – 141 с.