Оглавление

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Изучение окружающего мира является одним из важнейших направлений для человечества. На смену менее точным методам, таким как наблюдение, математический расчет и экспериментальное подтверждение пришли современные методы математического моделирования, позволяющие проследить изменение состояния мира через время или смоделировать интересующий процесс, воссоздание которого в реальности очень сложно либо ограничено технически и финансово.

Несмотря на огромные возможности математического моделирования человек не способен в полной мере их использовать из-за естественных физических ограничений, таких как человеческая внимательность и концентрация, ограниченность ресурсов, например, бумага и чернила, а также ограниченность времени.

С задачей математического моделирования в современном мире хорошо справляются информационные технологии. С учетом стремительного роста мощностей компьютеров можно сказать, что скорость выполнения расчетов будет повышаться.

Основной проблемой для изучения мира путем математического моделирования с использованием информационных технологий является внутренняя компьютеров архитектура, которая не может производить сложные математические вычисления интегралов, дифференциальных уравнений, уравнений в частных производных, а также выполнять сложные математические преобразования функций.

Для устранения этой проблемы используются численные методы, позволяющие переводить уравнения в вид, понятный компьютеру тем самым давая человечеству возможность использовать информационные технологии для изучения окружающего мира путем моделирования сложных систем.

Несмотря на высокую популярность численных методов в области компьютерного моделирования, они были разработаны задолго до появления первых компьютерных систем и используется для решения различных математических задач.

Темой курсовой работы является «Анализ распределения температур в стержне с боковым теплообменом и источником тепла», которая позволяет в полной мере изучить некоторые числовые методы. Результаты работы могу быть использованы во множестве областей, например, в строительстве как анализ теплоизоляции помещения от окружающей среды.

Целью курсовой работы является разработка программного средства для анализа распределения температур в стержне с боковым теплообменом.

Для достижения цели были поставлены следующие задачи:

1. Изучить численные методы в механике.
2. Изучить способы решения численных методов.
3. Разработать алгоритм решения задачи анализа распределения температур в стержне с боковым теплообменом и источником тепла.

4. Реализовать программу для решения поставленной задачи позволяющую выводить результат расчетов в графическом виде.

5. Сравнить результат работы программы с результатом решения одной из ведущих программ, представленных на современном рынке систем математического анализа.

1 ОБЗОР ЧИСЛЕННЫХ МЕТОДОВ МОДЕЛИРОВАНИЯ В МЕХАНИКЕ

1.1 Общая характеристика численных методов

Одним из доступных способов познания мира, в частности явлений, процессов и т.д., является математическое моделирование изучаемого объекта. Математическая модель является совокупностью уравнений, связывающих между собой параметры, характеризующие состояние исследуемого явления. В результате анализа составляется конкретная математическая задача, другими словами – набор уравнений, подлежащих решению.

За счет активного развития компьютерных технологий для решения анализа математических моделей прибегают к численным методам, позволяющим автоматизировать процесс.

Численные методы – это методы решения математических задач в численном виде, представление как исходных данных в задаче, так и её решения в виде числа или набора чисел.

Основами для вычислительных методов являются:

- решение систем линейных уравнений;
- интерполирование и приближённое вычисление функций;
- численное интегрирование;
- численное решение системы нелинейных уравнений;
- численное решение обыкновенных дифференциальных уравнений;
- численное решение уравнений в частных производных (уравнений математической физики);
- решение задач оптимизации.

Область математики, изучающая численные методы описывает основные алгоритмы построения моделей окружающего мира, а также способы его изучения, позволяя строить модели сложнейших физических, технических и биологических процессов.

Наблюдая за окружающим миром человек описывает их вербально, пытаясь понять суть явления, находя закономерности, после чего строиться математическая модель.

Однако расчет и анализ построенных моделей человеком займет огромное количество времени, кроме этого не исключена вероятность появления ошибок. Компьютерные технологии позволяют получить качественное и количественное представления об изучаемом процессе, увидеть в графическом виде.

Для примера возьмем такой сложны биологический процесс как кровообращение.

Сформулировав основные законы, которым подчиняется данный процесс, выделив закономерности получим вербальную модель.

После чего, выразив выявленные законы в математических уравнениях, таких как дифференциальные или интегральные уравнения, необходимо их решить.

Для решения дифференциальных или интегральных уравнений с помощью компьютера необходимо их преобразовать в численный вид. В результате решения уравнений компьютер позволит нам увидеть тот же процесс, но в виде чисел. Которые могут быть использованы для визуализации процесса на экране.

Большинство процессов мироздания описывается нелинейными уравнениями, которые лишь в первом приближении можно заменить линейными.

В редких случаях можно решить математические уравнения, описывающие реальные процессы, в явном виде. В некоторых случаях можно доказать единственность решения уравнения, но о свойствах явления, и о том, где находится решение в явном виде можно понять лишь в процессе наблюдения за поведением объекта исследования.

Важность решений в явном виде проявляется при моделировании молекулярных процессов, в частности – отдельных молекул.

Под численными методами в широком смысле можно понимать интерпретацию математической модели на языке, доступном компьютеру, например, численным дифференциального уравнения является разностное уравнение, приближающее исходное дифференциальное.

Из-за того, что численные методы приближают исходные уравнения, то решение является приближенным.

Для решения задачи на компьютере необходимо написать программу, реализующую необходимый численный метод. Важным аспектом при использовании компьютера является то, что всегда есть вероятность обнаружить свойства, о которых исследователь не догадывался, либо, подтвердить наличие предполагаемого.

В результате анализа темы видно, что основу вычислительного эксперимента составляют: математическая модель, метод и алгоритм решения, компьютерная программа.

При отсутствии хотя бы одной части вычислительного эксперимента провести глубокое исследование невозможно.

Важно заметить, что одной и той же модели можно сопоставить множество разнообразных дискретных моделей, однако не все они подходят для практического использования.

Вычислительные алгоритмы должны удовлетворять установленным требованиям. Дискретная модель должна быть комплементарна компьютеру.

Выделяют две основные группы требований к вычислительным алгоритмам: адекватность исходной задачи и возможность реализации на компьютере.

Адекватность включает в себя сходимость метода, выполнение математических аналогов законов сохранения, качественно правильное поведение метода, его соответствие модели.

Численный метод называют сходящимся, если при увеличении числа уравнений решение дискретной задачи стремится к решению исходной задачи.

При моделировании задачи важно помнить, что компьютер работает лишь с конкретным набором уравнений, поэтому необходимо оценивать погрешность дискретной модели в зависимости от числа уравнений.

Корректность численного метода должна соответствовать корректности исходной задачи, иными словами, однозначной разрешимости и непрерывной зависимости от исходных данных.

Компьютерная реализация метода включает требования по памяти и времени, метод должен быть реализуем за определенное время на данном компьютере с учетом его быстродействия и памяти.

1.2 Погрешности в численных методах

Численные методы являются методами для поиска приближенного решения задач, а значит – результат вычислений всегда имеет некоторую погрешность и чем она меньше, тем точнее метод. Существует несколько видов погрешностей.

Погрешность модели. Современный мир слишком сложен для изучения во всем многообразии, во всей полноте его взаимосвязей, в том числе и незначительных. Любая наука, изучающая природу, не изучает ее напрямую, как было сказано ранее, изучает модели, созданные на основе процесса или явления. Простыми словами модель – это идеальное описание явления, не включающее в себя незначительные свойства и становится очевидно, что процесс моделирования сопровождается упрощением явления, а значит – добавлением погрешностей в результат описания модели.

Математическая модель создается на языке математики, но оценка погрешности математической модели есть прерогатива не математики, а той науки, в рамках которой изучается явление.

Погрешность исходных данных. Как правило, математическая модель содержит некоторые параметры, зависящие от исходных данных, которые определяются экспериментальным путем, а значит – являются не точными.

Погрешности в решении, обусловленные моделированием и исходными данными, называются неустранимыми. Они не зависят от математики и присутствуют, даже если решение поставленной математической задачи найдено точно.

Погрешность метода. После создания математической модели анализировать ее можно множеством способов, при этом сложные математические задачи, созданные на основе модели заменяются более простыми, что приводит к возникновению погрешностей метода вычисления.

Погрешность округления. Любые математические вычисления, в особенности в области изучения математической модели, производятся с использованием не целых чисел. Количество знаков после запятой, в десятичной дроби, может быть очень большим, особенно в случаях с иррациональными числами. Как компьютерные системы, так и человек ограничены в вычислениях этим показателем, компьютеры – на уровне архитектуры, человек – на уровне времени необходимого для проведения расчетов. В связи с этим возникает

необходимость в округлении как промежуточных, так и результирующих значений.

Главная проблема погрешности округления заключается в том, что этот вид погрешности может накапливаться в процессе вычислений, а значит – обесценить результат в случае, если погрешность будет слишком большой.

Даже те результаты, которые получены точными аналитическими методами, подвержены погрешности округлений и могут оказаться приближенными.

Полная погрешность не может быть меньше, чем наибольшая из составляющих ее погрешностей, т.к. она является результатом взаимодействия разных видов погрешностей.

Для оценки погрешности вводятся понятия абсолютной и относительной погрешности.

Пусть x – точное значение некоторой величины; a – приближенное значение той же величины (a ≈ x). Абсолютная погрешность приближенного числа a определяется формулой $\Delta_a = |x - a|$ но т.к. x неизвестно, то и абсолютная погрешность неизвестна. Для решения этой задачи вводится понятие предельной абсолютной погрешности Δ_a^* – это значение, которое абсолютная погрешность не превзойдет при выбранном способе измерений т.е. $|x - a| \leq \Delta_a^*$.

Из предыдущего выражения следует, что $a - \Delta_a^* \leq x \leq a + \Delta_a^*$, поэтому наименьшее значение Δ_a^* уменьшит длину интервала, содержащего искомое значение x и, следовательно, понизит неопределенность в знаниях об этой величине.

В технике формулу предельной абсолютной погрешности часто записывают в виде $x = a \pm \Delta_a^*$, причем Δ_a^* называется допусками. Никакое изделие не может быть изготовлено с абсолютно точным соблюдением номинальных размеров, допуски показывают возможные (допустимые) отклонения от номинала.

Абсолютная погрешность оценивает точность измерений, но без учета размера изучаемого объекта она является не полной. Абсолютная погрешность в 1 см при измерении габаритного объекта, например, комнаты, является незначительной, в то время как при измерении человека является грубой.

Более информативным показателем качества измерений является относительная погрешность δ_a приближенного числа a как отношение абсолютной погрешности к модулю числа a

Относительная погрешность является величиной безразмерной, т. е. не зависит от выбора системы единиц измерения, что позволяет сравнивать качество измерений разнородных величин, измеряется в долях единицы или в процентах.

1.3 Численные методы в механике

Механика – раздел физики, наука, изучающая движение материальных тел и взаимодействие между ними; при этом движением в механике называют изменение во времени взаимного положения тел или их частей в пространстве.

Численные методы в механике применяются во множестве областей:

1. Аэромеханика.
2. Вычислительная механика.
3. Газовая и волновая динамика.
4. Геометрия и топология.
5. Гидромеханика.
6. Механика композитов.
7. Прикладная механика.
8. Теоретическая механика и мехатроника.
9. Теория пластичности.
10. Теория упругости.
11. Термодинамика.
12. и д.р.

В рамках курсовой работы изучаемой областью является термодинамика. В ней, с помощью численных методов, решаются как задачи теплопроводности отдельных материалов, так и задачи анализа отвода тепла радиатором, анализа эффективности охлаждения, моделирования сопряженного теплообмена и т.д.

1.4 Методы решения численных методов в термодинамике

Для решения численных методов в термодинамике используется уравнение теплопроводности, которое, в общем виде, в пространстве с декартовыми координатами $x = (x_1, \ldots, x_n)$ имеет вид, представленный на рисунке 1.1.

$$\frac{\partial u}{\partial t} - a^2 \left(\frac{\partial^2 u}{\partial x_1^2} + \cdots + \frac{\partial^2 u}{\partial x_n^2} \right) = f(x, t)$$

Рисунок 1.1 – Уравнение теплопроводности в общем виде

где a – коэффициент теплопроводности, f(x, t) – функция тепловых источников.

Уравнение теплопроводности называют однородным если f(x, t) = 0, т.е. система является теплоизолированной.

Для различных задач применяются различные модификации уравнения.

1.4.1 Задача Коши для уравнения теплопроводности. Одной из основных задач теории дифференциальных уравнений является задача Коши. Суть задачи состоит в нахождении интеграла уравнения, удовлетворяющего некоторым начальным условиям.

Задача Коши составляется при анализе процессов, изменяемых во времени, например – распределение тепла в стержне. Этим обуславливается выбор

терминологии: начальные значения и анализируемые значения, т.е. изменяющиеся во времени.

От краевых задач задача Коши отличается тем, что область, в которой определено искомое решение заранее не указывается. Тем не менее, задачу Коши можно рассматривать как одну из краевых задач.

Уравнение задачи Коши для однородного уравнения теплопроводности имеет вид, представленный на рисунке 1.2:

$$\frac{\partial u}{\partial t} - a^2 \Delta u = 0$$
$$x \in R^n, t > 0, u(x,0) = \varphi(x), x \in R^n$$

Рисунок 1.2 – Уравнение задачи Коши для однородного уравнения теплопроводности

где $\varphi(x)$ – начальная функция, непрерывная и ограниченная на всем пространстве, а искомая функция $u = u(x,t)$ является непрерывной и ограниченной при $t \geq 0$ и всех значениях аргумента x.

Однородная задача Коши имеет несколько свойств:

– принцип максимума. Данный принцип говорит о том, что решение однородной задачи Коши удовлетворяет неравенству $inf\ \varphi <= u(x,t) <= sup\ \varphi$ при всех $x \in R$ и всех $t > 0$;

– теорема существования и единственности. Для любого T > 0 решение однородной задачи Коши существует, единственно и непрерывно зависит от начальной функции φ(x) в полосе $S = \{(x,t) | 0 \leq t \leq T, x \in R^n\}$. Другими словами, данная задача является корректно поставленной.

– интеграл Пуассона. В пространстве с декартовыми координатами решение однородной задачи Коши задается в виде интегральной формулы, называемой интегралом Пуассона. Именно $u(x,t)$ при всех $t > 0$ есть свертка по пространственной переменной x ядра с начальной функцией.

Интеграл Пуассона задает единственное непрерывное и ограниченное решение данной задачи Коши. На рисунке 1.4 представлено уравнения интеграла Пуассона.

$$u(x,t) = \int\limits_{R^n} \text{ф}(\text{ч} - \text{у}, t)\varphi(y)dy = \frac{1}{(2a\sqrt{\pi t})^n} \int\limits_{R^n} \exp\left(-\frac{|x-y|^2}{4a^2 t}\right)\varphi(y)dy$$

Рисунок 1.4 – Интеграл Пуассона

– физический парадокс. Из формулы Пуассона следует, что если начальная функция φ равна нулю всюду, за исключением некоторой ограниченной области, например, заданной условием $x < \epsilon$, в которой она положительна, то через сколь угодно малый промежуток времени t > 0 решение u(x, t) будет строго положительным во всех точках пространства. Отсюда следует парадоксальное с физической точки зрения утверждение, что тепло распространяется с бесконечной скоростью. Объяснение парадокса состоит в том, что уравнение теплопроводности не вполне описывает реальный физический процесс распространения тепла.

1.4.2 Одномерное уравнение теплопроводности. Метод разделения переменных (Метод Фурье). Для случая одной пространственной переменной x (задача о нагревании или охлаждении стержня) уравнение теплопроводности принимает вид, представленные на рисунке 1.5.

$$u_t - a^2 u_{xx} = f(x,t)$$

Рисунок 1.5 – Одномерное уравнение теплопроводности

Для этого уравнения можно ставить и решать различные краевые задачи, один из методов решения которых предложен французским математиком Фурье.

Метод разделения переменных – метод решения дифференциальных уравнений, основанный на алгебраическом преобразовании исходного уравнения к равенству двух выражений, зависящих от разных независимых переменных.

В применении к уравнениям в частных производных схема разделения переменных приводит к нахождению решения в виде ряда или интеграла Фурье. В этом случае метод также называют методом Фурье и методом стоячих волн.

1.4.3 Метод конечных разностей. Для решения задачи Коши можно использовать метод конечных разностей.

Метод конечных разностей – численный метод решения дифференциальных уравнений, основанный на замене производных разностными схемами.

Разностная схема – это конечная система алгебраических уравнений, поставленная в соответствие какой-либо дифференциальной задаче, содержащей дифференциальное уравнение и дополнительные условия (например, краевые условия и/или начальное распределение). Таким образом, разностные схемы применяются для сведения дифференциальной задачи, имеющей континуальный характер, к конечной системе уравнений, численное решение которых принципиально возможно на вычислительных машинах. Алгебраические уравнения, поставленные в соответствие дифференциальному уравнению, получаются применением разностного метода, что отличает теорию разностных схем от других численных методов решения дифференциальных задач (например, проекционных методов, таких как метод Галёркина).

Решение разностной схемы называется приближенным решением дифференциальной задачи.

Хотя формальное определение не накладывает существенных ограничений на вид алгебраических уравнений, но на практике имеет смысл рассматривать только те схемы, которые каким-либо образом отвечают дифференциальной задаче. Важными понятиями теории разностных схем являются понятия сходимости, аппроксимации, устойчивости, консервативности.

Решение задач методом конечных разностей, когда процесс изменяется во времени, представляет собой итерационный процесс — на каждой итерации мы находим решение на новом временном слое. Для решения таких задач используются явные, неявные схемы и предиктор-корректор (пара из специально подобранных явной и неявной схемы). Явные схемы и схемы предиктор-корректор просто пересчитывают значение, используя информацию с предыдущих временных слоёв, использование неявной схемы приводит к решению уравнения (или системы уравнений).

Для параболических и гиперболических уравнений часто прибегают к смешиванию методов – производные по времени аппроксимируют с помощью разностной схемы, а оператор по пространству аппроксимируется с помощью конечно элементной постановки.

Для решения дифференциальных уравнений методом конечных разностей используются явная и не явная схемы Эйлера.

Метод Эйлера – простейший численный метод решения систем обыкновенных дифференциальных уравнений. Впервые описан Леонардом Эйлером в 1768 году в работе «Интегральное исчисление». Метод Эйлера основан на аппроксимации интегральной кривой кусочно-линейной функцией, так называемой ломаной Эйлера, изображение которой представлено на рисунке 1.6.

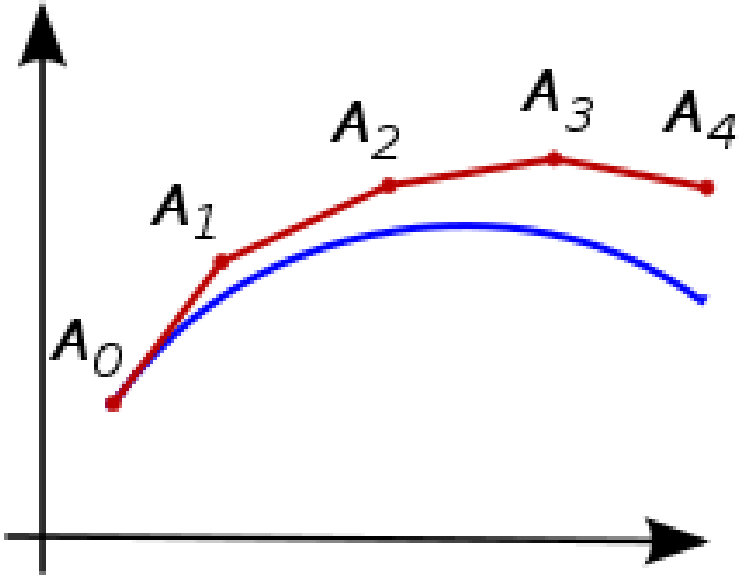

Рисунок 1.6 – Ломаная Эйлера в пяти точках

Метод Эйлера являлся исторически первым методом численного решения задачи Коши. О. Коши использовал этот метод для доказательства существования решения задачи Коши. Ввиду невысокой точности и вычислительной неустойчивости для практического нахождения решений задачи Коши метод Эйлера применяется редко. Однако в виду своей простоты метод Эйлера находит своё применение в теоретических исследованиях дифференциальных уравнений, задач вариационного исчисления и ряда других математических проблем.

1.4.4 Метод теплового баланса. Различные физические процессы (теплопроводности или диффузии, колебаний, газодинамики и т. д.) характеризуются некоторыми интегральными законами сохранения (тепла, массы, количества движения, энергии и т. д.). При выводе дифференциальных уравнений математической физики обычно исходят из некоторого интегрального соотношения (уравнения баланса), выражающего закон сохранения для малого объема. Дифференциальное уравнение получается из уравнения баланса при стягивании объема к нулю в предположении существования непрерывных производных, входящих в уравнение.

Метод конечных разностей физически означает переход от непрерывной среды к некоторой ее дискретной модели. При таком переходе естественно требовать, чтобы основные свойства физического процесса сохранялись. Такими свойствами, прежде всего, являются законы сохранения. Разностные схемы, выражающие на сетке законы сохранения, называют консервативными (или дивергентными). Законы сохранения для всей сеточной области («интегральные законы сохранения») для консервативных схем должны быть алгебраическим следствием разностных уравнений.

Для получения консервативных разностных схем естественно исходить из уравнений баланса, записанных для элементарных объемов (ячеек) сеточной области. Входящие в эти уравнения баланса интегралы и производные следует заменить приближенными разностными выражениями. В результате получаем однородную разностную схему. Такой метод получения консервативных однородных разностных схем называется интегро-интерполяционным методом (методом баланса). На рисунке 1.7 представлено уравнение теплового баланса на отрезке.

$$W_{i-1/2} - W_{i+1/2} + \int_{x_{i-1/2}}^{x_{i+1/2}} f(x)dx = \int_{x_{i-1/2}}^{x_{i+1/2}} q(x)u(x)dx, W = -ku'$$

Рисунок 1.7 – Уравнение теплового баланса на отрезке

2 АЛГОРИТМИЧЕСКИЙ АНАЛИЗ ЗАДАЧИ

Согласно теме курсовой работы, необходимо проанализировать распределение температур в стержне с боковым теплообменом и источником теплоты. Составим краевую задачу на основе одномерного уравнения теплопроводности. Начальная температура стержня равна T_0, на левой границе стержня коэффициент теплоотдачи равен a1, на правой – a2. Объемный тепловой поток Qv распределен по всей площади объекта; На рисунке 2.1. представлен общий вид задачи.

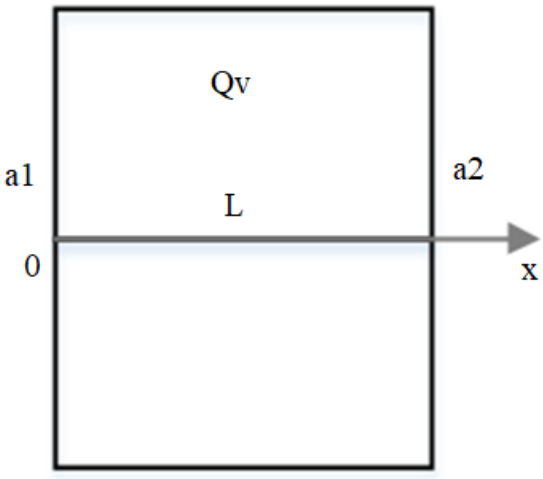

Рисунок 2.1 – Общий вид задачи теплопроводности

В заданных условиях температура может распространятся только по оси Ox, в то время как температура на осях Oy и Oz считается постоянной. Так как условиями задачи задан материал железобетон, то теплофизические характеристики принимаются независимыми от температуры, а значит дифференциальное уравнение теплопроводности примет вид, представленный на рисунке 2.2 вместе с начальными и граничными условиями.

$$pc\frac{\partial v}{\partial t} = \lambda\frac{\partial^2 v}{\partial x} - \alpha_v v + q_v, 0 < x < L$$

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

$$x = 0: \left[-\lambda\frac{\partial v}{\partial x} + \alpha_1 v\right] = q_1$$

$$x = L: \left[\lambda\frac{\partial v}{\partial x} + \alpha_2 v\right] = q_2$$

Рисунок 2.2 – Уравнение теплопроводности для анализируемой задачи с начальными и граничными условиями

Для составления математической модели процесса необходимо задать физические параметры среды, в данном случае – железобетонного стержня: λ = 1.69 Вт/(м·ºС), ρ = 2500 кг/м3, с = 840 Дж/(кг·ºС).

Поставленную задачу необходимо решать методом конечных разностей на равномерной сетке. Сетка представляет собой N-1 равных промежутков по оси Ox, называемых конечно-разностной сеткой. На рисунке 2.2 представлен общий вид сетки.

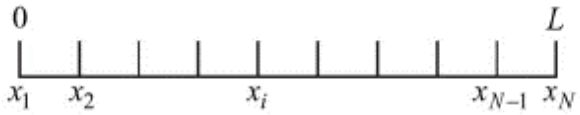

Рисунок 2.2 – Общий вид конечно-разностной сетки

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

Первым этапом преобразований является составление уравнения теплового баланса на основе имеющихся данных. На рисунке 2.3 представлено уравнение теплового баланса.

$$\int_{x_{n-1/2}}^{x_{n+1/2}} c\rho\left(T^i - T^{i-1}\right)dx = \int_{\tau_{j-1}}^{\tau_j}\left[-q_{n\,1/2} + q_{n-1/2} + \int_{x_{n-1/2}}^{x_{n+1/2}} (q_v - \alpha_v T)dx\right]d\tau$$

Рисунок 2.3 – Уравнение теплового баланса для задачи

Следующим этапом является аппроксимация левого выражения уравнения теплового баланса. В правой части уравнения для аппроксимации интегралов по пространственной переменной используется уравнение, полученное в результате предыдущей операции. Необходимо принять некоторые предположения о изменении температуре по времени на интервале, для аппроксимации интегралов по времени. Подставив выражения в уравнение баланса получается разностная схема, представленная на рисунке 2.4

$$\rho c \frac{(h_n + h_{n-1})}{2} * \frac{(u_n^j - u_n^{j-1})}{\Delta t} - \sigma \left[\lambda_{n+1/2} \frac{u_{n+1}^j - u_n^j}{h_n} + \lambda_{n-1/2} \frac{u_{n-1}^j - u_n^j}{h_{n-1}} \right]$$
$$+ (1-\sigma) \left[\lambda_{n+1/2} \frac{u_{n+1}^{j-1} - u_n^{j-1}}{h_n} + \lambda_{n-1/2} \frac{u_{n-1}^{j-1} - u_n^{j-1}}{h_{n-1}} \right]$$
$$+ q_{vn}^j \frac{(h_n + h_{n-1})}{2} - \alpha_{vn} \frac{(h_n + h_{n-1})}{2} \left[\sigma u_{vn}^j + (1-\sigma) u_n^{j-1} \right],$$

где $q_{vn}^j = \frac{1}{L_n \Delta \tau} \int_{\tau_{j-1}}^{\tau_j} \int_{x_{n-1/2}}^{x_{n+1/2}} q_c(x,\tau) dx d\tau$

Рисунок 2.4 – Разностная схема уравнения теплового баланса

Следующим этапом является аппроксимация граничных условий. Для определенности расчетов можно взять ячейку [0, $h_1/2$], прилегающую к границе x = 0. Воспользовавшись неявной схемой, а также полученными ранее выражениями тепловых потоков получается уравнение теплового потока, выходящего из ячейки через границу x = $h_1/2$, представленное на рисунке 2.5.

$$q_{1+1/2} = \lambda_{1+1/2} \frac{T_1^j - T_2^j}{h_1}, \text{где } \lambda_{1+1/2} = h_1 \left[\int_{x_1}^{x_2} \frac{dx}{\lambda(x)} \right]$$

Рисунок 2.5 – Уравнение теплового потока, выходящего из ячейки x = $h_1/2$

Получаем тепловой поток рассеиваемый в среду на границе – $q^1 = \alpha_0 T_1^j$, мощность, выделяемую внутренними источниками тепла – $q^2 = q_{v1}^j \frac{h_1}{2}$, тепловой поток, рассеиваемый с боковой поверхности – $q^3 = \alpha_{v1} T_1^j \frac{h_1}{2}$, мощность, расходуемая на нагрев элементарного объема – $q^4 = c\rho \frac{h_1 (T_1^j - T_1^{j-1})}{2\Delta\tau}$. Из закона сохранения энергии следует уравнение, представленное на рисунке 2.6.

$$-q_{1+1/2} - q^1 + q_n + q^2 - q^3 - q^4 = 0$$

Рисунок 2.6 – Уравнение закона сохранения энергии для текущей задачи

Полученные уравнения для граничных и внутренних точек являются консервативной неявной схемой численного решения. Если просуммировать все уравнения схему, то получится сеточный аналог закона сохранения энергии для всей области стержня. В результате суммирования будет получена равенство, представленное на рисунке 2.7.

$$h\left(\frac{q_{v1}^{j}}{2}+\frac{q_{vN}^{j}}{2}+\sum_{n-2}^{N-1}q_{vn}^{j}\right)+q_0+q_1 = \alpha_0 u_1^j+\alpha_1 u_N^j+h\left(\frac{\alpha_{v1}u_1^j}{2}+\frac{\alpha_{vN}u_N^j}{2}+\sum_{n-2}^{N-1}\alpha_{vn}u_n^j\right) + \frac{c\rho h}{\Delta\tau}\left[\frac{(u_1^j-u_1^{j-1})}{2}+\frac{(u_N^j-u_N^{j-1})}{2}+\sum_{n-2}^{N-1}(u_n^j-u_n^{j-1})\right]$$

Рисунок 2.7 – Итоговое равенство теплового баланса

Равенство 2.7 имеет следующий смысл: сумма мощности внутренних источников тепла и потоков, выделяющихся на границах, равна сумме тепловых потоков, уходящих в среду через границы и рассеиваемых через боковую поверхность и мощности, расходуемой на нагрев стержня.

Следующим этапом является приведение равенства 2.7 к уравнениям для первого, внутренних и конечного узлов. Уравнения представлены на рисунках 2.8, 2.9 и 2.10 соответственно.

$$u_2^j-\left(1+\frac{\alpha_0}{\lambda_{3/2}}+\frac{c\rho h^2}{2\Delta\tau\lambda_{3/2}}+\frac{\alpha_v h^2}{2\lambda_{3/2}}\right)u_1^j+q_0\frac{h}{\lambda_{3/2}}+\frac{h^2}{2\lambda_{3/2}}\left(q_{v1}^j+\frac{c\rho}{\Delta\tau}u_1^{j-1}\right)=0$$

Рисунок 2.8 – Уравнение для первого узла

$$u_{n+1}^j-\left(1+\frac{\lambda_{n-1/2}}{\lambda_{n+1/2}}+\frac{c\rho h^2}{\lambda_{n+1/2}}+\frac{\alpha_v h^2}{\lambda_{n+1/2}}\right)u_n^j+\frac{\lambda_{n-1/2}}{\lambda_{n+1/2}}u_{n-1}^j + \left(q_{vn}^j+\frac{c\rho}{\Delta\tau}u_n^{j-1}\right)\frac{h^2}{\lambda_{n+1/2}}=0$$

Рисунок 2.9 – Уравнение для промежуточных узлов

$$-\left(1+\frac{\alpha_2 h}{\lambda_{N-1/2}}+\frac{c\rho h^2}{2\Delta\tau\lambda_{N-1/2}}+\frac{\alpha_v h^2}{2\lambda_{N-1/2}}\right)u_N^j+u_{N-1}^j+q_2\frac{h}{\lambda_{N-1/2}}+\frac{h^2}{2\lambda_{N-1/2}}\left(q_{vN}^j+\frac{c\rho}{\Delta\tau}u_N^{j-1}\right)=0$$

Рисунок 2.10 – Уравнение для конечного узла

Для реализации решения системы в программном виде полученную систему необходимо свести к общему виду, представленному на рисунке 2.11. Уравнения A, B, C и F выводятся из уравнений 2.8 - 2.10.

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

Рисунок 2.11 – Общий вид системы уравнений для решения задачи

Полученные в результате сведения системы к общему виду были получены трехточечные разностные уравнения второго порядка. Важно заметить, что система 2.11 представлена в виде трехдиагональной матрицы и ее необходимо решать на каждом шаге по времени, т.к. она является нестационарной задачей.

Обращая внимание на формулу видно, что для решения уравнения необходимо знать значение T_{i-1}, для исключения этого оператора предположим, что существуют такие наборы чисел α_iи $\beta_i(i=\overline{1,N-1})$, при которых $T_i^{n+1}=\alpha_i T_{i+1}^{n+1}+\beta_i$,Тогда трехточечное уравнение второго порядка 2.7 преобразуется в двухточечное уравнение первого порядка. Уменьшив в уравнении значение i на единицу, полученное выражение $T_{i-1}^{n+1}=\alpha_{i-1}T_i^{n+1}+\beta_i$ необходимо подставить в уравнение 2.11. После преобразований уравнения получается формула для нахождения T_i^{n+1}, представленная на рисунке 2.12, а уравнения для нахождения α_i и β_i на рисунке 2.13.

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

Рисунок 2.12 – Уравнение нахождения значений

$$\alpha_i = \frac{A_i}{B_i - C_i\alpha_{i-1}}, \beta_i = \frac{C_i\beta_i - F_i}{B_i - C_i\alpha_{i-1}}$$

Рисунок 2.13 – Уравнения нахождения α_i и β_i

Для нахождения α_i и β_i по формуле 2.13 необходимо знать α_0 и β_0, которые находятся из левого граничного условия путем преобразований. На рисунке 2.14 представлены общее уравнение α_0 и β_0 и итоговые.

$$\text{при } \alpha_0 T_1^{n+1} + \beta_0 = -\frac{C_0}{B_0} T_0^{n+1} + \frac{F_0}{B_0} = \alpha_0^{n+1}$$

$$\alpha_0 = -\frac{C_0}{B_0}, \beta_0 = \frac{F_0}{B_0}$$

$$\text{где } B_0 = F_0 = \frac{a_1}{\lambda}, C_0 = -1$$

Рисунок 2.14 – Нахождение уравнений α_0 и β_0

Для определения $T_N^{n+1}, N = \overline{N-1, 2}$ необходимо знать значение T_N^{n+1}, которое находится из правого граничного условия путем преобразований. На рисунке 2.15 представлено уравнение переменной.

$$T_N^{n+1} = \frac{F_N + A_N\beta_{N-1}}{B_N + A_N\alpha_{N-1}}, \text{где } B_N = F_N = \frac{a_2}{\lambda}, A_N = -1$$

Рисунок 2.15 – Уравнение T_N^{n+1}

После нахождения уравнений и определения начальных значений последовательно находятся $T_{N-1}^{n+1}, T_{N-2}^{n+1}, \dots, T_2^{n+1}$.

Решение уравнений вида 2.10 способом, описанным выше, называется методом прогонки и сводится к вычислениям по трем формулам: нахождение прогоночных коэффициентов α_i, β_i по формулам 2.12 при i = 2, N −1 (прямая прогонка) и затем получение неизвестных T_i^{n+1}по формуле 2.6 при i= N-1, N-2, …, 2 (обратная прогонка).

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

Прогонка является корректной, если знаменатели прогоночных коэффициентов 2.12 не обращаются в нуль, и устойчивой, если $\alpha_i < 1$ при всех $i = \overline{1, N-1}$.

Аппроксимация дифференциальной задачи конечно-разностной выполнена с первым порядком точности по времени τ и вторым по пространственной координате h. При этом неявная разностная схема является абсолютно устойчивой, т.е. можно проводить интегрирование краевой задачи с любым разностным шагом по времени. Шаг по времени выбирается таким образом, чтобы весь интервал времени разбивался как минимум на 10 шагов.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ЗАДАЧИ

Для реализации программы анализа теплопроводности в стержне с боковым теплообменом и источником тепла необходимо спроектировать два блока – блок расчетов и блок вывода результата на экран. Весь функционал программы можно уместить в два класса, которые будут реализовывать функционал блоков (Thermo – для расчетов, Draw – для вывода на экран). На рисунке 3.1 представлена общая схема работы программы.

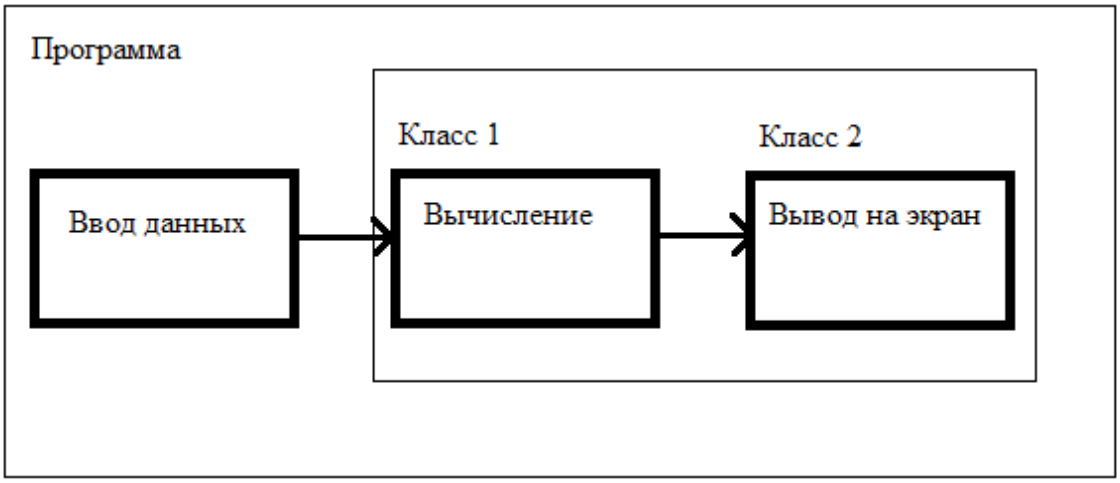

Рисунок 3.1 – общая схема работы программы

Вводимыми данными являются следующие параметры среды: c – теплоемкость, ρ – плотность материала, λ – теплопроводность, T_0 – начальная температура стержня, α_1 – коэффициент теплоотдачи с левой стороны, α_2 – коэффициент теплоотдачи с правой стороны, α_v – объемный коэффициент теплоотдачи, L – длина стержня, Qv – объемный коэффициент теплопроводности, N_x – количество узлов пространственной сетки, N_y – количество узлов временной сетки.

Общая часть алгоритма работы обоих методов заключается в инициализации массива разностной сетки, предназначенной для хранения данных о температурах, а также расчет шагов алгоритма по времени. На рисунках 3.2, 3.3 представлены фрагменты кода инициализации переменных метода.

```
var T = new double[Nx + 1];
var a = new double[Nx + 1];
var b = new double[Nx + 1];
var h = L / (Nx - 1);
var Dt = 0.2;
for (int i = 0; i < Nx; T[i++] = T0) ;
var time = 0;
```

Рисунок 3.2 – Инициализация известных переменных

```
    // из левого краевого условия
    var B0 = -a1 / l;
    var C0 = -1;
    var F0 = a1 / l;

    // из правого краевого условия
    double Ak = -1;
    var Bk = a2 / l;
    var Fk = a2 / l;
```

Рисунок 3.3 – Инициализация переменных для расчета начальных значений

После инициализации переменных и задания начальных значений необходимо производить расчеты, для удобства написания кода воспользуемся циклом while для выделения общего блока расчетов. Удобство заключается в том, что внутри общего цикла будет использовать несколько циклов for, что может привести к возникновению ошибок человеческого фактора. Внутри цикла while выполним операции расчетов согласно описанным в главе 2 алгоритмам. На рисунке 3.4 представлена реализация метода для расчета температуры стержня.

```
    while (time < Nt) {
      time++;
      // находим первоначальные значения a и b из левого
краевого условия
      a[0] = - (C0 / B0);
      b[0] = F0 / B0;
      // находим остальные a и b
      for (int i = 1; i < Nx; i++){
        double tmp = l / h;
        var Ai = tmp;
        var Bi = 2 * tmp + p * c / Dt;
        var Ci = tmp;
        var Fi = -p * c * T[i] / Dt + Qv;

        a[i] = Ai / (Bi - Ci * a[i - 1]);
        b[i] = (Ci * b[i - 1] - Fi) / (Bi - Ci * a[i - 1]);
      }

      // находим Tn из правого краевого условия
      T[Nx] = (Fk + Ak * b[Nx-1]) / (Ak*a[Nx-1] + Bk);

      // находим остальные Ti
      for (int i = Nx - 1; i >= 0; i--)
        T[i] = a[i] * T[i + 1] + b[i];
    }
```

Рисунок 3.4 – Цикл расчетов распределения тепла методом прогонки

После выполнения операции расчета метод возвращает массив значений температур в каждом узле сетки. Для вывода на экран необходимо отрисовать каждый элемент массива. Цвет узла определяется методом IntToColor, от синего – холодный, до красного – равный максимальной температуре источника тепла.

Принцип работы алгоритма определения цвета можно описать следующим образом:

1. Метод принимает два аргумента – значение температуры в ячейке и значение максимальной температуры для текущих данных.

2. Основываясь на том, значение цветов RGB проходят последовательно (++ - значение увеличивается, -- - значение уменьшается): FF++00, --FF00, 00FF++, 00--FF, 0000--, необходимо реализовать алгоритм.

3. Для определения цветов необходимо разделить максимальное значение температуры на количество возможных вариантов температур и определить в каком находится рассчитанное значение.

4. Завершающим этапом является определение процента дополнительного цвета, обозначенного ++ или – и расчет значения.

Код определения цвета температуры представлен на рисунке 3.5.

```
 private static Color IntToColor(Double Value, Double Max){
     double r = 0, g = 0, b = 0;
     double p = Max / 5;
     if ( Value < p) {
       p = Value * 100 / p;
       b = (100 - p) * 255 / 100;
     } else if(Value < p * 2) {
       p = (Value - p) * 100 / p;
       g = (100 - p) * 255 / 100;
       b = 255;
     } else if(Value < p * 3) {
       p = (Value - p * 2) * 100 / p;
       g = 255;
       b = (100 - p) * 255 / 100;
     } else if (Value < p * 4){
       p = (Value - p * 3) * 100 / p;
       r = (100 - p) * 255 / 100;
       g = 255;
     } else {
       p = (Value - p * 4) * 100 / p;
       r = 255;
       g = (100 - p) * 255 / 100;
     }
     return Color.FromArgb((int)Math.Ceiling(r),
            (int)Math.Ceiling(g), (int)Math.Ceiling(b));
   }
```

Рисунок 3.5 – Код метода IntToColor

4 ВЕРИФИКАЦИЯ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

Для подтверждения корректности работы программы было принято решение сравнить результаты разработанной программы с результатами программы ANSYS.

Первым этапом создания стержня в программе ANSYS является создание точек в декартовой системе координат в пространстве, так как значение глубины стержня не определено то использоваться будут координаты по x и y. Для задания точек в программе ANSYS необходимо в главном меню (Main Menu) выбрать пункт Preprocessor -> Modeling -> Create -> Keypoints -> In Active CS. После чего ввести в появившемся окне последовательно координаты точек. На рисунке 4.1 представлен результат создания точек.

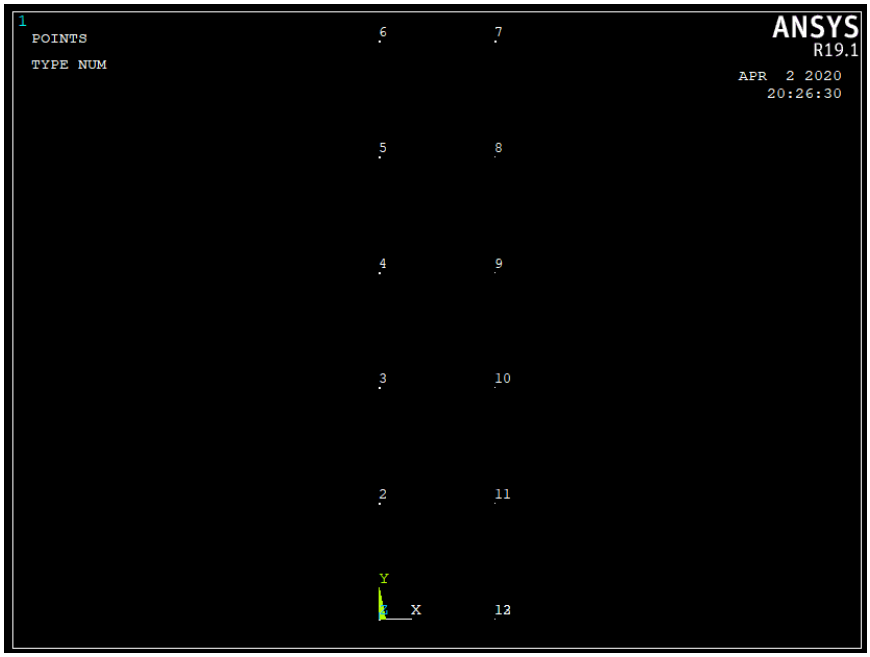

Рисунок 4.1 – Результат создания точек

Следующим этапом является соединение точек линиями. Для выполнения этого действия необходимо в главном меню (Main Menu) выбрать пункт Preprocessor -> Modeling -> Create -> Lines -> Lines -> In Active Coord.

В появившемся окне Lines in Active Coord необходимо установить переключатель на значение Pick, после чего выбирать пары точек для объединения. В случае если необходимо удалить случайно созданную линию достаточно нажать правую кнопку или установить значение переключателя на Unpick и выбрать лишнюю линию. На рисунке 4.2 представлено окно создания линий и результат выполнения операции.

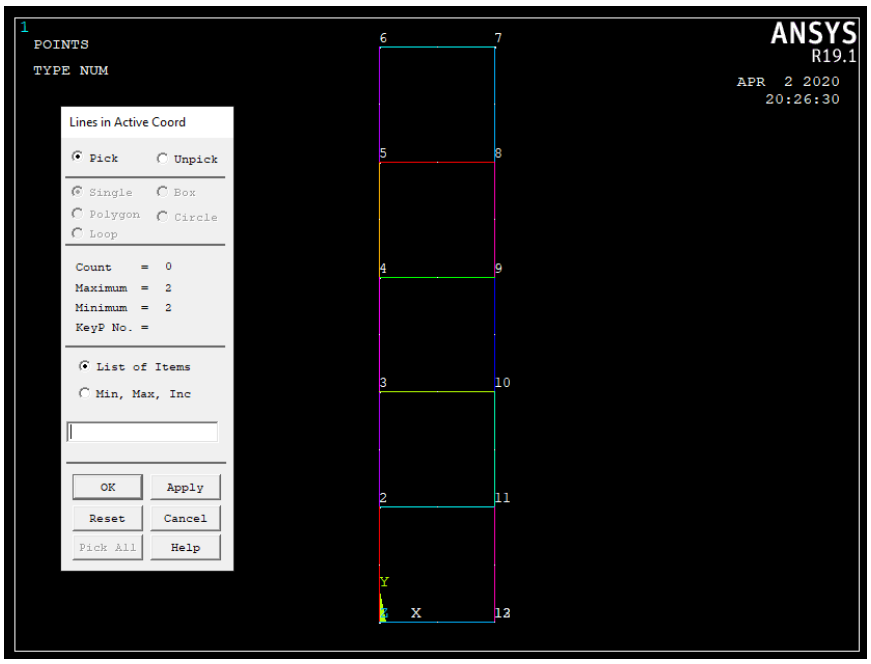

Рисунок 4.2 – Окно соединения точек и результат создания линий

После создания линий логичным продолжением является этап создания поверхности стержня. Если объект состоящий из линий определят только такие физические параметры стержня как ширина и высота, то поверхность определяет площадь. Для создания поверхности необходимо в главном меню (Main Menu) выбрать пункт Preprocessor -> Modeling -> Create -> Areas -> Arbutrary -> By Lines. Отличием интерфейса создания поверхностей является только то, что для создания поверхности из выбранных линий необходимо нажать на кнопку Apply или Ok, после чего поле закрасится бирюзовым цветом. На рисунке 4.3 представлен результат создания поверхности.

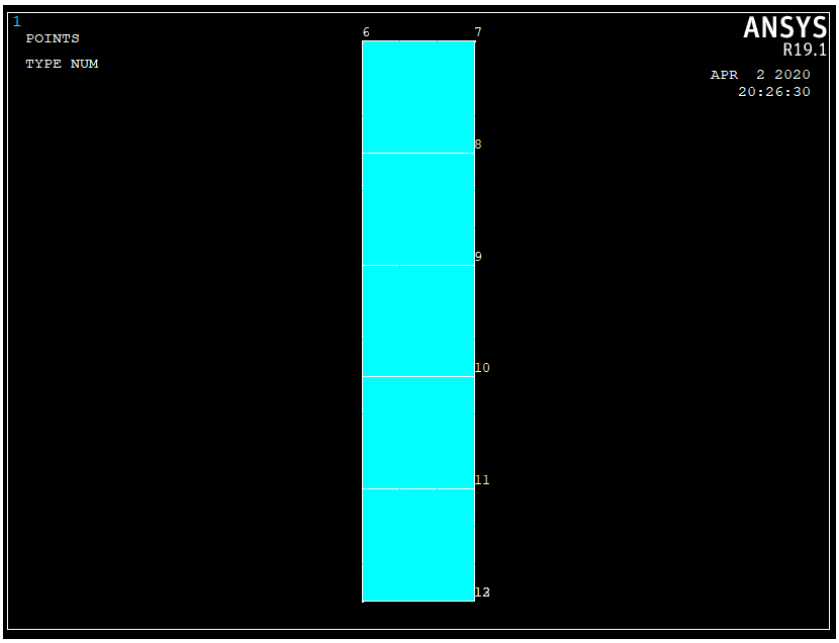

Рисунок 4.3 – Окно создания площади

Следующим этапом моделирования стержня для расчетов распределения температур является создание материала. Материал может задавать такую физическую величину как теплопроводность (KXX). Для задания параметром материала в главном меню (Main Menu) выбрать пункт Preprocessor -> Material Props -> Material Models и в появившемся окне выбрать подходящий материал, для текущего задания им является Thermal -> Conductivity(Проводимость) -> Isotopic. На рисунке 4.4 представлен скриншот окна выбора материала.

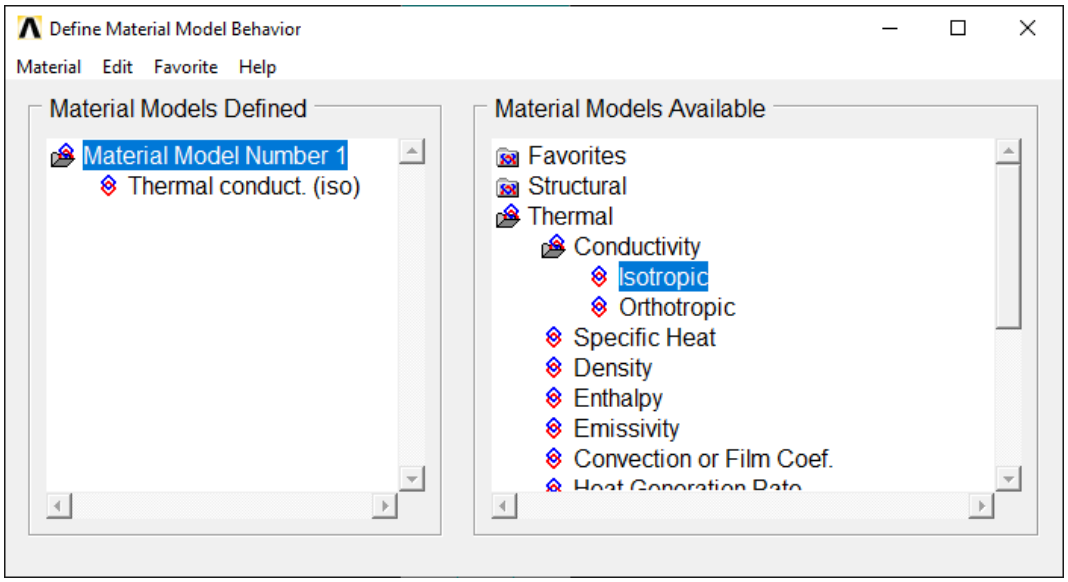

Рисунок 4.4 – Окно создания материала

После указания базовых параметров элемента необходимо приступить к непосредственному созданию элемента, для этого необходимо в главном меню (Main Menu) выбрать пункт Preprocessor -> Element Type -> Add/Edit/Delete и в появившемся окне нажать Add. После чего выбрать Thermal Mass -> Solid(Твердый) -> Triangl 6node 35. На рисунке 4.5 представлен интерфейс выбора элемента.

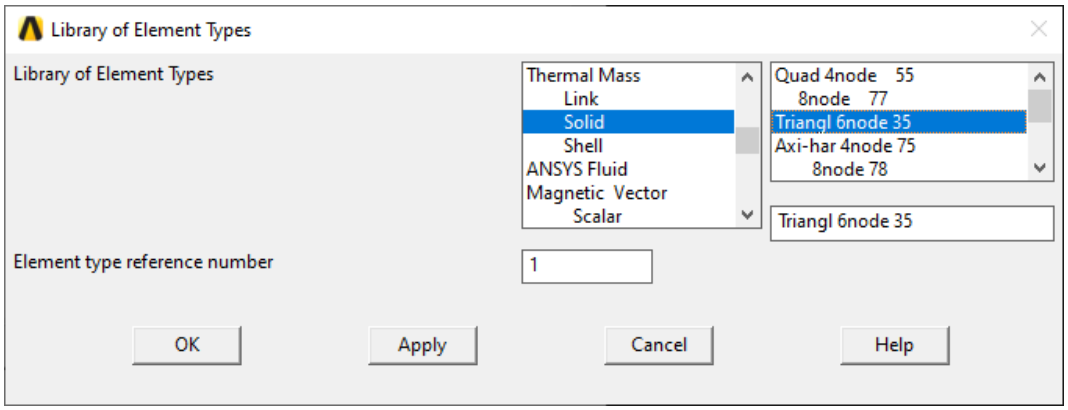

Рисунок 4.5 – Окно создания конечного элемента

После задания материала и создания конечного элемента необходимо разбить его на элементы, другими словами – создать сетку. Этого делается путем выбора в главном меню (Main Menu) пункта Preprocessor -> Meshing -> Mesh Tool. В появившемся окне задаться значение созданного ранее материала, а также некоторые дополнительные параметры, например, вид сетки Free(Свободный) или Mapped, который задается параметром, например, 3 or 4 sided. Кроме этого можно удалить ранее созданную сетку нажав на кнопку Clear. После указания всех параметров необходимо нажать кнопку Mesh. В отобразившемся окне, аналог которого появлялся при создании линий и поверхности, необходимо установить переключатель на Pick и выбрать все блоки модели, после чего нажать Ok. На рисунке 4.6 представлен результат создания сетки.

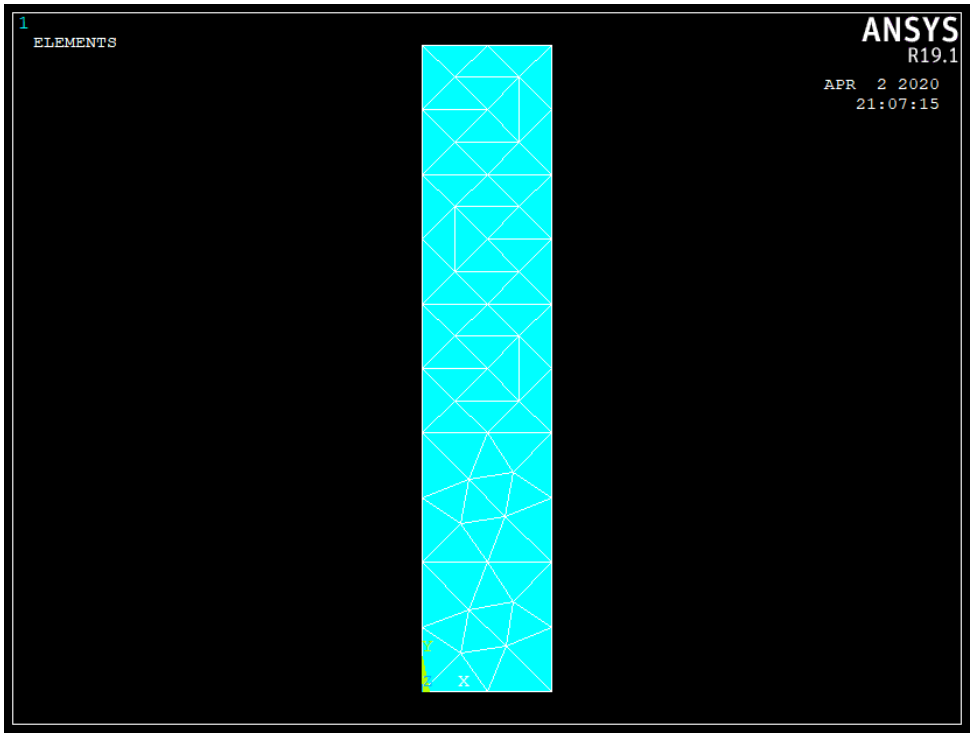

Рисунок 4.6 – Результат разбиения стержня на конечные элементы

После создания элемента необходимо задать температурные параметры стержня. К таким параметрам относится базовая температура объекта, изоляция и показатель теплообмена. Для задания температуры необходимо в главном меню (Main Menu) выбрать пункт Preprocessor -> Loads -> Define Loads -> Apply -> Thermal -> Themperature -> On Lines. Так как по условию задачи источник тепла находится внутри стержня, то выбираем центральные линии для задания равномерного прогрева стержня. На рисунке 4.7 представлено окно задания температуры.

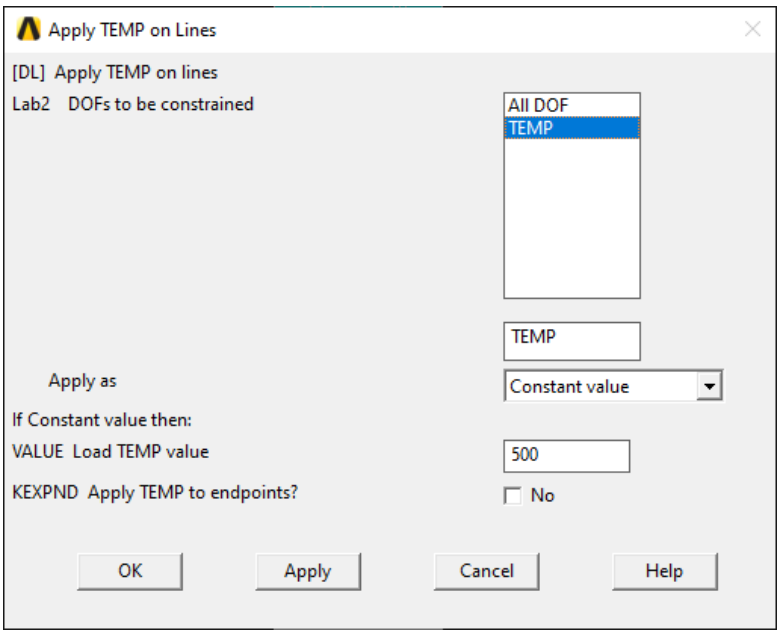

Рисунок 4.7 – Окно настройки температуры

После задания температуры необходимо задать коэффициенты теплоотдачи с обоих сторон, для смоделированного стержня это будут верхняя и нижняя стороны. Для задания конвекции необходимо в главном меню (Main Menu) выбрать пункт Preprocessor -> Loads -> Define Loads -> Apply -> Thermal -> Convection -> On Lines. На рисунке 4.8 представлено окно задания конвекции для одной стороны. Для второй этот процесс аналогичен.

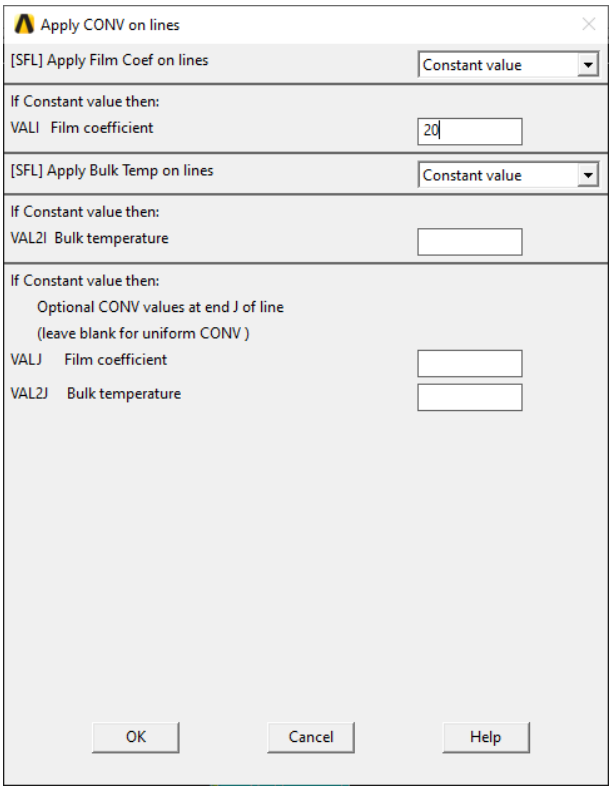

Рисунок 4.8 – Окно задания конвекции

После задания конвекции необходимо задать изоляцию для сторон, не принимающих участие в процессе теплообмена, для этого необходимо в главном меню (Main Menu) выбрать пункт Preprocessor -> Loads -> Define Loads -> Apply -> Thermal -> Heat Flux -> On Lines. Процесс задания изоляции похож на предыдущие пункты так как он использует уже рассмотренные окна. На рисунке 4.9 представлено окно задания изоляции.

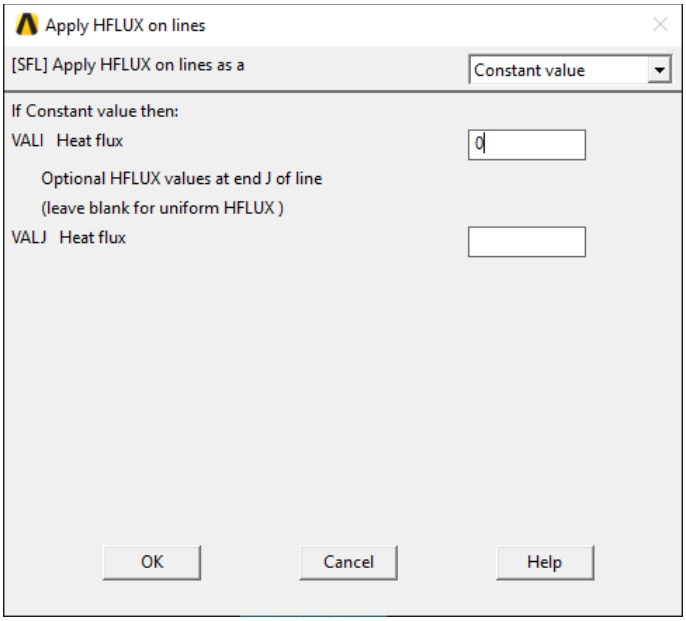

Рисунок 4.10 – Окно задания изоляции

После создания стержня необходимо произвести непосредственный расчет, для этого для этого необходимо в главном меню (Main Menu) выбрать пункт Solution -> Solve -> Curent LS, тем самым открыв окно для расчетов. Подтвердив намерения нажав на кнопку Ok в появившемся окне, в случае безошибочного задания всех параметров элемента появится сообщение об успешном завершении расчетов. На рисунке 4.11 представлено сообщение об успешном завершении расчетов.

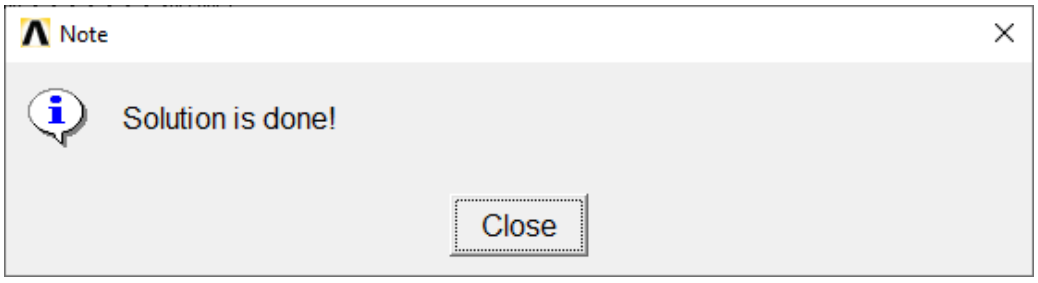

Рисунок 4.11 – Результат расчетов по заданным параметрам

Для визуального отображения результатов необходимо в главном меню (Main Menu) выбрать пункт General Postproc -> Plot Results -> Contour Plot -> Nodal Solu и в появившемся окне выбрать Nodal Solution -> DOF Solution -> Nodal

Temperature. На рисунке 4.12 представлен результат выполнения программы ANSYS.

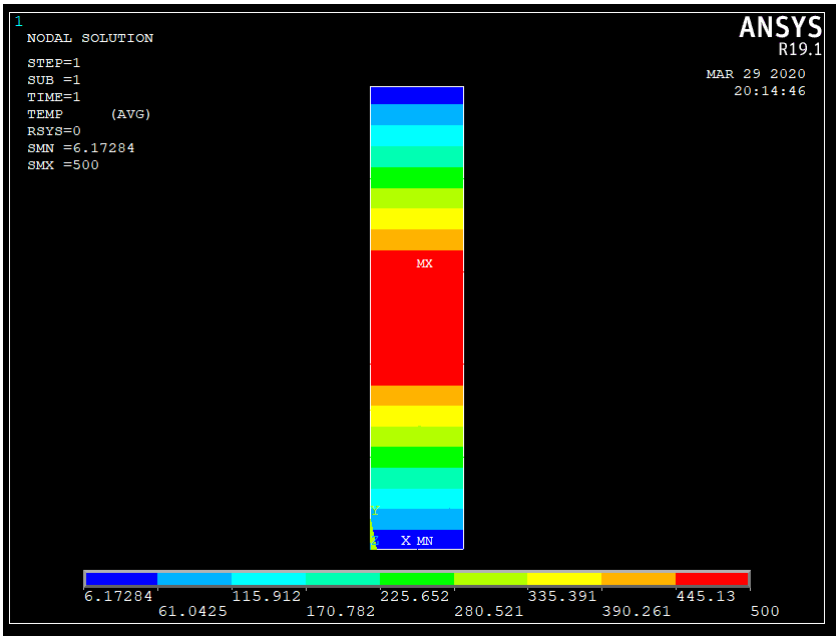

Рисунок 4.12 – Результат выполнения расчетов в программе ANSYS

Для выполнения аналогичных расчетов в разработанной программе необходимо ввести начальные данные в столбце слева и нажать кнопку Считать. Результат расчетов будет отображен в центре экрана, а в строке ниже – результат расчетов в численном виде. На рисунке 4.13 представлен результат выполнения программы.

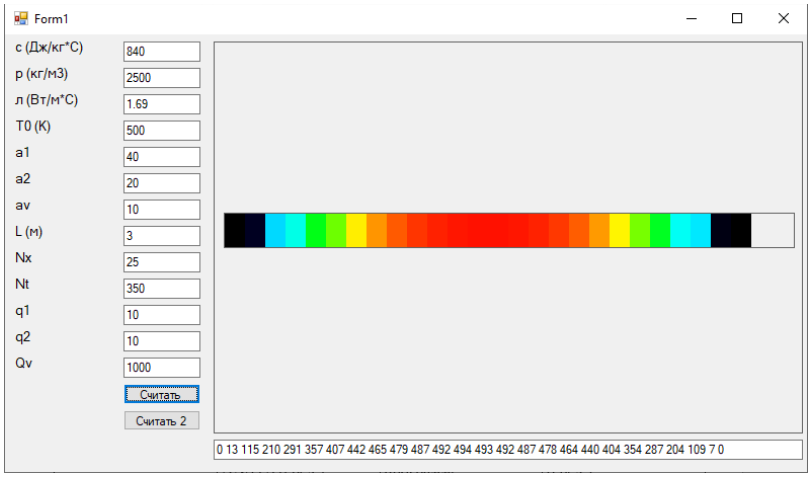

Рисунок 4.12 – Результат выполнения расчетов в разработанной программе

Корректность работы программы проверяется путем визуального, рисунки 4.11 и 4.12, и числового сравнения результатов расчётов приложения и

математического пакета ANSYS. В таблице 4.1 представлены данные для оценки погрешности расчётов программы.

Таблица 4.1– Оценка погрешности расчетов программы

Номер узла	Температура узла в ANSYS	Температура узла в программе	Погрешность измерений, %
3	116	115	0,0086
14	500	493	0,014
22	280	287	-0,025
25	6	7	-0,1667

В таблице 4.2 представлены данные для сравнения расхождения результатов.

Таблица 4.2 – Данные для сравнения

	ANSYS	Разработанное приложение	Расхождение результатов, %
Минимальное значение	6	7	-0,2%
Среднее значение	253	250,5	0,5%
Максимальное значение	500	494	1,2%

Процент погрешности результатов разработанной программы и ANSYS не превысил 1%, а расхождения не превышает 2%, что говорит о правильной работе и достаточной точности приложения.

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы по анализу распределения температуры в стержне с боковым теплообменом и источником тепла были проанализированы различные численные методы.

Большое внимание было уделено численным методам в области термодинамики, т.к. они напрямую связана с темой курсовой работы.

Стало известно, что численные методы не являются точными методами и имеют погрешность, в случае увеличения которой результаты могут стать неверными. Погрешности могут быть нескольких видов, а самой критически важной является погрешность округления.

Также были рассмотрены методы решения численных методов, один из которых был использован в разработанной программе, а именно – метод конечных разностей в неявном виде.

После анализа источников была сформирована и проанализирована задача для темы курсовой работы, составлен алгоритм ее решения.

На основании алгоритма решения задачи был написан код программы, выполняющей вычисления на компьютере и выводящей на экран результат распределения тепла в стержне с боковым теплообменом.

Для проверки корректности работы программы результат расчетов был сравнен с результатом расчетов программы ANSYS.

Список использованных источников

1. Troelsen A. W., Olsen A. Pro C# 5.0 and the. NET 4.5 Framework. – New York City, USA : Apress, 2012. – Т. 6.
2. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. Учебник. – БИНОМ. Лаборатория знаний, 2012.
3. Вержбицкий В. М. Численные методы (математический анализ и обыкновенные дифференциальные уравнения). – Издательский дом" ОНИКС 21 век", 2005.
4. Зенков А.В. Численные методы: учеб. пособие / А.В.Зенков. – Екатеринбург: Изд-во Урал. ун-та, 2016. – 124 с.
5. Кудайкулов А. К., Кенжегулов Б. З., Мырзашева А. Н. Математическая модель установившегося поля распределения температуры по длине стержня, ограниченной длины при наличии локальной температуры, теплового потока, теплообмена и теплоизоляции //Наука, новые технологии и инновации. – 2009. – №. 5. – С. 18-22.
6. Кузнецов Г.В., Шеремет М.А. Разностные методы решения задач теплопроводности: учебное пособие. / Г.В. Кузнецов, М.А. Шеремет. – Томск: Изд-во ТПУ, 2007. – 172 с.
7. Неймарк Ю. И. Математические модели в естествознании и технике. – Изд-во Нижегор. ун-та, 2004.
8. Рихтер Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# //Питер–2012.–896 с. – 2012.
9. Токочаков, В. И. Моделирование, оптимизация и управление теплотехническими системами: лабораторный практикум по одноименному курсу для студентов специальности 1-43 01 05 «Промышленная теплоэнергетика» дневной и заочной форм обучения: в 3 ч. Ч. 2 / В. И. Токочаков. – Гомель: ГГТУ им. П. О. Сухого, 2009. – 41 с.

Приложение А
Текст программы

Draw.cs

```
public static class Draw {

  public static Int32 mm_in_px = 10;

  public static void Grid(Graphics Canvas, Size CanvasSize,
                          Double[] Array, Double[] Array2 = null){

    var GraphSize = new Size(CanvasSize.Width, CanvasSize.Height);

    var Pen = new Pen(Color.Black);
    var gPen = new Pen(Color.Green);
    var rPen = new Pen(Color.Red);

    var sPos = new Point((CanvasSize.Width / 2) - (GraphSize.Width / 2),
                (CanvasSize.Height / 2) - (GraphSize.Height / 2));
    var ePos = new Point(sPos.X + GraphSize.Width, sPos.Y +
                GraphSize.Height);

    var cPos = new Point(sPos.X, ePos.Y);

    var Rectangle = new Rectangle(sPos, GraphSize);

    int sH = cPos.Y - sPos.Y;
    int sW=(int)Math.Ceiling((double)((ePos.X - cPos.X)/Array.Length));

    for(int i = 0; i < Array.Length; i++)
      Canvas.FillRectangle(
        new SolidBrush(Draw.IntToColor(Array[i], Array[0])),
        new Rectangle(new Point(sPos.X + (sW * i), sPos.Y),
        new Size(sW, sH)));

    Canvas.DrawRectangle(Pen, Rectangle);
 }

  private static Color IntToColor(Double Value, Double Max) {

    if (Value < 0)
      Value = 0;
    if (Value > Max)
      Value = Max;

    double r = 0, g = 0, b = 0;

    double p = Max / 5;

    if ( Value < p) {
      p = Value * 100 / p;
      b = (100 - p) * 255 / 100;
    } else if(Value < p * 2) {
      p = (Value - p) * 100 / p;
      g = (100 - p) * 255 / 100;
      b = 255;
    } else if(Value < p * 3) {
      p = (Value - p * 2) * 100 / p;
      g = 255;
      b = (100 - p) * 255 / 100;
    } else if (Value < p * 4){
```

```
            p = (Value - p * 3) * 100 / p;
            r = (100 - p) * 255 / 100;
            g = 255;
          } else {
            p = (Value - p * 4) * 100 / p;
            r = 255;
            g = (100 - p) * 255 / 100;
          }

          return Color.FromArgb(
            (int)Math.Ceiling(r),
            (int)Math.Ceiling(g),
            (int)Math.Ceiling(b));
        }
```

Thermo.cs

```
      public class Thermo2 {
        private Int32 Nx;
        private Int32 Nt;
        private Double L = 3;
        private Double H = 0.2;
        private Int32 T0 = 500;

        public Thermo2(Int32 Nx, Int32 Nt, Double L, Double H, Int32 T0)
        {
          this.Nx = Nx;
          this.Nt = Nt * 100;
          this.L = L;
          this.H = H;
          this.T0 = T0;
        }

        public Double[] Four(Double cp, Double Qv, Double a1, Double a2, Double
Av, Double q1, Double q2, Double Dt) {
          this.Qv = Qv;

          double[] U, A, B, C, D;

          U = new double[Nx];
          for (int i = 0; i < U.Length; U[i++] = T0) ;

          int N1 = Nx - 1;
          H = Len / N1;

          double R1 = cp * H * H / Dt;
          double R2 = Av * H * H;
          double R3 = H * H;

          A = new double[Nx];
          B = new double[Nx];
          C = new double[Nx];
          A[0] = 1;
          B[0] = -1 - (a1*H + R2/2 + R1/2) / AL(H/2);
          C[0] = 0;

          for(int i = 1; i < N1; i++){
            var Xi = (i - 1) * H;
            A[i] = 1;
            var AA = AL(Xi + H / 2);
            C[i] = AL(Xi - H/2) / AA;
            B[i] = -1 - C[i] - (R1 + R2) / AA;
          }
```

```
          A[N1] = 0;
          B[N1] = -1 - (a2 * H + R2 / 2 + R1 / 2) / AL(Len - H / 2);
          C[N1] = 1;

          var j = 0;
          D = new double[Nx];

          while (j++ < Nt){

           D[0] = (R1 * U[0] / 2 + QV(0, j * Dt) * R3 / 2 + q1 * H) / AL(H / 2);

           for (int i = 1; i < N1; i++){
             var Xi = (i - 1) * H;
             D[i] = (R1 * U[i] / 2 + QV(Xi, j * Dt) * R3 / 2) / AL(Xi - H / 2);
           }

           D[N1] = (R1 * U[N1] / 2 + QV(Len, j * Dt) * R3 / 2 + q2 * H) / AL(Len
- H / 2);

           U = SYSTRD(U, A, B, C, D);

           }
          return U;
        }

        public Double[] SYSTRD(Double[] X, Double[] A, Double[] B, Double[] C,
Double[] D) {
          double[] G = new double[Nx];

          var N1 = Nx - 1;

          X[0] = -A[0] / B[0];
          G[0] = -D[0] / B[0];

          for(int i = 1; i < N1; i++){
            var S = B[i] + C[i] * X[i-1];
            X[i] = -A[i] / S;
            G[i] = -(D[i] + C[i] * G[i-1]) / S;
          }

          X[N1] = -(C[N1] * G[N1 - 1] + D[N1]) / (B[N1] + C[N1] * X[N1 - 1]);

          for(int i = N1 - 1; i >= 0; i--)
            X[i] = X[i] * X[i + 1] + G[i];

          return X;
        }
```

Form1.Метод работы с классами

```
      private void button1_Click(object sender, EventArgs e) {

        var math = new Thermo2(
          l: Double.Parse(this.txtla.Text.Replace('.', ',')),
          Nx:Int32.Parse(this.txtNx.Text),
          Nt:Int32.Parse(this.txtNy.Text),
          L:Double.Parse(this.txtL.Text.Replace('.', ',')),
          H:0.02,
          T0: Double.Parse(this.txtT.Text.Replace('.', ','))
        );

        var image = new Bitmap(this.Page1.Width, this.Page1.Height);
        this.txtGraph1.Text = "";
```

```
        var res = math.Third(
          l:Double.Parse(this.txtla.Text.Replace('.', ',')),
          p:Double.Parse(this.txtP.Text),
          c:Double.Parse(this.txtC.Text),
          Qv: Double.Parse(this.txtQv.Text),
          a1: Double.Parse(this.txtA1.Text),
          a2: Double.Parse(this.txtA2.Text),
          av: Double.Parse(this.txtAv.Text),
          q1: Double.Parse(this.txtQ1.Text),
          q2: Double.Parse(this.txtQ2.Text)
        );

        foreach (double a in res)
          this.txtGraph1.Text += System.Math.Round(a, 0).ToString() + " ";
        Draw.Grid(Graphics.FromImage(image), image.Size, res,
Double.Parse(this.txtT.Text.Replace('.', ',')));
        this.Page1.Image = image;
        this.Page1.Update();
       }
```

Приложение Б
Блок-схемы алгоритмов

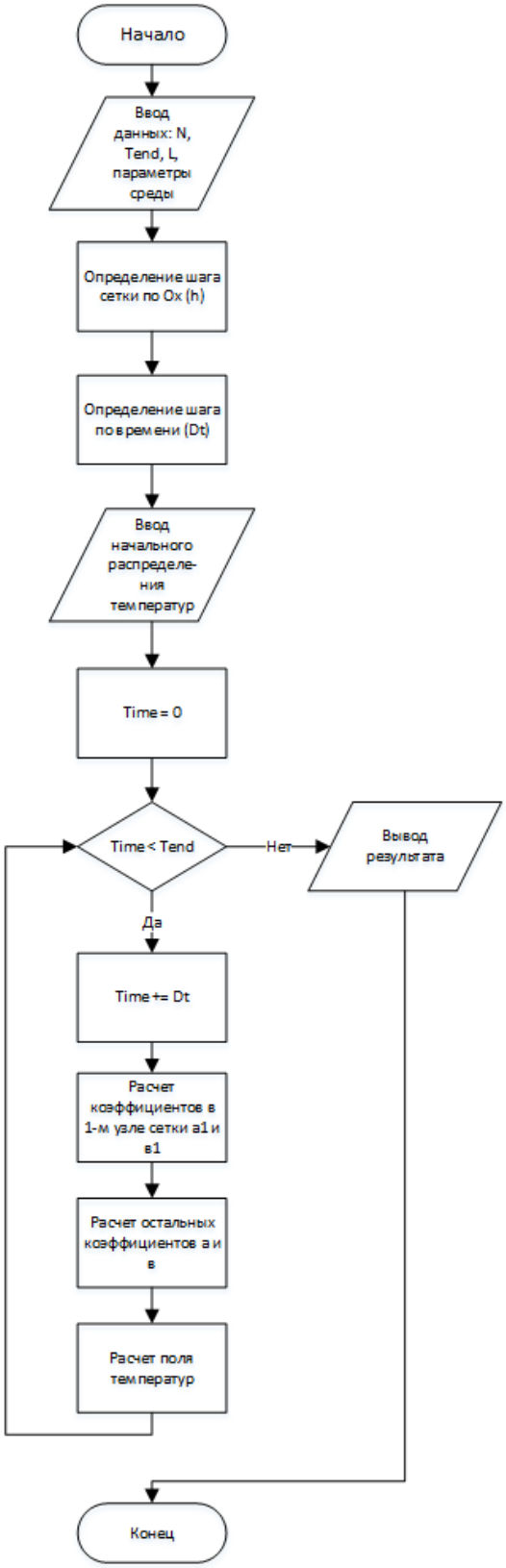

Рисунок Б1 – Блок-схема неявного метода

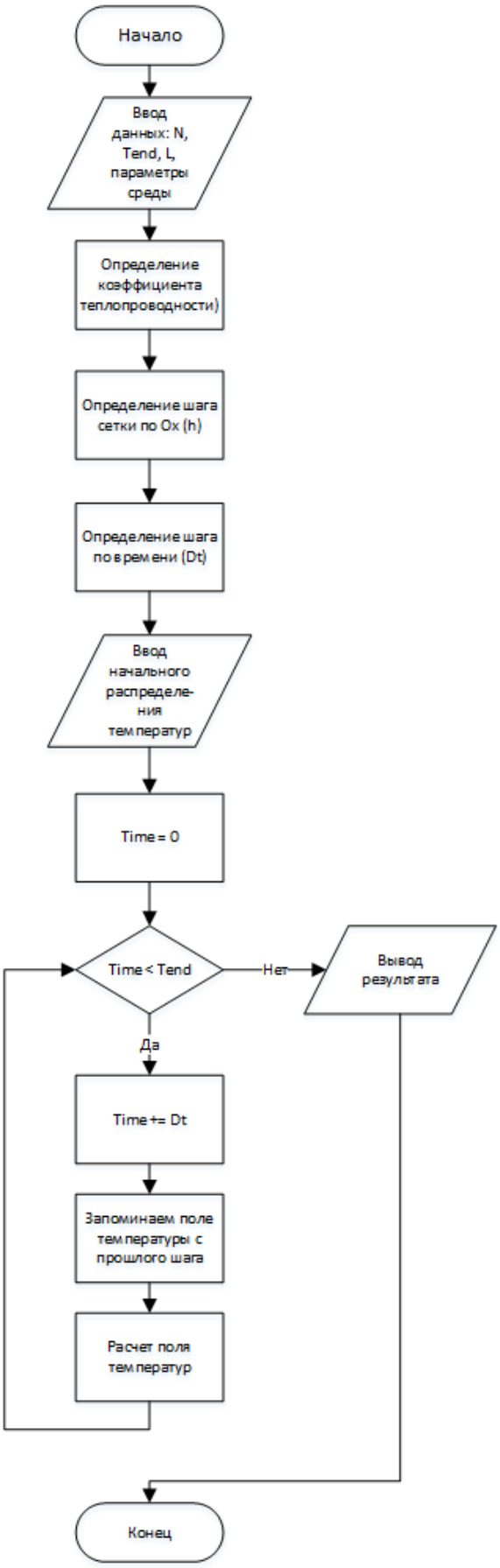

Рисунок Б2 – Блок-схема явного метода