**Содержание**

[**Введение 2**](#_3dy6vkm)

[**Постановка задачи 3**](#_md026fhu5t1)

[**Численное решение уравнения теплопроводности с постоянным запаздыванием в неоднородности сеточными методами 4**](#_2s8eyo1)

[**Заключение 27**](#_17dp8vu)

[**Список источников 27**](#_3rdcrjn)

# **Введение**

Многие физические явления нашего мира, которые описываются моделью, содержат два эффекта: распределенность параметров и динамику с эффектом наследственности. Математическим описанием таких явлений могут служить дифференциальные уравнения в частных производных запаздыванием (дифференциальное уравнение с отклоняющимся аргументом). Например, такие уравнения используются при изучении проблем с горением в ракетном двигателе.

Но у таких уравнений есть проблема - сложность аналитического решения, поэтому единственный эффективный способ получить ответ - получить приближенное решение уравнения, используя численные методы.

Такие методы основаны на алгоритмах, и их можно реализовать на компьютерах, используя различные языки программирования.

# **Постановка задачи**

В данной работе будет рассматриваться 2 численных способа решения уравнения теплопроводности с постоянным запаздыванием, с граничными условиями первого рода.

Общий вид уравнения теплопроводности с постоянным запаздыванием:

Граничные условия первого рода:

Начальные условия:

Рассмотрим 2 сеточных метода: явный и неявный.

Их алгоритмы будут такие:

1. Явный:

,

где ,

1. Неявный:

,

где

# **Численное решение уравнения теплопроводности с постоянным запаздыванием в неоднородности сеточными методами**

1) Перед решением основного уравнения, я решил простое дифференциальное уравнение с постоянным запаздывание, чтобы понять, как решать уравнения такого вида.

Общий вид такого уравнения:

Начальная функция:

Само уравнение было такое:

,

где запаздывание

Начальная функция:

Точное решение:

Точное решение известно и получено аналитически, оно нужно чтобы узнать точность метода.

Для решения используемый метод Эйлера.

Алгоритм таков:

,

где

- приближение точного значения в узле - , i - индекс

Для получения значений от до воспользуемся начальной функцией:

В нашем случае , число m я брал такое:

Для реализации метода был использован язык программирования Python и его библиотеки: matplotlib - для создания графиков, numpy - для вычислений, преобразований результата в матрицы для графиков и сохранения результатов.

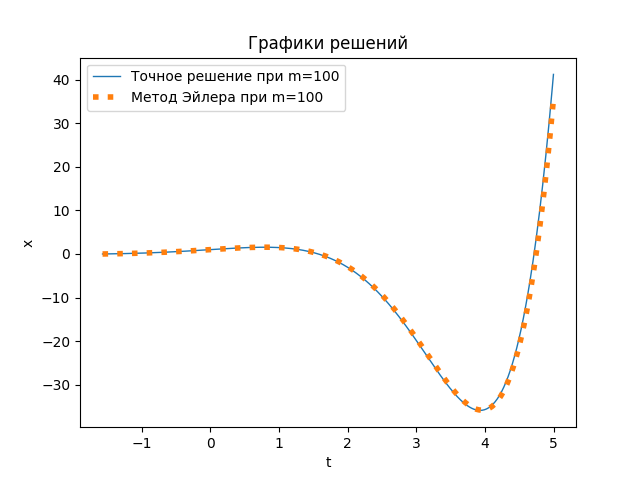
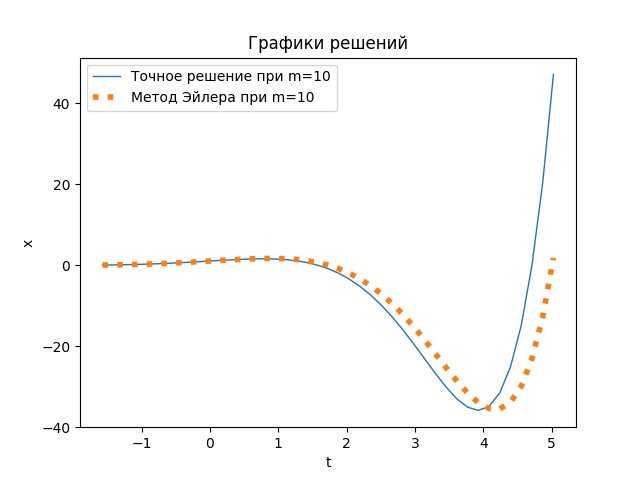
Для того, чтобы узнать точность метода при разных шагах, найдем такое значение: для разных , которые зависят от параметра m.

Эта формула даст нам значение максимального отклонения приближенного значения от точного.

| m=10 | m=50 | m=100 | m=500 |
| --- | --- | --- | --- |
|  |  |  |  |

Результат вычисления метода, выведенный на график и в таблице:

| m | t=0.5 | t=1 | t=2.5 | t=5 |
| --- | --- | --- | --- | --- |
| 10 |  |  |  |  |
| 50 |  |  |  |  |
| 100 |  |  |  |  |
| 500 |  |  |  |  |



На графике и в таблицах хорошо видно, что при увеличении числа m, которое влияет на число шагов, значения приближенного решения начинают совпадать с точным и погрешность уменьшается.

Метод Эйлера имеет первый порядок сходимости, поэтому его точность не высока и это хорошо заметно при значении m1.

Также порядок сходимости можно проверить с помощью формул вида:

Если значения этой формулы стремятся к 1 при увеличении числа m т.е. при уменьшении числа , то это покажет что метод сходится с 1 порядком.

| m=10 | m=100 | m=1000 | m=10000 |
| --- | --- | --- | --- |
| 0.776 | 0.954 | 0.997 | 0.999 |

Как видно по таблице, число стремится к 1, что соответствует порядку метода.

2) Перейдем теперь к основному уравнению. Решим его явным сеточным методом.

Само уравнение имеет вид:

Граничные условия первого рода:

Начальные условия:

Точное решение известно, и имеет вид:

Алгоритм явного метода таков:

Разобьем отрезок [0,l] на котором определено уравнение точками , где h - пространственный шаг, , n - число точек разбиения.

-временной шаг, ; , T - конец временного отрезка, у нас будет T = 10.

Точное значение - заменяем приближенным значением .

m - целое число, как в методе Эйлера, только здесь оно зависит от значения и : m = [/]

,

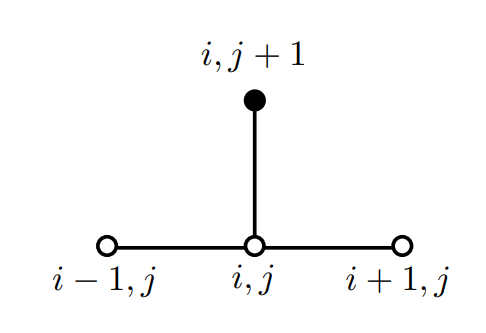
где ,

Для получения начальных и остальных значений воспользуемся условиями начальными и граничными:

Начальные условия:

Граничные условия:

Сам метод можно описать рисунком:



Где пустые кружки - это известные значения в узле; закрашенные - неизвестные, которые мы ищем.

По аналогу с методом Эйлера, посчитаем отклонение от точного решения по формуле: для разных и h. Результат в таблице:

|  |  |  |  |
| --- | --- | --- | --- |
| =0.01 |  |  | метод расходится |
| =0.001 |  |  | 0.412 |
| =0.0001 |  |  |  |

Результаты вычисления метода, выведенные на график и в таблице:

| t\x | x=0.2 | x=0.4 | x=0.6 | x=0.8 | x=1 |
| --- | --- | --- | --- | --- | --- |
| t=0.5 |  |  |  |  |  |
| t=1 |  |  |  |  |  |
| t=2.5 |  |  |  |  |  |
| t=5 |  |  |  |  |  |
| t=7.5 |  |  |  |  |  |
| t=10 |  |  |  |  |  |

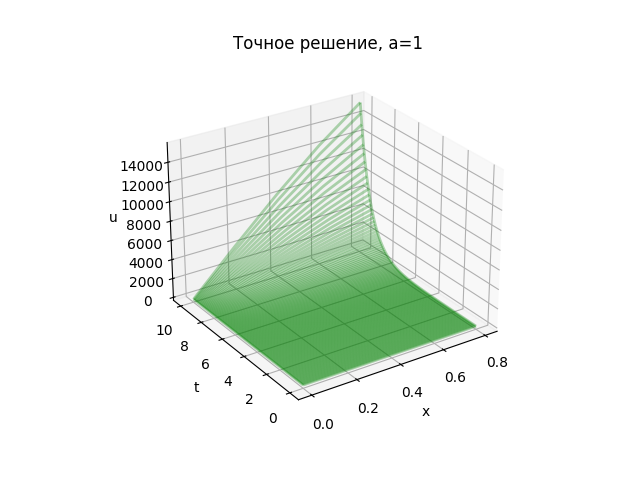
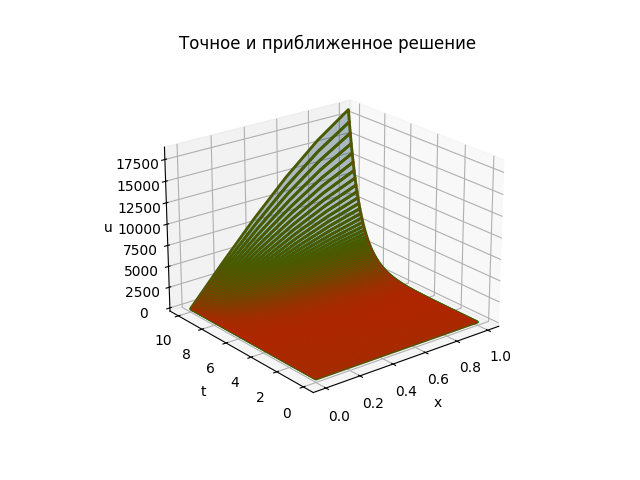
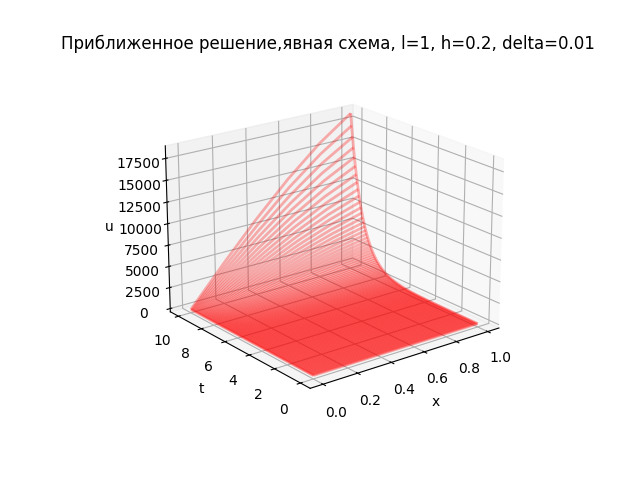
  
У явного метода есть проблема, если не соблюдать условие , то метод будет неустойчивый, и график будет такой:

Таблица такая:

| x\t | t=0.5 | t=1 | t=2.5 | t=5 | t=7.5 | t=10 |
| --- | --- | --- | --- | --- | --- | --- |
| x=0.2 |  |  |  |  |  |  |



Для решения этой проблемы есть неявный метод, который сходится всегда, т.е. абсолютно устойчив.

Сам метод сходится с порядком: т.е. 1 порядок по шагу времени, 2 порядок по шагу пространства. Снова по аналогу с методом Эйлера, проверим что сходится с помощью формул:

Таблица для :

|  |  |  |  |
| --- | --- | --- | --- |
| =0.01 |  |  | метод расходится |
| =0.001 |  |  |  |
| =0.0001 |  |  |  |

По таблице видно что порядок по - равен 1, причем слишком малые не улучшают точность.

Таблица для :

|  |  |  |  |
| --- | --- | --- | --- |
| =0.01 |  | при h=0.2/2  расходится | метод расходится |
| =0.001 |  |  |  |
| =0.0001 |  |  |  |

По таблице видно что порядок по - равен 1.

3) Решение уравнения неявным сеточным методом.

У неявного метода, в отличии от явного, единственное изменение в том, что мы ищем значения на следующем временном слое с помощью всего одного значения с предыдущего слоя. Причем нету явной зависимости следующего слоя от предыдущего, поэтому метод называется неявный.

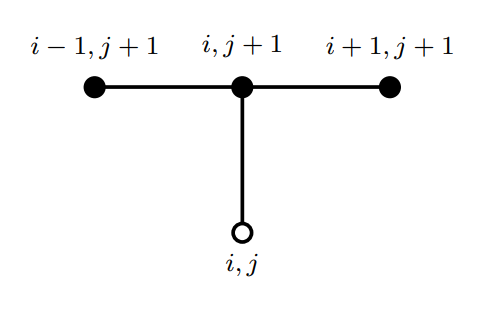
Алгоритм таков:

,

где

Начальные и граничные условия аналогичны явному методу

Рисунок метода будет выглядеть так:



т.е. у нас известно только 1 значение, а найти надо 3.

Если расписать уравнения метода на фиксированном слое j при каждом i=0,..,n, то получим систему, главная матрица которой - трехдиагональная, т.е. ненулевые значения присутствуют только на главной диагонали и на диагоналях выше и ниже главной.

Такие системы решаются методом прогонки. Причем метод будет сходится, если есть диагональное преобладание, т.е. модуль значения на главной диагонали превосходит сумму модулей значений на побочных диагоналях. В данной матрице это присутствует т.к. .

Метод прогонки основан на такой зависимости значений: , где - это неизвестные значения линейной трехдиагональной системы.

Подставив такое отношение в систему можно вывести способ расчета коэффициентов (их называют прогоночными). В нашей задаче они будут вычисляться таким образом:

, i=2,..,n

- т.е. граничное условие на j+1 слое

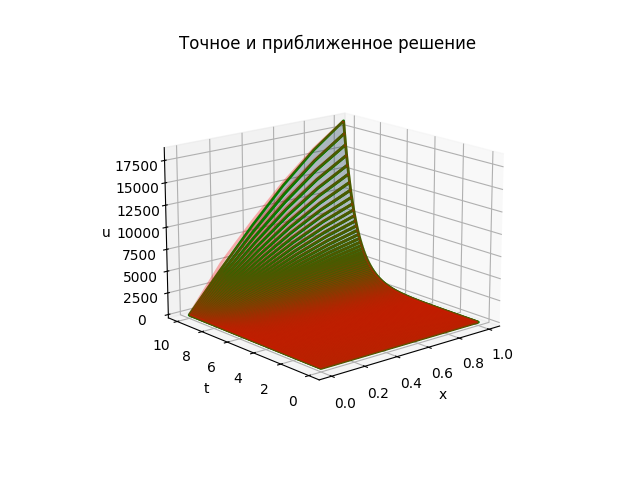
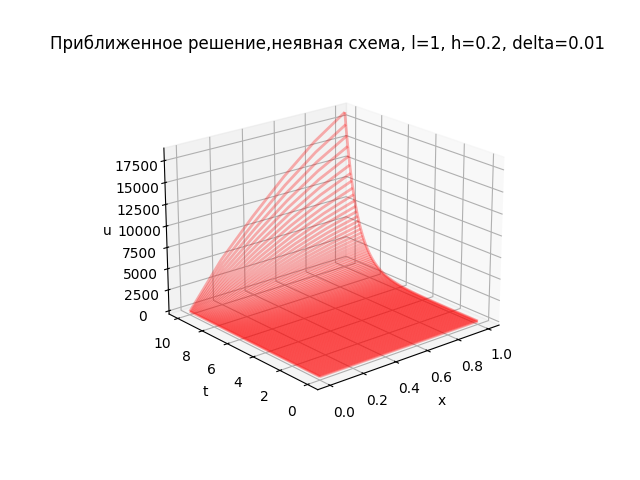
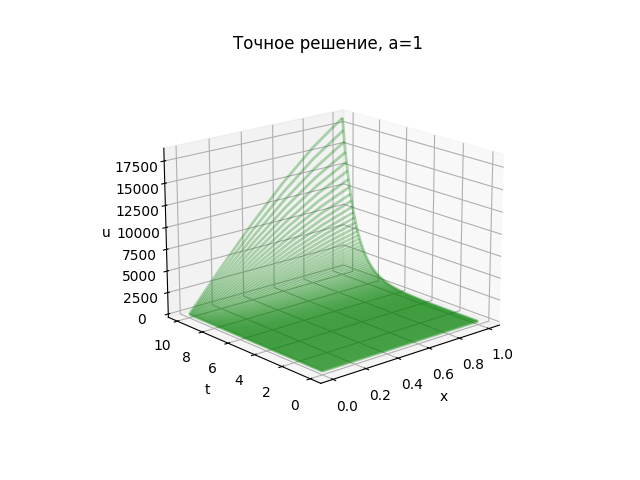
После получения значений прогоночных коэффициентов, на основе зависимости получаем значения в обратном порядке от , которое известно из граничных условий, до .Таблица отклонения приближенного решения от точного:

|  |  |  |  |
| --- | --- | --- | --- |
| =0.01 |  |  |  |
| =0.001 |  |  |  |
| =0.0001 |  |  |  |

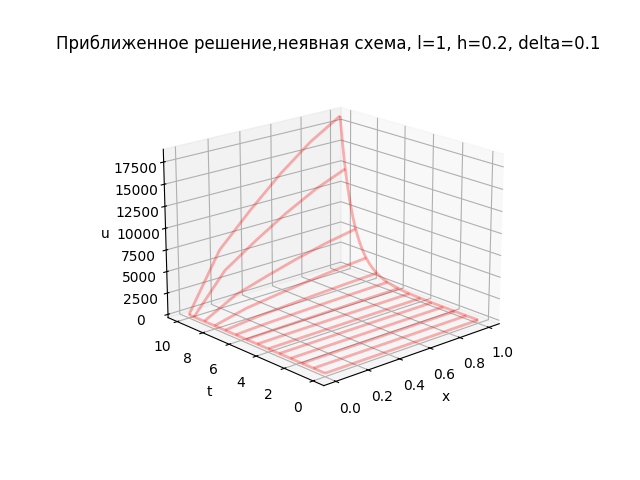
Сравнивая таблицу явного и неявного методов, можно заметить, что явный метод сходится быстрее т.к. его отклонение от точного решения гораздо меньше, в отличие от неявного метода при одинаковых шагах разбиения. Также в случае =0.01 и когда явный расходится, неявный сходится - это и есть главная особенность неявного метода.

Результаты вычисления метода, выведенные на график и в таблице:

| t\x | x=0.2 | x=0.4 | x=0.6 | x=0.8 | x=1 |
| --- | --- | --- | --- | --- | --- |
| t=0.5 |  |  |  |  |  |
| t=1 |  |  |  |  |  |
| t=2.5 |  |  |  |  |  |
| t=5 |  |  |  |  |  |
| t=7.5 |  |  |  |  |  |
| t=10 |  |  |  |  |  |



Проверим еще один случай, когда явный метод расходился:подставим параметры, при которых явный не работал, т.е. h=0.2, =0.1:



Видно, что метод сошелся.

Порядок сходимости неявного метода совпадает с порядком сходимости явного, т.е. он .

Аналогично проверим это используя значения и .

Таблица для :

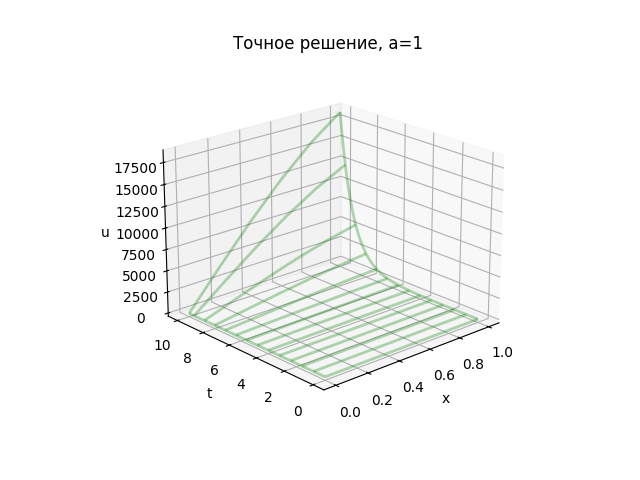
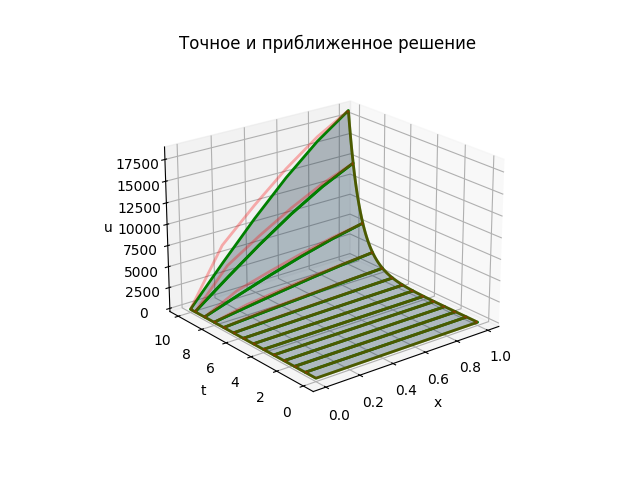
|  |  |  |  |
| --- | --- | --- | --- |
| =0.01 |  |  |  |
| =0.001 |  |  |  |
| =0.0001 |  |  |  |

Таблица для :

|  |  |  |  |
| --- | --- | --- | --- |
| =0.01 |  |  |  |
| =0.001 |  |  |  |
| =0.0001 |  |  |  |

Результаты точного решения и приближенного решения на 1 графике:

| t\x | x=0.2 | x=0.4 | x=0.6 | x=0.8 | x=1 |
| --- | --- | --- | --- | --- | --- |
| t=0.5 |  |  |  |  |  |
| t=1 |  |  |  |  |  |
| t=2.5 |  |  |  |  |  |
| t=5 |  |  |  |  |  |
| t=7.5 |  |  |  |  |  |
| t=10 |  |  |  |  |  |



# **Заключение**

Было разобрано 2 сеточных метода решения уравнения теплопроводности с постоянным запаздыванием в неоднородности: явный и неявный. Полученный результат совпадает с теорией: явный метод требует соблюдения условий сходимости, в то время как невинный - абсолютно устойчивый и сходится всегда. Но неявный метод хоть и имеет тот же порядок, что и явный, но у него нету заметна одна проблема: он может не стать точнее, если уменьшать только один шаг h. Это особенно заметно при значении , ведь отрицательные значения говорят о том, что при шаге h, погрешность A(h,) ниже, чем при шаге h/2. Также явный метод оказался более точен. Поэтому, хоть у явного метода есть недостаток, а именно условие устойчивости, если есть возможность условие соблюсти, метод будет выгоднее.

Помимо разбора методов, было изучена работа с библиотеками matplotlib и numpy, которые позволили создать графики по матрице значений.

# **Список источников**

<https://www.python.org/> - среда разработки Python

<https://matplotlib.org/2.0.2/index.html> - документация по библеотеке matplotlib

<https://numpy.org/> - документация по библеотеке numpy

**Литература**

Л. Э. Эльсгольц, С. Б. Норкин “Введение в теорию дифференциальных уравнений с отклоняющимся аргументом 1971”

В. Г. Пименов “Численные методы 2013”

В. Г. Пименов, А. В. Лекомцев “Современные проблемы математики. Моделирование динамических процессов 2023”

В. Г. Пименов “Разностные методы решения уравнений в частных производных с наследственностью 2014”

Г.В. Кузнецов, М.А. Шеремет “Разностные методы решения задач теплопроводности 2007”