МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 9**

**по дисциплине  
 «Методы вычислений»**

Выполнил студент группы 25/2                                             В.И.Яценко

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Курс    2

Краснодар

2025 г.

## **Постановка задачи**

Построить явную и неявную разностную схему решения уравнения, определить ее устойчивость. Сравнить полученные результаты графически и аналитически. Составить отчет по выполнению работы.

Дано уравнение теплопроводности:

*, ,*

с начальным условием:

,

и граничными условиями:

, ,

Известно точное значение для проверки:

Задача также подразумевает, что параметр Т выбирается самостоятельно.

## **Описание разработанного алгоритма**

Под разностной схемой понимается метод численного решения дифференциальных уравнений (обыкновенных или в частных производных), основанный на замене производных конечно-разностными аналогами. Сеточная функция аппроксимирует исходную непрерывную функцию, а производные заменяются разностными отношениями.

Начальное и граничные условия имеют вид:

, (1)

, , (2)

Разностная схема имеет вид:

*,* (3)

n – номер текущего слоя, – значение сеточной функции в точке на временном слое n, i – номер узла по пространству, h – шаг сетки по пространству, – шаг по времени, – источник тепла (внешнее воздействие)

Схема (3) представляет собой систему линейных алгебраических уравнений с числом уравнений, равным числу неизвестных. Находить решение такой системы следует по слоям.

Начальное условиезадает состояние системы в начальный момент времени . Граничные условия (2) определяют поведение решения на границах за все время t.

Решение на нулевом слое задано начальными условиями (1). Решениена слое находится по формуле:

которая получена следующим образом:

Умножаем обе части на :

Переносим вправо:

Для решения задачи была написана программа на языке C++. На вход программа получает значения T, и шаг по пространственной сетке h.

Рассмотрим уравнение вида:

(4)

Будем искать частные решения, имеющие вид:

(5)

где I – мнимая единица, – любое действительное число и q – число, подлежащее определению.

Подставляя (5) в (4) и сокращая на , найдем значения:

Если множитель q станет по модулю больше единицы, то решение вида (5) будет неограниченно возрастать при , то есть разностное уравнение вида (4) будет неустойчивым. В противном случае – устойчивым. В случае неустойчивости уравнения найти решение разностной задачи практически невозможно, так как погрешности, внесенные в начальный момент времени, будут неограниченно возрастать при увеличении n.

Уравнение вида (4) будет устойчивым тогда, когда выполняется неравенство:

Для вычислений вводится параметр T, и h, на основе которых вычисляются Nx (количество пространственных шагов) и Nt (количество временных шагов). В коде присутствует проверка на устойчивость:  
bool is\_stable = (tau <= h \* h / 2.0);

cout << "Условие устойчивости: " << (is\_stable ? "выполнено" : "нарушено") << endl;

При этом, если схема неустойчива, то ее значения вычисляться не будут.

Так как неявная схема безусловно устойчива, данная проверка распространяется только на явную схему.

Программа выводит в консоль результаты расчетов для явной и неявной схем для последнего временного слоя (, вводимый с клавиатуры. Результаты выводятся в консоль для дальнейшего построения графиков.

## **Расчёты и числовые результаты**

Код разработанной программы на выбранном языке программирования:

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <iomanip>

#define \_USE\_MATH\_DEFINES

#include <math.h>

#include <algorithm>

using namespace std;

const double PI = acos(-1);

// Точное решение

double exact\_solution(double x, double t) {

return exp(-t) \* sin(x);

}

// Явная схема

void explicit\_scheme(double h, double tau, int Nx, int Nt, vector<vector<double>>& u\_explicit) {

// Начальное условие

for (int i = 0; i <= Nx; ++i) {

double x = i \* h;

u\_explicit[i][0] = sin(x);

}

// Временные шаги

for (int n = 0; n < Nt; ++n) {

double t = (n + 1) \* tau;

// Внутренние узлы

for (int i = 1; i < Nx; ++i) {

u\_explicit[i][n + 1] = u\_explicit[i][n] + (tau / (h \* h)) \*

(u\_explicit[i + 1][n] - 2 \* u\_explicit[i][n] + u\_explicit[i - 1][n]);

}

// Граничные условия

u\_explicit[0][n + 1] = 0.0;

u\_explicit[Nx][n + 1] = exp(-t);

}

}

void jacobi(double T, int N, int M, vector<vector<double>>& u2) {

double eps = 1e-200;

//int max\_iterations = 1000;

double h = M\_PI / (2 \* N);

double k = T / M;

for (int i = 0; i <= N; i++) {

u2[i][0] = sin(i \* h); // u(x, 0) = sin(x)

}

double r = k / (h \* h);

for (int j = 0; j < M; j++) {

// u(0, t) = 0

u2[0][j + 1] = 0.0;

// u(PI/2, t) = e^(-t)

u2[N][j + 1] = exp(-(j + 1) \* k);

for (int i = 1; i < N; ++i) {

u2[i][j + 1] = u2[i][j];

}

int iteration = 0;

double max\_diff;

do {

max\_diff = 0.0;

vector<double> u\_old(N + 1);

for (int i = 0; i <= N; i++) {

u\_old[i] = u2[i][j + 1];

}

for (int i = 1; i < N; ++i) {

double u\_new = (u2[i][j] + r \* u2[i - 1][j + 1] + r \* u2[i + 1][j + 1]) / (1 + 2 \* r);

u2[i][j + 1] = u\_new;

max\_diff = max(max\_diff, abs(u\_new - u\_old[i]));

}

iteration++;

} while (max\_diff > eps);

}

}

int main() {

setlocale(LC\_ALL, "Ru");

// Параметры сетки

cout << "Введите параметры:" << endl;

cout << "T: ";

double T;

cin >> T;

//cout << "h: ";

double h = PI/20.0;

//cin >> h;

cout << "tau: ";

double tau;

cin >> tau;

double L = PI / 2;

int Nx = static\_cast<int>(round(L / h)); // Число узлов по пространству

int Nt = static\_cast<int>((T / tau)); // Число узлов по времени

cout << Nx << " " << Nt << endl;

// Выделение памяти для решений

vector<vector<double>> u\_explicit(Nx + 1, vector<double>(Nt + 1, 0.0));

vector<vector<double>> u\_implicit(Nx + 1, vector<double>(Nt + 1, 0.0));

double res = h \* h / 2.0;

bool is\_stable = (tau <= h \* h / 2.0);

cout << "Условие устойчивости: " << (is\_stable ? "выполнено" : "нарушено") << endl;

// Решение явной и неявной схемами

if (is\_stable) {

cout << "явная схема: \n";

explicit\_scheme(h, tau, Nx, Nt, u\_explicit);

for (int i = 0; i <= Nx; ++i) {

double x = i \* h;

cout << x << fixed << " " << u\_explicit[i][Nt] << " " << exact\_solution(x, T) << " " << abs(u\_explicit[i][Nt] - exact\_solution(x, T)) << endl;

}

}

cout << endl;

jacobi(T, Nx, Nt, u\_implicit);

cout << "неявная схема: \n";

for (int i = 0; i <= Nx; ++i) {

double x = i \* h;

cout << x <<fixed<< " " << u\_implicit[i][Nt] << " " << exact\_solution(x, T)<<" " << abs(u\_implicit[i][Nt] - exact\_solution(x, T)) << endl;

}

cout << endl;

return 0;

}

Результаты тестирования программы:

Заданные параметры: , h = .

|  |  |  |  |
| --- | --- | --- | --- |
| x | Найденное значение | Точное значение | Погрешность |
| 0 | 0 | 0 | 0.00000 |
| 0,15708 | 0,057591 | 0,057549 | 0,000042 |
| 0,314159 | 0,113762 | 0,113681 | 0,000081 |
| 0,471239 | 0,167128 | 0,167014 | 0,000114 |
| 0,628319 | 0,216372 | 0,216234 | 0,000138 |
| 0,785398 | 0,260281 | 0,26013 | 0,000151 |
| 0,942478 | 0,297772 | 0,297621 | 0,000151 |
| 1,099557 | 0,32792 | 0,327783 | 0,000137 |
| 1,256637 | 0,349981 | 0,349874 | 0,000107 |
| 1,413717 | 0,363411 | 0,36335 | 0,000061 |
| 1,570796 | 0,367879 | 0,367879 | 0 |

Таблица 1 – значения в момент времени t=T для явной схемы

|  |  |  |  |
| --- | --- | --- | --- |
| x | Найденное значение | Точное значение | Погрешность |
| 0 | 0 | 0 | 0 |
| 0,15708 | 0,057618 | 0,057549 | 0,000069 |
| 0,314159 | 0,113814 | 0,113681 | 0,000133 |
| 0,471239 | 0,167201 | 0,167014 | 0,000187 |
| 0,628319 | 0,216461 | 0,216234 | 0,000227 |
| 0,785398 | 0,260378 | 0,26013 | 0,000248 |
| 0,942478 | 0,297869 | 0,297621 | 0,000248 |
| 1,099557 | 0,328007 | 0,327783 | 0,000225 |
| 1,256637 | 0,350049 | 0,349874 | 0,000175 |
| 1,413717 | 0,36345 | 0,36335 | 0,0001 |
| 1,570796 | 0,367879 | 0,367879 | 0 |

Таблица 2 – значения в момент времени t=T для неявной схемы

## **Анализ результатов**

Графическое сравнение результатов:

Рисунок 1 – графическое сравнение найденного и точного значений для явной схемы в момент времени t=T

Рисунок 2 – графическое сравнение найденного и точного значений для неявной схемы в момент времени t=T

Рисунок 3 – графическое сравнение найденного и точного значений для неявной схемы в момент времени t=T

Рисунок 4 – графическое сравнение найденного и точного значений для неявной схемы в момент времени t=T

## **Выводы**

Точность численных методов зависит от параметров задачи и выбранных шагов дискретизации. Неявные схемы выгоднее для длительных расчётов благодаря безусловной устойчивости - они позволяют брать крупные шаги по времени, экономя вычислительные ресурсы, хотя и требуют решения систем уравнений на каждом шаге. Явные методы проще в реализации и точнее на коротких интервалах, но жёсткие ограничения на шаг (например, τ ≤ h²/2 для уравнения теплопроводности) делают их неэффективными для долгосрочного моделирования. Выбор схемы всегда представляет компромисс между точностью и вычислительными затратами.

Оптимальный подход определяется конкретными условиями задачи: при моделировании быстропротекающих процессов с высокими требованиями к точности предпочтительны явные схемы, тогда как для долгосрочных прогнозов и жёстких систем лучше подходят неявные методы.