**­­­­­МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**Лабораторная работа №3**

Сечко Марины Юрьевны

студентки 3 курса, специальность

«компьютерная безопасность»

Преподаватель:

Максим Александрович Полещук,

ассистент кафедры вычислительной математики

Минск, 2017

1. **Постановка задачи.**

**Задание.** Рассмотрим следующую смешанную задачу для линейного уравнения одномерной теплопроводности: тонкий плоский вертикальный электропроводящий слой, заключенный между двумя толстыми плоскими слоями обычного покрытия, нагревается проходящим через него электрическим током:



Начальные условия запишутся следующим образом:



Дифференциальное уравнение выбрать в соответствии с вариантом задания: положить параметр  равным номеру в списке академической группы, параметр  — номеру группы.

Построить алгоритм определения приближенного решения для явной разностной схемы



на двумерной сетке {(*mh*, *n*τ) *| m=*0,1,…,*M*, *Mh=*π, *n=*0,1,…,*N*, *N*τ*=*50} с числом шагов  по координате *x*, используя шаблон из четырех точек . Провести вычисления для двух значений τ:



Вычисление для второго значения τ остановить при накоплении погрешности до неприемлемой величины (условие устойчивости разностной схемы не выполняется).

В отчете для каждого проведенного вычисления представить рисунок: для первого вычисления — рисунок с шестью графиками приближенной функции  на слоях с номерами , для второго вычисления — рисунок аналогичный первому вычислению с числом графиков, соответствующих числу отработавших слоев до останова программы.

1. **Входные данные**
2. **Краткая теория**

Рассмотрим следующую смешанную задачу для линейного уравнения одномерной теплопроводности:

 (1)

Начальные условия:

 (2)

Явная разностная схема:

 (3)

 (4)

Значения  для любого j могут быть определены с помощью аппроксимации  начального условия:  (5)

Если задать n = 0, то, пользуясь выражением (4), можно определить значения . Значения  определяются с помощью аппроксимации граничных условий (5). Далее, задаём n = 1 и из выражения (4) определяем , а значения   - опять же с помощью аппроксимации граничных условий (5) и т. д. Таким образом, соотношение (4) позволяет рассчитать все значения искомой функции u в узлах разностной сетки, кроме значений, задаваемых с помощью начального и граничных условий.

1. **Листинг программы**

public class Test {

public static void func(float tay1, float tay2, float h) {

int M = 20, N = (int) (50f / tay1);

float sigma = 17f;

float max = 0f, eps = 1f; //

int[] n = new int[6];

for (int k = 0; k <= 5; k++) {

n[k] = (int) (10 \* k / tay1);

}

float[] y\_old\_1 = new float[M + 1];

float[] y\_new\_1 = new float[M + 1];

float[] y\_old\_2 = new float[M + 1]; //

float[] y\_new\_2 = new float[M + 1]; //

float x = 0f, pi = (float) Math.PI, pi2 = pi / 2;

for (int i = 0; i <= 20; i++) {

if (x <= pi2) {

y\_old\_1[i] = 9 \* x;

y\_old\_2[i] = 9 \* x; //

} else {

y\_old\_1[i] = 9 \* (pi - x);

y\_old\_2[i] = 9 \* (pi - x); //

}

x += h;

}

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

System.out.printf("(%f;%f) ", i \* h, y\_old\_1[i]);

}

System.out.println();

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

y\_new\_1[0] = 0;

y\_new\_1[M] = 0;

y\_new\_2[0] = 0; //

y\_new\_2[M] = 0; //

for (int j = 1; j < M; j++) {

y\_new\_1[j] = sigma \* tay1 / h / h \* (y\_old\_1[j - 1] - 2 \* y\_old\_1[j] + y\_old\_1[j + 1]) + y\_old\_1[j];

y\_new\_2[j] = sigma \* tay2 / h / h \* (y\_old\_2[j - 1] - 2 \* y\_old\_2[j] + y\_old\_2[j + 1]) + y\_old\_2[j]; //

}

boolean flag = false;

for (int k = 1; k <= 5; k++) {

if (n[k] == i) {

flag = true;

break;

}

}

if (flag) {

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

// System.out.println(j \* h);

System.out.println(y\_new\_1[j]);

}

System.out.println();

}

for (int k = 0; k <= M; k++) { /////////////

if (Math.abs(y\_new\_1[k] - y\_new\_2[k]) > max) {

max = Math.abs(y\_new\_1[k] - y\_new\_2[k]);

}

}

if (eps > max) {

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

System.out.print("(" + (j \* h) + ";" + y\_new\_2[j] + ") ");

}

System.out.println();

} else {

break;

}

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

y\_old\_1[j] = y\_new\_1[j];

y\_old\_2[j] = y\_new\_2[j]; //

}

}

}

public static void main(String[] args) {

try {

int M = 20;

float sigma = 17f, tay1, tay2, h = (float) Math.PI / M;

tay1 = 5f / 11 \* h \* h / sigma;

tay2 = 5f / 9 \* h \* h / sigma;

func(tay1, tay2, h);

} catch (Exception e) {

e.printStackTrace();

}

}

}

1. **Результаты**

Рисунок с шестью графиками приближенной функции  на слоях с номерами  для :

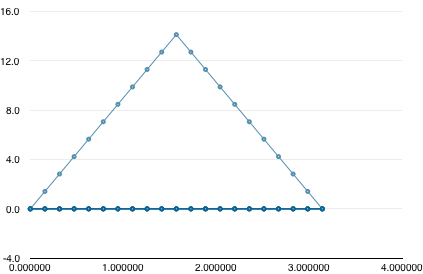
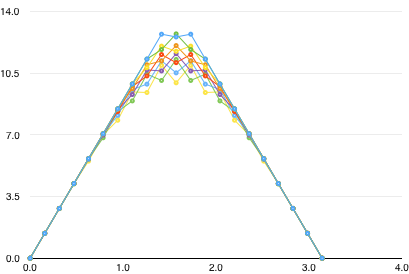


Рисунок с шестью графиками приближенной функции  на слоях с номерами для :



1. **Выводы**

Условие устойчивости данной разностной схемы, накладывающее ограничение на выбор интервала деления при создании разностной сетки (или, иначе говоря, ограничение на выбор расчётного шага по одной из независимых переменных):



Отметим, что это, безусловно, является недостатком явной разностной схемы (3). В то же время она имеет достаточно простой метод решения.