**Министерство науки и высшего образования РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ИС**

отчет

**по лабораторной работе №11**

**по дисциплине «Конструирование программ»**

Тема: Решение задачи Дирихле для уравнения Лапласа методом сеток

|  |  |  |
| --- | --- | --- |
| Студент гр. 8363 |  | Нерсисян А.С. |
| Преподаватель |  | Копыльцов А.В. |

Санкт-Петербург

2020

**Цель работы.**

С помощью подпрограммы Dirichlet найти приближенное решение уравнения Лапласа в заданной области с указанными граничными условиями.

**Основные теоретические положения.**

Найти приближенное решение уравнения Лапласа



в квадрате , принимающее на границе области  заданные краевые условия



Построим область решения, покроем ее сеткой с шагом , ,  и вычислим значения искомой функции  в граничных точках области по формулам (8.9.2). Введем обозначения  и аппроксимируем частные производные  и  в каждом внутреннем узле сетки центральными разностными производными второго порядка по формулам (8.6.7) и (8.6.8):



Уравнение Лапласа (8.9.1) заменим конечно-разностным уравнением

.

Погрешность замены дифференциального уравнения разностным составляет величину . Уравнение (8.9.4) вместе со значениями  в граничных узлах образуют систему линейных алгебраических уравнений относительно приближенных значений  в узлах сетки:

.

Численное решение задачи Дирихле для уравнения Лапласа в области  состоит в нахождении приближенных значений  искомой функции  во внутренних узлах сетки. Для определения величин  необходимо решить систему линейных алгебраических уравнений

В данной лабораторной работе система решается методом простых итераций (см. подразд. 5.5) по формулам

,

где верхним индексом  обозначен номер итерации. В качестве условия окончания итерационного процесса можно принять



Как следует из теории, изложенной в подразд. 8.6, описанная разностная схема обладает свойством устойчивости и сходимости. Это означает, что, выбрав достаточно малый шаг , можно сколь угодно точно решить исходную задачу.

Итак, решим по описанной процедуре модельный пример. Пусть надо найти решение  задачи Дирихле в квадрате со стороной, равной единице, для уравнения Лапласа (8.9.1) с краевыми условиями вида





Зададим одинаковый шаг по  и по , равный , и вычислим все краевые условия. На приведенном рисунке известные значения функции  в граничных точках помечены жирными точками, разыскиваемые приближенные решения внутри квадрата  - крестиками, внутри  выделен используемый шаблон.











*y*

1

*x*

1

    и так далее  









Поскольку система (8.9.5) будет решаться методом простых итераций, зададим начальное приближение, то есть вектор  в каждом горизонтальном слое. Сначала рассмотрим горизонталь с граничными точками (0, 0.1) и (1, 0.1). Будем считать, что функция  по горизонталям области  распределена равномерно. , , и так как отрезок разбит на десять частей (см. рисунок), то шаг изменения функции .

Тогда получим ,  и так далее. Аналогичным образом рассчитаем начальные значения функции во внутренних точках других горизонталей.

(0, 0.1)      (1, 0.1)

Выполним теперь эти же действия средствами пакета Mathcad:

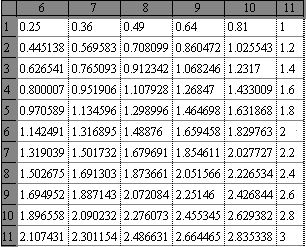


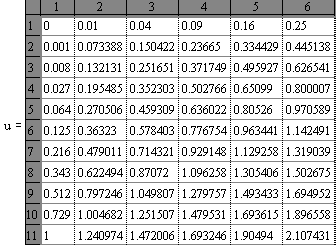
Теперь можно запустить вычисления по методу простых итераций. На практике вместо условий (8.9.7) окончания итерационного процесса применяют более надежный критерий



где .

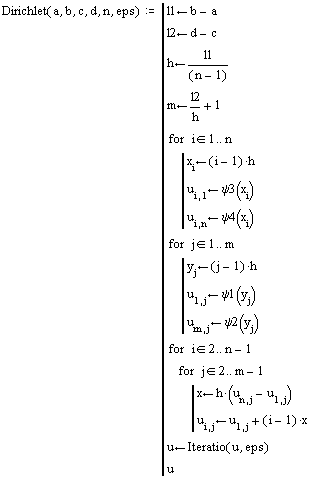
Метод простых итераций выгоднее всего оформить в виде небольшой подпрограммы. Попытка огранизовать вычисления с помощью дискретных аргументов приведет к неоправданной сложности математических выражений. Записать процесс последовательных приближений в векторно-матричной форме, не удастся, так как необходимо обеспечить выход из итерационного процесса по достижении заданной точности. Программа метода простых итераций очень проста и понятна и может быть, например, такой:





Число итераций для выполнения условия сходимости (8.9.8) в этом примере равно 111.

Наконец, описанный выше алгоритм можно оформить в виде единственной подпрограммы Dirichlet:



Входные параметры этой подпрограммы: - отрезок по оси , - отрезок по оси , - число шагов по . Так как использована формула (8.9.5), где , то число шагов сетки по оси  вычисляется в самой программе. Желающие могут легко усовершенствовать алгоритм для случая .

Итак, решим еще раз уравнение Лапласа для этого примера.



Если распечатать еще раз матрицу , получим точно такую же таблицу, как на предыдущей странице. Как видно, результаты вычислений полностью совпадают.

**Экспериментальные результаты.**

**Задание № 1**

С помощью подпрограммы Dirichlet найти приближенное решение уравнения Лапласа в заданной области с указанными граничными условиями.:

**Дано:** Вариант 11

**Обработка результатов эксперимента.**

**Задание № 1. решение:**

#include <iostream>

#include <conio.h>

using namespace std;

double psi1(double y)

{

return y\*y;

}

double psi2(double y)

{

return y\*y + 2 \* y;

}

double psi3(double x)

{

return 2 \* (x\*x - x);

}

double psi4(double x)

{

return x\*x - 1;

}

int main()

{

double x1 = -1, x2 = 1, y1 = -2, y2 = 0, h = 0.2, u[11][11], E = 0.001, l, next;

bool rep;

cout.precision(3);

for (int i = 0; i < 11; ++i)

u[0][i] = psi1(y1 + h \* i);

for (int i = 0; i < 11; ++i)

u[10][i] = psi2(y1 + h \* i);

for (int i = 0; i < 11; ++i)

u[i][0] = psi3(x1 + h \* i);

for (int i = 0; i < 11; ++i)

u[i][10] = psi4(x1 + h \* i);

for (int i = 1; i < 10; ++i)

{

l = (u[10][i] - u[0][i]) / 10;

for (int j = 1; j < 10; ++j)

u[j][i] = u[0][i] + j\*l;

}

do

{

rep = false;

for (int i = 1; i < 10; ++i)

for (int j = 1; j < 10; ++j)

{

next = (u[i][j - 1] + u[i][j + 1] + u[i - 1][j] + u[i + 1][j]) / 4;

if (abs(next - u[i][j]) > E) rep = true;

u[i][j] = next;

}

} while (rep);

for (int i = 0; i < 11; ++i)

{

for (int j = 0; j < 11; ++j)

cout << u[j][i] << " ";

cout << endl;

}

\_getch();

return 0;

}

**Выводы.**

В ходе выполнения данной лабораторной была написана программа, которая с помощью подпрограммы Dirichlet находит приближенное решение уравнения Лапласа в заданной области с указанными граничными условиями.