**Санкт-Петербургский политехнический университет Петра Великого**

Физико-механический институт

Высшая школа теоретической механики и математической физики

Индивидуальное задание №3

**Решение уравнения Лапласа методом конечных разностей**

Вариант 21

|  |  |  |
| --- | --- | --- |
| Выполнил  студент 5030103/10201 |  | Елизавета Сорокопудова |
| Проверил |  | Витохин Евгений Юрьевич |

Санкт-Петербург

2023 г

1. **Постановка задачи**

Используя МКР, составить приближенное решение задачи Дирихле для

уравнения Лапласа в квадрате ABCD с вершинами

A (0; 0), B (0; 1), C (1; 1), D (1; 0), шаг h = 0.2

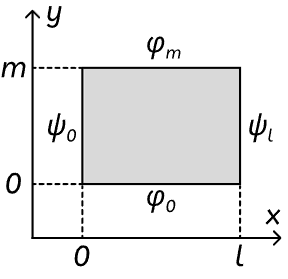
При решении задачи использовать итерационный процесс усреднения

Либмана до получения ответа с точностью до 0,01

Задаются граничные условия на сторонах квадрата:

Округление до 4 знаков после запятой

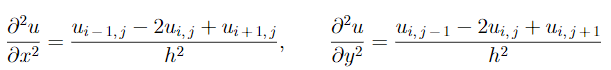
1. **Метод решения**

Рассмотрим уравнение Лапласа в прямоугольной области длиной l и шириной m в декартовой системе координат:

Зададим граничные условия первого рода:

Вводим конечно-разностную сетку, разбивая ось x на N узлов, а ось y – на M узлов:

Частные производные второго порядка будем аппроксимировать следующими соотношениями:



Подставив выражения в исходное дифференциальное уравнение и выразив получим систему линейных алгебраических уравнений вида



Воспользуемся итерационным методом. Введем

начальное приближение, которое будем уточнять на каждой итерации:

Начальное приближение выбирается произвольным, но чем ближе оно к истинному распределению искомой величины, тем меньше понадобится итераций для нахождения решения. На каждой новой итерации k значение ис комой функции будем определять так

где *ω* – константа метода, – начальное приближение, уточненное на *k*-ой итерации.

В зависимости от значения константы *ω* данный метод имеет следующие названия:

– метод Зейделя, *ω = 1*

– метод последовательной верхней релаксации, *ω > 1*

– метод последовательной нижней релаксации, *ω < 1*

Условием останова итерационной процедуры является условие, согласно

которому норма разности значений искомой функции на следующей k + 1

итерации и текущей k итерации должна быть меньше заданной точности

(1)

В качестве можно использовать, например, бесконечную норму. В таком случае условие (1) запишется в таком виде:

1. **Полученные результаты**.

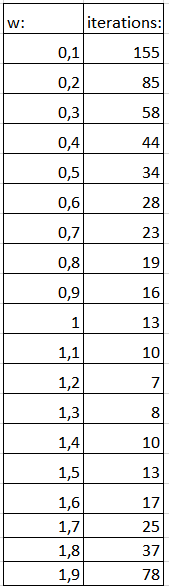


Таблица 1. Зависимость количества итераций от константы ω

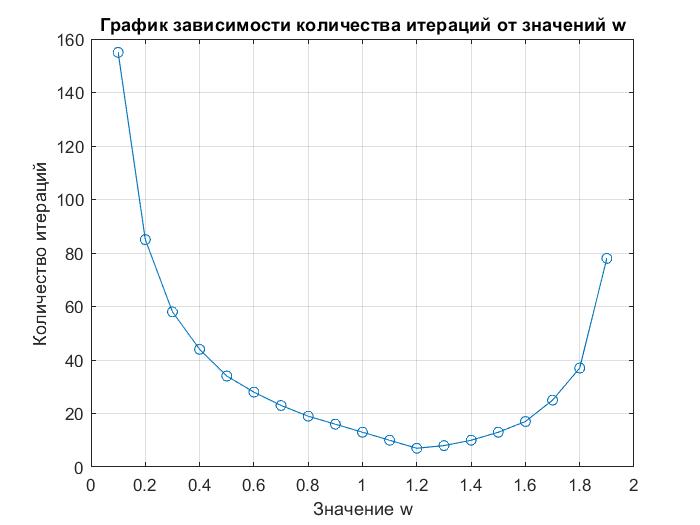


Рисунок 1. График зависимости количества итераций от константы ω

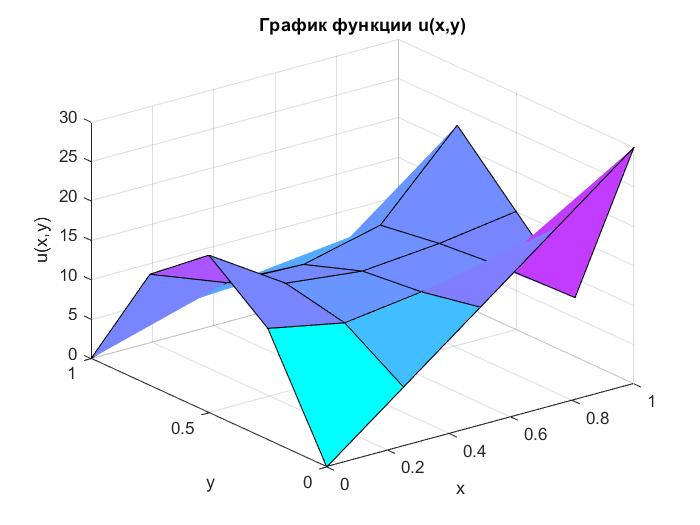


Рисунок 2. График поверхности 𝑢=𝑢(𝑥,𝑦)

**3. Код программы, MatLab**

clc;

clear all;

a = 0;

b = 1;

h = 0.2;

n = (b-a)/h;

x = linspace(a, b, n);

y = linspace(a, b, n);

U = zeros(n, n);

for i = 1:n

U(i, 1) = 20.\*sin(pi.\*y(i));

U(i, n) = 30.\*x(i);

U(1, i) = 30 \* y(i);

U(n, i) = 20.\*x(i).\*(1-x(i));

end

w = 0;

eps = 0.01;

U\_k = zeros(n, n);

U\_k = U;

U\_k1 = zeros(n, n);

U\_k1 = U;

w = 0;

W = zeros(1, 19);

iter = 0;

ITER = zeros(1, 19);

for k = 1:19

w = k \* 0.1;

U\_k = U;

U\_k1 = zeros(n, n);

iter = 0;

while norm(U\_k1 - U\_k) > eps

iter = iter + 1;

U\_k1 = U\_k;

for i = 2:(n-1)

for j = 2:(n-1)

U\_k(i, j) = U\_k(i, j)\*(1-w) + w\*0.25\*(U\_k(i-1, j) + U\_k(i+1, j) + U\_k(i, j-1) + U\_k(i, j+1));

end

end

end

ITER(1, k) = iter;

W(1, k) = w;

fprintf('w: %.1f, iterations: %d\n', w, iter);

end

w1 = 0.1;

iter = 0;

U\_k = U;

U\_k1 = zeros(n, n);

while norm(U\_k1 - U\_k) > eps

iter = iter + 1;

U\_k1 = U\_k;

for i = 2:(n-1)

for j = 2:(n-1)

U\_k(i, j) = U\_k(i, j)\*(1-w1) + w1\*0.25\*(U\_k(i-1, j) + U\_k(i+1, j) + U\_k(i, j-1) + U\_k(i, j+1));

end

end

end

fprintf('w: %.1f, iterations: %d\n', w1, iter);

surf(x, y, U\_k);

title('График функции u(x,y)');

xlabel('x');

ylabel('y');

zlabel('u(x,y)');

colormap('cool')

figure;

plot(W, ITER, '-o');

xlabel('Значение w');

ylabel('Количество итераций');

title('График зависимости количества итераций от значений w');

grid on