

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«Национальный исследовательский  
Нижегородский государственный университет им. Н.И. Лобачевского»  
(ННГУ)**

**Институт информационных технологий, математики и механики**

Направление подготовки: «Фундаментальная информатика и информационные  
технологии»

Магистерская программа: «Инженерия программного обеспечения»

**ОТЧЕТ**

по лабораторной работе

**«Волновая схема на задаче Дирихле для уравнения Пуассона»**

**Выполнил:** студент группы 382006-  
1м

\_\_\_\_\_ А.А. Солуянов

**Проверил:**

к.ф.-м. н., доц., доцент каф. МОСТ

\_\_\_\_\_ К.А. Баркалов

Нижний Новгород  
2021

## Оглавление

Введение .....	3
Постановка задачи .....	4
Погрешность аппроксимации .....	5
Метод верхних релаксаций.....	5
Реализация.....	7
Описание алгоритма .....	7
Схема распараллеливания .....	8
Результаты.....	9
Тестовая инфраструктура .....	9
Эксперименты .....	9
Заключение .....	11
Литература .....	12

Математическое моделирование задач механики, физики и других отраслей науки и техники сводятся к дифференциальным уравнениям. В связи с этим решение дифференциальных уравнений является одной из важнейших математических задач. В вычислительной математике изучаются численные методы решения дифференциальных уравнений, которые особенно эффективны в сочетании с использованием вычислительной техники.

Задача Дирихле для уравнения Пуассона является одной из классических задач математической физики. Для решения уравнений с частными производными как правило используются сеточные методы. Нередко с помощью компьютера в области определения строится сетка, и составляется разностное уравнение, в котором искомыми неизвестными являются значения функции в узлах сетки. Решение разностного уравнения также можно искать по-разному. На практике широко применяются итерационные методы. Вычислительная схема в этом случае описывает, как следующее состояние сетки зависит от предыдущего. В результате счета на компьютере получается приближенное решение уравнений с частными производными.

Бывают сложные модели, в которых строятся сетки с большим количеством узлов: десятки миллионов и даже больше. Актуальной задачей для таких моделей является распараллеливание вычислений с целью сокращения времени счета.

В данной лабораторной работе будет рассмотрен метод релаксации с оптимальным параметром для задачи Дирихле для уравнения Пуассона с волновой схемой параллелизации вычислений.

## Постановка задачи

Рассмотрим уравнение Пуассона

$$\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = -f(x_1, x_2) \quad (1)$$

Будем искать его решение, непрерывное в прямоугольнике

$$\bar{G} = G \cup \Gamma = \{x = (x_1, x_2): 0 \leq x_\alpha \leq l_\alpha, \alpha = 1, 2\}$$

и принимающее на границе  $\Gamma$  заданные значения

$$u|_\Gamma = \mu(x) \quad (2)$$

Задача, определяемая уравнением (1) и условием (2), называется задачей Дирихле.

Физическая интерпретация задачи – изгиб упругой пластины.

Определим разностную схему для задачи (1), (2). Введем в  $\bar{G}$  сетку  $\bar{\omega}_h = \omega_h \cup \gamma_h = \{x_i = (i_1 h_1, i_2 h_2), i_\alpha = 0, 1, \dots, N_\alpha, h_\alpha = \frac{l_\alpha}{N_\alpha}, \alpha = 1, 2\}$  и обозначим через  $y_i = y_{i_1 i_2} = y(i_1, i_2) = y(x_i)$  сеточную функцию заданную на  $\bar{\omega}_h$ ;  $h_1$  и  $h_2$  – шаги сетки по координатами  $x_1$  и  $x_2$ .

Чтобы написать разностную схему для (1), (2), аппроксимируем каждую из производных  $\frac{\partial^2 u}{\partial x_\alpha^2}$  на трехточечным шаблоне, полагая

$$\frac{\partial^2 u}{\partial x_1^2} \sim \frac{u(x_1 - h_1, x_2) - 2u(x_1, x_2) + u(x_1 + h_1, x_2)}{h_1^2} \quad (3)$$

$$\frac{\partial^2 u}{\partial x_2^2} \sim \frac{u(x_1, x_2 - h_2) - 2u(x_1, x_2) + u(x_1, x_2 + h_2)}{h_2^2} \quad (4)$$

Пользуясь выражениями (3), (4), заменим (1) разностным уравнением

$$\begin{aligned} & \frac{y(i_1 - 1, i_2) - 2y(i_1, i_2) + y(i_1 + 1, i_2)}{h_1^2} + \frac{y(i_1, i_2 - 1) - 2y(i_1, i_2) + y(i_1, i_2 + 1)}{h_2^2} \\ & = -f(i_1, i_2) \end{aligned} \quad (5)$$

К этому уравнению надо присоединить краевые условия

$$y = \mu(x), x = (i_1 h_1, i_2 h_2) \in \gamma_h \quad (6)$$

Уравнения (5) и (6) в совокупности образуют систему линейных алгебраических уравнений (СЛАУ), где число уравнений системы равно  $(M - 1) \times (N - 1)$ , т.е. столько, сколько и неизвестных, это значения  $u_{mn}$  при  $m = 1, 2, \dots, M - 1$ ,  $n = 1, 2, \dots, N - 1$ .

## Погрешность аппроксимации

Пусть  $u = u(x)$  – решение задачи Дирихле (1), (2), а  $y = y(i_1, i_2)$  – решение разностной задачи (5), (6). Рассмотрим погрешность  $z(x) = y(x) - u(x)$ ,  $x = (i_1 h_1, i_2 h_2) \in \omega_h$

Подставляя  $y = z + u$  в (5) и (6), получаем для погрешности  $z = z(x)$  неоднородное уравнение

$$\Delta z = z_{x_1 x_1} + z_{x_2 x_2} = -\psi(x), x \in \omega_h(G) \quad (7)$$

с однородным краевым условием

$$z = 0 \text{ при } x \in \gamma_h \quad (8)$$

Здесь

$$\psi(x) = \Delta u + f(x) = u_{x_1 x_1} + u_{x_2 x_2} + f(x) \quad (9)$$

Есть невязка или погрешность аппроксимации для схемы (5) на решении  $u = u(x)$  уравнения (1).

Можно показать, что

$$|\psi| \leq M_4 \frac{h_1^2 + h_2^2}{24} \quad (10)$$

где

$$M_4 = \max\left(\left|\frac{\partial^4 u}{\partial x_1^4}\right|, \left|\frac{\partial^4 u}{\partial x_2^4}\right|\right) \quad (11)$$

А значит схема (5) имеет второй порядок аппроксимации.

## Метод верхних релаксаций

Среди явных одношаговых итерационных методов наибольшее распространение получил метод верхних релаксаций. Это связано с тем, что метод верхних релаксаций содержит свободный параметр  $\omega$ , изменяя который можно получать различную скорость сходимости итерационного процесса.

Наиболее эффективно этот метод применяется при решении множества близких алгебраических систем линейных уравнений. На первом этапе проводится решение одной из систем с различными значениями итерационного параметра  $\omega$  и из анализа скорости сходимости итерационного процесса выбирается оптимальное значение этого параметра. Затем все остальные системы решаются с выбранным значением  $\omega$ .

Еще одно достоинство итерационного метода верхних релаксаций состоит в том, что при его реализации на ЭВМ алгоритм вычислений имеет простой вид и позволяет использовать всего один массив для неизвестного вектора.

Основная вычислительная формула имеет вид

$$a_{ii}x_i^{s+1} = -\omega \sum_{j=1}^{i-1} a_{ij}x_j^{s+1} + (1-\omega)a_{ii}x_i^s - \omega \sum_{j=i+1}^n a_{ij}x_j^s + \omega b_i \quad (12)$$

Метод релаксации для разностной схемы имеет вид

$$Dv_{ij}^{s+1} = \frac{\omega}{h^2}v_{i-1,j}^{s+1} + \frac{\omega}{k^2}v_{i,j-1}^{s+1} + \frac{\omega}{h^2}v_{i+1,j}^s + \frac{\omega}{k^2}v_{i,j+1}^s + (1-\omega)Dv_{ij}^s + \omega f_{ij} \quad (13)$$

где  $D = 2(\frac{1}{h^2} + \frac{1}{k^2})$

Оптимальный параметр метода релаксации:

$$\omega_{\text{opt}} = 2/(1 + 2 \sin\left(\frac{\pi h}{2}\right)) \quad (14)$$

## Реализация

### Описание алгоритма

Входными параметрами алгоритма является объект типа *heat\_task*, который содержит информацию о размерности сетки, размерах пластины и функции, определяющие граничные условия и внешнее воздействие.

Формат выхода должен выглядеть как указатель на массив, в котором содержится решение задачи.

Предварительными действиями является подготовка выходного вектора и матрицы внешнего воздействия, так как она неизменна и часто используется в алгоритме, поэтому выгодно посчитать ее заранее.

Таким образом основное тело алгоритма (Листинг 1) можно представить, как итеративное вычисление элементов результирующего вектора с помощью метода верхней релаксации.

```
for (int iteration = 0; iteration < max_iter; iteration++) {
    for (int k = 0; k < task.n + task.m - 3; ++k) {

        int start = min(1 + k, task.n - 1);
        int finish = max(1, k - task.m + 3);

#pragma omp parallel for private(prev, curr_index, top_index, bot_index,
left_index, right_index)
        for (int i = start; i >= finish; --i) {

            int j = task.m - (k - i + 2);

            curr_index = i * (task.m + 1) + j;

            top_index = curr_index + 1;
            bot_index = curr_index - 1;
            left_index = curr_index - (task.m + 1);
            right_index = curr_index + (task.m + 1);

            prev = v[curr_index];

            v[curr_index] = omega * (h_ * (v[left_index] +
v[right_index]) + k_ * (v[top_index] + v[bot_index]) + f[i][j]) + \
                (1 - omega) * D * prev;
            v[curr_index] /= D;

        }
    }
}
```

Листинг 1. Основное тело алгоритма

## Схема распараллеливания

Рассмотрим возможность построения параллельного алгоритма, который выполнял бы только те вычислительные действия, что и последовательный метод (может быть только в несколько ином порядке) и, как результат, обеспечивал бы получение точно таких же решений исходной вычислительной задачи. Как уже было отмечено выше, в последовательном алгоритме каждое очередное  $k$ -ое приближение значения  $u_{ij}$  вычисляется по последнему  $k$ -му приближению значений  $u_{i-1j}$  и  $u_{ij-1}$  и предпоследнему  $k$ -му приближению значений  $u_{i+1j}$  и  $u_{ij+1}$ .

Таким образом, при требовании совпадения результатов вычислений последовательных и параллельных вычислительных схем в начале каждой итерации метода только одно значение  $u_{11}$  может быть пересчитано (возможности для распараллеливания нет). Но далее после пересчета  $u_{11}$  вычисления могут выполняться уже в двух узлах сетки  $u_{12}$  и  $u_{21}$  (в этих узлах выполняются условия последовательной схемы), затем после пересчета узлов  $u_{12}$  и  $u_{21}$  — в узлах  $u_{13}$ ,  $u_{22}$  и  $u_{31}$  и т.д. Обобщая сказанное, можно увидеть, что выполнение итерации метода сеток можно разбить на последовательность шагов, на каждом из которых к вычислениям окажутся подготовленными узлы вспомогательной диагонали сетки с номером, определяемым номером этапа (Рисунок 1).

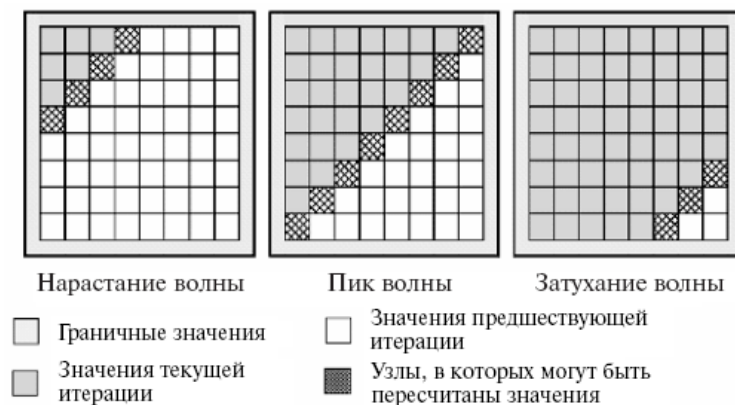


Рисунок 1. Волновая схема обработки данных

Такой алгоритм получил название волновой схемы. Следует отметить, что в нашем случае размер волны (степень возможного параллелизма) динамически изменяется в ходе вычислений — волна нарастает до своего пика, а затем затухает при приближении к правому нижнему узлу сетки.



## Результаты

### Тестовая инфраструктура

Вычислительные эксперименты проводились с использованием следующей инфраструктуры (Таблица 1).

Таблица 1. Тестовая инфраструктура

Процессор	4 ядра, Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz
Память (RAM, Cache)	8,00 ГБ, L1 – 256 Kb, L2 – 1 Mb, L3 – 8 Mb
Операционная система	Windows 10
Среда разработки	Visual Studio 2019
Компилятор	Intel(R) oneAPI DPC++/C++ Compiler 2021.1
Библиотеки	OpenMP

### Эксперименты

Испытания проводились на задаче со следующими данными (Рисунок 2,3):

- Размер пластины  $10 \times 10$
- Разбиения на сетку с числом узлов  $1000 \times 1000$ ,  $1500 \times 1500$ ,  $2000 \times 2000$
- Граничные условия
  - Левая граница:  $1 + y^2$
  - Левая граница:  $4 + y^2$
  - Левая граница:  $4 + x^2$
  - Левая граница:  $9 + x^2$
- Функция внешнего воздействия  $f(x, y) = 4$

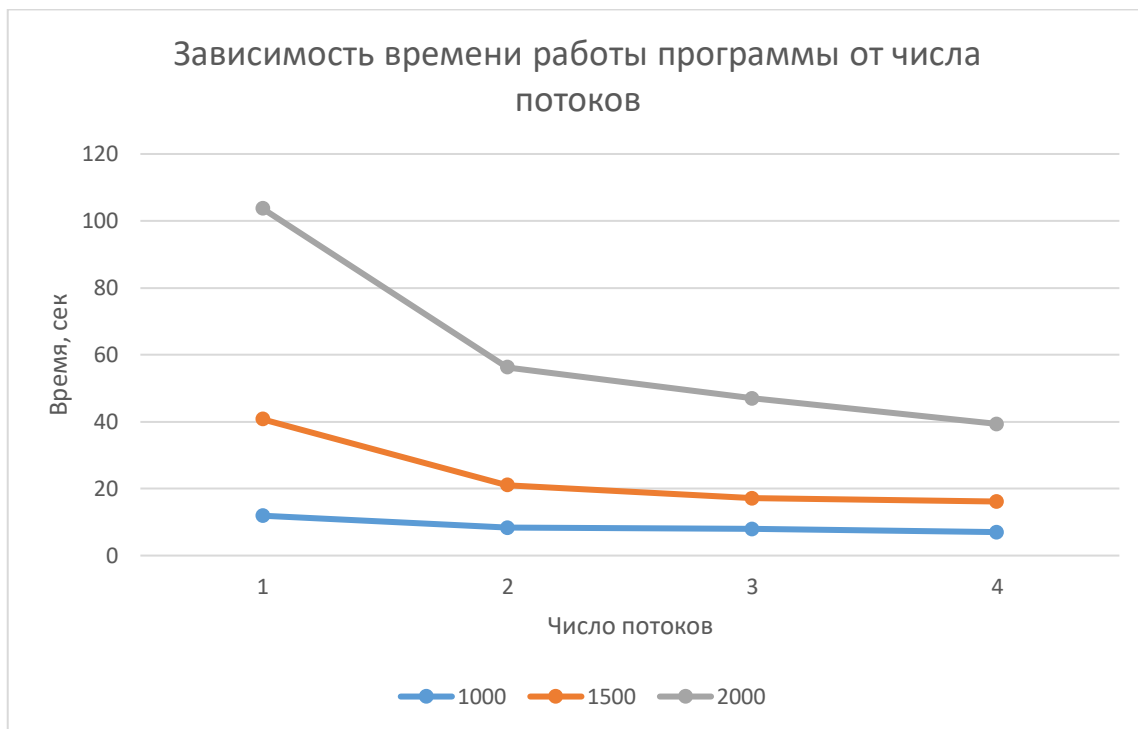


Рисунок 2. Зависимость времени работы программы от числа потоков

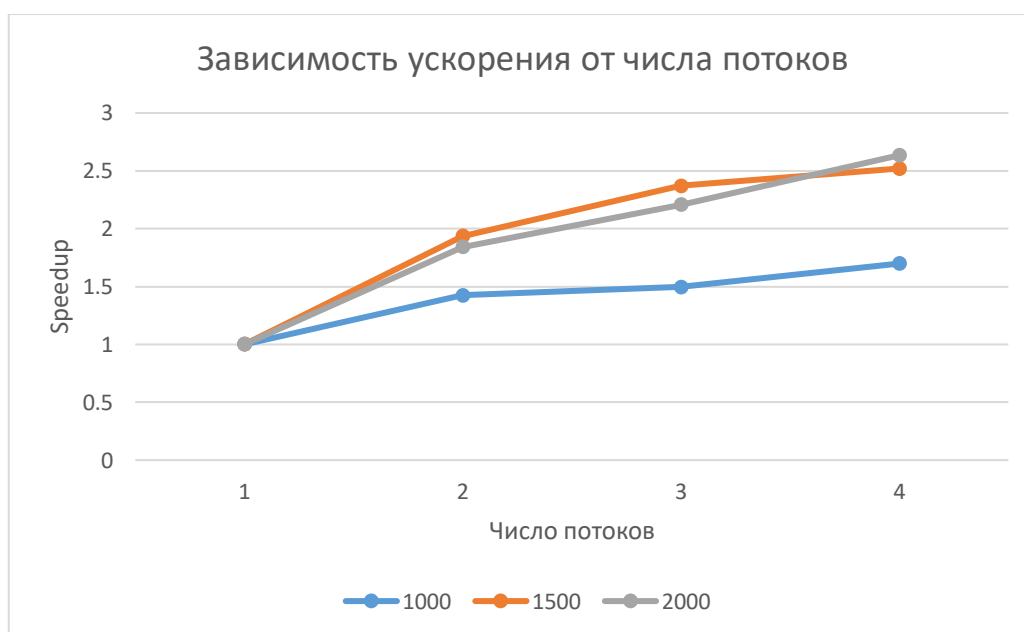


Рисунок 3. Зависимость ускорения от числа потоков

## Заключение

В ходе данной лабораторной работы был изучен метод релаксации с оптимальным параметром для задачи Дирихле для уравнения Пуассона с волновой схемой параллелизации вычислений.

Использование волновой схема обработки данных удалось достигнуть существенного ускорения алгоритма и избежать гонки данных, которая была бы неизбежна при попытке параллелизации последовательного обхода элементов матрицы.

## Литература

1. Самарский А.А. Введение численные методы. – СПб.: Лань, 2005
2. Самарский А.А., Гулин А.В. Численные методы. – М.: Наука, 1989.
3. Е.А. Рындин, И.В. Куликова, И.Е. Лысенко. Основы численных методов: теория и практика, 2015
4. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – М.: Наука, 1987
5. Тихонов А.Н., Самарский А.А. Уравнения математической физики. – М.: Наука, 1977.

