Московский Государственный Университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

**Отчет по второму заданию в рамках курса**

**«Суперкомпьютерное моделирование и технологии»**

**Выполнила:** Яковлева Маргарита Александровна

студентка 614 группы

Вариант 6

# Математическая постановка задачи

В области , ограниченной контуром γ, рассматривается дифференциальное уравнение Пуассона

, (1)

в котором оператор Лапласа

, q

функция .

Для выделения единственного решения уравнение дополняется граничными условием Дирихле:

. (2)

Требуется найти функцию , удовлетворяющую уравнению (1) в области *D* и краевому условию (2) на ее границе, где область *D* – квадрат с отсеченной вершиной :

# Численный метод решения поставленной задачи

## Метод фиктивных областей

Для решения поставленной задачи предлагается использовать метод

фиктивных областей. Пусть область *D* принадлежит прямоугольнику

.

Обозначим через , замыкание области *D* и прямоугольника Π соответственно, через Γ – границу прямоугольника. Разность множеств

.

В прямоугольнике П рассмотрим задачу Дирихле:

(3)

с кусочно-постоянным коэффициентом:

и правой частью:

Требуется найти непрерывную в функцию *v(x, y)*, удовлетворяющую дифференциальному уравнению всюду в , равную нулю на границе Г прямоугольника, и такую, чтобы вектор потока:

имел непрерывную нормальную компоненту на общей части криволинейной границы области *D* и прямоугольника П.

Переход к новой задаче позволяет получить решение исходной задачи

с любой наперед заданной точностью , решая при этом

задачу Дирихле в прямоугольнике П, содержащем исходную область.

Для *D* - квадрата с отсеченной вершиной выберем

## Разностная схема решения задачи

В замыкании прямоугольника определим равномерную прямоугольную сетку , где

Множество внутренних узлов сетки обозначим .

Рассмотрим линейное пространство *H* функций, заданных на сетке . Обозначим через значение сеточной функции *H* в узле сетки . Определим скалярное произведение и норму в пространстве сеточных функций *H*:

Будем использовать метод конечных разностей, который заключается в замене дифференциальной задачи математической физики на

конечно-разностную операторную задачу вида:

Дифференциальное уравнение задачи (3) во всех внутренних точках сетки аппроксимируется разностным уравнением:

в котором коэффициенты при

и правая часть

Краевые условия Дирихле в задаче (3) аппроксимируются точно равенством

Полученная система является линейной относительно неизвестных величин

и может быть представлена в виде c самосопряженным и положительно определенным оператором *A*. Построенная разностная схема линейна и имеет единственное решение при любой правой части.

Интегралы вычисляем аналитически: ,

где – длина части отрезка , которая принадлежит области *D* и вычисляется путем поиска точек пересечения прямой c границами *D.* Правую часть схемы приближенно заменяем на значение в центре квадрата : .

## Метод минимальных невязок

Приближенное решение разностной схемы предлагается вычислять методом

наименьших невязок.

Метод позволяет получить последовательность сеточных функций , сходящуюся по норме пространства Н к решению разностной схемы.

Начальное приближение выберем равным нулю во всех точках сетки. Итерация вычисляется по формуле: , где невязка , итерационный параметр .

В качестве критерия останова используется условие .

# Описание программной реализации

Для выполнения задания был разработан последовательный код, представляющий собой программу на языке Cи, реализующую описанный численный метод. Были выполнены расчеты на сгущающихся сетках

(M, N) = (10, 10), (20, 20), (40, 40) и построены графики

полученных приближенных решений. Для написания параллельной программы вложенные циклы в функциях, вызывающихся на каждой итерации метода минимальных

невязок, были размечены с помощью директивы OpenMP: pragma omp parallel for collapse(2).

Были проведены расчеты на сетках (40, 40), (80, 80), (160, 160) на разном

числе потоков. Полученные приближенные решения совпали с соответствующими решениями при последовательных вычислениях (на сетках (40, 40), (80, 80), и (160, 160)), но время их вычисления удалось уменьшить за счет использования параллелизма. Результаты вычислений приведены в таблице.

# Зависимости времени решений от числа нитей (компиляция с флагом О2)

|  |  |  |  |
| --- | --- | --- | --- |
| Число OpenMP-нитей | Число точек сетки M x N | Время решения | Ускорение |
| 2 | 40 x 40 | 1.659481 | 1.543403 |
| 4 | 40 x 40 | 1.189708 | 2.152838 |
| 6 | 40 x 40 | 1.120341 | 2.286133 |
| 8 | 40 x 40 | 1.057279 | 2.422490 |
| 16 | 40 x 40 | 1.094842 | 2.339377 |
| 2 | 80 x 80 | 43.785937 | 1.415766 |
| 4 | 80 x 80 | 19.366747 | 3.200882 |
| 6 | 80 x 80 | 14.629144 | 4.237478 |
| 8 | 80 x 80 | 12.495609 | 4.960996 |
| 16 | 80 x 80 | 11.153633 | 5.557890 |
| 2 | 160 x 160 | 86.547483 | 1.933513 |
| 4 | 160 x 160 | 45.297374 | 3.694268 |
| 6 | 160 x 160 | 31.794382 | 5.263214 |
| 8 | 160 x 160 | 25.095805 | 6.668072 |
| 16 | 160 x 160 | 16.825990 | 9.945367 |

## Графики зависимостей ускорений от числа потоков (компиляция с флагом О2)

Изображение выглядит как линия, График, диаграмма, скат

Автоматически созданное описание

# Зависимости времени решений от числа нитей (компиляция с флагом О3)

|  |  |  |  |
| --- | --- | --- | --- |
| Число OpenMP-нитей | Число точек сетки M x N | Время решения | Ускорение |
| 2 | 40 x 40 | 1.881945 | 1.051024 |
| 4 | 40 x 40 | 1.381757 | 1.431489 |
| 6 | 40 x 40 | 1.243437 | 1.590728 |
| 8 | 40 x 40 | 1.138108 | 1.737946 |
| 16 | 40 x 40 | 1.183627 | 1.671109 |
| 2 | 80 x 80 | 25.888349 | 1.896905 |
| 4 | 80 x 80 | 13.115685 | 3.744199 |
| 6 | 80 x 80 | 11.555275 | 4.249810 |
| 8 | 80 x 80 | 9.108469 | 5.391436 |
| 16 | 80 x 80 | 8.853903 | 5.546450 |
| 2 | 160 x 160 | 50.414782 | 1.902636 |
| 4 | 160 x 160 | 26.820982 | 3.576340 |
| 6 | 160 x 160 | 19.427241 | 4.937446 |
| 8 | 160 x 160 | 15.787640 | 6.075699 |
| 16 | 160 x 160 | 13.315621 | 7.203641 |

## Графики зависимостей ускорений от числа потоков (компиляция с флагом О2)

Изображение выглядит как линия, График, диаграмма, скат

Автоматически созданное описание

# График невязки в зависимости от числа итераций

Изображение выглядит как линия, диаграмма, График, снимок экрана

Автоматически созданное описание

# График ошибки в зависимости от числа итераций

Изображение выглядит как линия, График, диаграмма, скат

Автоматически созданное описание

# График L1 нормы невязки в зависимости от количества итераций

## Изображение выглядит как линия, График, диаграмма, скат Автоматически созданное описание

# График L2 нормы невязки в зависимости от количества итераций

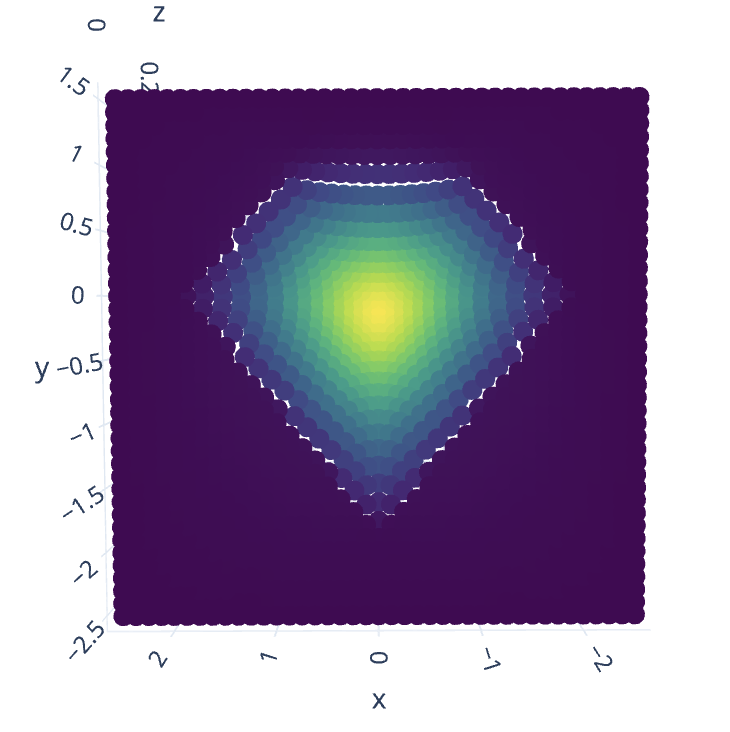
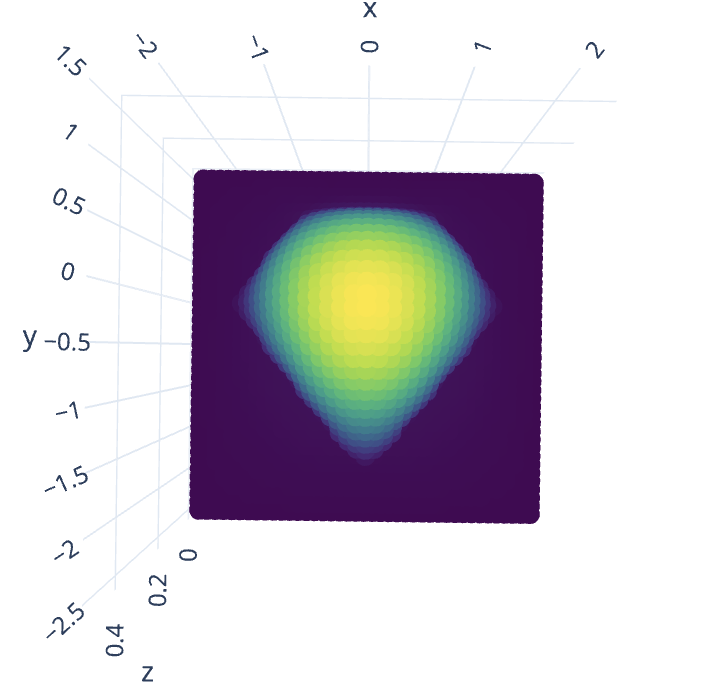
Изображение выглядит как линия, График, скат, диаграмма

Автоматически созданное описание

# Рисунки приближенного решения на сетке 40\*40

Изображение выглядит как рисунок, Графика, иллюстрация, дизайн

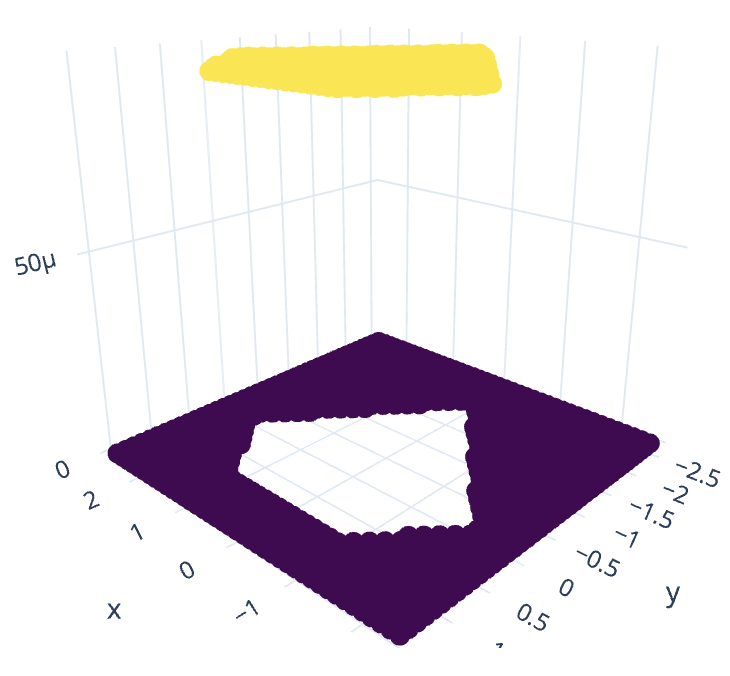
Автоматически созданное описаниеИзображение выглядит как снимок экрана, Графика, графический дизайн, дизайн

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Графика, графический дизайн

Автоматически созданное описаниеИзображение выглядит как снимок экрана, текст, дизайн

Автоматически созданное описание

## Промежуточные результаты сходимости на сетке 40\*40



## Промежуточные результаты сходимости на сетке 80\*80

Изображение выглядит как диаграмма, линия, Прямоугольник, дизайн

Автоматически созданное описание

## Промежуточные результаты сходимости на сетке 160\*160

Изображение выглядит как диаграмма, текст, линия, дизайн

Автоматически созданное описание