Бахышов Вахид, 409 группа

Отчёт по задаче "Численное двумерное интегрирование, триангуляция".

Численное интегрирование - 2D

Пусть T - треугольник на плоскости, $S(T)$ - его площадь, A, B, C - середины сторон. Несложно показать, что квадратурная формула

$$I(f) = \iint_T f(x)dx \approx S(f)\frac{1}{3}S(T)(f(A) + f(B) + f(C)),$$

где $x = (x_1, x_2)\,, dx = dx_1 dx_2$, точна для всех многочленов второй степени вида

$$a_0 + a_1x_1 + a_2x_2 + a_{11}x_1^2 + a_{12}x_1x_2 + a_{22}x_2^2$$

Задача 3. Для заданного прямоугольника со сторонами Lx, Ly (это шаг сетки по осям ох, оу) постройте его триангуляцию и результат сохраните в отдельном файле в формате:

```
    <число вершин> их (N + 1)(N + 1) = (N+1)^2
    <число треугольников> их N * N * 2
    <число внутренних ребер> их 0.5 * (N * N * 2 * 3 - 4 * N)
    <число граничных (внешних) ребер> их 4 * N
(для каждой вершины)
    <номер вершины>:<x y> (координаты вершины)
    . . .
(для каждого треугольника)
```

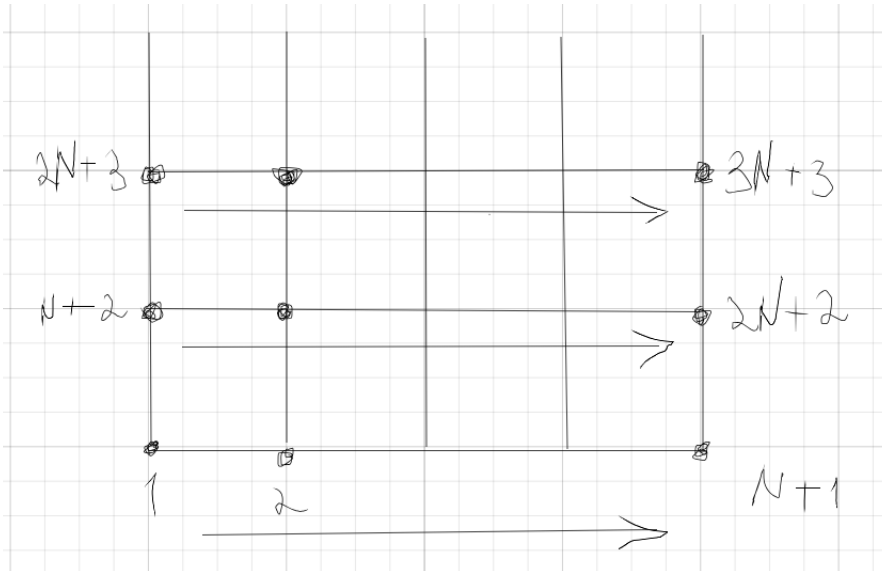

Рис. 1: Нумерация для каждой вершины, нумеруем так

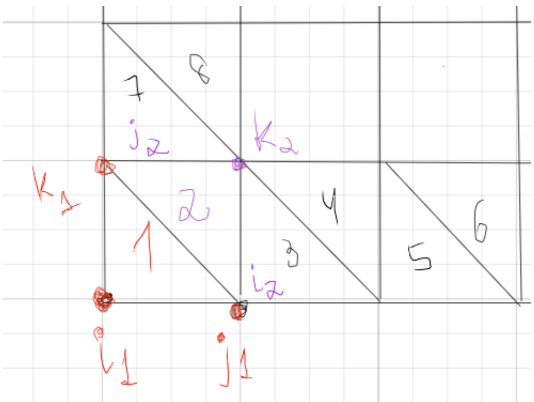

Рис. 2: i, j, k - номера вершин; идем по маленьким прямоугольникам и для каждого описываем два треугольника

```
    <номер треугольника>:<i j k> (номера вершин)
    . . .
(для каждого внутреннего ребра)
    <номер внутреннего ребра>:<m n> (номера вершин)
    . . .
(для каждого граничного ребра)
    <номер граничного ребра>:<m n> (номера вершин)
    . . .
```

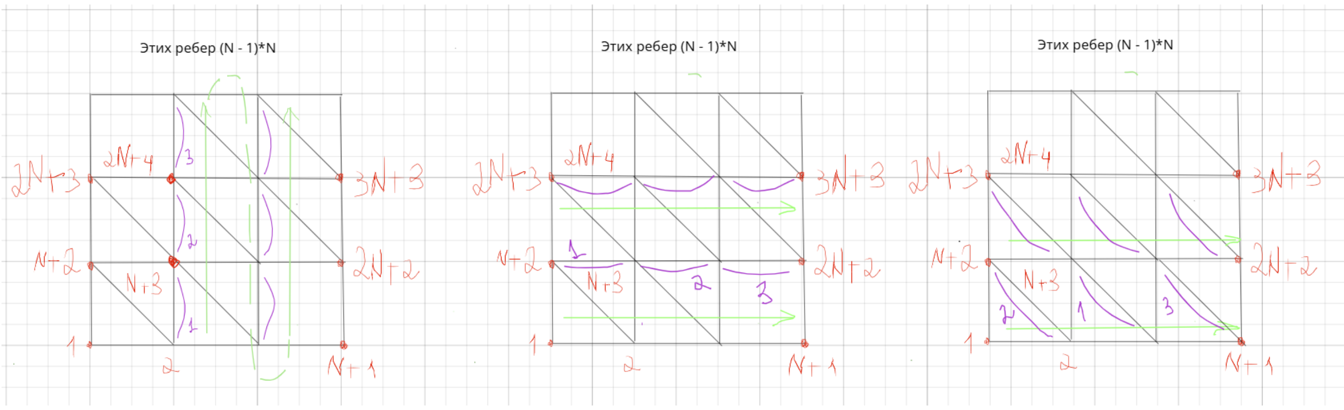

Рис. 3: для каждого внутреннего ребера m и n - номера вершин

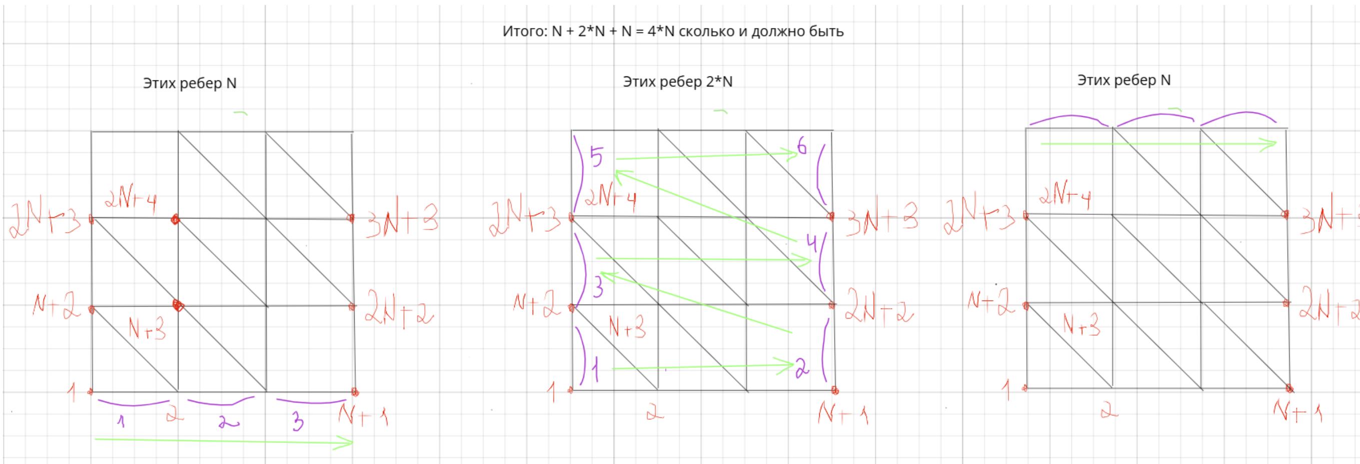

Рис. 4: для каждого граничного (внешнего) ребера m и n - номера вершин

Описание задачи. Этот код решает задачу численного интегрирования функции на прямоугольной области с разбиением на треугольники. Используются два подхода для вычисления интегралов: 1) Квадратура центра масс.
2) Квадратура из файла (с заранее заданными узлами и весами).

Основные шаги:
1) Разбиение области на треугольники (triangulation).
2) Численное вычисление интеграла по каждому треугольнику.
3. Сравнение полученного значения с истинным.

Указание. На первом шаге исходный прямоугольник делится на $Nx \times Ny$ равновеликих прямоугольников. На втором шаге каждый прямоугольник либо делится на два треугольника построением "северо-западной"/"северо-восточной"диагональю, либо делится на четыре треугольника проведением двух диагоналей. Полученный набор вершин и треугольников сохраняется в файл в указанном формате.

Описание функций:

1) **double trueValue(double xa, double xb, double ya, double yb, int k)** вычисляет точное значение интеграла для заданного полинома (или функции) k на прямоугольнике [xa, xb] × [ya, yb]; xa, xb — границы по x, ya, yb — границы по y, k — степень полинома.

2) **void triangulationFixed(double xa, double xb, double ya, double yb, int N)** функции разбивают прямоугольную область на треугольники и записывают результаты в файл

В цикле xa по строке. ya по столбцу, $(N+1)^2$ вершин (вершину нумеруешь для каждой вершины вычисляем координату);

двойной цикл по i и j проходит по сетке и разбивает каждый квадрат на два треугольника. vert left down, vert left up, vert right down, vert right up — индексы вершин текущих треугольников в нашем квадрате, координаты вершин, triangle index — счётчик треугольников.

3) **void getVertexCoordinates(double xa, double xb, double ya, double yb, int m, int N, double & xm, double & ym)** функция вычисляет координаты вершины треугольника по её индексу m в сетке. Индексы i и j находятся из m, после чего координаты вычисляются с учётом шага. m — индекс вершины, xm, ym — координаты вершины.

4) **double integrateAccordingToTheQuadratureOfTheCenterOfMass(dou (*f)(double, double, int), double xa, double xb, double ya, double yb, int k, int N, double *VertexCoords)** функция выполняет численное интегрирование функции с использованием квадратуры центра масс для каждого треугольника. res — накопленное значение интеграла, VertexCoords — массив для хранения координат вершин треугольников, xm, ym — координаты центра масс.

std::stringstream ss(line); n, v1, v2, v3 - номер треугольника и номер 1ой 2ой и 3ей вершине будут в данной строке line,

в треугольнике 3 вершины у каждой 2 координаты; центр по 2-ум координатам как средне-арифметическое 3 координат; 0.5 * hx * hy формула произведения площади треугольника, f(xm, ym, k) - это значение ф-ции треугольника в центре масс, формула вычисления: 0.5 * hx * hy формула произведения площади треугольника, f(xm, ym, k) - это значение ф-ции треугольника в центре масс

5) **void readingQuadratureFile(double *CoordsAndWeights)** Считывает координаты и веса для квадратурной формулы из файла и сохраняет их в массив. CoordsAndWeights — массив для хранения координат и весов, считываем числа в массив

6)**void searchAffineTransformation(double* VertexCoords, double *EquationCoefficients)** Вычисляет коэффициенты аффинного преобразования для перехода из эталонного треугольника в произвольный.

В файле этот треугольник хотим перевести в маленькие треугольники, вычисляем коэффициенты аффинного преобразования для перехода из эталонного треугольника в произвольный, VertexCoords — координаты вершин треугольника, EquationCoefficients — массив для коэффициентов аффинного преобразования.
Находим коэффициенты афинного преобразования $x' = ax + by + e$, $y' = cx + dy + f$, переводящее точки треугольника (0, 0), (1, 0), (0, 1) $x, \ y$ в точки i-ого треугольника триангуляции

7) **double integrateAccordingToTheQuadratureFromTheFile(double (*f)(double, double, int), double xa, double xb, double ya, double yb, int k, int N, double *VertexCoords, double *CoefOfTransform, double *CoordsAndWeights)** функция выполняет интегрирование на основе квадратурной формулы из файла.

В цикле для каждой точки из файла находим преобразование которое переволит точки большого треугольника в маленький, т.е координаты x, y

Если просуммировать 3-ий столбец, то получаем что суммарный вес треугольника = 0.5 (а у нас должен быть суммарный вес треугольника = 1), поэтому в res умножаем на 2 помимо умножения на res, и на площадь треугольника hx * hy * 0.5

8) **void writeToFileForP(double (*f)(double, double, int), double xa, double xb, double ya, double yb, int k, int N, int numTests, double *VertexCoords, double *CoefOfTransform, double *CoordsAndWeights, int typeOfQuadrature)** выполняет многократное численное интегрирование для разного числа делений N и записывает результаты (аппроксимация, точное значение, ошибка) в файл.

Задача. Используя файлы с указанной триангуляцией, построенные для $Lx = Ly = 1$ и различных $Nx = Ny = N$, численно найдите на примере задачи

$$I(f) = \int_{[0,1]\times[0,1]} \int_1 \left(x_1^4 + x_1^2 x_2^2 + x_2^4\right) dx$$

асимптотику $R_N^{[0,1]^2}(f) = \mid (I(f) - S_N(f) \mid \sim C/N^p$ для погрешности полученной составной квадратуры.

```
Convergence index for the centher of mass method p =  1.9999999999894682
```

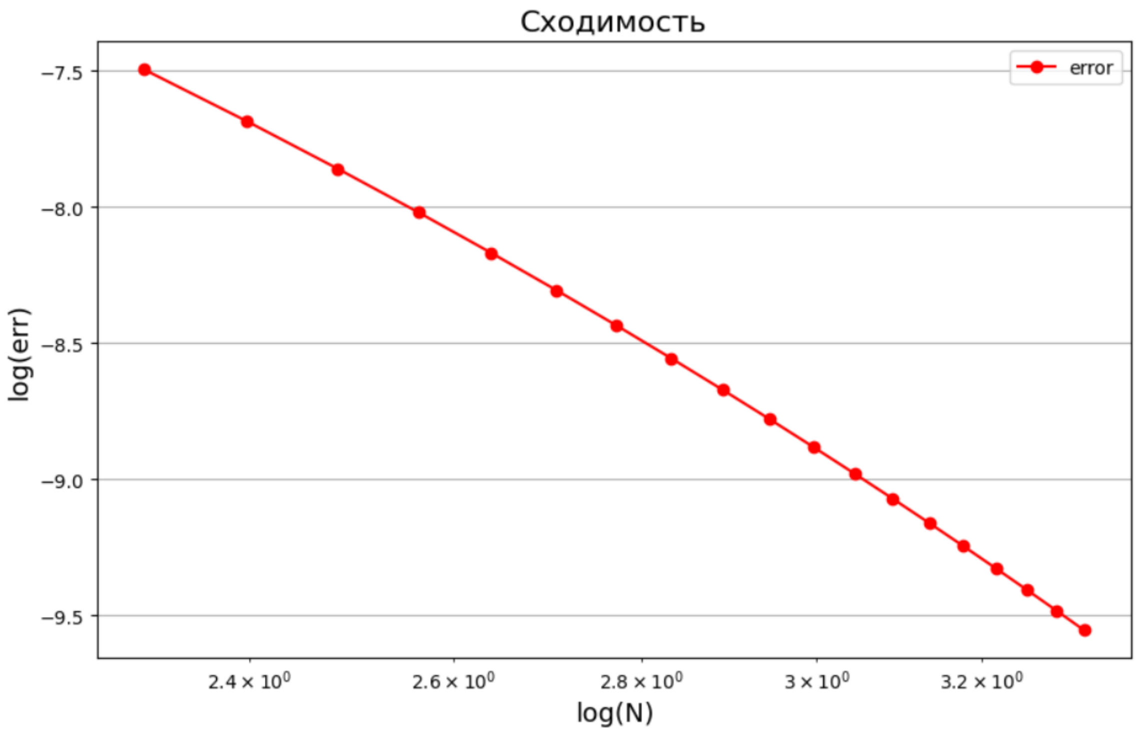

Рис. 5: N $=10$, $I(f)=\int_{[0,1]\times[0,1]}\int_1\left(x_1^4+x_1^2x_2^2+x_2^4\right)dx$

```
Convergence index for the centher of mass method p =  1.5042514941387628
```

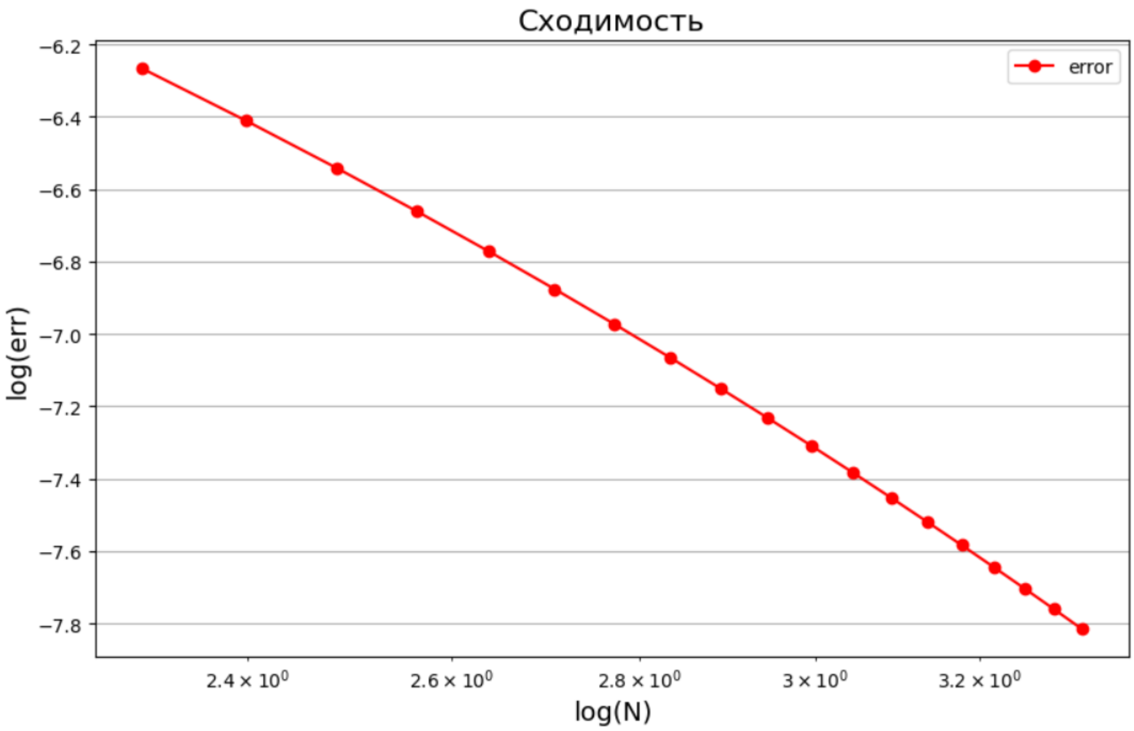

Рис. 6: N = 10, $I(f) = \int_{[0,1]\times[0,1]}\int_1 \sqrt{x*y}dx$