Отчёт

1 Постановка задачи

Задана область $[0,1]\times[0,1]$ и набор точек $\{(x_i, y_j)\}_{i=0,\dots,N}^{j=0,\dots,N}$, где

$$\{x_i\}_{i=0}^{N} = \left\{i * \frac{1}{N}\right\}_{i=0}^{N} \qquad \{y_i\}_{i=0}^{N} = \left\{(i * \frac{2}{2N-1} - \frac{1}{2N-1})\right\}_{i=0}^{N}$$

Задана функция $f : [0,1]\times[0,1] \to \mathbb{R}, f \in C^{\infty}([0,1]\times[0,1])$, удовлетворяющая краевым условиям:

$$\begin{cases} f'_x(0,y) = 0 \\ f'_x(1,y) = 0 \\ f(x,0) = 0 \\ f'_y(x,1) = 0 \end{cases}$$

Необходимо по значениям функции $f(x,y)$ в точках $\{(x_i, y_j)\}_{i=0,\dots,N}^{j=0,\dots,N}$ построить её разложение в дискретный ряд Фурье

$$F_N(x,y) = \sum_{m=0}^{N}\sum_{n=0}^{N-1} c_{mn}\cdot\phi_{mn}(x,y),$$

удовлетворяющий тем же граничным условиям, что и исходная функция, причём $\{\phi_{mn}(x,y)\}_{m=0,\dots,N}^{n=0,\dots,N}$ - полная и ортонормированная относительно скалярного произведения система тригонометрических функций на пространстве дискретных функций над $\{(x_i, y_j)\}_{i=0,\dots,N}^{j=0,\dots,N}$.

В точках $(x_i, y_0)_{i=0,\dots,N}$ доопределяем функцию как $f(x_i, y_0) = -f(x_i, y_1)$. Скалярное произведение выбирается таким образом, чтобы

$$\langle f, g\rangle \xrightarrow{N\to\infty} \int\limits_{[0,1]\times[0,1]} f(x,y)g(x,y)dxdy$$

Затем нужно исследовать изменение погрешности интерполяции функции построенным дискретным рядом Фурье при росте числа узлов интерполяции.

2 Алгоритм решения

2.1 Одномерная задача

Пусть даны узлы $\{x_i\}_{i=0,\dots,N}$, значения удовлетворяющей заданным краевым условиям функции $f \in F$ в этих узлах и полная ортогональная относительно скалярного произведения $\langle\ ,\ \rangle$ система дискретных функций $\{\phi_i\}_{i=0,\dots,N}$. Требуется найти коэффициенты c_i, при которых:

$$f(x) = \sum_{i=0}^{N} c_i \phi_i(x),$$

Скалярно домножаем обе части уравнения на ϕ_i. Из условия ортогональности следует, что в правой части остаётся только одно ненулевое слагаемое и $\langle f, \phi_i\rangle = c_i \cdot \langle \phi_i, \phi_i\rangle$. Отсюда:

$$c_i = \frac{\langle f, \phi_i\rangle}{\langle \phi_i, \phi_i\rangle}$$

Вычислительная сложность алгоритма вычисления коэффициентов в одномерной задаче - $O(N^2)$: N коэффициентов, на каждый из которых нужно два скалярных произведения.

2.2 Двумерная задача

Пусть даны узлы $\{(x_i, y_j)\}_{i=0,\dots,N}^{j=0,\dots,N} \subseteq X \times Y$, значения функции f в этих узлах (f удовлетворяет краевым условиям). Пусть $\{\phi_i\}$ и $\{\psi_j\}$ - системы ортогональных функций на X и Y соответственно.
Необходимо найти коэффициенты c_{mn}, удовлетворяющие:

$$f(x_i, y_j) = \sum_{m=0}^{N} \sum_{n=0}^{N-1} c_{mn} \cdot \phi_m(x_i)\psi_n(y_j)$$

Обозначим:

$$d_n(x_i) = \sum_{m=0}^{N} c_{mn} \cdot \phi_m(x_i) \tag{1}$$

Тогда:

$$f(x_i, y_j) = \sum_{n=0}^{N-1} d_n(x_i) \cdot \psi_n(y_j) \tag{2}$$

На данном этапе получаем N одномерных задач при фиксированных x_i. Решая эти одномерные задачи, находим матрицу коэффициентов $(d_n(x_i))_{i=0,\dots,N}^{n=0,\dots,N}$. Каждая из задач требует $O(N^2)$ операций, следовательно, вычисление матрицы требует

$O(N^3)$. Возвращаясь к определению функций $d_n(x)$, при фиксированных n получаем задачи на восстановление c_{mn}, каждая из которых требует $O(N^2)$ операций, следовательно, вычисление всех c_{mn} требует $O(N^3)$.

Общая сложность алгоритма: $O(N^3) + O(N^3) = O(N^3)$.

3 Решение задачи

При данных краевых условиях, необходимо взять следующие системы собственных функций в качестве базисных:

$$\{\cos(\pi n x)\}_{n=0,\dots,N} \quad \{\sin(\pi(m+\frac{1}{2})y)\}_{m=0,\dots,N-1}$$

Из курса анализа известно, что первая система ортогональна и полна на множестве дискретных функций над $\{x_i\}_{i=0}^N = \left\{\frac{1}{N}\right\}_{i=0}^N$, удовлетворяющих краевым условиям:

$$\begin{cases} f'_x(0) = 0 \\ f'_x(1) = 0 \end{cases}$$

со скалярным произведением, заданным как:

$$\langle f, g \rangle = f(x_0)g(x_0) \cdot \frac{1}{2N} + \sum_{i=1}^{N-1} f(x_i)g(x_i) \cdot \frac{1}{N} + f(x_N)g(x_N) \cdot \frac{1}{2N}$$

Вторая система ортогональна и полна на множестве дискретных функций над точками $\{y_j\}_{j=0}^N = \left\{(j * \frac{2}{2N-1} - \frac{1}{2N-1})\right\}_{j=0}^N$, удовлетворяющих краевым условиям:

$$\begin{cases} f(0) = 0 \\ f'_y(1) = 0 \end{cases}$$

со скалярным произведением, заданным как:

$$\langle f, g \rangle = \sum_{i=0}^{N-1} f(x_i)g(x_i) \cdot \frac{1}{N} + f(x_N)g(x_N) \cdot \frac{1}{2N}$$

Следовательно, мы попадаем в ситуацию, описанную в пункте 2.2 и можем решить задачу.

4 Описание работы программы

При запуске программы первым из входного файла **input.txt** считывается значение **n** - количество отрезков разбиения. Затем вызывается функция

void uniform_partition(FILE* in_part, const int n);

Она заполняет файл **input_part.txt** значениями целевой функции в узлах разбиения. Целевая функция задаётся функцией

double func(double x, double y);

Так как узлы разбиения по оси Oy слева сдвинуты, а краевое условие с этой стороны - $f(x, 0) = 0$, то значение функции в 0-й точке (вне отрезка $[0, 1]$) определяем как $f(x_i, y_0) = -f(x_i, y_1)$.

Далее, точки из файла считываются в массив f и вызывается функция

void f2c(double* C, double* f, double* d, const int n);

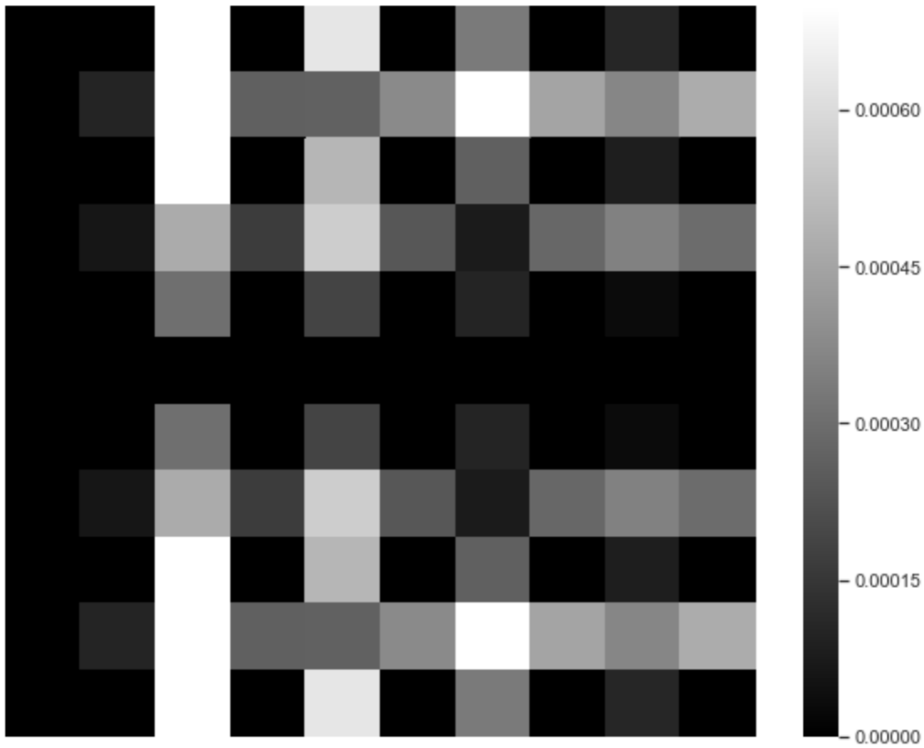
которая работает по алгоритму, описанному в разделах 2 и 3. Результатом её работы является заполненный двумерный массив коэффициентов c_{mn}. Вычислительная сложность выполнения функции - $O(N^3)$ арифметических операций. Найденные коэффициенты записываются в файл **output_coef.txt**.

Теперь надо собрать функцию обратно как сумму базисных функций с найденными коэффициентами. Для этого используются функции

void c2d(double *C, double *d, const double x0, const int n);
double d2f(double *d, const double y0, const int n);

Первая собирает коэффициенты d_n для данного x по формуле (1), вторая из dn для данного y собирает f по формуле (2). Полученные значения записываются в файл **output.txt**. Временная сложность выполнения функции - $O(N^3)$ арифметических операций. Далее считаем модуль разности значений, получившихся как сумма ряда, и реальных значений функции в узлах интерполяции и между ними. Заносим их в файл **output_deviation.txt**.

В узлах должен получиться 0 с точностью до машинной точности, между узлами может быть незначительное отклонение. Для наглядности построим тепловую карту по таблице из файла **output_deviation.txt**:



Как видно - все узлы чёрные, значит всё в порядке.