Отчёт по Интегрированию

1 Квадратурная Формула Ньютона-Котеса

1.1 Постановка задачи и теоретическое обоснование

Пусть $-\infty < a < b < \infty$, и на отрезке $[a, b]$ задана функция $f : [a, b] \to \mathbb{R}$, т.ч. $f \in \mathbb{C}^{(n+1)}$. Требуется решить задачу приближённого численного интегрирования, а именно построить квадратурную формулу Ньютона-Котеса $S_n(f)$, которая по значениям функции в равноотстоящих узлах $\{x_i = a + i(b-a)\}_{i=0}^n$ восстанавливает значение интеграла, т.е.

$$S_n(f) = \sum_{i=0}^{n} c_i f(x_i) \approx \int_a^b p(x) f(x) dx = I(f)$$

Весовую функцию $p(x)$ положим равной единице. Функцию $f(x)$ можно приблизить с помощью интерполяционного многочлена Лагранжа, тогда:

$$\int_a^b f(x)dx \approx \int_a^b \sum_{i=0}^{n} f(x_i) \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)} dx = \sum_{i=0}^{n} f(x_i) \int_a^b \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)} dx$$

Положим

$$c_i = \int_a^b \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)} dx \tag{1}$$

Оценка погрешности квадратурной формулы в этой ситуации имеет вид:

$$R_n = |I(f) - S_n(f)| \leq \frac{\|f^{(n+1)}\|}{(n+1)!} \cdot \int_a^b |\omega_{n+1}(x)| dx$$

Тогда для n = 3:

$$R_3(f) = |I(f) - S_3(f)| \leq \|f^{(4)}\| \frac{(b-a)^5}{480}$$

Разобьём отрезок $[a, b]$ на $\{[a_k, b_k]\}$, т.ч. $a_1 = a,\ b_n = b,\ b_{k-1} = a_k, k = \overline{2, n-1}$.

$|b_k - a_k| = \frac{b-a}{n}$. Тогда:

$$I(f) = \int_a^b f(x)dx = \sum_{k=1}^{n} \int_{a_k}^{b_k} f(x)dx = \sum_{k=1}^{n} I_k(f)$$

Норму ошибки можно переписать в виде:

$$|I(f) - S(f)| = |\sum_{k=1}^{n}(I_k(f) - S_k(f))| \leq \sum_{k=1}^{n} |I_k(f) - S_k(f)| \leq \sum_{k=1}^{n} \|f^{(4)}\| \frac{(b_k - a_k)^5}{480} =$$
$$= \sum_{k=1}^{n} \|f^{(4)}\| \frac{((b-a)/n)^5}{480} = \frac{\|f^{(4)}\|(b-a)^5}{480 \cdot n^4}$$

Значит идея построения составной квадратуры, то есть разбиения отрезка на части и построения на каждой своей квадратуры, будет вести к сходимости общей квадратурной формулы к точному значению интеграла.

1.2 Алгоритм решения

Выполняем разбиение отрезка $[a, b]$ на n подотрезков, как было описано в предыдущем пункте. Для каждого из них строим квадратурную формулу по четырём равноотстоящим узлам:

$$S_k(f) = \sum_{i=0}^{3} c_{ik} f(x_{ik}), \text{ где } c_{ik} = \int_{a_k}^{b_k} \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)} dx, \ i = 0, 1, 2, 3.$$

Эти интегралы можно в явном виде посчитать, откуда:

$$c_{0k} = \frac{b_k - a_k}{8}, \quad c_{1k} = \frac{3(b_k - a_k)}{8}, \quad c_{2k} = \frac{3(b_k - a_k)}{8}, \quad c_{3k} = \frac{b_k - a_k}{8}$$

Далее просто складываем значения и при некотором n получаем необходимую степень приближения.

$$S(f) = \sum_{k=1}^{n} S_k(f)$$

1.3 Описание работы программы

При запуске программы первым из входного файла **input.txt** считывается значение **n** - количество отрезков разбиения и границы отрезка **a** и **b** (**a** < **b**). Затем вызывается функция

double integration_func(const int n, const double a, const double b);

Она производит все операции, описанные в предыдущем пункте. Далее значение, возвращённое этой функцией, выводится на экран.

2 Квадратура Гаусса

2.1 Постановка задачи и теоретическое обоснование

Пусть даны функция $f \in \mathbb{C}^{(4)}$ и отрезок $[a, b]$. Требуется построить квадратуру для поиска $\int\limits_a^b f(x)dx$, точную для многочленов максимально возможной степени. (Весовую функцию в данной ситуации полагаем равной единице).

Теорема 1. *Пусть ψ_n - ортогональный многочлен степени n на отрезке $[a, b]$ с весовой функцией $p(x) \equiv 1$. Пусть $a < x_0 < \cdots < x_{n-1} < b$ - его корни и $c_0, \ldots, c_{n-1}$ - коэффициенты квадратурной формулы S_n, т.ч. $I(P_{n-1}) = S_n(P_{n-1}) \ \forall \ P_{n-1}$. Тогда $I(P_{2n-1}) = S_n(P_{2n-1}) \ \forall \ P_{2n-1}$. Причём более точную квадратуру построить нельзя.*

Теорема 2. *Для квадратуры Гаусса, точной для многочленов степени $2n-1$, верна следующая оценка погрешности:*

$$R_n = |I(f) - S_n(f)| \leq \frac{\|f^{(2n)}\|}{(2n)!} \cdot 2^{1-2n} \cdot \left(\frac{b-a}{2}\right)^{2n} \cdot \left(\int\limits_a^b |p|dx + \sum_{i=0}^{n} |c_i|\right) \qquad (2)$$

Ортогональными многочленами на [-1, 1] с весом 1 являются многочлены Лежандра. Перенося их на отрезок $[a, b]$ с помощью формулы замены переменной получаем ортогональный многочлен на $[a, b]$. Далее находим его корни, узлы построения нашей квадратуры, и из условий точности квадратуры на многочленах можно решив СЛУ найти коэффициенты c_i.

2.2 Алгоритм решения

Выполняем разбиение отрезка $[a, b]$ на n подотрезков аналогично первой задаче. Для каждого из них строим Квадратуру Гаусса в четырёх точках и суммируем результаты. Для начала, многочлен Лежандра 4-й степени на $[-1, 1]$:

$$P_4(y) = \frac{1}{8}(35y^4 - 30y^2 + 3)$$

Его корни на $[-1, 1]$:

$$y_0 = -\sqrt{\frac{3}{7} + \frac{2\sqrt{6/5}}{7}} = -y_3, \ y_1 = -\sqrt{\frac{3}{7} - \frac{2\sqrt{6/5}}{7}} = -y_2$$

Используя формулу $x_{ik} = \frac{b_k - a_k}{2} \cdot y_i + \frac{b_k + a_k}{2}$ можно перенести эти корни на любой из подотрезков. Коэффициенты c_i симметричны, т.е. $c_i = c_{3-i}$, поэтому составлять систему на них можно из двух уравнений:

$$\begin{cases} c_{0k} + c_{1k} + c_{2k} + c_{3k} = 2c_{0k} + 2c_{1k} = \int\limits_{a_k}^{b_k} x^0 dx = b_k - a_k \\ c_{0k}x_{0k}^2 + c_{1k}x_{1k}^2 + c_{2k}x_{2k}^2 + c_{3k}x_{3k}^2 = c_{0k}(x_{0k}^2 + x_{3k}^2) + c_{1k}(x_{1k}^2 + x_{2k}^2) = \int\limits_{a_k}^{b_k} x^2 dx = \frac{b_k^3 - a_k^3}{3} \end{cases}$$

Отсюда

$$c_{1k} = \frac{b_k - a_k}{2(x_{1k}^2 + x_{2k}^2 - x_{0k}^2 - x_{3k}^2)} \cdot \left(\frac{2(b_k^2 + a_k b_k + a_k^2)}{3} - (x_{0k}^2 + x_{3k}^2) \right), \; c_{0k} = \frac{b_k - a_k}{2} - c_{1k}$$

$$c_{2k} = c_{1k}, \quad c_{3k} = c_{0k}$$

И теперь можем собрать все квадратуры:

$$S_k(f) = c_{0k}f(x_{0k}) + c_{1k}f(x_{1k}) + c_{2k}f(x_{2k}) + c_{3k}f(x_{3k}), \quad S(f) = \sum_{k=1}^{n} S_k(f)$$

2.3 Описание работы программы

При запуске программы первым из входного файла **input.txt** считывается значение **n** - количество отрезков разбиения и границы отрезка **a** и **b** (**a** < **b**). Затем вызывается функция

double integration_func(const int n, const double a, const double b);

Она производит все операции, описанные в предыдущем пункте. Далее значение, возвращённое этой функцией, выводится на экран.

3 Двумерная задача

3.1 Постановка задачи и алгоритм решения

Пусть на квадрате $[0,1]\times[0,1]$ задана функция $f : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$. Будем интегрировать её по этому квадрату. Для этого разбиваем квадрат на тругольники и считаем интеграл по приведённому ниже алгоритму.

Равномерно разбиваем осевые отрезки на **n** подотрезков длиной $\frac{1}{n}$ каждый. Проводя через каждую из точек линии, параллельные второй оси, получаем n^2 квадратов размера $1/n \times 1/n$. Проводя во всех квадратах главные диагонали, получаем триангуляцию квадрата $[0,1]\times[0,1]$, содержащую $2n^2$ треугольников, $3n^2+2n$ рёбер, $(n+1)^2$ вершин.

Чтобы вычислить значение интеграла, необходимо проинтегрировать функцию по всем треугольникам, где интеграл по треугольнику аппроксимируется по формуле:

$$\iint\limits_{\triangle} f(x,y)dxdy \approx \frac{f(M_1)+f(M_2)+f(M_3)}{3} \cdot |\triangle|$$

где M_1, M_2, M_3 - середины рёбер треугольника. Из свойства аддитивности интеграла имеем, что интеграл по всему квадрату находится как сумма интегралов по всем треугольникам:

$$\iint\limits_{[0,1]^2} f(x,y)dxdy \approx \sum_{\triangle} \iint\limits_{\triangle} f(x,y)dxdy$$

3.2 Описание работы программы

При запуске программы в первую очередь с клавиатуры считывается значение **n** - количество отрезков разбиения. Далее запускается функция

void file_fill(FILE* out_triangles, const int n);

которая записывает в файл **output.txt** номера всех вершин и их координаты, номера всех треугольников с номерами вершин, которые им принадлежат и номера всех рёбер с номерами их вершин.

Далее вершины и треугольники считываются и записываются в массивы **nodes** и **triangles**. Запускается функция

double integration_func(const int n, double* nodes, int* triangles);

которая считает интегралы по треугольникам, складывает их и возвращает интеграл по квадрату как сумму интегралов по треугольникам. Полученное значение выводится на экран.