Московский государственный технический университет

им. Н.Э. Баумана

Факультет: «Информатика и системы управления»

Кафедра: «Программное обеспечение ЭВМ и информационные технологии»

Расчётно-пояснительная записка

к курсовому проекту по моделированию на тему

«Алгоритм и программная реализация метода последовательного интегрирования при вычислении двойных и тройных интегралов»

Студент:

Группа ИУ7-88

Преподаватель: Градов В. М.

Москва

2019

Содержание

1. Постановка задачи

Реализовать на программном уровне алгоритм последовательного интегрирования при вычислении кратных интегралов: двойных и тройных.

В качестве метода численного интегрирования выбран метод Симпсона.

2. Метод Симпсона для кратных интегралов

2.1 Метод Симпсона для K-мерного интеграла

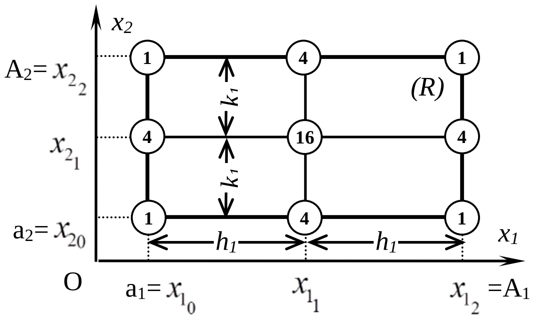

Рис. 1

Пусть сначала область интегрирования есть K-мерный пространственный параллелепипед $R\{a_1 \le x_1 \le A_1; a_2 \le x_2 \le A_2; \ldots; a_k \le x_k \le A_k;\}$ (рис. 5), стороны которого параллельны осям координат. Каждый из промежутков $[a_i, A_i]$ $(i = \overline{1,k})$ разобьём пополам точками:

$$x_{i_0} = a_i,\ x_{i_1} = a_i + h_i,\ \mathrm{x}_{\mathrm{i}_2} = \mathrm{a_i} + 2\mathrm{h_i} = \mathrm{A_i} \quad (i = \overline{1,k}),$$

где $h_i = \dfrac{A_i - a_i}{2} \quad (i = \overline{1,k})$.

Всего таким образом, получим 3^k точек сетки.

Имеем:

$$\underbrace{\int \ldots \int}_{k \text{ раз}} f(x_1, x_2, \ldots, x_k) dx_1 \ldots dx_k = \underbrace{\int_{a_1}^{A_1} dx_1 \int_{a_2}^{A_2} dx_2 \ldots \int_{a_k}^{A_k}}_{k \text{ раз}} f(x_1, x_2, \ldots, x_k) dx_k$$

Находим K-мерный интеграл, вычисляя каждый внутренний интеграл по формуле Симпсона на соответствующем отрезке. Проведём полностью все вычисления для случая K=2:

$$\iint f(x_1,x_2)dx_1dx_2 = \int\limits_{a_1}^{A_1} dx_1 \cdot \frac{h_2}{3}\left[f(x_1,x_{2_0}) + 4f(x_1,x_{2_1}) + f(x_1,x_{2_2})\right] =$$

$$= \frac{h_2}{3}\left[\int\limits_{a_1}^{A_1} f(x_1,x_{2_0})dx_1 + 4\int\limits_{a_1}^{A_1} f(x_1,x_{2_1})dx_1 + \int\limits_{a_1}^{A_1} f(x_1,x_{2_2})dx_1\right].$$

Применяя к каждому интегралу снова формулу Симпсона, получим:

$$\iint f(x_1,x_2)dx_1dx_2 = \frac{h_1h_2}{9}\Big\{\left[f(x_{1_0},x_{2_0}) + 4f(x_{1_1},x_{2_0}) + f(x_{1_2},x_{2_0})\right] +$$
$$+ 4\left[f(x_{1_0},x_{2_1}) + 4f(x_{1_1},x_{2_1}) + f(x_{1_2},x_{2_1})\right] + \left[f(x_{1_0},x_{2_2}) + 4f(x_{1_1},x_{2_2}) + f(x_{1_2},x_{2_2})\right]\Big\}$$

или

$$\iint f(x_1,x_2)dx_1dx_2 = \frac{h_1h_2}{9}\Big\{\left[f(x_{1_0},x_{2_0}) + f(x_{1_2},x_{2_0}) + f(x_{1_0},x_{2_2}) + f(x_{1_2},x_{2_2})\right] +$$
$$+ 4\left[f(x_{1_1},x_{2_0}) + f(x_{1_0},x_{2_1}) + f(x_{1_2},x_{2_1}) + f(x_{1_1},x_{2_2})\right] + 16f(x_{1_1},x_{2_1})\Big\}.$$

2.2 Метод Симпсона для двойного интеграла

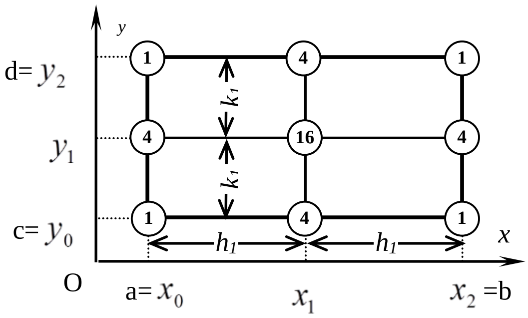

Рис. 2

Рассмотрим случай K=2, то есть двумерный интеграл. Предположим, что стороны прямоугольника R мы разделили соответственно на n равных частей (каждая часть, как на рис. 2); в результате получилась относительно крупная сеть прямоугольников (на рис. 3 вершины этих прямоугольников отмечены более крупными кружками).

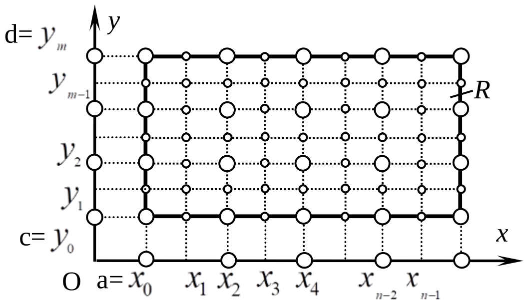

Рис. 3

Каждый из этих прямоугольников в свою очередь разделим на четыре равные части.

Пусть $h_1 = \dfrac{b-a}{2n}$ и $h_2 = \dfrac{d-c}{2n}$. Тогда сеть узлов будет иметь следующие координаты:

$$x_i = x_0 + ih_1 \quad (x_0 = a\ ;\ \ i = \overline{0,2n}\ )$$

$$y_j = y_0 + jh_2 \quad (y_0 = c;\ \ j = \overline{0,2n}).$$

Применяя формулу Симпсона для K-мерного интеграла, получим:

$$\iint f(x,y)dxdy = \frac{h_1 h_2}{9}\Big\{\big[f(x_0,y_0) + f(x_2,y_0) + f(x_0,y_2) + f(x_2,y_2)\big] +$$

$$+ 4\big[f(x_1,y_0) + f(x_0,y_1) + f(x_2,y_1) + f(x_1,y_2)\big] + 16 f(x_1,y_1)\Big\}.$$

Или

$$\iint\limits_{(R)} f(x,y)dxdy = \frac{h_1 h_2}{9}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}[(f_{2i,2j} + f_{2i+2,2j} + f_{2i+2,2j+2} + f_{2i,2j+2}) +$$

$$+ 4(f_{2i+1,2j} + f_{2i+2,2j+1} + f_{2i+1,2j+2} + f_{2i,2j+1}) + 16 f_{2i+1,2j+1}].$$

2.3 Метод Симпсона для тройного интеграла

По аналогии для интеграла, где K=2, найдем уравнение для тройного интеграла, где K=3. Тогда можно говорить про разбиение объемной фигуры на сеть более мелких кубов.

Пусть $h_1 = \frac{b-a}{2n}$, $h_2 = \frac{d-c}{2n}$ и $h_3 = \frac{f-e}{2n}$. Тогда сеть узлов будет иметь следующие координаты:

$$x_i = x_0 + ih_1 \quad (x_0 = a\ ;\ \ i = \overline{0,2n}\ )$$

$$y_j = y_0 + jh_2 \quad (y_0 = c;\ \ j = \overline{0,2n})$$

$$z_k = z_0 + kh_3 \quad (z_0 = e\ ;\ \ k = \overline{0,2n}\ ).$$

Применяя формулу Симпсона для K-мерного интеграла получим:

$$\iiint f(x,y,z)dxdydz =$$

$$= \frac{h_1 h_2 h_3}{27}\left\{\begin{bmatrix} f(x_0,y_0,z_0) + f(x_2,y_0,z_0) + f(x_0,y_2,z_0) + \\ + f(x_2,y_2,z_0) + f(x_0,y_0,z_2) + f(x_2,y_0,z_2) + \\ + f(x_0,y_2,z_2) + f(x_2,y_2,z_2) \end{bmatrix} + \right.$$

$$+4\begin{bmatrix} f(x_0,y_1,z_2) + f(x_2,y_1,z_2) + f(x_1,y_0,z_0) + \\ + f(x_0,y_1,z_0) + f(x_2,y_1,z_0) + f(x_1,y_2,z_0) + \\ + f(x_0,y_0,z_1) + f(x_2,y_0,z_1) + f(x_0,y_2,z_1) + \\ + f(x_2,y_2,z_1) + f(x_1,y_0,z_2) + f(x_1,y_2,z_2) \end{bmatrix} +$$

$$\left. +16\begin{bmatrix} f(x_1,y_1,z_2) + f(x_1,y_1,z_0) + f(x_1,y_0,z_1) + \\ + f(x_0,y_1,z_1) + f(x_2,y_1,z_1) + f(x_1,y_2,z_1) \end{bmatrix} + 64 f(x_1,y_1,z_1) \right\}.$$

Или

$$\iiint f(x,y,z)dxdydz =$$

$$= \frac{h_1 h_2 h_3}{27} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \sum_{k=0}^{p-1} \{ (f_{2i,2j,2k} + f_{2i+2,2j,2k} + f_{2i,2j+2,2k} + f_{2i+2,2j+2,2k} +$$

$$+ f_{2i,2j,2k+2} + f_{2i+2,2j,2k+2} + f_{2i,2j+2,2k+2} + f_{2i+2,2j+2,2k+2}) +$$

$$+ 4(f_{2i,2j+1,2k+2} + f_{2i+2,2j+1,2k+2} + f_{2i+1,2j,2k} + f_{2i,2j+1,2k} + f_{2i+2,2j+1,2k} +$$

$$+ f_{2i+1,2j+2,2k} + f_{2i,2j,2k+1} + f_{2i+2,2j,2k+1} + f_{2i,2j+2,2k+1} + f_{2i+2,2j+2,2k+1} +$$

$$+ f_{2i+1,2j,2k+2} + f_{2i+1,2j+2,2k+2}) + 16(f_{2i+1,2j+1,2k+2} + f_{2i+1,2j+1,2k} +$$

$$+ f_{2i+1,2j,2k+1} + f_{2i,2j+1,2k+1} + f_{2i+2,2j+1,2k+1} + f_{2i+1,2j+2,2k+1}) +$$

$$+ 64 f_{2i+1,2j+1,2k+1}.$$

3. Программная реализация метода

В качестве языка программирования был выбран языка Kotlin.

3.1 Реализация метода интегрирования двойного интеграла

```
fun simpsonDouble(
    function: FunctionDouble,
    a: Double,
    b: Double,
    c: Double,
    d: Double,
    n: Int
): Double {
    var sum = 0.0           //значение интеграла после всех сложений

    if((n%2 != 0) && n > 0) {
        throw IllegalArgumentException("Argument {n} must be even and more
than 0")
    }

    val h1=(b-a)/(2*n)
    val h2=(d-c)/(2*n)

    for (i in 0 until n){
        for (j in 0 until n) {

            sum += ((function.getFunction(a+2*i*h1,c+2*j*h2) +
                    function.getFunction(a+2*(i+1)*h1,c+2*j*h2) +
                    function.getFunction(a+2*i*h1,c+2*(j+1)*h2) +
                    function.getFunction(a+2*(i+1)*h1,c+2*(j+1)*h2)) +
                    4 * (function.getFunction(a+(2*i+1)*h1,c+2*j*h2) +
                    function.getFunction(a+2*i*h1,c+(2*j+1)*h2) +
                    function.getFunction(a+2*(i+1)*h1,c+(2*j+1)*h2) +
                    function.getFunction(a+(2*i+1)*h1,c+2*(j+1)*h2)) +
                    16 * function.getFunction(a+(2*i+1)*h1,c+(2*j+1)*h2))
        }
    }
    sum *= h1 * h2 / 9

    return sum
}
```

3.2 Реализация метода интегрирования тройного интеграла

```
fun simpsonTriple(
    function: FunctionTriple,
    a: Double,
    b: Double,
    c: Double,
    d: Double,
    e: Double,
    f: Double,
    n: Int
) : Double {

    if((n%2 != 0) && n > 0) {
        throw IllegalArgumentException("Argument {n} must be even and more than 0")
    }

    var sum = 0.0           //значение интеграла после всех сложений
    val h1=(b-a)/(2*n)
    val h2=(d-c)/(2*n)
    val h3=(f-e)/(2*n)

    for (i in 0 until n){
        for (j in 0 until n)
            for (k in 0 until n) {

               sum+=((function.getFunction(a+2*i*h1,c+2*j*h2, e+2*k*h3) +
                       function.getFunction(a+2*(i+1)*h1,c+2*j*h2, e+2*k*h3) +
                       function.getFunction(a+2*i*h1,c+2*(j+1)*h2, e+2*k*h3) +
                       function.getFunction(a+2*(i+1)*h1,c+2*(j+1)*h2, e+2*k*h3) +
                       function.getFunction(a+2*i*h1,c+2*j*h2, e+2*(k+1)*h3) +
                       function.getFunction(a+2*(i+1)*h1,c+2*j*h2, e+2*(k+1)*h3) +
                       function.getFunction(a+2*i*h1,c+2*(j+1)*h2, e+2*(k+1)*h3) +
                       function.getFunction(a+2*(i+1)*h1,c+2*(j+1)*h2, e+2*(k+1)*h3)) +
                       4 * (function.getFunction(a+2*i*h1,c+(2*j+1)*h2, e+2*(k+1)*h3) +
                       function.getFunction(a+2*(i+1)*h1,c+(2*j+1)*h2, e+2*(k+1)*h3) +
                       function.getFunction(a+(2*i+1)*h1,c+2*j*h2, e+2*k*h3) +
                       function.getFunction(a+2*i*h1,c+(2*j+1)*h2, e+2*k*h3) +
                       function.getFunction(a+2*(i+1)*h1,c+(2*j+1)*h2, e+2*k*h3) +
                       function.getFunction(a+(2*i+1)*h1,c+2*(j+1)*h2, e+2*k*h3) +
                       function.getFunction(a+2*i*h1,c+2*j*h2, e+(2*k+1)*h3) +
                       function.getFunction(a+2*(i+1)*h1,c+2*j*h2, e+(2*k+1)*h3) +
                       function.getFunction(a+2*i*h1,c+2*(j+1)*h2, e+(2*k+1)*h3) +
                       function.getFunction(a+2*(i+1)*h1,c+2*(j+1)*h2, e+(2*k+1)*h3) +
                       function.getFunction(a+(2*i+1)*h1,c+2*j*h2, e+2*(k+1)*h3) +
                       function.getFunction(a+(2*i+1)*h1,c+2*(j+1)*h2, e+2*(k+1)*h3)) +
                       16 * (function.getFunction(a+(2*i+1)*h1,c+(2*j+1)*h2,
e+2*(k+1)*h3) +
                       function.getFunction(a+(2*i+1)*h1,c+(2*j+1)*h2, e+2*k*h3) +
                       function.getFunction(a+(2*i+1)*h1,c+2*j*h2, e+(2*k+1)*h3) +
                       function.getFunction(a+2*i*h1,c+(2*j+1)*h2, e+(2*k+1)*h3) +
                       function.getFunction(a+2*(i+1)*h1,c+(2*j+1)*h2, e+(2*k+1)*h3) +
                       function.getFunction(a+(2*i+1)*h1,c+2*(j+1)*h2, e+(2*k+1)*h3)) +
                       64 * function.getFunction(a+(2*i+1)*h1,c+(2*j+1)*h2,
e+(2*k+1)*h3))
            }
    }
    sum *= h1 * h2 * h3 / 27

    return sum
}
```

Список использованных источников

1. Wikipedia. Формула Симпсона. — https://en.wikipedia.org/wiki/Simpson%27s_rule

2. Градов В.М. – Компьютерные технологии в практике математического моделирования: Учебное пособие. – Ч. 1 — М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 108 с: ил.

3. Градов В. М. – Лекции.