Университет ИТМО
Факультет программной инженерии и компьютерной техники

Вычислительная математика

Лабораторная работа №2
Вариант – метод Симпсона

Выполнила: Екатерина Машина

Группа P3210

Преподаватель: Ольга Вячеславовна Перл

Санкт-Петербург
2020 г.

Цель работы

Реализовать метод Симпсона (метод парабол) для решения определенных интегралов и оценить погрешность по методу Рунге.

Описание использованного метода

Суть метода заключается в том, что на определенном промежутке дуга некоторой параболы в общем случае теснее прилегает кривой *y = f(x),* чем хорда, соединяющая концы дуги этой кривой (как в методе трапеций).
В связи с этим значения площадей соответствующих элементарных трапеций, ограниченных сверху дугами парабол, являются более близкими к значениям площадей соответствующих частичных криволинейных трапеций, ограниченных сверху дугой кривой *y = f(x),* чем значения площадей соответствующих прямолинейных трапеций.

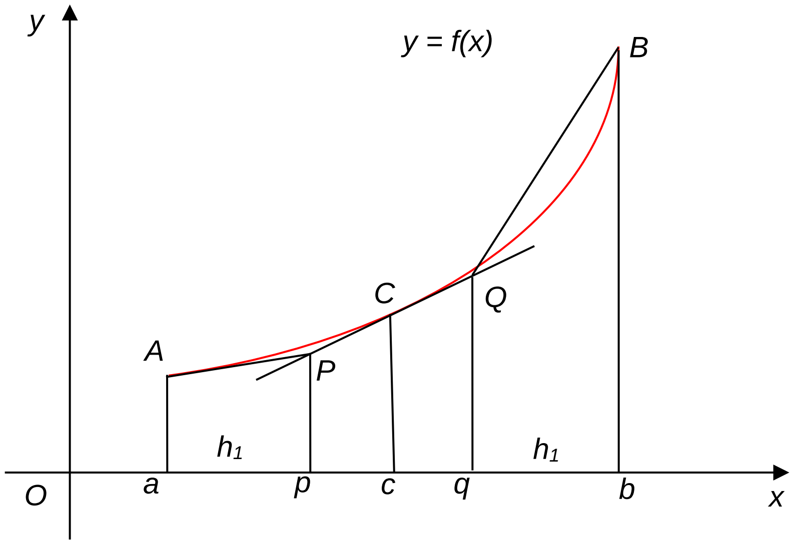


Рис1. Геометрическая интерпретация метода парабол.

Рассмотрим функцию *y = f(x)* такую, что на отрезке [*a; b*] она положительна и непрерывна. Найдем площадь криволинейной трапеции aABb (Рис1.).

Разобьём отрезок [*a*, *b*] точкой $c = \frac{a+b}{2}$ пополам и в точке *C*(*c*, *f*(*c*)) и проведем касательную к графику *y = f(x).*
После этого разделим [*a*, *b*] точками *p* и *q* на три равные части и проведем через них прямые *x = p* и *x = q*.
Пусть *P* и *Q* – точки пересечения этих прямых с касательной. Соединив *A* с *P* и *B* с *Q*, получим три трапеции *aAPp, pPQq, qQBb*. Тогда площадь трапеции *aABb* можно приближенно посчитать по следующей формуле

$$I \approx \frac{aA + pP}{2} \cdot h_1 + \frac{pP + qQ}{2} \cdot h_1 + \frac{qQ + bB}{2} \cdot h_1$$, где $$h_1 = \frac{b-a}{3}$$.

Откуда получаем

$$I \approx \frac{b-a}{6} \cdot \left(aA + 2\left(pP + qQ\right) + bB\right).$$

Заметим, что *aA = f(a), bB = f(b)*, а *pP + qQ = 2f(c)* (как средняя линия трапеции), в итоге получаем формулу Симпсона:

$$I \approx \frac{b-a}{6} \cdot \left(f(a) + 4f(c) + f(b) \right)$$

Общий вид формулы Симпсона:

Если кратко, то суть метода заключается в том что мы делим подынтегральную функцию на n(четное) равных четное) равных отрезков, на каждом из которых аппроксимируем значение этой функции параболой, затем можем вычислить искомый интеграл по формуле:

$$\int_a^b f(x)dx \approx \frac{h}{3} \sum_{k=1,2}^{n-1} \left(f(x_{k-1}) + 4f(x_k) + f(x_{k+1}) \right)$$

При этом мы вычисляем значение интеграла по формуле для разбиения на n и 2n отрезков, после чего вычисляем погрешность по формуле Рунге.

$$\Delta_{2n} \approx \theta |I_{2n} - I_n| ,$$

где для метода Симпсона $\theta = \frac{1}{15}$

При этом если найденная погрешность оказывается больше заданной точности то увеличиваем n в два раза и снова вычисляем интегралы. Затем повторяем увеличение n до тех пор погрешность не окажется меньше нужной точности.

Выводы

Выполнив данную лабораторную работу я пришла к выводу, что все приведенные в вариантах методы интегрирования схожи между собой в том, что всегда производится разбиение функции на n отрезков, а затем интерполирование на каждом из них.

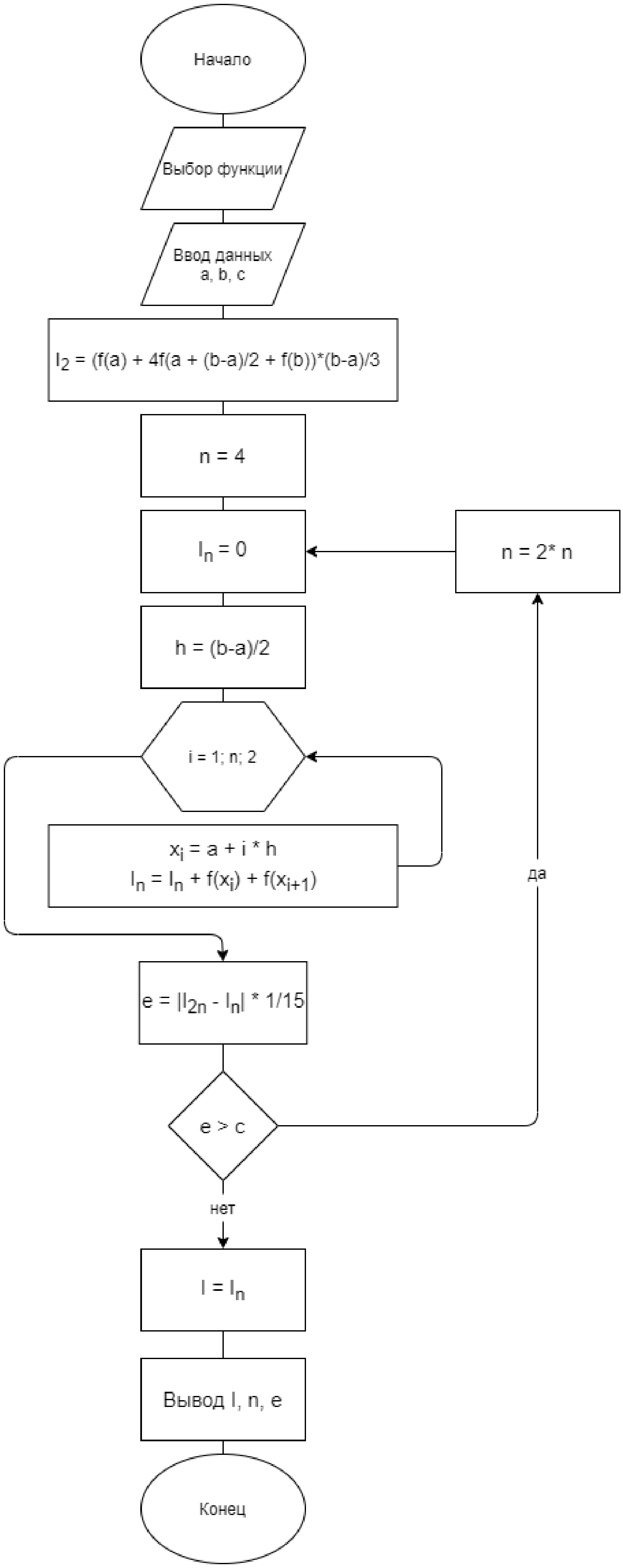
Рассмотрю подробнее все три метода, чтобы это доказать.

1. Метод прямоугольников:
   1.1 Метод левых прямоугольников
   1.2 Метод средних прямоугольников
   1.3 Метод правых прямоугольников

   Метод заключается в том, что мы разбиваем фигуру под графиком на прямоугольники и считаем интеграл как сумму площадей этих прямоугольников с учетом значения функции в левой точке (метод левых прямоугольников) в средней точке (метод средних прямоугольников) и в правой точке (метод правых прямоугольников). В силу того что в методе средних прямоугольников симметрия не нарушается, то погрешность у данного метода будет меньше чем у левых и правых прямоугольников.

2. Метод трапеций:
   В данном методе мы аппроксимируем функцию к прямой и считаем интеграл как площадь многоугольника образованного получившимися трапециями. Данный метод менее точный чем средние прямоугольники так как значение в средней точке точнее чем полусумма значений на концах.
3. Метод Симпсона:
   Метод Симпсона является самым точным из всех представленных методов, так как мы аппроксимируем функцию параболой (которая зачастую находится гораздо ближе к графику (в сравнении с прямой)).

Блок-схема


Начало
Выбор функции
Ввод данных
a, b, c
I2 = (f(a) + 4f(a + (b-a)/2 + f(b))*(b-a)/3
n = 4
In = 0
n = 2* n
h = (b-a)/2
i = 1; n; 2
xi = a + i * h
In = In + f(xi) + f(xi+1)
да
e = |I2n - In| * 1/15
e > c
нет
I = In
Вывод I, n, e
Конец

Листинг численного метода

```java
public static double[] integrate(Function function, double lowLimit,
double upLimit, double accuracy) {
        int countOfDivisions = 2;
        double measurementError = 2;
        double step = (upLimit - lowLimit)/countOfDivisions;

        double lowValue = function.getY(lowLimit);
        if (!Double.isFinite(lowValue))
            lowValue = function.getY(lowLimit + EPS);

        double upValue = function.getY(upLimit);
        if (!Double.isFinite(lowValue))
            upValue = function.getY(upLimit - EPS);

        double value = 4 * function.getY(lowLimit + step);
        if (!Double.isFinite(value))
            value = 4 *(function.getY((lowLimit + step + EPS)) +
function.getY(lowLimit + step - EPS))/2;

        double previousValue = (step / 3)*(lowValue + value + upValue);
        double currentValue = 0;

        while (measurementError > accuracy && countOfDivisions <
MAX_VALUE_OfDivisions) {

            countOfDivisions *= 2;
            step = (upLimit - lowLimit)/countOfDivisions;
            currentValue = (step / 3)*(countSum(function,countOfDivisions,
step, lowLimit));
            measurementError = (Math.abs(currentValue -
previousValue)/15);
            previousValue = currentValue;
        }

        return new double[]{ currentValue, countOfDivisions,
measurementError};
    }
```

```java
private static double countSum(Function function, double stepCounter,
double step, double lowLimit){
        double result = 0;

        double currentValue;
        for (int i = 0; i < stepCounter; i+=2){

            double tmp = 0;

            currentValue = function.getY(lowLimit + step*(i-1));
            if (!Double.isFinite(currentValue))
                if (i==0)
                    currentValue = function.getY(lowLimit + EPS);
                else
                    currentValue = (function.getY(lowLimit + step*(i-1) +
EPS) + function.getY(lowLimit + step*(i-1) - EPS))/2;
```

```
            tmp += currentValue;


            currentValue = function.getY(lowLimit + step*i);
            if (!Double.isFinite(currentValue))
                currentValue = (function.getY(lowLimit + step*i + EPS) +
function.getY(lowLimit + step*i - EPS))/2;
            tmp += 4 * currentValue;


            currentValue = function.getY(lowLimit + step*(i+1));
            if (!Double.isFinite(currentValue))
                if (i==stepCounter)
                    currentValue = function.getY(lowLimit + EPS);
                else
                    currentValue = (function.getY(lowLimit + step*(i+1) +
EPS) + function.getY(lowLimit + step*(i+1) - EPS))/2;
            tmp += currentValue;

            result += tmp;
        }
        return result;
    }
```

Примеры

Изначально выводимый при старте программы текст:

To choose function type 'input' here,
you choose from some functions:
1) y = 2x + 5
2) y = 1 / x
3) y = x^2
4) y = sqrt(x)
5) y = sin(x) / x
6) y = e^x
to exit enter 'exit'
to get help use 'help'.

Примеры работы:

Пример 1

```
> input
Choose on of these functions. Type number of
function you want to integrate
1) y = 2x + 5
2) y = 1 / x
3) y = x^2
4) y = sqrt(x)
5) y = sin(x) / x
6) y = e^x

>>> 6
Type low limit  of the integration here
>>> 12
Type up limit  of the integration here
>>> 1
Type wanted accuracy
>>> 0.001
 Integral: -162752.05979862803
 Amount of divisions: 1.34217728E8
 Measurement error: 8.89239328292509E-4
```

Пример 2

```
> input
Choose on of these functions. Type number of
function you want to integrate
1) y = 2x + 5
2) y = 1 / x
3) y = x^2
4) y = sqrt(x)
5) y = sin(x) / x
6) y = e^x

>>> 4
Type low limit  of the integration here
>>> 5
Type up limit  of the integration here
>>> -1
Type wanted accuracy
>>> 0.001
Integral does not converge
```

Пример 3

```
> input
Choose on of these functions. Type number of
function you want to integrate
1) y = 2x + 5
2) y = 1 / x
3) y = x^2
4) y = sqrt(x)
5) y = sin(x) / x
6) y = e^x

>>> 2
Type low limit  of the integration here
>>> -5
Type up limit  of the integration here
>>> 5
Type wanted accuracy
>>> 0.001
 Integral: -0.007812539737036321
 Amount of divisions: 512.0
 Measurement error: 5.208518782537581E-4
```

Пример 4

```
> input
Choose on of these functions. Type number of
function you want to integrate
1) y = 2x + 5
2) y = 1 / x
3) y = x^2
4) y = sqrt(x)
5) y = sin(x) / x
6) y = e^x

>>> 2
Type low limit  of the integration here
>>> -2
Type up limit  of the integration here
>>> 1
Type wanted accuracy
>>> 0.001
Cannot get accuracy
```