НАУЧНО–ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ

ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Лабораторная работа **№ 2 (Интегрирование)**

по курсу «**Вычислительная математика**»

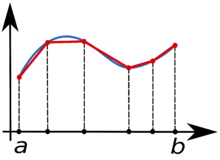
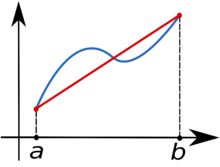
Вариант: метод трапеций

Санкт-Петербург

2016 г.

Описание метода (метод трапеций)

Метод трапеций использует линейную интерполяцию, т.е. график функции у=f(х) представляется в виде ломаной, соединяющей точки (хi, уi). В этом случае площадь всей криволинейной трапеции складывается из площадей элементарных прямоугольных трапеций.

Площадь каждой такой трапеции определяется по формуле:

*; ,* где n – число интервалов разбиения*.*

Складывая все эти равенства, получим формулу трапеций для численного интегрирования:

, или

Также данные формулы можно представить в виде:

*; .*

Вычисление погрешности по оценке Рунге:

Интеграл вычисляется по выбранной формуле (прямоугольников, трапеций, парабол Симпсона) при числе шагов, равном n, а затем при числе шагов, равном 2n. Погрешность вычисления значения интеграла при числе шагов, равном 2n, определяется по **формуле Рунге**:

{\displaystyle \Delta \_{2n}\approx \Theta |I\_{2n}-I\_{n}|}Для формул средних прямоугольников и трапеций {\displaystyle \Theta ={\frac {1}{3}}} , а для формулы Симпсона {\displaystyle \Theta ={\frac {1}{15}}}.

Блок-схема метода

D:\ITMO\3_sem\ComputationalMathematics\labs\lab2\Untitled Diagram (2).png

Листинг класса, где реализован метод трапеций (ConsoleApp.Integral)

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.Linq;
4. **using** System.Text;
5. **using** System.Threading.Tasks;
7. **namespace** ConsoleApp
8. {
9. **class** Integral
10. {
11. **public** **static** **double** EPS = 0.0001;
12. **public** **static** CalculatedFunction Calculate(**int** f, **double** xLower, **double** xUpper, **double** precision) {
13. //initialization
14. **int** numOfIter = 1;
15. **double** previousIter = 0;
16. **double** tmpIter = 0;
17. **double** tmpPrecision = 0;
18. **double** h = (xUpper - xLower);
20. // main cycle for Integral calculation
21. **do** {
22. previousIter = tmpIter;
23. **double** sum = 0;
24. **if** (!conditionIsTrue(f, xLower, xUpper)) {
25. **return** **null**;
26. }
27. **for** (**int** i = 1; i < numOfIter+1; i++) {
28. **double** w = FindFunctionValue(f, xLower + i \* h);
29. sum += w;
30. }
31. tmpIter = h \* (((FindFunctionValue(f, xLower) + FindFunctionValue(f, xUpper)) \* 0.5) + sum);
32. tmpPrecision = Math.Abs(tmpIter - previousIter) / 3;
33. **if**( tmpPrecision > precision) {
34. h /= 2;
35. numOfIter \*= 2;
36. }
37. } **while** (tmpPrecision > precision && numOfIter < Math.Pow(2,26)); // 2^26 - ONLY MY RESTRICTION!
39. **return** **new** CalculatedFunction(f, tmpIter, tmpPrecision, numOfIter);
40. }
42. **private** **static** **double** FindFunctionValue(**int** function\_num, **double** x) {
43. **double** res = **double**.NaN;
44. **switch** (function\_num) {
45. **case** 1: {
46. res = x \* x \* x + 3;
47. **break**;
48. }
49. **case** 2: {
50. res = 1 / Math.Log(x);
51. **break**;
52. }
53. **case** 3: {
54. res = x \* Math.Exp(-x);
55. **break**;
56. }
57. }
58. **return** res;
59. }
61. // handling exceptions
62. **private** **static** **bool** conditionIsTrue (**int** function\_num, **double** xLower, **double** xUpper) {
63. **switch** (function\_num) {
64. **case** 1: {
65. **return** **true**;
66. }
67. **case** 2: {
68. **if** (xUpper < EPS || (xUpper < 1 && xLower > EPS) || xLower > 1)
69. **return** **true**;
70. **else** **return** **false**;
71. }
72. **case** 3: {
73. **return** **true**;
74. }
75. **default**: {
76. **return** **false**;
77. }
78. }
79. }
80. }
81. }

Примеры работы программы

-----------------------------------MAIN MENU-----------------------------------

Choose the function:

[1] f = x^3 + 3

[2] f = 1 / ln(x)

[3] f = x \* exp(-x)

[9] leave the program

> 1

Enter the value of the lower bound:

> 45

Enter the value of the upper bound:

> -45

Enter the value of the precision:

> 1

Serial number: 1

Number of sections: 4194304

Final result: 271,955394741976

Precision: 0,651798249203125

-----------------------------------MAIN MENU-----------------------------------

Choose the function:

[1] f = x^3 + 3

[2] f = 1 / ln(x)

[3] f = x \* exp(-x)

[9] leave the program

> 2

Enter the value of the lower bound:

> -100

Enter the value of the upper bound:

> 100

Enter the value of the precision:

> 50

Error! Check that the function is continuous in the range you have entered!

-----------------------------------MAIN MENU-----------------------------------

Choose the function:

[1] f = x^3 + 3

[2] f = 1 / ln(x)

[3] f = x \* exp(-x)

[9] leave the program

> 3

Enter the value of the lower bound:

> 15

Enter the value of the upper bound:

> -15

Enter the value of the precision:

> 1

Serial number: 3

Number of sections: 65536

Final result: -45766244,1279682

Precision: 0,913352467119694

Выводы по работе:

В ходе выполнения работы я ознакомился с принципами численного вычисления интегралов с заданной точностью и основными методами, в особенности с методом трапеции и вычислением погрешности с помощью правила Рунге.