|  |  |
| --- | --- |
| Изображение выглядит как текст, керамические изделия, фарфор  Автоматически созданное описание | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

Факультет «Фундаментальные науки»

Кафедра «Техническая физика»

**Лабораторная работа №5**

по курсу «Вычислительная физика»

Выполнили: Коберник Т. Н., Плетенёв Б. А.

Группа: ФН4-71Б

Преподаватели: Хасаншин Р.Х., Ивлиев П.А.

Москва, 2022 г.

Содержание

[**1.** **Теоретическая часть** 3](#_Toc116841460)

[**1.1** **Метод прямоугольников** 3](#_Toc116841461)

[**1.2** **Метод трапеций** 3](#_Toc116841462)

[**1.3** **Метод Симпсона** 3](#_Toc116841463)

[**1.4** **Экстраполяция Ричардсона** 3](#_Toc116841464)

[**2.** **Постановка задачи** 3](#_Toc116841465)

[**3.** **Программа** 3](#_Toc116841466)

[**4.** **Результаты** 5](#_Toc116841467)

5. Выводы

1. **Теоретическая часть**

## **Метод прямоугольников**

## **Метод трапеций**

## **Метод Симпсона**

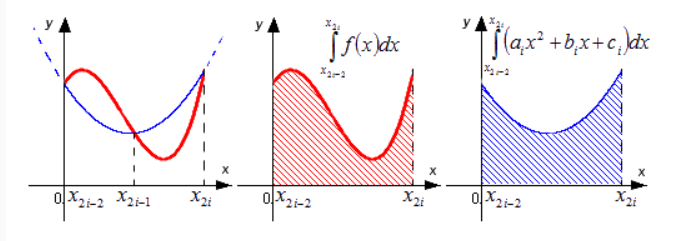
В данном методе в качестве приближенной функции используется полином Лагранжа 2-го рода.

Пускай задана функция вида , имеющая непрерывность на интервале  [a; b], необходимо произвести вычисление определенного интеграла .

Необходимо разбить отрезок [a; b] на n отрезков вида , i=1,..,n с длиной  и точками . Тогда точки   считаются  серединами отрезков . Данный случай показывает, что определение узлов производится через

Каждый интервал подынтегральной функции приближен при помощи параболы, заданной , проходящей через точки  с координатами .

Приближённое значение интеграла будем искать в виде



*Рис. 1. Графическая иллюстрация метода парабол (Симпсона)*

Очевидно, что

Пускай , тогда

Рассмотрим интеграл:

Таким образом для метода парабол получаем формулу:

## **Экстраполяция Ричардсона**

При использовании формул численного интегрирования результат часто получается недостаточно точным. Если требуется увеличить точность значений, применяют экстраполяцию Ричардсона. Предположим, что нужно вычислить значение интеграла.

Предположим, что нужно вычислить значение интеграла

Вычислим значение интеграла с шагом ℎ, и значение с шагом . Пусть при интегрировании использовались формулы порядка 𝑝. Тогда получим:

Константа 𝑐 определяется значением производной с точностью до интервала. Поэтому 𝑐1 ≈ 𝑐 с точностью до ℎ. Отсюда следует, что

Такой метод называется экстраполяцией Ричардсона. Формулу можно использовать в различных случаях:

1. Для увеличения порядка аппроксимации формулы интегрирования.

2. Для практического оценивания погрешности (правило Рунге ) по величине

Заметим, что результат интегрирования находится вне отрезка значений от до 𝐼(ℎ). Поэтому такой метод называется экстраполяцией. Интеграл можно записывать различными способами:

Аналогично можно выписать интеграл и на более редкой сетке. Вспомним, что симметричные формулы обладают свойством, что их ошибки оказываются на порядок лучше, чем ожидается из построения. Это относится к формуле прямоугольников с центральной точкой (второго порядка сходимости, а не первого) и к формуле Симпсона (четвертого порядка аппроксимации, а не третьего). Это свойство имеет и алгоритм экстраполяции Ричардсона. Применив алгоритм к формулам интегрирования, получим формулу с порядком аппроксимации, улучшенным на два.

1. **Постановка задачи**

Используя формулы прямоугольников, трапеций, Симпсона и экстраполяцию Ричардсона, вычислить значение интеграла:

Вариант №6 

Получить значения интеграла при трех вариантах сетки на отрезке интегрирования.

Получить точное значение интегралов и сравнить с результатами численного интегрирования.

1. **Программа**
2. **import** numpy as np
3. **import** matplotlib.pyplot as plt
4. **import** math
6. **def** function(x):
7. **return** x **\*** math.atan(x)
9. **def** rectangle\_method(x):
10. sum **=** 0
11. **for** i **in** range(len(x)**-**1):
12. sum **+=** (x[i**+**1] **-** x[i]) **\*** function((x[i**+**1] **+** x[i]) **/** 2)
13. **return** sum
15. **def** trapezoid\_method(x):
16. h **=** x[1] **-** x[0]
17. sum **=** 0
18. **for** i **in** range(len(x)**-**1):
19. sum **+=** (function(x[i**+**1]) **+** function(x[i])) **/** 2 **\*** h
20. **return** sum
22. **def** Simpson\_method(x):
23. h **=** x[1] **-** x[0]
24. sum **=** 0
25. **for** i **in** range(round((len(x)**-**2)**/**2)):
26. sum **+=** h **/** 3 **\*** (function(x[2 **\*** i]) **+** 4 **\*** function(x[2 **\*** i **+** 1]) **+** function(x[2 **\*** (i **+** 1)]))
27. **return** sum
29. **def** Richardson\_method(x, integration\_method):
30. **if** integration\_method **in** [trapezoid\_method, rectangle\_method]:
31. p **=** 2
32. **else**:
33. p **=** 2
34. h **=** x[1] **-** x[0]
35. x0 **=** np.linspace(0, math.sqrt(3), round((2 **\*** math.sqrt(3) **/** h)))
36. int\_cur **=** integration\_method(x)
37. int\_next **=** integration\_method(x0) **+** (integration\_method(x0) **-** integration\_method(x)) **/** (2 **\*\*** p **-** 1)
38. eps **=** h **\*\*** p
39. **while** abs(int\_next **-** int\_cur) > eps:
40. x **=** x0
41. h **/=** 2
42. x0 **=** np.linspace(0, math.sqrt(3), round((2 **\*** math.sqrt(3)**/**h)))
43. int\_cur **=** int\_next
44. int\_next **=** integration\_method(x0) **+** (integration\_method(x0) **-** integration\_method(x)) **/** (2 **\*\*** p **-** 1)
45. **return** int\_next
47. # должно быть четное кол-во отрезков в сетке
48. **for** length **in** [500, 1000 ,1500]:
49. x **=** np.linspace(0, math.sqrt(3), length)
50. print(f'Количество точек = {length}')
51. print(f'Интегрирование методом прямоугольников: I = {rectangle\_method(x)}')
52. print(f'Интегрирование методом трапеций: I = {trapezoid\_method(x)}')
53. print(f'Интегрирование методом Симпсона: I = {Simpson\_method(x)}')
55. print(f'Интегрирование методом прямоугольников + экстраполяция Ричардсона: I = {Richardson\_method(x, rectangle\_method)}')
56. print(f'Интегрирование методом трапеций + экстраполяция Ричардсона: I = {Richardson\_method(x, trapezoid\_method)}')
57. print(f'Интегрирование методом Симпсона + экстраполяция Ричардсона: I = {Richardson\_method(x, Simpson\_method)}')
58. **Результаты**

Точное значение интеграла:

***I =1.22836969860875684554470575143339907266004363934488023328872623***

В ходе работы приведенной выше программы были получены следующие результаты:

Количество точек = 500  
Интегрирование методом прямоугольников: I = 1.2283689555342665  
Интегрирование методом трапеций: I = 1.2283711847577046  
Интегрирование методом Симпсона: I = 1.22208283776855

Интегрирование методом прямоугольников + экстраполяция Ричардсона:

I = 1.2283696981116428  
Интегрирование методом трапеций + экстраполяция Ричардсона:

I = 1.2283696996029914  
Интегрирование методом Симпсона + экстраполяция Ричардсона:

I = 1.2283615009826305

Количество точек = 1000  
Интегрирование методом прямоугольников: I = 1.2283695132118642  
Интегрирование методом трапеций: I = 1.2283700694025395  
Интегрирование методом Симпсона: I = 1.2252271858717845

Интегрирование методом прямоугольников + экстраполяция Ричардсона:

I = 1.2283696985468504  
Интегрирование методом трапеций + экстраполяция Ричардсона:

I = 1.2283696987325712  
Интегрирование методом Симпсона + экстраполяция Ричардсона:

I = 1.2283676512536617

Количество точек = 1500  
Интегрирование методом прямоугольников: I = 1.2283696162651003  
Интегрирование методом трапеций: I = 1.2283698632960736  
Интегрирование методом Симпсона: I = 1.2262748944032358

Интегрирование методом прямоугольников + экстраполяция Ричардсона:

I = 1.2283696985904335  
Интегрирование методом трапеций + экстраполяция Ричардсона:

I = 1.2283696986453974  
Интегрирование методом Симпсона + экстраполяция Ричардсона:

I = 1.2283690163846757

1. **Выводы**

По приведённым результатам вычислений видно, что метод прямоугольников дает более точные результаты, нежели метод трапеции. Самым точным является метод Симпсона.  
При увеличении количества узловых точек, точность каждого из приближений значительно увеличивается.

Также, при применении экстраполяции Ричардсона удается существенно улучшить значения каждого из методов. Однако применение данной экстраполяции к методу Симпсона требует большой вычислительной мощности, что необходимо учитывать при выполнении практических задач.