Отчет по лабораторной работе № 4
Вариант № 18(3)

Щербаков Илья Анатольевич

Группа: Б9122-02-03-01сцт2

Цель работы

1. Вычислить интеграл: $\int_a^b f(x)dx$ с использованием метода центральных прямоугольников.;
2. Получить формулу для численого интегрирования методом центральных прямоугольников в виде $I = \sum_{i=0}^{n} c_i \cdot f(x_i)$;
3. Изучить, как точно метод аппроксимирует интеграл. Оценить, насколько близко полученное значение к точному.
4. Провести численное исследование с различными значениями n для метода центральных прямоугольников.
5. Сделать вывод о том, как меняется ошибка при изменении параметра n.
6. Сравнить метод центральных прямоугольников с другими известными методами, такими как метод прямоугольников, метод трапеций и формула Симпсона для n = 10000.
7. Сделать вывод о том, насколько эффективен метод центральных прямоугольников по сравнению с другими методами.
8. Представить заключение на основе проведенного анализа.

Дано:

1. **Функция:** $y = x^2 + \ln(x) - 4$
2. **Отрезок** $[1.5, 2.0]$

Вычисление интеграла $\int_a^b f(x)dx$:

Подставим известные значения **a**, **b** в нашу функцию:

$$\int_{1.5}^{2}(x^2+\ln(x)-4)dx = \int x^2dx + \int \ln x dx - \int 4dx = (\frac{x^3}{3} + \ln n \cdot x - 5x)\Big|_{1.5}^{2} = \frac{2^3}{3} + \ln 2 \cdot 2 - 5 \cdot 2 - \left(\frac{1.5^3}{3} + \ln 1.5 \cdot 1.5 - 5 \cdot 1.5\right) =$$

$$= -\frac{23}{24} + \ln\frac{8\sqrt{6}}{9} \approx -0.180237$$

Получаем формулу $I = \sum_{i=0}^{n} c_i \cdot f(x_i)$:

$$\sum_{i=0}^{n-1} f(x_{i+\frac{1}{2}}) \int_{x_i}^{x_i+1} \frac{x - x_i}{x_{i+\frac{1}{2}} - x_i} dx = \sum_{i=0}^{n-1} f(x_{i+\frac{1}{2}}) \int_{x_i}^{x_i+1} \frac{2(x - x_i)}{h} dx = \sum_{i=0}^{n-1} f(x_{i+\frac{1}{2}}) \cdot \frac{1}{h}(x_{i+1} - x_i)^2 = \sum_{i=0}^{n-1} f(x_{i+\frac{1}{2}}) \cdot h$$

Определение порядка аппроксимации:

$$R_n(x) = \int_a^b f(x)\,dx - \sum_{i=0}^{n} f(x_{i+\frac{1}{2}}) \cdot h = \int_a^b f(x)dx - \sum_{i=0}^{n} f(x_{i+\frac{1}{2}})(x_i - x_{i-1}) =$$

$$= \int_a^b f(x)dx - \sum_{i=0}^{n} \int_{x_{i-1}}^{x_i} f(x_{i-\frac{1}{2}})dx = \sum_{i=0}^{n} \int_{x_{i-1}}^{x_i} f(x_{i-\frac{1}{2}})dx$$

$$\sum_{i=0}^{n} \int_{x_{i-1}}^{x_i} \left(f(x_{i-\frac{1}{2}}) + f'(x_{i-\frac{1}{2})\cdot(x-x_{i-\frac{1}{2}}}) + \frac{f''(x_{i-\frac{1}{2}})}{2} \cdot (x - x_{i-\frac{1}{2}})^2 \right) =$$

$$= \sum_{i=0}^{n} \left(\frac{f'(x_{i-\frac{1}{2}})}{2} \left((x_{i+1} - x_{i-\frac{1}{2}})^2 - (x_i - x_{i-\frac{1}{2}})^2 \right) + \frac{f''(x_{i-\frac{1}{2}})}{6} \cdot \left((x_{i+1} - x_{i-\frac{1}{2}})^3 - (x_i - x_{i-\frac{1}{2}})^3 \right) \right) =$$

$$\sum_{i=0}^{n} \frac{f''(x_{i-\frac{1}{2}})}{2} \cdot \left(\left(\frac{h}{2}\right)^3 - \left(-\frac{h}{2}\right)^3 \right) = \sum_{i=0}^{n} \frac{f''(x_{i-\frac{1}{2}})}{6} \cdot \left(\frac{h^3}{8} + \frac{h^3}{8} \right) \leq \frac{M}{24} \cdot (b-a) \cdot h^2$$

$$\sup_{a \leq x \leq b} |f''(x)| = M$$

Соответственно можно сделать вывод - порядок аппроксимации второй.

Реализация на Питоне:

Опредение основных функций

```
def custom_function(x):
    return x ** 2 + log(x) - 4

def custom_function_derivative(x, k):
    if k == 1:
        return 2 * x + 1 / x
    elif k == 2:
        return 2 - 1 / x ** 2
    return (-1) ** ((k % 2) + 1) * factorial(k - 1) / x ** k

def middle_rectangle_method(func, a, b, n):
    h = (b - a) / n
    return sum(func(a + h * (i + 0.5)) * h for i in range(n))

def middle_rectangle_error(func, a, b, n):
    m = max(abs(custom_function_derivative(a + (b - a) * i / 1000, 2))
    for i in range(1001))
    return m / 24 * (b - a) ** 3 / n ** 2

def left_rectangle_method(func, a, b, n):
    h = (b - a) / n
    return sum(func(a + h * i) * h for i in range(n))

def right_rectangle_method(func, a, b, n):
    h = (b - a) / n
    return sum(func(a + h * i) for i in range(1, n + 1)) * h

def trapezoidal_method(func, a, b, n):
    h = (b - a) / n
    return ((func(a) + func(b)) / 2 + sum(func(a + h * i)
    for i in range(1, n))) * h

def simpson_method(func, a, b, n):
    h = (b - a) / n
    return sum(func(a + h * (i - 1)) + 4 * func(a + h * (i - 0.5)) +
```

```
    func(a + h * (i)) for i in range(1, n + 1)) * h / 6

def left_rectangle_error(func, a, b, n):
    m = max(abs(custom_function_derivative(a + (b - a) * i / 1000, 1))
    for i in range(1001))
    return m * (b - a) / 2

def right_rectangle_error(func, a, b, n):
    m = max(abs(custom_function_derivative(a + (b - a) * i / 1000, 1))
    for i in range(1001))
    return m * (b - a) / 2

def trapezoidal_error(func, a, b, n):
    m = max(abs(custom_function_derivative(a + (b - a) * i / 1000, 2))
    for i in range(1001))
    return m / 12 * (b - a) ** 3 / n ** 2

def simpson_error(func, a, b, n):
    m = max(abs(custom_function_derivative(a + (b - a) * i / 1000, 4))
    for i in range(1001))
    return m / 2880 * (b - a) ** 5 / n ** 4
```

Составление таблицы значений для метода центральных прямоугольников

Для составления таблицы значений для метода центральных прямоугольников мы:

1. Сначала установим начальные значения для интеграла, количества прямоугольников и других параметров.
2. Затем, с помощью цикла итераций, будем увеличивать количество прямоугольников, вычисляя значение интеграла методом центральных прямоугольников на каждой итерации.
3. Для каждой итерации также будем рассчитывать абсолютную разницу между точным значением интеграла и вычисленным значением, относительную погрешность и теоретическую погрешность метода.

Питон:

```
result = {'j': [], 'n': [], 'I_n': [], 'delta_I_n': [], 'relative_I_n': [],
          'R_n': [], 'growth': [0]}
for i in range(15):
    n *= 2
    I_n = middle_rectangle_method(custom_function, a, b, n)
    result['j'].append(i + 1)
    result['n'].append(n)
    result['I_n'].append(I_n)
    result['delta_I_n'].append(abs(exact_integral - I_n))
    result['relative_I_n'].append(result['delta_I_n'][i] / abs(exact_integral) * 100)
    result['R_n'].append(middle_rectangle_error(custom_function, a, b, n))
    if i > 0:
        result['growth'].append(result['delta_I_n'][i] / result['delta_I_n'][i - 1])

df_middle_rect = pd.DataFrame({
    'Iteration': result['j'],
    'n': result['n'],
    'I_n': result['I_n'],
    'delta_I_n': result['delta_I_n'],
    'Relative Error (%)': result['relative_I_n'],
    'R_n': result['R_n'],
    'Growth': result['growth']
})

```

```python
print("Table of values for the middle rectangle method:")
print(df_middle_rect)
```

Получившаяся таблица:

```
Table of values for the middle rectangle method:
    Iteration      n       I_n  ...  Relative Error (%)           R_n    Growth
0           1      2 -0.182408  ...        1.204942e+00  2.278646e-03  0.000000
1           2      4 -0.180779  ...        3.010687e-01  5.696615e-04  0.249862
2           3      8 -0.180372  ...        7.525664e-02  1.424154e-04  0.249965
3           4     16 -0.180271  ...        1.881350e-02  3.560384e-05  0.249991
4           5     32 -0.180245  ...        4.703333e-03  8.900960e-06  0.249998
5           6     64 -0.180239  ...        1.175831e-03  2.225240e-06  0.249999
6           7    128 -0.180237  ...        2.939574e-04  5.563100e-07  0.250000
7           8    256 -0.180237  ...        7.348920e-05  1.390775e-07  0.250000
8           9    512 -0.180237  ...        1.837217e-05  3.476938e-08  0.249998
9          10   1024 -0.180237  ...        4.592913e-06  8.692344e-09  0.249993
10         11   2048 -0.180237  ...        1.148099e-06  2.173086e-09  0.249972
11         12   4096 -0.180237  ...        2.868955e-07  5.432715e-10  0.249887
12         13   8192 -0.180237  ...        7.159406e-08  1.358179e-10  0.249548
13         14  16384 -0.180237  ...        1.776920e-08  3.395447e-11  0.248194
14         15  32768 -0.180237  ...        4.313282e-09  8.488617e-12  0.242739
```

Рис. 1: Таблица значений для метода центральных прямоугольников

Вывод о поведении ошибки

- Из табличных значений видно, что абсолютная ошибка метода центральных прямоугольников примерно соответствует теоретической ошибке, но немного меньше по значению. При увеличении числа прямоугольников n в степени, соответствующей порядку аппроксимации, абсолютная ошибка изменяется пропорционально. Например, с $j = 1$ до $j = 7$ ошибка заметно увеличивается, что означает незначительное уменьшение абсолютной ошибки. Затем, на $j = 7$ и $j = 8$, ошибка достигает максимального значения (0.25), после чего начинает уменьшаться до $j = 15$.

Составление сравнительной таблицы :

Для составления сравнительной таблицы различных методов численного интегрирования используется следующий алгоритм:

1. Для каждого из пяти методов определяются соответствующие функции и оценки погрешности.
2. Вычисляются значения интегралов и оценки погрешности для каждого метода в цикле.
3. Данные о значениях интегралов, абсолютных различиях, относительных погрешностях и теоретических погрешностях собираются в таблицу для каждого метода.

Питон (полностью не поместилось в рамки страницы, полный код на GitHub(ссылка в конце отчёта):

```python
calculation_results = {'method': ['Left Rectangular', "Right Rectangular",
                                  "Middle Rectangular", "Trapezoidal", "Simpson's"],
                       'I_n': [], 'delta_I_n': [], 'relative_I_n': [], 'R_n': []}
for i, (formula, error) in enumerate([(left_rectangle_method, left_rectangle_error),
                                      (right_rectangle_method, right_rectangle_error),
```

```
                                                    (middle_rectangle_method, middle_rectangle_error
                                                    (trapezoidal_method, trapezoidal_error),
                                                    (simpson_method, simpson_error)]):
     calculation_results['I_n'].append(formula(custom_function, a, b, 10000))
     calculation_results['delta_I_n'].append(abs(exact_integral - calculation_results['
     calculation_results['relative_I_n'].append(calculation_results['delta_I_n'][i] / a
     calculation_results['R_n'].append(error(custom_function, a, b, 10000))

table_headers = ['Method', 'I_n', 'delta_I_n', 'relative_I_n', 'R_n']
table_data = [["Left Rectangular", calculation_results['I_n'][0], calculation_results[
                    calculation_results['relative_I_n'][0], calculation_results['R_n'][0]],
                   ["Right Rectangular", calculation_results['I_n'][1], calculation_results
                    calculation_results['relative_I_n'][1], calculation_results['R_n'][1]],
                   ["Middle Rectangular", calculation_results['I_n'][2], calculation_result
                    calculation_results['relative_I_n'][2], calculation_results['R_n'][2]],
                   ["Trapezoidal", calculation_results['I_n'][3], calculation_results['delt
                    calculation_results['relative_I_n'][3], calculation_results['R_n'][3]],
                   ["Simpson's", calculation_results['I_n'][4], calculation_results['delta_
                    calculation_results['relative_I_n'][4], calculation_results['R_n'][4]]

print(tabulate(table_data, headers=table_headers, tablefmt='grid'))
```

Получившаяся таблица:

Method	I_n	delta_I_n	relative_I_n	R_n
Left Rectangular	-0.180288	5.09419e-05	0.0282639	1.125
Right Rectangular	-0.180186	5.09422e-05	0.0282641	1.125
Middle Rectangular	-0.180237	8.6495e-11	4.79897e-08	9.11458e-11
Trapezoidal	-0.180237	1.73922e-10	9.64964e-08	1.82292e-10
Simpson's	-0.180237	3.10807e-13	1.72444e-10	1.28601e-21

Рис. 2: Сравнительная таблица

Вывод о эффективности выбранного метода

- Среди всех рассмотренных методов численного интегрирования метод центральных прямоугольников демонстрировал второй по эффективности результат. Он предоставил более точные значения по сравнению с методами левых и правых прямоугольников, а также методом трапеций. Тем не менее, он оказался менее точным, чем метод Симпсона.

Заключение

- В ходе проведения лабораторной работы был проанализирован эффективность различных методов численного интегрирования, таких как метод левых, правых и центральных прямоугольников, метод трапеций и метод Симпсона. Результаты показали, что метод центральных прямоугольников оказался высокоэффективным по сравнению с методами левых и правых прямоугольников, а также методом трапеций. Однако он значительно уступает методу Симпсона, который демонстрирует лучшие результаты.

- Благодаря вычеисленным данным можно сделать вывод о том, что выбор метода численного интегрирования зависит от требуемой точности вычислений.

- Также в результате лабораторной работы были получены таблицы значений для метода центральных прямоугольников и сравнительного анализа различных методов численного интегрирования, что позволяет обосновать выводы.

Ссылка на GitHub:

IlushkaDV