Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

Высшего образования

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа 3 по вычислительной математике

Вариант №13

Группа: P3216

Выполнил:

Сиразетдинов А.Н.

Проверил:

Малышева Т. А.

Г. Санкт-Петербург

2024

Оглавление

Цель

Найти приближенное значение определенного интеграла с требуемой точностью различными численными методами.

Порядок выполнения работы

Обязательное задание (до 80 баллов)

Исходные данные:

1. Пользователь выбирает функцию, интеграл которой требуется вычислить (3-5 функций), из тех, которые предлагает программа.
2. Пределы интегрирования задаются пользователем.
3. Точность вычисления задается пользователем.
4. Начальное значение числа разбиения интервала интегрирования: n=4.
5. Ввод исходных данных осуществляется с клавиатуры.

Программная реализация задачи:

1. Реализовать в программе методы по выбору пользователя:
2. Метод прямоугольников (3 модификации: левые, правые, средние)
3. Метод трапеций
4. Метод Симпсона
5. Методы должны быть оформлены в виде отдельной(ого) функции/класса.
6. Вычисление значений функции оформить в виде отдельной(ого) функции/класса.
7. Для оценки погрешности и завершения вычислительного процесса использовать правило Рунге.
8. Предусмотреть вывод результатов: значение интеграла, число разбиения интервала интегрирования для достижения требуемой точности.

Вычислительная реализация задачи:

1. Вычислить интеграл, приведенный в таблице 1, точно.
2. Вычислить интеграл по формуле Ньютона – Котеса при $n = 6$.
3. Вычислить интеграл по формулам средних прямоугольников, трапеций и Симпсона при $n = 10$.
4. Сравнить результаты с точным значением интеграла.
5. Определить относительную погрешность вычислений для каждого метода.
6. В отчете отразить последовательные вычисления.

Рабочие формулы методов

Метод прямоугольников

На каждом шаге интегрирования функция аппроксимируется полиномом нулевой степени – отрезком, параллельным оси абсцисс. Площадь криволинейной трапеции приближенно заменяется площадью многоугольника, составленного из

n - прямоугольников. Таким образом, вычисление определенного интеграла сводится к нахождению суммы

n- элементарных прямоугольников.

$$\int_a^b f(x)dx \approx S_n = \sum_{i=1}^{n} f(\xi_i)\Delta x_i$$

Различают метод левых, правых и средних прямоугольников.

В качестве точек ξ_i могут выбираться левые ($\xi_i = x_{i-1}$) или правые ($\xi_i = x_i$) границы отрезков, получим формулы левых и правых прямоугольников.

УНИВЕРСИТЕТ ИТМО Лекция №3. Численное интегрирование

Метод трапеций

Подынтегральную функцию на каждом отрезке $[x_i; x_{i+1}]$ заменяют интерполяционным многочленом первой степени:

$$f(x) \approx \varphi_i(x) = a_i x + b$$

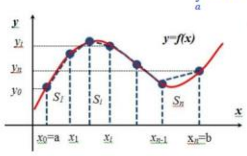
Используют линейную интерполяцию, т.е. график функции $y = f(x)$ представляется в виде ломаной, соединяющий точки (x_i, y_i) . Площадь всей фигуры (криволинейной трапеции):

$$S_{общ} = S_1 + S_2 + \dots + S_n = \frac{y_0 + y_1}{2}h_1 + \frac{y_1 + y_2}{2}h_2 + \dots + \frac{y_{n-1} + y_n}{2}h_n$$

$$y_0 = f(a), \quad y_n = f(b), \quad y_i = f(x_i), \quad h_i = x_i - x_{i-1}$$

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

$$\int_a^b f(x)dx = \frac{1}{2}\sum_{i=1}^{n} h_i(y_{i-1} + y_i)$$



При $h_i = h = \frac{b-a}{n} = const$ формула трапеций:

$$\int_a^b f(x)dx = h \cdot \left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i\right)$$

или $$\int_a^b f(x)dx = \frac{h}{2} \cdot \left(y_0 + y_n + 2\sum_{i=1}^{n-1} y_i\right)$$

Формула Симпсона

$$\int_a^b f(x) = \frac{h}{3}\left[(y_0 + 4(y_1 + y_3 + \dots + y_{n-1}) + 2(y_2 + y_4 + \dots + y_{n-2}) + y_n)\right]$$

Вычислительная реализация задачи

Вычисление интеграла точно

$$\int_2^4 (2x^3 - 2x^2 + 7x - 14)dx = \left(\frac{x^4}{2} - \frac{2x^3}{3} + \frac{7x^2}{2} - 14x\right)|_2^4 = \frac{290}{3} = 96,(6)$$

Вычисление интеграла по формуле Ньютона-Котеса

$$n = 6$$

$$h = \frac{4-2}{6} = \frac{1}{3}$$

$$x = \left[2; 2\frac{1}{3}; 2\frac{2}{3}; 3; 3\frac{1}{3}; 3\frac{2}{3}; 4\right]$$

$$\int_2^4 (2x^3 - 2x^2 + 7x - 14)dx = f(2)c_6^0 + f\left(2\frac{1}{3}\right)c_6^1 + f\left(2\frac{2}{3}\right)c_6^2 + f(3)c_6^3 + f\left(3\frac{1}{3}\right)c_6^4 + f(3\frac{2}{3})c_6^5 + f(4)c_6^6 =$$

$$c_6^0 = c_6^6 = \frac{41}{420}$$

$$c_6^1 = c_6^5 = \frac{18}{35}$$

$$c_6^2 = c_6^4 = \frac{9}{140}$$

$$c_6^3 = \frac{68}{105}$$

$$= 8 * \frac{41}{420} + 16.85 * \frac{18}{35} + 28.37 * \frac{9}{140} + 43 * \frac{68}{105} + 61.18 * \frac{9}{140} + 83.37 * \frac{18}{35} + 110 * \frac{41}{420} = 96.6651(6)$$

Относительная погрешность: 0.001566

Вычисление интеграла по формуле средних прямоугольников

$$x = [2.1, 2.3, 2.5, 2.7, 2.9, 3, 3.5, 3.7, 3.9]$$

$$y = [10.4, 15.85, 22.25, 29.69, 38.26, 43, 71.75, 85.83, 101.52]$$

$$I = sum(y) * 0.2 = 96.56$$

Относительная погрешность: 0.10(6)

Вычисление интеграла по формуле трапеций

$$x = [2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4, 3.6, 3.8, 4]$$

$$y = [8.0, 13.016, 18.928, 25.832, 33.824, 43.0, 53.456, 65.288, 78.592, 93.464, 110]$$

$$I = 0.2\left(\frac{8+110}{2} + (13.016 + 18.928 + 25.832 + \ldots + 93.464)\right) = 96,88$$

Относительная погрешность: 0,22

Вычисление интеграла по формуле Симпсона

$$x = [2.0,\ 2.2,\ 2.4,\ 2.6,\ 2.8,\ 3.0,\ 3.2,\ 3.4,\ 3.6,\ 3.8]$$

$$y = [8.0,13.016,18.928,25.832,33.824,43.0,53.456,65.288,78.592,93.464]$$

$$y_1 + y_3 + .. + y_{n-1} = 240.6$$

$$y_2 + y_4 + \cdots + y_{n-2} = 184.8$$

$$I = \frac{0.2}{3}(8.0 + 4 * 240.6 + 2 * 184.8 + 93.464) = 95.5642(6)$$

Относительная погрешность: 1.1024

Листинг программы

```python
class LeftRectangles(Method):
    name = "метод левых прямоугольников"
    k=1

    Azat222
    def solve(self) -> None:
        table = PrettyTable(["N", "n", "I", "runge"])
        table.title = self.name.capitalize()
        f = self.equation.function
        h = (self.right - self.left) / self.n
        x_linspace = [self.left + h * i for i in range(self.n)]
        f_linspace = [f(i) for i in x_linspace]
        I_prev = 10000
        I = sum([y * h for y in f_linspace])
        self.n *= 2
        iter_count = 2
        while abs(self.runge(I, I_prev)) > self.accuracy:
            h = (self.right - self.left) / self.n
            x_linspace = [self.left + h * i for i in range(self.n)]
            f_linspace = [f(i) for i in x_linspace]
            I_prev = I
            I = sum([y * h for y in f_linspace])
            table.add_row([iter_count, self.n, I, self.runge(I, I_prev)])
            self.n *= 2
            iter_count += 1
        self.n //= 2
        print(table)
        print("Значение интеграла: ", I)
        print("Число разбиений:", self.n)
```

```python
class MiddleRectangles(Method):
    name = "метод средних прямоугольников"
    k=2

    Azat222 *
    def solve(self) -> None:
        table = PrettyTable(["N", "n", "I", "runge"])
        table.title = self.name.capitalize()
        f = self.equation.function
        h = (self.right - self.left) / self.n
        x_linspace = [self.left + h / 2 + h * i for i in range(self.n)]
        f_linspace = [f(i) for i in x_linspace]
        I_prev = 10000
        I = sum([y * h for y in f_linspace])
        self.n *= 2
        iter_count = 2
        while abs(self.runge(I, I_prev)) > self.accuracy:
            h = (self.right - self.left) / self.n
            x_linspace = [self.left + h / 2 * i for i in range(1, self.n + 1)]
            f_linspace = [f(i) for i in x_linspace]
            I_prev = I
            I = sum([y * h for y in f_linspace])
            table.add_row([iter_count, self.n, I, self.runge(I, I_prev)])
            self.n *= 2
            iter_count += 1
        self.n //= 2
        print(table)
        print("Значение интеграла: ", I)
        print("Число разбиений:", self.n)
```

```python
class RightRectangles(Method):
    name = "метод правых прямоугольников"
    k=1

    Azat222
    def solve(self) -> None:
        table = PrettyTable(["N", "n", "I", "runge"])
        table.title = self.name.capitalize()
        f = self.equation.function
        h = (self.right - self.left) / self.n
        x_linspace = [self.left + h + h * i for i in range(self.n)]
        f_linspace = [f(i) for i in x_linspace]
        I_prev = 10000
        I = sum([y * h for y in f_linspace])
        self.n *= 2
        iter_count = 2
        while abs(self.runge(I, I_prev)) > self.accuracy:
            h = (self.right - self.left) / self.n
            x_linspace = [self.left + h + h * i for i in range(self.n)]
            f_linspace = [f(i) for i in x_linspace]
            I_prev = I
            I = sum([y * h for y in f_linspace])
            table.add_row([iter_count, self.n, I, self.runge(I, I_prev)])
            self.n *= 2
            iter_count += 1
        self.n //= 2
        print(table)
        print("Значение интеграла: ", I)
        print("Число разбиений:", self.n)
```

```python
2 usages  Azat222 *
class Trapezoid(Method):
    name = "метод трапеций"
    k=2

    Azat222
    def solve(self) -> None:
        table = PrettyTable(["N", "n", "I", "runge"])
        table.title = self.name.capitalize()
        f = self.equation.function
        h = (self.right - self.left) / self.n
        x_linspace = [self.left + h * i for i in range(self.n)]
        f_linspace = [f(i) for i in x_linspace]
        I_prev = 10000
        I = sum([(f_linspace[i] + f_linspace[i - 1]) * h for i, y in enumerate(f_linspace[1:])]) / 2
        self.n *= 2
        iter_count = 2
        while abs(self.runge(I, I_prev)) > self.accuracy:
            h = (self.right - self.left) / self.n
            x_linspace = [self.left + h * i for i in range(self.n)]
            f_linspace = [f(i) for i in x_linspace]
            I_prev = I
            I = sum([(f_linspace[i] + f_linspace[i - 1]) * h for i, y in enumerate(f_linspace[1:])]) / 2
            table.add_row([iter_count, self.n, I, self.runge(I, I_prev)])
            self.n *= 2
            iter_count += 1
        self.n //= 2
        print(table)
        print("Значение интеграла: ", I)
        print("Число разбиений:", self.n)
```

```python
Azat222
def solve(self) -> None:
    table = PrettyTable(["N", "n", "I", "runge"])
    table.title = self.name.capitalize()
    f = self.equation.function
    h = (self.right - self.left) / self.n
    x_linspace = [self.left + h * i for i in range(self.n)]
    f_linspace = [f(i) for i in x_linspace]
    I_prev = 10000
    I = h / 3 * (
        f_linspace[0] +
        4 * sum(f_linspace[1::2]) +
        2 * sum(f_linspace[2::2]) +
        f_linspace[-1]
    )
    self.n *= 2
    iter_count = 2
    while abs(self.runge(I, I_prev)) > self.accuracy:
        h = (self.right - self.left) / self.n
        x_linspace = [self.left + h * i for i in range(self.n)]
        f_linspace = [f(i) for i in x_linspace]
        I_prev = I
        I = h / 3 * (
                f_linspace[0] +
                4 * sum(f_linspace[1::2]) +
                2 * sum(f_linspace[2::2]) +
                f_linspace[-1]
        )
        table.add_row([iter_count, self.n, I, self.runge(I, I_prev)])
        self.n *= 2
        iter_count += 1
    self.n //= 2
    print(table)
    print("Значение интеграла: ", I)
    print("Число разбиений:", self.n)
```

Результаты выполнения программы

```
Выберите уравнение:
0: sin(x) + 4
1: cos(x) - x*sin(x)
2: x^3 - 1.89x^2  - 2x + 1.76
0
Введите левый предел интегрирования 0
Введите правый предел интегрирования 2
Введите точность интегрирования 0.001
Выберите метод:
0: метод прямоугольников
1: метод трапеций
2: метод симпсона
0
+----------------------------------------------------+
|               Метод левых прямоугольников           |
+---+------+-------------------+----------------------+
| N |  n   |         I         |        runge         |
+---+------+-------------------+----------------------+
| 2 |  8   | 9.295101198880884 |  -0.13590544387280623  |
| 3 |  16  | 9.357471325805385 | -0.062370126924500724  |
| 4 |  32  |  9.38727027664563 | -0.029798950840245197  |
| 5 |  64  | 9.401823816052605 | -0.014553539406975347  |
| 6 | 128  | 9.409014138701684 | -0.007190322649078595  |
| 7 | 256  | 9.412587690570998 | -0.0035735518693140733 |
| 8 | 512  | 9.414369064286092 | -0.0017813737150937925 |
| 9 | 1024 | 9.415258400597773 | -0.0008893363116815323 |
+---+------+-------------------+----------------------+
```

```
Значение интеграла:  9.415258400597773
Число разбиений: 1024
+----------------------------------------------------+
|             Метод средних прямоугольников           |
+---+-----+-------------------+----------------------+
| N |  n  |         I         |        runge         |
+---+-----+-------------------+----------------------+
| 2 |  8  |  9.02338182008443 |  0.13587494088975363  |
| 3 |  16 | 8.971688022974924 |  0.017231265703168575 |
| 4 |  32 | 8.945616534732917 |  0.008690496080668927 |
| 5 |  64 |  8.93252466717817 |  0.004363955851582446 |
| 6 | 128 | 8.925964704041016 | 0.0021866543790511153 |
| 7 | 256 | 8.922681215226136 | 0.0010944962716266105 |
| 8 | 512 | 8.921038594012987 | 0.0005475404043829476 |
+---+-----+-------------------+----------------------+
Значение интеграла:  8.921038594012987
Число разбиений: 512
+----------------------------------------------------+
|              Метод правых прямоугольников           |
+---+------+-------------------+----------------------+
| N |  n   |         I         |        runge         |
+---+------+-------------------+----------------------+
| 2 |  8   | 9.522425555587304 |  0.09141891283361225  |
| 3 |  16  | 9.471133504158594 |  0.051292051428710295 |
| 4 |  32  | 9.444101365822235 |  0.027032138336359424 |
| 5 |  64  | 9.430239360640908 |  0.013862005181326964 |
| 6 | 128  | 9.423221910995835 |  0.00701744964507256  |
| 7 | 256  | 9.419691576718074 |  0.003530334277760616 |
```

```
| 6 | 128  | 9.423221910995835 |  0.00701744964507256  |
| 7 | 256  | 9.419691576718074 |  0.003530334277760616 |
| 8 | 512  | 9.417921007359631 | 0.0017705693584435522 |
| 9 | 1024 | 9.417034372134543 | 0.0008866352250880283 |
+---+------+-------------------+----------------------+
Значение интеграла:  9.417034372134543
Число разбиений: 1024
```

Вывод

В процессе выполнения работы я узнал про численные методы решения интегралов, написал программу использующую их и применил их самостоятельно на практике