Клепиков Глеб 215
22 декабря 2024 г.

1. Постановка задачи

Задача 3.2

Найдите приближенное значение интеграла методом трапеций, разбив интервал интегрирования на n равных частей, где $n = 16, 32, 64$.

Интеграл: $\displaystyle\int_a^b \frac{1}{(25x^2+1)\sqrt{3x-x^2}}\,dx$

Рассмотрите два отрезка интегрирования:

1. $a = 1$, $b = 2$
2. $a = 0$, $b = 3$

Сравните результаты с аналитическим значением интеграла.

Подберите более эффективный численный метод вычисления интеграла для второй задачи.

2. Методы решения

Для решения поставленной задачи была создана программа на языке программирования Python. Задача была методами:

1. Метод трапеций

$$T_n = \int\limits_a^b g_n(x)dx = \sum_{i=1}^{n} \int\limits_{x_{i-1}}^{x_i} g_n(x)dx =$$

$$= \frac{b-a}{n}\left\{\frac{1}{2}f(a) + f(x_1 + f(x_2) + \ldots + f(x_{n-1}) + \frac{1}{2}f(b)\right\}$$

$$I = \int\limits_a^b f(x)dx = T_n + \beta_n$$

2. Метод Симпсона

$$S_n = \int_a^b g_n(x)dx = \sum_{j=1}^{n/2} \int_{x_{2j-2}}^{x_{2j}} g_n(x)dx =$$

$$= \frac{b-a}{3n}\{f(a) + 4f(x_1 + 2f(x_2) + \ldots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(b)\}$$

$$I = \int_a^b f(x)dx = S_n + \gamma_n$$

В качестве критерия выбора метода была избрана точность вычисления относительно аналитического значения интеграла. Точность вычислялась по формуле Рунге: $\Delta_{2n} \approx \Theta|I_{2n} - I_n|$, где $\Theta = \frac{1}{3}$ для метода трапеций и $\Theta = \frac{1}{15}$ для метода Симпсона.

На отрезке от 0 до 3 интеграл становится интегралом 2 рода с особыми точками 0 и 3. Поэтому интеграл разделяется на 3 отрезка: $\left[0, \frac{3}{n}\right]$, $\left[\frac{3}{n}, 3 - \frac{3}{n}\right]$ и $\left[3 - \frac{3}{n}, 3\right]$. Второй отрезок вычисляется методами трапеции и Симпсона для n-2 отрезков, а 1 и 3 вычисляются отдельно. Для вычисления интеграла на отрезке $\left[0, \frac{3}{n}\right]$ используется формула Маклорена с остаточным членом в форме Лагранжа.

$$R_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!}(x - x_0)^{n+1}$$

$$\frac{1}{25x^2+1} = 1 + R_0(x)$$

Таким образом вычисляется значение всего интеграла на отрезке $\left[0, \frac{3}{n}\right]$:

$$I = 2 * arcsin(\sqrt{\frac{1}{N}})$$

а его погрешность

$$R_n(x) = \frac{\frac{90\sqrt{3}}{8} arcsin(\sqrt{\frac{1}{n}})}{n}$$

Аналогично находится значение на отрезке $\left[3 - \frac{3}{n}, 3\right]$, только вместо точки 0 в формуле Тейлора берется точка 3:

$$I = \frac{1}{226}(\pi - 2 * arcsin\sqrt{1 - \frac{1}{N}})$$

$$R_n(x) = \frac{(150 - \frac{150}{n}) * 3\, arcsin(\sqrt{1 - \frac{1}{n}})}{n * (226 - \frac{450}{n} + \frac{225}{n^2})}$$

3. Результаты вычислений

	Значение методом трапеций	Значение методом Симпсона	Погрешность методом трапеций	Погрешность методом Симпсона
$n = 16$	**0,013325**	**0,013308**	**0,00001746820**	**0,00000006368**
$n = 32$	**0,013312**	**0,013308**	**0,00000438263**	**0,00000000415**
$n = 64$	**0,013309**	**0,013308**	**0,00000109664**	**0,00000000026**

Таблица 1. Значения и погрешности методом трапеций и Симпсона на инт. (1, 2)

	Значение методом трапеций	Значение методом Симпсона	Погрешность методом трапеций	Погрешность методом Симпсона
$n = 16$	**0,668975**	**0,656013**	**0,53559999100**	**0,49344354669**
$n = 32$	**0,6165775**	**0,606427**	**0,21724861306**	**0,20135077834**
$n = 64$	**0,600067**	**0,594909**	**0,08939986101**	**0,08464570873**

Таблица 2. Значения и погрешности методом трапеций и Симпсона на инт. (0, 3)

Аналитические значения интегралов на интервалах:

Интервал $[1,\ 2]$	Интервал $[0,\ 3]$
0.013307665	0.59168705

Исходя из результатов программы видно, что метод Симпсона выдает значение интеграла более приближенное к значению, полученному аналитическим путем, чем метод трапеций на обоих интервалах и при всех n. Кроме того, метод Симпсона имеет меньшую погрешность. Таким образом, можно сделать вывод, что метод Симпсона выгоднее в данных условиях.

4. Реализация методов

```
import math
import numpy
```

```
#Функция

def f(x):
    return 1 / ((25 * x**2 + 1) * ((3 * x - x**2)**0.5))

# Метод трапеций

def trapezoid(a, b, N):
    h = (b - a) / N
    if a == 0 and b == 3:
        a += h
        b -= h
        return 2*numpy.arcsin(math.sqrt(1/N)) + (1/226)*(math.pi - 
2*numpy.arcsin(math.sqrt(1-1/N))) + trapezoid(a, b, N - 2)
    result = 0.5 * (f(a) + f(b))
    for i in range(1, N):
        x = a + i * h
        result += f(x)
    result *= h
    return result

# Метод Симпсона
def sympson(a, b, N):
    h = (b - a) / N
    if a == 0 and b == 3:
        a += h
        b -= h
        return 2*numpy.arcsin(math.sqrt(1/N)) + (1/226)*(math.pi - 
2*numpy.arcsin(math.sqrt(1-1/N))) + sympson(a, b, N-2)
    result = f(a) + f(b)
    for i in range(1, N):
        x = a + i * h
        if i % 2 != 0:
            result += 4 * f(x)
        else:
            result += 2 * f(x)
```

```
    return result * h / 3

intervals = [(1, 2), (0, 3)]
n_values = [16, 32, 64]
for a, b in intervals:
    print(f"Интеграл на отрезке [{a}, {b}]:")
    for n in n_values:
        trap_result = trapezoid(a, b, n)
        simp_result = sympson(a, b, n)
        print(f"n = {n}, Метод трапеций: {trap_result:.6f}, Метод
Симпсона: {simp_result:.6f}”)
        if a == 0:
            print("Погрешность трапеций -",
(90*math.sqrt(3)/8)*numpy.arcsin(1/math.sqrt(n))/n + ((150 - 150/n)/(226 -
450/n + 225/(n**2)))*3/n * numpy.arcsin(math.sqrt(1 - 1/n)) + abs(trapezoid(a,
b, n-2)-trapezoid(a, b, (n-2)//2))/3)
            print("Погрешность Симпсона -",
(90*math.sqrt(3)/8)*numpy.arcsin(1/math.sqrt(n))/n + ((150 - 150/n)/(226 -
450/n + 225/(n**2)))*3/n * numpy.arcsin(math.sqrt(1 - 1/n)) + abs(sympson(a,
b, n-2)-sympson(a, b, (n-2)//2))/15)
        else:
            print("Погрешность трапеций -", (abs(trapezoid(a, b,
n)-trapezoid(a, b, n//2))/3))
            print("Погрешность Симпсона -", (abs(sympson(a, b,
n)-sympson(a, b, n//2))/15))
print()
```

5. Вывод

Точность методов: Проведенные расчеты показали, что метод Симпсона обеспечивает более высокую точность при вычислении определенного интеграла по сравнению с методом трапеций, особенно при меньшем количестве разбиений.

Зависимость от разбиений: Для обоих методов точность увеличивается с ростом количества разбиений интервала, что подтверждает их сходимость к точному значению интеграла.

Простота реализации: Метод трапеций проще в реализации, но требует большего числа разбиений для достижения приемлемой точности.

Выбор метода: Метод Симпсона предпочтителен в задачах, где требуется высокая точность при ограниченных вычислительных ресурсах.

Практическая значимость: Полученные результаты могут быть полезны для выбора оптимального численного метода в зависимости от требований к точности и доступных ресурсов.