Московский авиационный институт
(национальный исследовательский университет)

Институт компьютерных наук и прикладной математики

Кафедра вычислительной математики и программирования

Курсовой проект по курсу «Численные методы»
«Вычисление несобственных интегралов численными методами»

Студент: М. И. Меджидли
Преподаватель: Д.Е. Пивоваров
Группа: М8О-408Б-20
Дата: 11.02.2024

Москва, 2024

1 Теоретические сведения

Определённый интеграл называется несобственным, если выполняется по крайней мере одно из следующих условий.

1. Область интегрирования является бесконечной - интеграл 1 рода.

2. Подынтегральная функция является неограниченной в окрестности некоторых точек области интегрирования - интеграл 2 рода.

Несобственный интеграл 2 рода можно свести к интегралу 1 рода с помощью замены переменной. Поэтому в данной работе будем рассматривать несобственные интегралы 1 рода

1. Сведение к определенному интегралу

Рассмотрим преобразование из мат анализа, выполненное с помощью замены переменной:

$\int\limits_a^b f(x)dx = \int\limits_{1/b}^{1/a} \frac{1}{t^2} f(\frac{1}{t})dt$ при $ab > 0$

Можем разложить несобственный интеграл на сумму интегралов.

$\int\limits_{-\infty}^{+\infty} f(x)dx = \int\limits_{-\infty}^{-A} f(x)dx + \int\limits_{-A}^{B} f(x)dx + \int\limits_{B}^{+\infty} f(x)dx$ при $-A < 0$ и $B > 0$

Первый и последний интегралы можем преобразовать с помощью формулы выше. Так мы можем посчитать каждый из этих трех интегралов (например, методом прямоугольников) и сложить получившиеся результаты.

2. Предельный переход

Запишем предельный переход для несобственного интеграла 1 рода:

$\int\limits_a^{+\infty} f(x)dx = \lim\limits_{b->\infty} \int\limits_a^b f(x)dx$

Будем вычислять правый интеграл (например, методом прямоугольников) до тех пор, пока следующее слагаемое не станет меньше заданного эпсилон.

2 Исходный код

```
INF = 1e10


def f(x):
    """
    Function to integrate
    """
    return 1. / (1 + x**2)


def integrate_rectangle_method(f, l, r, h):
    """
    Stolen from lab 3.5
    Calculate integral f(x)dx at interval [l; r] using rectangle method with step=h
    """
    result = 0
    cur_x = l
    while cur_x < r:
        result += h * f((cur_x + cur_x + h) * 0.5)
        cur_x += h
    return result


def integrate_with_definite_integral(f, l, r, h=0.01, eps=1e-6):
    """
    Calculate improper integral (type 1) transforming to definite integrals
    """

    def f_new(t):
        return (1. / t ** 2) * f(1. / t)

    result = 0
    if r == INF:
        new_r = max(eps, l)
        result += integrate_rectangle_method(f_new, eps, 1. / new_r - eps, h)
    else:
        new_r = r
    if l == -INF:
        new_l = min(-eps, r)
        result += integrate_rectangle_method(f_new, 1. / new_l + eps, -eps, h)
    else:
        new_l = l
    if new_l < new_r:
        result += integrate_rectangle_method(f, new_l, new_r, h)
    return result
```

```
def integrate_lim(f, l, r, h=0.1, eps=1e-6):
    """
    Calculate improper integral f(x)dx (type 1) using limit transition.
    Returns: integral result, number of iterations
    """
    result = 0
    iters = 0
    if r == INF:
        finish = False
        cur_x = max(l, 0)
        while not finish:
            iters += 1
            new_result = result + h * f((cur_x + cur_x + h) * 0.5)
            cur_x += h
            if abs(new_result - result) < eps:
                finish = True
            result = new_result
    else:
        result += integrate_rectangle_method(f, 0, r, h)
    if l == -INF:
        finish = False
        cur_x = min(0, r)
        while not finish:
            iters += 1
            new_result = result + h * f((cur_x - h + cur_x) * 0.5)
            cur_x -= h
            if abs(new_result - result) < eps:
                finish = True
            result = new_result
    else:
        result += integrate_rectangle_method(f, l, 0, h)
    return result, iters


if __name__ == '__main__':
    a = -INF
    b = INF
    h = 0.1
    eps = 1e-3
    print('Transforming to definite intrgral')
    res_definite = integrate_with_definite_integral(f, a, b, h, eps)
    print('Integral =', res_definite)
    print()

    print('Limit method')
    res_limit, iters_limit = integrate_lim(f, a, b, h, eps)
    print('Integral =', res_limit)
    print('Iterations:', iters_limit)
    print()
```

3 Результат работы программы

Для примера будем вычислять следующий интеграл: $\int\limits_l^r \frac{1}{1+x^2}$

1. l = 3, r = ∞

```
Transforming to definite integral
Integral = 0.3277407823690935

Limit method
Integral = 0.32075031473059007
Iterations: 99701
```

2. l = -∞, r = -9

```
Transforming to definite integral
Integral = 0.11954685365990542

Limit method
Integral = 0.10965722035305618
Iterations: 99101
```

3. l = -∞, r = 10

```
Transforming to definite integral
Integral = 3.08832958701484

Limit method
Integral = 3.042588970590539
Iterations: 31624
```

4. l = -∞, r = ∞

```
Transforming to definite integral
Integral = 3.2867676296096793

Limit method
Integral = 3.140960222545892
Iterations: 63248
```

4 Выводы

Выполнив данную работу, я познакомился с численными методами решения несобственных интегралов:

1. Сведение к сумме определенных интегралов

2. Предельный переход

Я реализовал два этих метода и протестировал их работу на разных функциях с разными пределами интегрирования. Полученные значения были довольно близки к ответам, полученным аналитически.