**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Информационные технологии и прикладная математика»

**Курсовая работа по дисциплине «Численные методы»**

**По теме  
«Вычисление несобственных интегралов численными методами»**

Студент: Бухарин А.И.

Группа: М80-402Б-20

Преподаватель: Пивоваров Д.Е.

Оценка:

Дата:

Москва, 2024

1. **Теоретические сведения**

Определение несобственного интеграла включает в себя выполнение хотя бы одного из следующих условий:

1. Область интегрирования бесконечна, что соответствует интегралу первого рода.
2. Подынтегральная функция становится неограниченной в некоторых точках области интегрирования, что определяет интеграл второго рода.

В некоторых случаях интеграл второго рода можно свести к интегралу первого рода с помощью замены переменной. Поэтому в этом контексте мы будем рассматривать несобственные интегралы первого рода.

*Сведение к определенному интегралу*

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

Можем разложить несобственный интеграл на сумму интегралов:

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.

*Предельный переход*

Запишем предельный переход для несобственного интеграла 1 рода:

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

1. **Листинг кода**

INF = 1e9

def f(x):

"""

Подинтегральная функция

"""

return 1 / (1 + x\*\*2)

def integrate\_rectangle\_method(f, l, r, h):

"""

Расчет интеграла f(x)dx на интервале [l; r] используя метод прямоугольников с шагом 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):

"""

Расчет несобственного интеграла первого типа методом перехода к определенному интегралу

"""

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):

"""

Расчет несобственного интеграла первого типа методом перехода к пределу

"""

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 = 3

b = INF

h = 0.1

eps = 1e-9

print('Переход к определенному интегралу')

res\_definite = integrate\_with\_definite\_integral(f, a, b, h, eps)

print('Интеграл =', res\_definite)

print()

print('Предельный метод')

res\_limit, iters\_limit = integrate\_lim(f, a, b, h, eps)

print('Интеграл =', res\_limit)

print('Итерации:', iters\_limit)

print()

1. **Результат работы программы**

Для примера будем вычислять следующий интеграл:

|  |  |  |
| --- | --- | --- |
|  |  | Результат |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Выводы**

Выполнив данную работу, я изучил численные методы для решения несобственных интегралов. В частности, я ознакомился с двумя методами:

1. Приведение к сумме определенных интегралов.
2. Использование предельного перехода.

Я реализовал оба этих метода и проверил их работоспособность на различных функциях с различными пределами интегрирования. Полученные численные значения были в значительной мере согласованы с аналитическими результатами.