БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Отчет

Методы численного анализа

Лабораторная работа 4

Выполнила

Юрковская Екатерина Артуровна

Студентка 2 курса 3 группы

Минск 2020

1) Постановка задачи.

Постановка задачи. Для указанного вида интеграла и указанной подынтегральной функции:

- Используя базовую квадратурную формулу из варианта, построить составную квадратурную формулу и включить ее в отчет.
- Применить полученную ранее составную квадратурную формулу для вычисления указанного в варианте интеграла с точностью $\epsilon = 10^{-4}$, $\epsilon = 10^{-6}$, $\epsilon = 10^{-8}$. Для оценки погрешности воспользоваться правилом Рунге. В каждом эксперименте следует подсчитать количество вычислений подынтегральной функции и включить в отчет.
- Используя какую-либо систему компьютерной алгебры (Wolphram Mathematica, Maple, MatLab) произвести вычисление указанного в варианте интеграла и сравнить с ранее полученным. Достигнута ли требуемая точность?

- Для указанного типа интегралов построить квадратурную формулу наивысшей алгебраической степени точности (НАСТ) с указанным количеством узлов и включить ее в отчет.
- Используя ранее полученную формулу НАСТ произвести вычисление интеграла из варианта.
- Результаты эксперимента оформить в таблицу:

Тип квадратурной формулы	Требуемая точность, испольуемая в правиле Рунге	Достигнутая точность	Количество вычислений подынтегральной функции
Составная	10^{-4}	...	...
Составная	10^{-6}	...	...
Составная	10^{-8}	...	...
НАСТ	–	...	...

Примечание:

12.	$(1-x)^3(1+x)^2$	$x^2cos(5-sin(e^{x^5}))$	[-1, 1]	Правых прямо-угольников	5

2) Теоретические сведения

Метод прямоугольников — метод численного интегрирования функции одной переменной, заключающийся в замене подынтегральной функции на многочлен нулевой степени, то есть константу, на каждом элементарном отрезке. Если рассмотреть график подынтегральной функции, то метод будет заключаться в приближённом вычислении площади под графиком суммированием площадей конечного числа прямоугольников, ширина которых будет определяться расстоянием между соответствующими соседними узлами интегрирования, а высота — значением подынтегральной функции в этих узлах. Алгебраический порядок точности равен 0. (Для формулы средних прямоугольников равен 1).

Если отрезок [a , b] является элементарным и не подвергается дальнейшему разбиению, значение интеграла можно найти по

2. **Формуле правых прямоугольников**: $\int_a^b f(x)\,dx \approx f(b)(b-a).$

В случае разбиения отрезка интегрирования на n элементарных отрезков приведённые выше формулы применяются на каждом из этих элементарных отрезков между двумя соседними узлами. В результате, получаются составные квадратурные формулы:

2. Для **правых прямоугольников**: $\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} f(x_i)(x_i - x_{i-1}).$

[1]

Для получения коэффициентов A_i, x_i КФ НАСТ воспользуемся Wolfram Mathematica (файл прилагается).

3) Полученные коэффициенты.

$$A_0 \to 0.027410178066337099441770 49,$$

$$x_0 \to 0.6904577501267610633887031,$$

$$A_1 \to 0.2129178606036482785254871,$$

$$x_1 \to 0.3265199313490006526499769,$$

$$A_2 \to 0.4390843794439508053922088,$$
$$x_2 \to -0.08233784955203490532166915,$$
$$A_3 \to 0.3222065654722182153781323,$$
$$x_3 \to -0.4751788706128316398343835,$$
$$A_4 \to 0.06504768308051226792906797,$$
$$x_4 \to -0.7927942946442285042159607$$

4) Исходный код программы.

```
import numpy as np
from math import *

I = -0.0738937049447472783 #Значение полученное через Wolfram
a=-1
b=1
global iterations

#подынтегральная ф-ия
def f(x):
     return cos(5-sin(exp(x**5)))* x**2

#весовая функция
def p(x):
    return (1-x)**3 * (1+x)**2


#формула правых прямоугольников
def right_rect(h):
    sum = 0
    for x in np.arange(a+h/2, b , h):
        sum +=  f(x) * p(x) * h
        global iterations
        iterations += 1
    return sum


def cacl_integral(eps):
    global iterations
    iterations=0

    prev = 0
    cur = right_rect(0.9)
    step = 1
    runge_delta = abs(cur - prev) / 3
    while (runge_delta > eps):
        step = step / 2
        prev = cur
        cur = right_rect(step)
        runge_delta = abs(cur - prev) / 3
    return cur

eps=10**-4
print('погрешность при eps = 10**-4 ')
print(abs(cacl_integral(eps)-I))
print(iterations)
```

```
eps=10**-6
print('погрешность при eps = 10**-6 ')
print(abs(cacl_integral(eps)-I))
print(iterations)

eps=10**-8
print('погрешность при eps = 10**-8 ')
print(abs(cacl_integral(eps)-I))
print(iterations)

A=[ 0.02741017806633709944177049, 0.2129178606036482785254871,
   0.4390843794439508053922088,0.3222065654722182153781323, 0.06504768308051226792906797,]
x=[0.6904577501267610633887031, 0.3265199313490006526499770,
-0.08233784955203490532166915, -0.4751788706128316398343835, -0.7927942946442285042159607]

Q=0
iterations=0
for i in range (5):
    Q+=f(x[i])*A[i]
    iterations+=1

print('погрешность НАСТ = ')
print(abs(Q-I))
print(iterations)
```

5) Таблица

Тип кф	Требуемая точность	Достигнутая точность	Число вычислений подынтегральной функции
Составная	10**-4	1.4923404591110634e-06	62
Составная	10**-6	8.678746075962973e-08	126
Составная	10**-8	3.311191598687202e-10	510
НАСТ	-	2.6233215295184964e-06	5

6) Список использованной литературы

1) https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BF%D1%80%D1%8F%D0%BC%D0%BE%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2