**Министерство науки и высшего образования РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ИС**

отчет

**по лабораторной работе №5**

**по дисциплине «Конструирование программ»**

Тема: Численное дифференцирование и численное интегрирование функций

|  |  |  |
| --- | --- | --- |
| Студент гр. 8363 |  | Нерсисян А.С. |
| Преподаватель |  | Копыльцов А.В. |

Санкт-Петербург

2020

**Цель работы.**

Заработать навыки численного дифференцирования, в тех случаях, когда либо невозможно, либо очень сложно или дорого продифференцировать функцию аналитически, а также повышения точности вычислений, аппроксимации исходной функции какой-либо гладкой функцией, значение производной от которой принимают за приближенное значение искомой производной.

**Основные теоретические положения.**

* 1. **Простейшие формулы численного дифференцирования для первой производной**

Из определения первой производной  естественно использовать для ее вычисления две простейшие приближенные формулы

,







α+

α-



  

,

соответствующие выбору фиксированных значений  и . Здесь  - малый пара-  
метр - шаг. Формулы и называют правой и левой разностными производными. Оценим их погрешности:

 и , воспользовавшись формулой Тейлора:



Подставив в  выражение, получим

Аналогично,  Таким образом,



Итак, формулы (4.1.1) и (4.1.2) имеют первый порядок точности по . Геометрическая интерпретация этих формул показана на предыдущем рисунке. Естественно предположить, что лучшим по сравнению с (4.1.1) и (4.1.2) приближением  является тангенс угла наклона  секущей к графику , проведенной через точки  и . Соответствующая формула приближения имеет вид



, полученную по формуле, называют центральной разностной производной. Оценим опять погрешность формулы (4.1.5). Для этого подставим в выражение для погрешности  соответствующие разложения в ряд Тейлора:





 





  

 Получим



Следовательно, справедлива оценка погрешности



Таким образом, центральная разностная производная аппроксимирует производную  со вторым порядком точности относительно параметра .

Для вычисления первой производной можно получить и еще более сложные и точные формулы. Однако в таких формулах с ростом порядка точности возрастает и число используемых значений функции. Например,



## Формулы численного дифференцирования для второй производной

Наиболее простой и широко применяемой для приближенных вычислений второй производной является следующая формула



Она выводится из формулы , в которой первые производные рассчитываются по формуле (4.2.1) по трем точкам . Формулу (4.2.1) часто называют второй разностной производной. Покажем, что она имеет второй порядок точности относительно . Итак, 

причем 

Тогда

Следовательно, 

Для получения  можно использовать формулы любого порядка точности. Например, формула



имеет четвертый порядок точности относительно параметра , но требует наличия значений функции в пяти точках.

## 1.3. Формулы численного дифференцирования, основанные на интерполяции алгебраическими многочленами

Предположим, что в окрестности точки  функция  аппроксимируется некоторой другой функцией , причем  в точке  легко вычисляется. Естественно в такой ситуации попытаться воспользоваться приближенной формулой 

Пусть  - интерполяционный многочлен степени  с узлами интерполяции . В этом случае  Поскольку



то для аппроксимации производных в общем случае при наличии неравномерной сетки узлов можно воспользоваться связью производных и разделенных разностей:



Формула имеет по крайней мере первый порядок точности. В частности при   - первая разностная производная; при   - вторая разностная производная.

Если шаг сетки узлов постоянен, то можно вместо разделенных разностей использовать конечные:



## 1.4. Обусловленность формул численного дифференцирования

Несмотря на внешнюю простоту формул численного дифференцирования, их применение требует особой осторожности. Так как табличные значения  функции  непременно содержат ошибки, и эти ошибки являются неустранимыми, они прибавляются к погрешностям аппроксимации. Для уменьшения этой погрешности обычно уменьшают шаг , но именно при малых шагах формулы численного дифференцирования становятся плохо обусловленными и результат их применения может быть полностью искажен неустранимой ошибкой. Важно понимать, что действительная причина этого явления лежит не в несовершенстве методов вычисления производных, а в некорректности самой операции дифференцирования приближенно заданных функций.

Рассмотрим  Это полная погрешность, она складывается из погрешности аппроксимации  и неустрани-  
мой погрешности  Пусть  Тогда  можно оценить следующим образом:  Фактически это число будет числом обусловленности формулы (4.1.1), то есть



Видно, что при  Поэтому, несмотря на то, что погрешность аппроксимации стремится к нулю при , полная погрешность  будет неограниченно возрастать. Найдем , при котором  Для этого необходимо, чтобы  Отсюда









.

Тогда 

Таким образом, при использовании формул численного дифференцирования необходимо обращать внимание на выбор шага  Даже при оптимальном выборе шага полная погрешность является величиной, пропорциональной  При  формулы для вычисления  обладают еще большей чувствительностью к ошибкам задания функций. Поэтому значения производных высокого порядка, найденные с помощью таких формул, могут быть очень неточными.

**Экспериментальные результаты.**

**Задание № 1**

Для данных функций построить правую, левую и центральную первые разностные производные, вторую разностную производную на указанном интервале с данными шагами сетки и сравнить полученные значения с точными значениями производных:

**Дано:** Вариант 11

**Задание № 2.**

Найти численное значение интеграла от функций, указанных в задании № 1, на заданном промежутке. При вычислении по подпрограмме parab выбрать , соответствующее меньшему значению шага.

**Обработка результатов эксперимента.**

**Задание № 1. решение:**

#include <iostream>

#include <conio.h>

using namespace std;

double arcsin(double x)

{

double result = x, prev, temp, a = x;

int i = 1;

do

{

prev = result;

temp = 1;

a \*= x\*x;

for (int j = 1; j <= i; ++j)

temp = temp \* ((2 \* j - 1) \* (2 \* j) )/ (4 \* j \* j);

temp /= 2 \* i + 1;

result += temp \* a;

++i;

} while ((result - prev) > 0.00001);

return result;

}

double func(double x)

{

return arcsin(x) / sqrt(1 - x\*x);

}

int main()

{

setlocale(LC\_ALL, "rus");

double x1, x2, h1, h2, x;

do

{

cout << "x1 = ";

cin >> x1;

cout << "x2 = ";

cin >> x2;

cout << "h1 = ";

cin >> h1;

cout << "h2 = ";

cin >> h2;

cout << "Задание 1" << endl;

cout << "Шаг равен " << h1 << endl;

x = x1;

do

{

cout << "x = " << x << "; f'r = " << (func(x + h1) - func(x)) / h1 << "; f'l = " << (func(x) - func(x - h1)) / h1 << "; f' = " << (func(x + h1) - func(x - h1)) / (h1 + h1) << "; f'' = " << (func(x - h1) - 2 \* func(x) + func(x + h1)) / (h1 \* h1) << endl;

x += h1;

} while (x <= x2);

cout << "Шаг равен " << h2 << endl;

x = x1;

do

{

cout << "x = " << x << "; f'r = " << (func(x + h2) - func(x)) / h2 << "; f'l = " << (func(x) - func(x - h2)) / h2 << "; f' = " << (func(x + h2) - func(x - h2)) / (h2 + h2) << "; f'' = " << (func(x - h2) - 2 \* func(x) + func(x + h2)) / (h2 \* h2) << endl;

x += h2;

} while (x <= x2);

cout << "Задание 2" << endl;

double result = 0;

x = x1;

do

{

result += func(x + h2 / 2);

x += h2;

} while (x < x2);

cout << "Формула центральных прямоугольников: Интеграл равен " << result \* h2 << endl;

result = func(x1) / 2;

x = x1 + h2;

do

{

result += func(x);

x += h2;

} while (x < x2 - h2);

result += func(x) / 2;

cout << "Формула трапеций: Интеграл равен " << result \* h2 << endl;

result = func(x1);

x = x1 + h2;

do

{

result += 4 \* func(x - h2 / 2) + 2 \* func(x);

x += h2;

} while (x < x2 - h2);

result += func(x) + 4 \* func(x - h2 / 2);

cout << "Формула парабол: Интеграл равен " << result \* h2 / 6 << endl;

x = \_getch();

} while (x != 27);

return 0;

}

**Выводы.**

В ходе выполнения данной лабораторной былo выпполненo численнoe дифференцированиe, в тех случаях, когда либо невозможно, либо очень сложно или дорого продифференцировать функцию аналитически, а также для повышения точности вычислений, аппроксимации исходной функции какой-либо гладкой функцией, значение производной от которой принимались за приближенное значение искомой производной.