Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

ОТЧЕТ

по лабораторной работе № 6

Название: Построение и программная реализация алгоритмов численного дифференцирования

Дисциплина: Вычислительные алгоритмы

Студент ИУ7И - 46Б ____________ Андрич К.
(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель ____________ В.М. Градов
(Подпись, дата) (И.О. Фамилия)

Москва, 2021

Цель работы

Получение навыков построения алгоритма вычисления производных от сеточных функций.

Задание

Задана табличная (сеточная) функция. Имеется информация, что закономерность, представленная этой таблицей, может быть описана формулой

$$y = \frac{a_0 x}{a_1 + a_2 x},$$

параметры функции неизвестны и определять их не нужно.

x	y	1	2	3	4	5
1	0.571					
2	0.889					
3	1.091					
4	1.231					
5	1.333					
6	1.412					

Вычислить первые разностные производные от функции и занести их в столбцы (1)-(4) таблицы:

1 - односторонняя разностная производная,

2 - центральная разностная производная,

3 - 2-я формула Рунге с использованием односторонней производной,

4 - введены выравнивающие переменные.

В столбец 5 занести вторую разностную производную.

Код программы

В программе есть 4 файлов: 2 заголовочных файла (functions.h и errors.h) и 2 файла кода в СИ (main.c, functions.c)

errors.h

```
#ifndef ERRORS_H
#define ERRORS_H

#define OK 0
#define NO_VALUE -1

#endif //ERRORS_H
```

functions.h

```
#ifndef FUNCTIONS_H
#define FUNCTIONS_H

#include <stdio.h>
#include <stdlib.h>

#define N 6

typedef struct
{
	float y1;
	float y2;
	float y3;
	float y4;
	float y5;
} struct_t;

float left_diffder(float Y[N], int step, int inx);
float center_diffder(float Y[N], int step, int inx);
float runge_second(float Y[N], int step, int inx);
float alignment_vars(float Y[N], float X[N], int inx);
float second_diffder(float Y[N], int step, int inx);
void solve(struct_t results[N], float X[N], float Y[N]);
void check_print(float x);
void print(struct_t results[N], float Y[N]);

#endif //FUNCTIONS_H
```

functions.c

```c
#include "functions.h"
#include "errors.h"

float left_diffder(float Y[N], int step, int inx)
{
    float result;
    if (inx > 0 && inx < N)
    {
        result = (Y[inx] - Y[inx - 1]) / step;
        return result;
    }
    else
        return NO_VALUE;
}

float center_diffder(float Y[N], int step, int inx)
{
    float result;
    if (inx > 0 && inx < N - 1)
    {
        result = (Y[inx + 1] - Y[inx - 1]) / (2 * step);
        return result;
    }
    else
        return NO_VALUE;
}

float runge_second(float Y[N], int step, int inx)
{
    float result, y1, y2;
    if (inx >= 2)
    {
        y1 = left_diffder(Y, step, inx);
        y2 = (Y[inx] - Y[inx - 2]) / (2 * step);
        result = 2 * y1 - y2;
        return result;
    }
    else
        return NO_VALUE;
}

float alignment_vars(float Y[N], float X[N], int inx)
{
```

```c
    float result, d;
    if (inx <= N - 2)
    {
        d = ((1 / Y[inx + 1]) - (1 / Y[inx])) / ((1 / X[inx + 1]) - (1 /
X[inx]));
        result = (d * (Y[inx] * Y[inx])) / (X[inx] * X[inx]);
        return result;
    }
    else
        return NO_VALUE;
}

float second_diffder(float Y[N], int step, int inx)
{
    float result;
    if (inx > 0 && inx < N - 1)
    {
        result = (Y[inx - 1] - 2 * Y[inx] + Y[inx + 1]) / (step * step);
        return result;
    }
    else
        return NO_VALUE;
}

void solve(struct_t results[N], float X[N], float Y[N])
{
    int step = 1;
    for (int i = 0; i < N; i++)
    {
        results[i].y1 = left_diffder(Y, step, i);
        results[i].y2 = center_diffder(Y, step, i);
        results[i].y3 = runge_second(Y, step, i);
        results[i].y4 = alignment_vars(Y, X, i);
        results[i].y5 = second_diffder(Y, step, i);
    }
}

void check_print(float x)
{
    if (x == NO_VALUE)
        printf("| %11s ", "-");
    else
        printf("| %11f ", x);
```

```c
}

void print(struct_t results[N], float Y[N])
{
    printf("| %11s ", "X");
    printf("| %11s ", "Y");
    for (int i = 0; i < N - 1; i++)
        printf("| %11d ", i + 1);
    printf("|\n");
    for (int i = 0; i < N; i++)
    {
        printf("| %11d ", i + 1);
        printf("| %11f ", Y[i]);
        check_print(results[i].y1);
        check_print(results[i].y2);
        check_print(results[i].y3);
        check_print(results[i].y4);
        check_print(results[i].y5);
        printf("|\n");
    }
}
```

main.c

```c
#include "functions.h"
#include "errors.h"

int main(void)
{
    float X[N] = {1.00, 2.00, 3.00, 4.00, 5.00, 6.00};
    float Y[N] = {0.571, 0.889, 1.091, 1.231, 1.333, 1.412};
    struct_t results[N];
    solve(results, X, Y);
    print(results, Y);
    return OK;
}
```

Результаты работы

```
katarina@LAPTOP-I1VEUM2H:/mnt/c/Users/katar/Desktop/MSTU Bauman/main/Algorithm/LR6$ ./app.exe
|           X |           Y |           1 |           2 |           3 |           4 |           5 |
|           1 |    0.571000 |           - |           - |           - |    0.408499 |           - |
|           2 |    0.889000 |    0.318000 |    0.260000 |           - |    0.246899 |   -0.116000 |
|           3 |    1.091000 |    0.202000 |    0.171000 |    0.144000 |    0.165437 |   -0.062000 |
|           4 |    1.231000 |    0.140000 |    0.121000 |    0.109000 |    0.117744 |   -0.038000 |
|           5 |    1.333000 |    0.102000 |    0.090500 |    0.083000 |    0.089496 |   -0.023000 |
|           6 |    1.412000 |    0.079000 |           - |    0.067500 |           - |           - |
```

X	Y	1	2	3	4	5
1	0.571	-	-	-	0.408499	-
2	0.889	0.318000	0.260000	-	0.2468899	-0.11600
3	1.091	0.202000	0.171000	0.144000	0.165437	-0.06200
4	1.231	0.140000	0.121000	0.109000.	0.117744	-0.038000
5	1.333	0.102000	0.090500	0.083000	0.089496	-0.023000
6	1.412	0.079000	-	0.067500	-	-

Левая разностная производная с точностью O(h)

$$y'_n = \frac{y_n - y_{n-1}}{h} + O(h).$$

Не возможно вычислить первое значение

Центральная разностная производная с точностью O(h^2)

$$y'_n = \frac{y_{n+1} - y_{n-1}}{2h} + O(h^2).$$

Не возможно вычислить первое и шестое значение

2ая формула Рунге с точностью O(h^2)

$$y'_0 = \frac{-3y_0 + 4y_1 - y_2}{2h} + O(h^2).$$

Не возможно вычислить первое и второе значение

Выравнивающие переменные

$$y_x{}' = \frac{\eta_\xi{}' \xi_x{}'}{\eta_y{}'} = \frac{\eta_\xi{}' y^2}{x^2}$$

Не возможно вычислить шестое значение

2ая разностная производная с точностью O(h^2)

$$y''_n = \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} + O(h^2)$$

Не возможно вычислить первое и шестое значение

Вопросы при защите лабораторной работы

1. *Получить формулу порядка точности $O(h^2)$ для первой разностной производной N y' в крайнем правом узле x_N.*

$$y_{N-1} = y_N - \frac{h}{1!}y'_N + \frac{h^2}{2!}y''_N - \frac{h^3}{3!}y'''_N + \frac{h^4}{4!}y^{IV}_N - \ldots$$

$$y_{N-2} = y_N - \frac{2h}{1!}y'_N + \frac{4h^2}{2!}y''_N - \frac{8h^3}{3!}y'''_N + \frac{16h^4}{4!}y^{IV}_N - \ldots$$

$$2h^2 y''_N = y_{N-2} - y_N + 2hy'_N$$

$$y''_N = \frac{y_{N-2} - y_N + 2hy'_N}{2h^2}$$

$$y_{N-1} = y_N - hy'_N + \frac{h^2}{2}\,\frac{y_{N-2} - y_N + 2hy'_N}{2h^2}$$

$$y_{N-1} = y_N - hy_N + \frac{y_{N-2} - y_N + 2hy'_N}{4}$$

$$y_{N-1} = \frac{4y_N - 4hy'_N + y_{N-2} - y_N + 2hy'_N}{4}$$

$$4y_{N-1} = 3y_N - 2hy'_N + y_{N-2}$$

$$y'_N = \frac{3y_N - 4y_{N-1} + y_{N-2}}{2h} + O(h^2)$$

2. *Получить формулу порядка точности $O(h^2)$ для второй разностной производной y_0'' в крайнем левом узле x_0.*

$$y_1 = y_0 + \frac{hy_0'}{1!} + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' + \frac{h^4}{4!}y_0^{IV} + \dots$$

$$y_2 = y_0 + \frac{2hy_0'}{1!} + \frac{4h^2}{2!}y_0'' + \frac{8h^3}{3!}y_0''' + \frac{16h^4}{4!}y_0^{IV} + \dots$$

$$2h^2 y_0'' = y_2 - 2hy_0$$

$$y_0'' = \frac{y_2 - 2hy_0}{2h^2}$$

$$hy_0' = y_1 - y_0 - \frac{h^2}{2}y_0''$$

$$y_0' = \frac{y_1 - y_0 - \frac{h^2}{2}y_0''}{h}$$

$$y_0'' = \frac{y_2 - 2h\,\frac{y_1 - y_0 - \frac{h^2}{2}y_0''}{h}}{2h^2}$$

$$y_0'' = \frac{y_2 - 2\left(y_1 - y_0 - \frac{h^2}{2}y_0''\right)}{2h^2}$$

$$y_0'' = \frac{y_2 - 2y_1 + 2y_0 + h^2 y_0''}{2h^2}$$

$$y_0'' - \frac{y_0''}{2} = \frac{y_2 - 2y_1 + 2y_0}{2h^2}$$

$$y_0'' = \frac{y_2 - 2y_1 + 2y_0}{h^2} + O(h^2)$$

3. *Используя 2-ую формулу Рунге, дать вывод выражения (9) из Лекции №7 для первой производной y_0' в левом крайнем узле*

$$y'_0 = \frac{-3y_0 + 4y_1 - y_2}{2h} + O(h^2).$$

$$\Omega = \phi(h) + \frac{\phi(h) - \phi(mh)}{m^p - 1} + O(h^{p+1})$$

$$m = 2,\ p = 1$$

$$\phi(h) + \phi(h) - \phi(2h) + O(h^2) = 2\phi(h) - \phi(2h) + O(h^2)$$

$$2\left(\frac{y_1 - y_0}{h} - \frac{y_0}{2} y_0''\right) - \left(\frac{y_2 - y_0}{2h} - h y_0''\right) + O(h^2) = \frac{-3y_0 + 4y_1 - y_2}{2h} + O(h^2)$$

4. *Любым способом из Лекций №7, 8 получить формулу порядка точности $O(h^3)$ для первой разностной производной y_0' в крайнем левом узле x_0.*

$2h$

$$y_1 = y_0 + \frac{h}{1!} y_0' + \frac{h^2}{2!} y_0'' + \frac{h^3}{3!} y_0''' + \frac{h^4}{4!} y^{IV} + \ldots$$

$$y_2 = y_0 + \frac{2h}{1!} y_0' + \frac{4h^2}{2!} y_0'' + \frac{8h^3}{3!} y_0''' + \frac{16h^4}{4!} y^{IV} + \ldots$$

$$y_3 = y_0 + \frac{3h}{1!} y_0' + \frac{9h^2}{2!} y_0'' + \frac{27h^3}{3!} y_0''' + \frac{81h^4}{4!} y_0^{IV} + \ldots$$

$$y_0' = \frac{y_1 - y_0 - \frac{h^2}{2} y_0'' - \frac{h^3}{6} y_0'''}{h}$$

$$y_0'' = \frac{y_2 - y_0 - 2h y_0' - \frac{8h^3}{6} y_0'''}{4h^2}$$

$$y_0' = \frac{4y_1 - 3y_0 - y_2}{2h} + \frac{h^2}{3} y_0'''$$

$$y_0''' = \frac{y_3 - y_0 - 3h y_0' - \frac{9}{2} h^2 y_0''}{27h^3}$$

$$y_0' = \frac{4y_3 - 27y_2 + 108y_1 - 85y_0}{66h} + O(h^3)$$