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)

Вычислительные алгоритмы.
Лабораторная работа №6.
«Построение и программная реализация алгоритмов численного дифференцирования»

Студент **Трошкин Николай Романович**

Группа **ИУ7-46Б**

Студент ____________________ Трошкин Н.Р.
подпись, дата *фамилия, и.о.*

Преподаватель ____________________ Градов В.М.
подпись, дата *фамилия, и.о.*

2021 г.

Цель работы

Получение навыков построения алгоритма вычисления производных от сеточных функций.

Задание

Исходные данные

Табличная функция:

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					

Таблица представляет закономерность: $y = \frac{a_0 x}{a_1 + a_2 x}$, где a_0, a_1, a_2 – некоторые коэффициенты, не подлежащие определению.

Требуемый результат

Заполнить таблицу производными заданной функции, вычисленными различными методами:

1 - первая односторонняя разностная производная,

2 - первая центральная разностная производная,

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

4 - с использованием выравнивающих переменных,

5 - вторая разностная производная.

Полученный результат

x	y	1	2	3	4	5
1	0.571	-	-	-	0.408	-
2	0.889	0.318	0.260	-	0.248	-0.116
3	1.091	0.202	0.171	0.144	0.166	-0.062
4	1.231	0.140	0.121	0.109	0.119	-0.038
5	1.333	0.102	0.090	0.083	0.089	-0.023
6	1.412	0.079	-	0.068	0.069	-

Комментарии:

1. Односторонняя разностная производная. Я выбрал левостороннюю производную. Для каждого узла таблицы она вычисляется следующим образом:

$$y'_n = \frac{y_n - y_{n-1}}{h} + O(h),\ \text{где } h - \text{шаг аргумента сетки}$$

Эта формула обладает низким, первым порядком точности относительно h. Также ее нельзя использовать для первого узла сетки, так как для него не найдется предыдущего. Формула получается из разложения в ряд Тейлора функции y(x) в точке x_{n-1} или x_{n+1} при центре разложения в точке x_n, и затем выражения из него производной.

2. Центральная разностная производная. Вычисляется по формуле:

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

Формула получается вычитанием из разложения функции в ряд Тейлора в точке x_{n+1} разложения этой же функции в точке x_{n-1}

Порядок точности этой формулы уже второй, но она не применима как к первому, так и к последнему узлам функции, так как требует наличия и предыдущего и следующего значения. Для этих узлов требуется выводить дополнительные формулы из рядов Тейлора.

3. Вторая формула Рунге имеет вид:

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

Здесь Ω - формула порядка точности $p + 1$, которую можно найти, зная формулу Φ, которая имеет порядок точности p. Используем левостороннюю формулу для нахождения более точной формулы производной по формуле Рунге. Положим m=2. Это означает, что будет использоваться формула для сетки с шагом h и для сетки с шагом 2h. Тогда:

$$\Omega = \frac{y_n - y_{n-1}}{h} + \frac{\frac{y_n - y_{n-1}}{h} - \frac{y_n - y_{n-2}}{2h}}{1} = \frac{3y_n - 4y_{n-1} + y_{n-2}}{2h} + O(h^2)$$

Так как формула для односторонней разностной производной имеет порядок точности 1, то полученная формула обладает вторым порядком точности. При этом она не применима к первым двум узлам функции, так как требует наличия двух предыдущих узлов.

4. Формула с использованием выравнивающих переменных.

Наша функция имеет вид $y = \frac{a_0 x}{a_1 + a_2 x}$.

Можно сделать следующую замену:

$$\eta = \frac{1}{y},\ \xi = \frac{1}{x},\ \text{тогда}\ \eta = \frac{a_1 \xi}{a_0} + \frac{a_2}{a_0}$$

Получается, что функция $\eta(\xi)$ - является линейной и для нее можно точно посчитать первую производную с помощью односторонней формулы. Более того, производная η'_ξ постоянная на всей области определения, так как функция линейна. Посчитав эту производную, перейдем обратно к исходным переменным по формуле:

$$y'_x = \frac{\eta'_\xi \xi'_x}{\eta'_y},\ \text{где, очевидно,}\ \xi'_x = \frac{-1}{x^2},\ \eta'_y = \frac{-1}{y^2}$$

Так как производная η'_ξ постоянна и переменные x и y не обращаются в 0 в данной таблице, рассчитать производную мы можем для любого узла.

5. Вторая разностная производная. Вычисляется по формуле:

$y_n'' = \frac{y_{n-1} - 2y_n + y_{n+1}}{h^2} + O(h^2)$. Обладает вторым порядком точности и не применяется к первому и последнему узлам таблицы. Получается сложением двух разложений функции в ряд Тейлора (тех, которые описаны в пункте 1).

Ответы на вопросы

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

Напишем формулы Тейлора для узлов y_{N-1} и y_{N-2} относительно y_N.

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

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

Исключим из этих формул h^2. Для этого умножим первую формулу на 4 и вычтем из нее вторую:

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

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

2. Получить формулу порядка точности $O(h^2)$ для второй разностной производной y''_0 в крайнем левом узле x_0.

Напишем формулы Тейлора для узлов y_1, y_2, y_3.

$$y_1 = y_0 + \frac{h}{1!}y'_0 + \frac{h^2}{2!}y''_0 + \frac{h^3}{3!}y'''_0 + \frac{h^4}{4!}y^{IV}_0 + \dots$$

$$y_2 = y_0 + \frac{2h}{1!}y'_0 + \frac{(2h)^2}{2!}y''_0 + \frac{(2h)^3}{3!}y'''_0 + \frac{(2h)^4}{4!}y^{IV}_0 + \dots$$

$$y_3 = y_0 + \frac{3h}{1!}y'_0 + \frac{(3h)^2}{2!}y''_0 + \frac{(3h)^3}{3!}y'''_0 + \frac{(3h)^4}{4!}y^{IV}_0 + \dots$$

Исключаем из них h и h^3, сначала h:

$$y_1 + y_2 - y_3 = y_0 - \frac{4h^2}{2!}y''_0 - \frac{18h^3}{3!}y'''_0 - \frac{64h^4}{4!}y^{IV}_0 + \dots$$

$2y_1 - y_2 = y_0 - \frac{2h^2}{2!}y''_0 - \frac{6h^3}{3!}y'''_0 - \frac{14h^4}{4!}y^{IV}_0 + ...$

Теперь исключаем h^3 из полученных двух формул, для этого вычтем из первой формулы три вторых:

$y_1 + y_2 - y_3 - 6y_1 + 3y_2 =- 2y_0 + \frac{2h^2}{2!}y''_0 - \frac{22h^4}{4!}y^{IV}_0 + ... =- 2y_0 + h^2y''_0 + O(h^2)$

Отсюда $y''_0 = \frac{2y_0 - 5y_1 + 4y_2 - y_3}{h^2} + O(h^2)$

3. Используя 2-ую формулу Рунге, дать вывод выражения (9) из Лекции №7 для первой производной y'_0 в левом крайнем узле

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

Левосторонняя формула первой разностной производной берется за Φ:

$\Phi(h) = \frac{y_1 - y_0}{h} + O(h);\ \Phi(2h) = \frac{y_2 - y_0}{h} + O(h),\ m = 2,\ p = 1$

$\Omega = \Phi(h) + \frac{\Phi(h) - \Phi(mh)}{m^p - 1} + O(h^{p+1}) = \frac{y_1 - y_0}{h} + \frac{y_1 - y_0}{h} - \frac{y_2 - y_0}{2h} + O(h^2) =$

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

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.

Напишем формулы Тейлора для узлов $y_1,\ y_2,\ y_3$.

$y_1 = y_0 + \frac{h}{1!}y'_0 + \frac{h^2}{2!}y''_0 + \frac{h^3}{3!}y'''_0 + \frac{h^4}{4!}y^{IV}_0 + ...$

$y_2 = y_0 + \frac{2h}{1!}y'_0 + \frac{(2h)^2}{2!}y''_0 + \frac{(2h)^3}{3!}y'''_0 + \frac{(2h)^4}{4!}y^{IV}_0 + ...$

$y_3 = y_0 + \frac{3h}{1!}y'_0 + \frac{(3h)^2}{2!}y''_0 + \frac{(3h)^3}{3!}y'''_0 + \frac{(3h)^4}{4!}y^{IV}_0 + ...$

Исключаем из них h^2 и h^3, сначала h^2:

$$5y_1 + y_2 - y_3 = 5y_0 + \frac{4h}{1!}y'_0 - \frac{14h^3}{3!}y'''_0 - \frac{60h^4}{4!}y^{IV}_0 +...$$

$$4y_1 - y_2 = 3y_0 + \frac{2h}{1!}y'_0 - \frac{4h^3}{3!}y'''_0 - \frac{12h^4}{4!}y^{IV}_0 +...$$

Теперь исключаем h^3 из полученных двух формул, для этого вычтем из двух первых формул семь вторых:

$$10y_1 + 2y_2 - 2y_3 - 28y_1 + 7y_2 =- 11y_0 - \frac{6h}{1!}y'_0 - \frac{36h^4}{4!}y^{IV}_0 +... =$$

$$=- 11y_0 - 6hy'_0 + O(h^3)$$

Отсюда $y'_0 = \frac{-11y_0 + 18y_1 - 9y_2 + 2y_3}{6h} + O(h^3)$

Код программы

Основная функция приложения:

```
int main(void)
{
    show_info(); // информация о программе
    FILE *file = get_file();    // получение файла с данными от пользователя

    size_t size = 0;   // size - количество узлов
    // считывание табличной функции в массив
    record_t *data = export_to_array(file, &size);    // record_t содержит два поля x и y
    fclose(file);

    // проверка успешности выделения памяти
    if (!data)
        return EXIT_FAILURE;

    // расчет производных каждым методом и вывод их на экран
    one_side_diff_derivative(data, size);
    center_diff_derivative(data, size);
    runge_derivative(data, size);
    alignment_derivative(data, size);
    second_diff_derivative(data, size);
    free(data);
    return EXIT_SUCCESS;
}
```

Функции, использованные в начинке программы:

```c
// вычисление и вывод на экран левосторонней первой производной
void one_side_diff_derivative(record_t *data, size_t size)
{
    printf("FIRST LEFT SIDE DIFFERENTIAL DERIVATIVE:\n");
    printf("y'(%.3lf) = undefined\n", data[0].x);    // не можем вычислить
    double h = data[1].x - data[0].x;
    for (size_t i = 1; i < size; i++)
    {
        double dy = (data[i].y - data[i - 1].y) / h;
        printf("y'(%.3lf) = %.3lf\n", data[i].x, dy);
    }
    printf("\n");
}

// вычисление и вывод на экран центральной первой производной
void center_diff_derivative(record_t *data, size_t size)
{
    printf("FIRST CENTER DIFFERENTIAL DERIVATIVE:\n");
    printf("y'(%.3lf) = undefined\n", data[0].x);
    double h = data[1].x - data[0].x;
    for (size_t i = 1; i < size - 1; i++)
    {
        double dy = (data[i + 1].y - data[i - 1].y) / 2.0 / h;
        printf("y'(%.3lf) = %.3lf\n", data[i].x, dy);
    }
    printf("y'(%.3lf) = undefined\n", data[size - 1].x);
    printf("\n");
}

// вычисление производной по второй формуле Рунге
void runge_derivative(record_t *data, size_t size)
{
    printf("FIRST RUNGE DERIVATIVE:\n");
    printf("y'(%.3lf) = undefined\n", data[0].x);
    printf("y'(%.3lf) = undefined\n", data[1].x);
    double h = data[1].x - data[0].x;
    for (size_t i = 2; i < size; i++)
    {
        double dy = (3 * data[i].y - 4 * data[i - 1].y + data[i - 2].y) / 2.0 / h;
```

```
        printf("y'(%.3lf) = %.3lf\n", data[i].x, dy);
    }
    printf("\n");
}

// вычисление с использованием выравнивающих переменных
void alignment_derivative(record_t *data, size_t size)
{
    printf("FIRST ALIGNMENT DERIVATIVE:\n");
    // вычисление производной линейной функции
    double delta_etha = 1.0 / data[1].y - 1.0 / data[0].y;
    double delta_xee = 1.0 / data[1].x - 1.0 / data[0].x;
    double d_etha = delta_etha / delta_xee;

    // вычисление производной в исходных координатах
    for (size_t i = 0; i < size; i++)
    {
        double dy = d_etha / data[i].x / data[i].x * data[i].y * data[i].y;
        printf("y'(%.3lf) = %.3lf\n", data[i].x, dy);
    }
    printf("\n");
}

// вычисление и вывод на экран второй разностной производной
void second_diff_derivative(record_t *data, size_t size)
{
    printf("SECOND DIFFERENTIAL DERIVATIVE:\n");
    printf("y'(%.3lf) = undefined\n", data[0].x);
    double h = data[1].x - data[0].x;
    for (size_t i = 1; i < size - 1; i++)
    {
        double dy = (data[i - 1].y - 2 * data[i].y + data[i + 1].y) / h / h;
        printf("y'(%.3lf) = %.3lf\n", data[i].x, dy);
    }
    printf("y'(%.3lf) = undefined\n", data[size - 1].x);
    printf("\n");
}
```