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»

Лабораторная работа № 6
по дисциплине "Вычислительные алгоритмы"

Тема Численное дифференцирование.

Студент Романов А.В.

Группа ИУ7-43Б

Оценка (баллы) _______________

Преподаватель Градов В.М.

Москва — 2020 г.

1. Тема работы

Построение и программная реализация алгоритмов численного дифференцирования.

2. Цель работы

Получение навыков построения алгоритма вычисления производных от сеточных функций.

3. Задание

Задана табличная функция. Имеетя информация, что законамерность, представленная этой таблицей, может быть описана формулой:

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

Вычислить значение производных и занести их в столбцы таблицы.

1 столбец – односторонняя разностная производная.
2 столбец – центральная разностная производная.
3 столбец – вторая формула Рунге с использованием односторонней производной.
4 столбец – введены выравнивающие переменные.
5 столбец – вторая разностная производная.

4. Описание алгоритма

Используя ряд Тейлора, можно получить разностные формулы для вычисления производных:

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

или

$$y'_n = \frac{y_n - y_{n-1}}{h}$$

Первое выражение – **правая** разностная производная, второе – **левая** разностная производная. Эти формулы имею самый низкий, первый порядок точности.

Таким же образом, можно получить **центральную** формулу:

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

Эта формула имеет второй порядок точности.

Используя ряд Тейлора, таким же образом можно полуичить **разностный аналог второй производной**:

$$y_n'' = \frac{y_{n-1} - 2y_n + y_{n+1}}{h^2}$$

Используя преобразования в рядах Тейлора, можем прийти к **первой формуле Рунге**:

$$\psi(x)h^p = \frac{\Phi(h) - \Phi(mh)}{m^p - 1}$$

Аналогично, можем получить **вторую формулу Рунге**:

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

Стоит отметить, что формулы Рунге справедливы не только для операции дифференцирования, но и для любых других приближенных вычислений. Важно, чтобы погрешность применяемых формул имела вид $R = \psi(x)h^p$

Так же, существует метод ввода так называемых **выравнивающих переменных**. При удачном подборе этих переменных исходная кривая может быть преобразована в прямую линию, производная от которой вычисляется точно по самым простым формулам. Пусть задана функция $y(x)$, и введены выравнивающие переменные $\xi = \xi(x)$ и $\eta = \eta(y)$. Тогда, возврат к заданным переменным осуществляется этой формулой:

$$y_x' = \frac{\eta_\xi' \, \xi_x'}{\eta_y'}$$

А η_ξ' можно вычислить по любой односторонней формуле.

5. Результаты

x	y	1	2	3	4	5
1	0.571	-	-	-	0.409	-
2	0.889	0.318	0.260	-	0.247	-0.116
3	1.091	0.201	0.171	0.144	0.165	-0.061
4	1.231	0.140	0.121	0.109	0.117	-0.038
5	1.333	0.102	0.091	0.083	0.089	-0.023
6	1.412	0.078	-	0.067	-	-

В **первом столбце** использовал левостороннюю формулу, поэтому нет значения при x = 1. Как и ожидалось, точность $O(h)$.
Второй столбец – центральная формула. Точность уже выше, чем в первом столбце - $O(h^2)$
Третий столбец – вторая формула рунге с использованием односторонней производной (брал слева).
Четвертый столбец – введены выравнивающие переменные. Так как неизвестны параметры, нет возможности точно оценить погрешность, но предположу, что она минимальна. Итоговая формула:

$$y'_x = \frac{\eta'_\xi \, \xi'_x}{\eta'_y} = \frac{\eta'_\xi \, y^2}{x^2}$$

η'_ξ искал с помощью правосторонней формулы:

$$\frac{-\frac{1}{y_{i+1}} + \frac{1}{y_i}}{-\frac{1}{x_{i+1}} + \frac{1}{x_i}}$$

Пятый столбец - вторая разностная производная.

6. Ответы контрольные вопросы

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

Имеем:

$$y_{N-1} = y_N - hy'_n + \frac{h^2}{2!}y''_N - \frac{h^3}{3!}y'''_N ...$$

$$y_{N-2} = y_N - 2hy'_n + \frac{4h^2}{2!}y''_N - \frac{8h^3}{3!}y'''_N ...$$

Отсюда получаем:

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

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

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

Имеем:

$$y_1 = y_0 + hy_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0'''...$$

$$y_2 = y_0 + 2hy_0' + \frac{4h^2}{2!}y_0'' + \frac{8h^3}{3!}y_0'''...$$

Отсюда получаем:

$$y_0'' = \frac{y_0 - 2y_1 + y_2}{h^2} - hy_0'''$$

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

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

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

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

$$-(\frac{y_2 - y_0}{2h} - hy_0'') + O(h^2) = \frac{4y_1 - 3y_0 - y_2}{2h} + O(h^2)$$

$\Phi(h)$ и $\Phi(2h)$ выведены из рядов Тейлора.

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

Имеем:

$$y_1 = y_0 + hy_0' + \frac{h^2}{2}y_0'' + \frac{h^3}{3!}y_0'''...$$

$$y_2 = y_0 + 2hy_0' + \frac{(2h)^2}{2!}y_0'' + \frac{(2h)^3}{3!}y_0'''...$$

$$y_3 = y_0 + 3hy_0' + \frac{(3h)^2}{2!}y_0'' + \frac{(3h)^3}{3!}y_0'''...$$

Выражая y_0' и y_0'', получаем:

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

Теперь выразим y_0''' и подставим:

$$y_0' = \frac{108y_1 - 85y_0 - 27y_2 + 4y_3}{66h} - \frac{3h^2}{11}y_0'''$$

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

7. Код программы

Файл Main.hs:

```
import System.IO
import Text.Printf
import Differentation

f :: [Double]
f = [0.571, 0.889, 1.091, 1.231, 1.333, 1.412]

x :: [Double]
x = [1..6]

main :: IO ()
main = do
    putStrLn "!!! Results are not balanced (by x value) !!!"
    putStr "\nLeft side: " >> mapM_ (printf "%.4f ") (leftSide f 1)
    putStr "\nCenter: " >> mapM_ (printf "%.4f ") (centerDiff f 1)
    putStr "\nRunge: " >> mapM_ (printf "%.4f ") (rungeCenter f 1 1)
    putStr "\nAlignment variables: " >> mapM_ (printf "%.4f ") (alignment f x 1)
    putStr "\nSecond derivative: " >> mapM_ (printf "%.4f ") (differential2 f 1)
```

Файл Differentation.hs:

```
module Differentation (
    leftSide,
    centerDiff,
    rungeCenter,
    alignment,
    differential2
) where

leftValue :: Double -> Double -> Double -> Double
leftValue ly y h = (ly - y) / h

leftSide :: [Double] -> Double -> [Double]
leftSide ys h = map (\y -> (fst y - snd y) / h) $ zip (tail ys) ys

centerDiff :: [Double] -> Double -> [Double]
centerDiff ys h = map (\y -> (fst y - snd y) / (2 * h)) $ zip (tail $ tail ys)
  ys

rungeCenter :: [Double] -> Double -> Int -> [Double]
rungeCenter ys h p = runge
    where yh = tail $ leftSide ys h
          ymh = centerDiff ys h
          runge = map (\y -> fst y + (fst y - snd y) / (2^p - 1)) $ zip yh ymh

fst' :: (Double, Double, Double) -> Double
fst' (x, _, _ ) = x

snd' :: (Double, Double, Double) -> Double
snd' (_, x, _) = x
```

```haskell
thd' :: (Double, Double, Double) -> Double
thd' (_, _, x) = x

differential2 :: [Double] -> Double -> [Double]
differential2 ys h = map (
                      \y ->
                          (fst' y - 2 * snd' y  + thd' y) / (h * h)) $ zip3 ys (
                              tail ys) (tail $ tail ys)

alignment :: [Double] -> [Double] -> Double -> [Double]
alignment ys xs h = a
    where ksi = map (\y -> -1 / y) ys
          phi = map (\x -> -1 / x) xs
          a = map (
            \i ->
                ((ys !! i) ^2 / (xs !! i) ^2) *
                    leftValue (ksi !! (i + 1)) (ksi !! i) (phi !! (i + 1) - phi
                        !! i)) [0..length ys - 2]
```