**Индивидуальное домашнее задание № 2**

**Численное дифференцирование**

Составить таблицу значений производной функции на промежутке [0,2] с шагом 0,1, вычисленной по приближенным формулам правой, левой и центральной производной

с точностью 0,0001.

Начальное значение положить равным 0,01, последующие значения определять по формуле . Итеративный процесс закончить, когда для всех табличных значений будет выполнятся неравенство

Для каждого численного метода нахождение производной указать конечный шаг .

Вычислить точное значение производной в точках табуляции и определить фактическое отклонение приближенных результатов от истинного, результаты представить в табличном виде. Выделить те значения, в которых отклонение превышает заданную погрешность.

**Задание:**

**Решение**

Программа выполнена на языке программирования Scala.

import math.\_

import scala.math.Numeric.DoubleAsIfIntegral

/\*\*

\* Created with IntelliJ IDEA.

\* User: SabLeOn

\* Date: 18.05.13

\* Time: 11:13

\* To change this template use File | Settings | File Templates.

\*/

object NumericalDifferentiation {

def f(x:Double):Double = (exp(x + 5) \* cos(7 \* pow(x, 2)) \* (x - 2)) / (1 + pow(atan(3 + x), 2))

def left\_f\_deriv(x:Double, h:Double):Double = (f(x) - f(x - h)) / h

def right\_f\_deriv(x:Double, h:Double):Double = (f(x + h) - f(x)) / h

def mid\_f\_deriv(x:Double, h:Double):Double = (f(x + h) - f(x - h)) / (2 \* h)

def f\_deriv(x:Double):Double = {

val COS = cos(7 \* pow(x, 2))

val DEN = 1 + pow(atan(x + 3), 2)

val B = COS \* (x - 2)

val dB = x \* COS - 14 \* x \* (x - 2) \* sin(7 \* pow(x, 2))

val C = DEN

val dC = 2 \* atan(3 + x) / (1 + pow(x + 3, 2))

val A = (x - 2) \* COS / DEN

val dA = (B \* dC - dB \* C)/pow(C, 2)

exp(x+5) \* (A + dA)

}

val valueX:List[Double] = List.range[Double](0, 2, 0.1)(new DoubleAsIfIntegral {

def compare(x: Double, y: Double): Int = {

if(x < y) return -1

else if (x > y) return 1

else return 0

}

})

val h0 = 0.01

val MINIMUM\_DELTA:Double = 0.0001

def isEnd(a:Array[Double], b:Array[Double]):Boolean = {

var result = true

(a, b).zipped foreach { (j, k) =>

result = result && (math.abs(j - k) <= MINIMUM\_DELTA)

}

return result

}

def calculateArray(delta:Double, func: (Double, Double) => Double):Array[Double] = {

val result:Array[Double] = new Array[Double](valueX.length)

(valueX, (0 to valueX.length - 1)).zipped foreach {(x, j) =>

result.update(j, func(x, delta))

}

return result

}

def iterate(deriv: (Double, Double) => Double):(Array[Double], Double) = {

var h = h0

var previousStep = calculateArray(h, deriv)

var currentStep = previousStep

var over = false

do {

//write previous step into buffers

previousStep = currentStep

//calculate next step

h = h/2.0

currentStep = calculateArray(h, deriv)

if(isEnd(currentStep, previousStep)) over = true

// this barrier I need to escape an infinity loop

else if(h < 1e-12) over = true

} while(!over)

return (currentStep, h)

}

def printArray(output: (Array[Double], Double)) = {

Console.println("Конечный шаг h = %e".format(output.\_2))

(valueX, output.\_1).zipped foreach { (x, fx) => {

Console.print("f(%.1f) = %.4f".format(x, fx))

if (math.abs(fx - f\_deriv(x)) > MINIMUM\_DELTA) Console.println(" \*")

else Console.println()

}

}

}

def main(args: Array[String]):Unit = {

println("Результаты вычислений по формулам левой производной: ")

printArray(iterate(left\_f\_deriv))

println("Результаты вычислений по формулам правой производной: ")

printArray(iterate(right\_f\_deriv))

println("Результаты вычислений по формулам центральной производной: ")

printArray(iterate(mid\_f\_deriv))

println("Точное значение производной:")

valueX foreach {x => println("f(%.1f) = %.4f".format(x, f\_deriv(x)))}

}

}

**Результат работы программы**

Результаты вычислений по формулам левой производной:

Конечный шаг h = 2,384186e-09

f(0,0) = -46,6579 \*

f(0,1) = -34,1273 \*

f(0,2) = 53,7822 \*

f(0,3) = 285,9256 \*

f(0,4) = 656,3326 \*

f(0,5) = 953,0439 \*

f(0,6) = 713,3244 \*

f(0,7) = -377,2376 \*

f(0,8) = -1581,6689 \*

f(0,9) = -1065,6734 \*

f(1,0) = 1351,7567 \*

f(1,1) = 1805,0165 \*

f(1,2) = -1481,3508 \*

f(1,3) = -1614,8912 \*

f(1,4) = 2335,5255 \*

f(1,5) = -226,5450 \*

f(1,6) = -1762,4563 \*

f(1,7) = 2034,9892 \*

f(1,8) = -1199,7289 \*

f(1,9) = 434,5520 \*

Результаты вычислений по формулам правой производной:

Конечный шаг h = 5,820766e-13

f(0,0) = -46,6553 \*

f(0,1) = -34,2773 \*

f(0,2) = 53,8574 \*

f(0,3) = 285,9375 \*

f(0,4) = 656,3477 \*

f(0,5) = 953,0396 \*

f(0,6) = 713,4277 \*

f(0,7) = -377,2461 \*

f(0,8) = -1581,6895 \*

f(0,9) = -1065,5029 \*

f(1,0) = 1351,9287 \*

f(1,1) = 1804,6387 \*

f(1,2) = -1480,8838 \*

f(1,3) = -1614,3799 \*

f(1,4) = 2335,3638 \*

f(1,5) = -226,3428 \*

f(1,6) = -1762,4512 \*

f(1,7) = 2034,2834 \*

f(1,8) = -1199,5361 \*

f(1,9) = 434,4604 \*

Результаты вычислений по формулам центральной производной:

Конечный шаг h = 9,765625e-06

f(0,0) = -46,6579 \*

f(0,1) = -34,1273 \*

f(0,2) = 53,7823 \*

f(0,3) = 285,9256 \*

f(0,4) = 656,3326 \*

f(0,5) = 953,0439 \*

f(0,6) = 713,3244 \*

f(0,7) = -377,2376 \*

f(0,8) = -1581,6689 \*

f(0,9) = -1065,6733 \*

f(1,0) = 1351,7567 \*

f(1,1) = 1805,0165 \*

f(1,2) = -1481,3509 \*

f(1,3) = -1614,8912 \*

f(1,4) = 2335,5256 \*

f(1,5) = -226,5450 \*

f(1,6) = -1762,4563 \*

f(1,7) = 2034,9891 \*

f(1,8) = -1199,7289 \*

f(1,9) = 434,5520 \*

Точное значение производной:

f(0,0) = -127,2559

f(0,1) = -149,4693

f(0,2) = -240,8491

f(0,3) = -452,1364

f(0,4) = -750,9551

f(0,5) = -912,2139

f(0,6) = -517,1528

f(0,7) = 620,6092

f(0,8) = 1643,0776

f(0,9) = 837,0242

f(1,0) = -1572,3287

f(1,1) = -1628,5103

f(1,2) = 1733,6269

f(1,3) = 1370,0763

f(1,4) = -2474,1131

f(1,5) = 579,0003

f(1,6) = 1545,7455

f(1,7) = -2105,0354

f(1,8) = 1490,1087

f(1,9) = -810,5724

**Результирующая таблица**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | левая | правая | центральная |  |
| Конечный шаг h | 2,384186e-09 | 5,820766e-13 | 9,765625e-06 |  |
| 0,0 | -46,6579 | -46,6553 | -46,6579 | -127,2559 |
| 0,1 | -34,1273 | -34,2773 | -34,1273 | -149,4693 |
| 0,2 | 53,7822 | 53,8574 | 53,7823 | -240,8491 |
| 0,3 | 285,9256 | 285,9375 | 285,9256 | -452,1364 |
| 0,4 | 656,3326 | 656,3477 | 656,3326 | -750,9551 |
| 0,5 | 953,0439 | 953,0396 | 953,0439 | -912,2139 |
| 0,6 | 713,3244 | 713,4277 | 713,3244 | -517,1528 |
| 0,7 | -377,2376 | -377,2461 | -377,2376 | 620,6092 |
| 0,8 | -1581,6689 | -1581,6895 | -1581,6689 | 1643,0776 |
| 0,9 | -1065,6734 | -1065,5029 | -1065,6733 | 837,0242 |
| 1,0 | 1351,7567 | 1351,9287 | 1351,7567 | -1572,3287 |
| 1,1 | 1805,0165 | 1804,6387 | 1805,0165 | -1628,5103 |
| 1,2 | -1481,3508 | -1480,8838 | -1481,3509 | 1733,6269 |
| 1,3 | -1614,8912 | -1614,3799 | -1614,8912 | 1370,0763 |
| 1,4 | 2335,5255 | 2335,3638 | 2335,5256 | -2474,1131 |
| 1,5 | -226,5450 | -226,3428 | -226,5450 | 579,0003 |
| 1,6 | -1762,4563 | -1762,4512 | -1762,4563 | 1545,7455 |
| 1,7 | 2034,9892 | 2034,2834 | 2034,9891 | -2105,0354 |
| 1,8 | -1199,7289 | -1199,5361 | -1199,7289 | 1490,1087 |
| 1,9 | 434,5520 | 434,4604 | 434,5520 | -810,5724 |
| 2,0 | -365,7455 | -365,8236 | -365,7455 | 731,4910 |