**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, 4)) \* (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, 4))

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

val B = COS \* (x - 2)

val dB = COS - 28 \* pow(x, 3) \* (x - 2) \* sin(7 \* pow(x, 4))

val C = DEN

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

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

val dA = (dB \* C - B \* dC)/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-13) 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 = 7,275958e-14

f(0,0) = -46,2891 \*

f(0,1) = -46,2891 \*

f(0,2) = -45,3125 \*

f(0,3) = -36,7188 \*

f(0,4) = 3,1250 \*

f(0,5) = 171,8750 \*

f(0,6) = 651,3672 \*

f(0,7) = 1370,9473 \*

f(0,8) = 579,6875 \*

f(0,9) = -2965,2832 \*

f(1,0) = 2695,5078 \*

f(1,1) = -3978,1250 \*

f(1,2) = 6353,2227 \*

f(1,3) = 7649,9023 \*

f(1,4) = 9680,1758 \*

f(1,5) = -8663,0859 \*

f(1,6) = 11215,5273 \*

f(1,7) = 10993,2129 \*

f(1,8) = -9732,9590 \*

f(1,9) = -1095,0195 \*

f(2,0) = 173,4163 \*

Результаты вычислений по формулам правой производной:

Конечный шаг h = 7,275958e-14

f(0,0) = -46,6797 \*

f(0,1) = -46,2891 \*

f(0,2) = -44,1406 \*

f(0,3) = -37,8906 \*

f(0,4) = 3,5156 \*

f(0,5) = 171,0938 \*

f(0,6) = 650,9766 \*

f(0,7) = 1371,9482 \*

f(0,8) = 579,2969 \*

f(0,9) = -2965,4785 \*

f(1,0) = 2695,8984 \*

f(1,1) = -3974,6094 \*

f(1,2) = 6343,6523 \*

f(1,3) = 7649,8047 \*

f(1,4) = 9667,8223 \*

f(1,5) = -8636,9141 \*

f(1,6) = 11225,3906 \*

f(1,7) = 11000,9277 \*

f(1,8) = -9733,0078 \*

f(1,9) = -1094,3359 \*

f(2,0) = 173,4163 \*

Результаты вычислений по формулам центральной производной:

Конечный шаг h = 6,103516e-07

f(0,0) = -46,6579

f(0,1) = -46,0452

f(0,2) = -44,4706

f(0,3) = -37,1362

f(0,4) = 3,5233

f(0,5) = 171,4007

f(0,6) = 651,8963

f(0,7) = 1372,3040

f(0,8) = 579,5640

f(0,9) = -2967,3742

f(1,0) = 2697,2758

f(1,1) = -3973,5516

f(1,2) = 6342,9744

f(1,3) = 7642,8851

f(1,4) = 9664,9272

f(1,5) = -8642,7308

f(1,6) = 11210,0243

f(1,7) = 10990,5357

f(1,8) = -9722,2336

f(1,9) = -1093,2934

f(2,0) = 173,2471

Точное значение производной:

f(0,0) = -46,6579

f(0,1) = -46,0452

f(0,2) = -44,4706

f(0,3) = -37,1362

f(0,4) = 3,5233

f(0,5) = 171,4007

f(0,6) = 651,8963

f(0,7) = 1372,3040

f(0,8) = 579,5640

f(0,9) = -2967,3742

f(1,0) = 2697,2758

f(1,1) = -3973,5516

f(1,2) = 6342,9744

f(1,3) = 7642,8851

f(1,4) = 9664,9272

f(1,5) = -8642,7308

f(1,6) = 11210,0243

f(1,7) = 10990,5357

f(1,8) = -9722,2336

f(1,9) = -1093,2934

f(2,0) = 173,2471

**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 |