



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 № 4 по дисциплине "Вычислительные алгоритмы"

Тема Среднеквадратичное приближение.

Студент Романов А.В.

Группа ИУ7-43Б

Оценка (баллы) \_\_\_\_\_

Преподаватель Градов В.М.

Москва — 2020 г.

## 1. Тема работы

Построение и программная реализация алгоритма наилучшего среднеквадратичного приближения.

## 2. Цель работы

Получение навыков построения алгоритма метода наименьших квадратов с использованием полинома заданной степени при аппроксимации табличных функций с весами.

## 3. Входные данные

1. Таблица функции с весами  $p_i$  с количеством узлов  $N$ .

$x$	$y$	$\rho$
$x_i$	$y_i$	$\rho_i$

2. Степень аппроксимирующего полинома –  $n$ .

## 4. Выходные данные

График, на котором изображён аппроксимирующий полином, и точки из исходной таблицы значений.

## 5. Описание алгоритма

Под близостью в среднем исходной и аппроксимирующей функций будем понимать результат оценки суммы

$$I = \sum_{i=1}^N \rho_i [y(x_i) - \varphi(x_i)]^2 \quad (1)$$

$y(x)$  - исходная функция

$\varphi(x)$  - множество функций, принадлежащих линейному пространству функций

$\rho_i$  - вес точки

Нужно найти наилучшее приближение, т.е

$$\sum_{i=1}^N \rho_i [y(x_i) - \varphi(x_i)]^2 = \min \quad (2)$$

Разложим функцию  $\varphi(x)$  по системе линейно независимых функций  $\varphi_k(x)$ :

$$\varphi(x) = \sum_{k=0}^N a_k \varphi_k(x) \quad (3)$$

Подставляя (3) в условие (2) получим:

$$((y - \varphi), (y - \varphi)) = (y, y) - 2 \sum_{k=0}^n a_k (y, \varphi_k) + \sum_{k=0}^n \sum_{m=0}^n a_k a_m (\varphi_k, \varphi_m) = \min \quad (4)$$

Дифференцируя по  $a_k$  получаем:

$$\sum_{i=0}^n (x^k, x^m) a_m = (y, x^k) \quad (5)$$

где

$$(x^k, x^m) = \sum_{i=1}^N \rho_i x_i^{k+m}$$
$$(y, x^k) = \sum_{i=1}^N \rho_i y_i x_i^k$$

Итоговый алгоритм:

1. Выбирается степень полинома  $n < N$ .
2. Составляется система линейных алгебраических уравнений типа.
3. В результате решения СЛАУ находятся коэффициенты полинома.

## 6. Результаты работы программы

### 1. Веса точек равны.

Исходная таблица:

$x_i$	$y_i$	$\rho_i$
1	1	1
2	4	1
3	9	1
4	16	1
5	25	1
6	36	1
7	49	1
8	64	1
9	81	1
10	100	1

../../src/n1w1.png

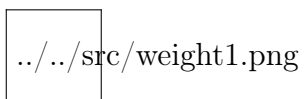
../../src/n2w1.png

../../src/n6w1.png

## 2. Веса точек разные.

Исходная таблица:

$x_i$	$y_i$	$\rho_i$
1	1	0.1
2	4	0.5
3	9	1
4	16	2
5	25	5
6	36	1.9
7	49	3
8	64	0.3
9	81	5
10	100	0.1



../../src/weight1.png

Синяя прямая - веса точек равны единице.

Красная прямая - веса точек разные.

## 7. Ответы на вопросы для защиты ЛР

1. Что произойдет при задании степени полинома  $n = N - 1$

Полином будет построен по всем точкам, независимо от того, какие будут веса у точек.

2. Будет ли работать Ваша программа при  $n \geq N$ ? Что именно в алгоритме требует отдельного анализа данного случая и может привести к аварийной остановке?

Программа работать будет. Но, по  $N$  точкам нельзя построить полином степени  $n$ , так как в данном случае определитель будет равен нулю. Программа будет работать из-за погрешностей, будут операции с действительными числами.

3. Получить формулу для коэффициента полинома  $a_0$  при степени полинома  $n = 0$ . Какой смысл имеет величина, которую представляет данный коэффициент?

Формула:

$$\frac{\sum_{i=1}^N y_i \rho_i}{\sum_{i=1}^N \rho_i}$$

Значение: математическое ожидание

4. Записать и вычислить определитель матрицы СЛАУ для нахождения коэффициентов полинома для случая, когда  $n = N = 2$ .

Пусть есть таблица:

$x_i$	$y_i$	$\rho_i$
$x_0$	$y_0$	$\rho_1$
$x_1$	$y_1$	$\rho_2$

Тогда имеем СЛАУ вида:

$$\begin{cases} (\rho_0 + \rho_1)a_0 + (\rho_0x_0 + \rho_1x_1)a_1 + (\rho_0x_0^2 + \rho_1x_1^2)a_2 = \rho_0y_0 + \rho_1y_1 \\ (\rho_0x_0 + \rho_1x_1)a_0 + (\rho_0x_0^2 + \rho_1x_1^2)a_1 + (\rho_0x_0^3 + \rho_1x_1^3)a_2 = \rho_0y_0x_0 + \rho_1y_1x_1 \\ (\rho_0x_0^2 + \rho_1x_1^2)a_0 + (\rho_0x_0^3 + \rho_1x_1^3)a_1 + (\rho_0x_0^4 + \rho_1x_1^4)a_2 = \rho_0y_0x_0^2 + \rho_1y_1x_1^2 \end{cases}$$

$$\begin{aligned} \Delta &= (\rho_0 + \rho_1)(\rho_0x_0^2 + \rho_1x_1^2)(\rho_0x_0^4 + \rho_1x_1^4) + (\rho_0x_0 + \rho_1x_1)(\rho_0x_0^3 + \rho_1x_1^3)(\rho_0x_0^2 + \rho_1x_1^2) + \\ &(\rho_0x_0^2 + \rho_1x_1^2)(\rho_0x_0 + \rho_1x_1)(\rho_0x_0^3 + \rho_1x_1^3) - (\rho_0x_0^2 + \rho_1x_1^2)(\rho_0x_0^2 + \rho_1x_1^2)(\rho_0x_0^2 + \rho_1x_1^2) - \\ &(\rho_0 + \rho_1)(\rho_0x_0^3 + \rho_1x_1^3)(\rho_0x_0^3 + \rho_1x_1^3) - (\rho_0x_0 + \rho_1x_1)(\rho_0x_0 + \rho_1x_1)(\rho_0x_0^4 + \rho_1x_1^4) = 0 \end{aligned}$$

Так как  $\Delta = 0$ , система решений не имеет, как я и говорил в ответе на вопрос 2.

## 8. Код программы

Файл Main.hs:

```
import Parse
import Gauss
import Plot
import Approximation
import System.IO

printRow :: ((Double, Double), Double) -> IO ()
printRow row = putStrLn $ show (fst $ fst row) ++ "_" ++ show (snd $ fst row) ++
    "_" ++ show (snd row)

main :: IO ()
main = do
    table <- openFile "table.csv" ReadMode >>= hGetContents >>= return .
        parseTable . lines
    putStrLn "X___Y___P" >> mapM_ printRow (zip (xy table) $ weight table) >>
        putStrLn "Enter_n:"
    coeffs <- fmap toInt getLine >>= return . (+ 1) >>= return .
        quadraticApproximation table
    print coeffs
    plotApproximation (f coeffs) $ xy table
```

Файл Gauss.hs:

```
module Gauss(
    gauss
) where

type Matrix = [[Double]]
type Coeffs = [Double]

subtractRow :: [Double] -> [Double] -> [Double]
subtractRow subRow row = map (\x -> fst x - snd x * (head row / head subRow)) $
    zip row subRow

triangulation :: Matrix -> Matrix
triangulation matrix
    | length matrix == 0 = matrix
    | otherwise = head matrix :
        triangulation (map tail (map (subtractRow $ head matrix) $ tail matrix))

gauss :: Matrix -> Coeffs
gauss = coeffs . reverse . triangulation
    where coeffs = foldl (\x y -> (last y - (sum $ zipWith (*) (init $ tail y) x
        )) / (head y) : x) []
```

Файл Approximation.hs:

```
module Approximation(
    quadraticApproximation,
    f
) where

import Gauss
import Parse

type Coeffs = [Double]
type Weights = [Double]

f :: Coeffs -> Double -> Double
f coeffs x = sum $ zipWith (*) coeffs (map (\y -> x ^ y) [0..length coeffs - 1])
```

```

mult3 :: Double -> Double -> Double -> Double
mult3 x y z = x * y * z

quadraticApproximation :: Table -> Int -> Coeffs
quadraticApproximation table n = gauss matrix
  where
    xs = map fst $ xy table
    ys = map snd $ xy table
    x_coeffs =
      map (\k -> sum $ zipWith (*) (map (^ k) xs) $ weight table) [0..n *
        2 - 2]
    y_coeffs =
      map (\k -> sum $ zipWith3 mult3 (map (^ k) xs) ys $ weight table)
        [0..n - 1]
    matrix = zipWith (\x y -> x ++ [y]) (map (\x -> take n $ drop x x_coeffs
      ) [0..n - 1]) y_coeffs

```

### Файл Plot.hs:

```

module Plot (
  plotApproximation
) where

import Graphics.Rendering.Chart.Easy
import Graphics.Rendering.Chart.Backend.Cairo

signal :: (Double -> Double) -> [Double] -> [(Double, Double)]
signal f xs = [ (x, f x) | x <- xs ]

plotApproximation f pts = toFile def "test.png" $ do
  layout_title .= "n=_9"
  setColors [opaque blue, opaque red]
  plot (line "polynom" [signal f [-1,(-0.9)..11]])
  plot (points "points" pts)

```