

Факультет программной инженерии и компьютерной техники

Вычислительная математика

Лабораторная работа №4

Выполнил: Орехов Сергей Владимирович

Р32151

Преподаватель: Машина Екатерина Алексеевна

Санкт-Петербург

2023

Цель работы

Найти функцию, являющуюся наилучшим приближением заданной табличной функции по методу наименьших квадратов.

Листинг методов:

import math  
from aprox.dop import ln  
from gauss import gauss  
  
  
class Power:  
 name = "Power"  
 x\_array = []  
 y\_array = []  
 phi\_array = []  
 n = 0  
 dm = 0  
 sko = 0  
  
 sum\_xi = 0  
 sum\_xi2 = 0  
  
 sum\_yi = 0  
 sum\_xi\_yi = 0  
  
 a = 0  
 b = 0  
  
 def \_\_init\_\_(self, X, Y, N):  
 self.x\_array, self.y\_array = X.copy(), Y.copy()  
 self.n = N  
  
 def calculate\_sums(self):  
 for i in range(self.n):  
 self.sum\_xi += ln(self.x\_array[i])  
 self.sum\_xi2 += ln(self.x\_array[i]) \*\* 2  
  
 self.sum\_yi += ln(self.y\_array[i])  
 self.sum\_xi\_yi += ln(self.x\_array[i]) \* ln(self.y\_array[i])  
  
 def calculate\_coeff(self):  
 matrix = [[self.sum\_xi2, self.sum\_xi, self.sum\_xi\_yi], [self.sum\_xi, self.n, self.sum\_yi]]  
 gauss\_answer = gauss(matrix)  
 self.a, self.b = math.e \*\* gauss\_answer[1], gauss\_answer[0]  
  
 def func(self, x):  
 return self.a \* x \*\* self.b  
  
 def func\_string(self):  
 return str(self.a) + " \* x \*\* " + str(self.b)

from gauss import gauss  
  
  
class Cube:  
 dm = 0  
 sko = 0  
 name = "Cube"  
 x\_array = []  
 y\_array = []  
 phi\_array = []  
 n = 0  
  
 sum\_xi = 0  
 sum\_xi2 = 0  
 sum\_xi3 = 0  
 sum\_xi4 = 0  
 sum\_xi5 = 0  
 sum\_xi6 = 0  
  
 sum\_yi = 0  
 sum\_xi\_yi = 0  
 sum\_xi2\_yi = 0  
 sum\_xi3\_yi = 0  
  
 a = 0  
 b = 0  
 c = 0  
 d = 0  
  
 def \_\_init\_\_(self, X, Y, N):  
 self.x\_array, self.y\_array = X.copy(), Y.copy()  
 self.n = N  
  
 def calculate\_sums(self):  
 for i in range(self.n):  
 self.sum\_xi += self.x\_array[i]  
 self.sum\_xi2 += self.x\_array[i] \*\* 2  
 self.sum\_xi3 += self.x\_array[i] \*\* 3  
 self.sum\_xi4 += self.x\_array[i] \*\* 4  
 self.sum\_xi5 += self.x\_array[i] \*\* 5  
 self.sum\_xi6 += self.x\_array[i] \*\* 6  
  
 self.sum\_yi += self.y\_array[i]  
 self.sum\_xi\_yi += self.x\_array[i] \* self.y\_array[i]  
 self.sum\_xi2\_yi += self.x\_array[i] \*\* 2 \* self.y\_array[i]  
 self.sum\_xi3\_yi += self.x\_array[i] \*\* 3 \* self.y\_array[i]  
  
 def calculate\_coeff(self):  
 matrix = [[self.sum\_xi6, self.sum\_xi5, self.sum\_xi4, self.sum\_xi3, self.sum\_xi3\_yi],  
 [self.sum\_xi5, self.sum\_xi4, self.sum\_xi3, self.sum\_xi2, self.sum\_xi2\_yi],  
 [self.sum\_xi4, self.sum\_xi3, self.sum\_xi2, self.sum\_xi, self.sum\_xi\_yi],  
 [self.sum\_xi3, self.sum\_xi2, self.sum\_xi, self.n, self.sum\_yi]]  
 gauss\_answer = gauss(matrix)  
 self.a, self.b, self.c, self.d = gauss\_answer  
  
 def func(self, x):  
 return self.a \* x \*\* 3 + self.b \* x \*\* 2 + self.c \* x + self.d  
  
 def func\_string(self):  
 return str(self.a) + " \* x \*\* 3 + " + str(self.b) + " \* x \*\* 2 + " + str(self.b) + " \* x + " + str(self.d)

Пример работы:

Пример 1:

Input c - for console, f - for file: c

Enter N:

5

Enter values:

1 2 3 4 5

2 3 4 5 6

Linear

0.9999999999999998 \* x + 1.0000000000000009

[2.000000000000001, 3.0000000000000004, 4.0, 5.0, 6.0]

DM:9.860761315262648e-31

SKO: 4.440892098500626e-16

plt.cla()

plt.show()

Square

-6.800116025829084e-16 \* x \*\* 2 + 1.000000000000004 \* x + 0.9999999999999959

[1.9999999999999991, 3.0000000000000013, 4.000000000000002, 5.000000000000001, 5.999999999999998]

DM:9.663546088957395e-30

SKO: 1.3902191258184729e-15

Cube

-3.3861802251067274e-15 \* x \*\* 3 + 3.11972669919669e-14 \* x \*\* 2 + 3.11972669919669e-14 \* x + 1.0000000000000617

[2.0000000000000053, 2.999999999999991, 3.9999999999999987, 5.000000000000007, 5.999999999999998]

DM:1.6270256170183368e-28

SKO: 5.7044291861996796e-15

Exponent

1.654394667446235 \* e \*\* (x \* 0.27080502011022034)

[2.1689435423954007, 2.8435271115570075, 3.727919273191352, 4.887374574678087, 6.4074429950736045]

DM:0.30574788789901286

SKO: 0.2472844062608934

Power

1.9425953940475515 \* x \*\* 0.6807587004469576

[1.9425953940475515, 3.1139464751699033, 4.103805791081051, 4.991601792084609, 5.810483422647173]

DM:0.06304179323788428

SKO: 0.11228694780595319

Logarithm

2.4215720670630736 \* ln(x) + 1.681348744876776

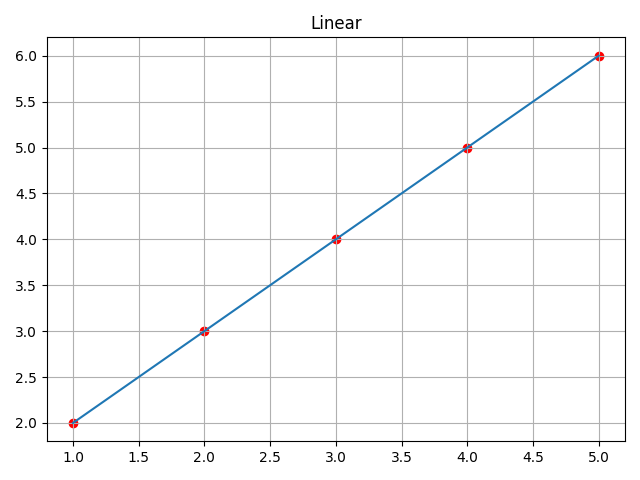
[1.681348744876776, 3.359854595684264, 4.341717575647705, 5.038360446491753, 5.578718637299499]

DM:0.5267543643470737

SKO: 0.32457799196713066

Coeff Pirson: 1.0

Best aprox is: Linear



Пример 2:

Input c - for console, f - for file: f

Linear

1.6853827687383325 \* x + 1.2167884769271708

[3.070709522539337, 5.093168845025335, 7.452704721259002, 8.801010936249668, 10.317855428114166, 12.67739130434783, 13.857159242464665]

DM:0.47301979194451904

SKO: 0.2599504875780658

plt.cla()

plt.show()

Square

-0.0588529246286681 \* x \*\* 2 + 2.1973859448783544 \* x + 0.37425996066499967

[2.7201724612305016, 5.11691566259956, 7.698891418548445, 9.070724988887067, 10.523992780836153, 12.595125151008196, 13.544177536890077]

DM:0.06900821293943576

SKO: 0.09928905344601241

Cube

-0.006041946721071601 \* x \*\* 3 + 0.01910703984150773 \* x \*\* 2 + 0.01910703984150773 \* x + 0.6397716678960742

[2.757914189278382, 5.064652923627468, 7.669249667665119, 9.079563878973712, 10.56968031028133, 12.62426422873726, 13.504674801436728]

DM:0.05939871933907765

SKO: 0.09211694379512354

Exponent

2.7309451573402184 \* e \*\* (x \* 0.23455048222905683)

[3.534787877599819, 4.683819314207406, 6.504436906314325, 7.846950074758005, 9.691220424727504, 13.458233029318226, 15.859621922259143]

DM:10.707089854491738

SKO: 1.2367636253251433

Power

2.5420901787906547 \* x \*\* 0.8380361310314193

[2.753464734158213, 5.108953488908082, 7.609647285228917, 8.966163221224862, 10.446321450059482, 12.672535048391234, 13.757005419231742]

DM:0.1543956454706496

SKO: 0.14851437903961143

Logarithm

5.6500370035357905 \* ln(x) + 1.1988754276365265

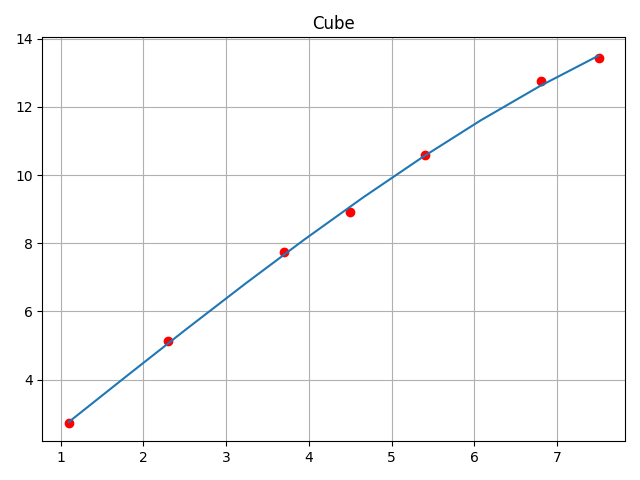
[1.737381470344612, 5.904842792802404, 8.591004271600355, 9.696968375604259, 10.727091918032356, 12.029559119379659, 12.583152052236358]

DM:4.199777952402651

SKO: 0.7745761930983062

Coeff Pirson: 0.9974189309974396

Best aprox is: Cube



Пример 3:

Input c - for console, f - for file: c

Enter N:

25.12

Wrong value

Вывод

Приятная лабораторная работа: очень понятные идеи, аппроксимация всех видов строится на основе решения СЛАУ, много разноцветных графиков и текста в выводе - эти и многие другие причины наслаждаться итогом этой лабораторной.