Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Системное и прикладное программное обеспечение.
Программная инженерия.

Лабораторная работа №4.

Дисциплина: Вычислительная математика.
Преподаватель: Малышева Татьяна Алексеевна.
Выполнил: Бусыгин Иван.
Группа: P3212.
Вариант: 6.

Санкт-Петербург
2022 год

Цель работы.

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

Вычисления вручную.

Аппроксимация функции $y(x) = \frac{2x}{x^4+2}$ строится по десяти точкам:

$$x_1 = 0 \qquad y_1 = \frac{2x_1}{x_1^4+2} = \frac{2\cdot 0}{0^4+2} = 0$$

$$x_2 = 0{,}2 \qquad y_2 = \frac{2x_2}{x_2^4+2} = \frac{2\cdot 0{,}2}{0{,}2^4+2} = \frac{250}{1251} \approx 0{,}19984013$$

$$x_3 = 0{,}4 \qquad y_3 = \frac{2x_3}{x_3^4+2} = \frac{2\cdot 0{,}4}{0{,}4^4+2} = \frac{250}{633} \approx 0{,}39494471$$

$$x_4 = 0{,}6 \qquad y_4 = \frac{2x_4}{x_4^4+2} = \frac{2\cdot 0{,}6}{0{,}6^4+2} = \frac{750}{1331} \approx 0{,}56348610$$

$$x_5 = 0{,}8 \qquad y_5 = \frac{2x_5}{x_5^4+2} = \frac{2\cdot 0{,}8}{0{,}8^4+2} = \frac{500}{753} \approx 0{,}66401062$$

$$x_6 = 1 \qquad y_6 = \frac{2x_6}{x_6^4+2} = \frac{2\cdot 1}{1^4+2} = \frac{2}{3} \approx 0{,}66666667$$

$$x_7 = 1{,}2 \qquad y_7 = \frac{2x_7}{x_7^4+2} = \frac{2\cdot 1{,}2}{1{,}2^4+2} = \frac{750}{1273} \approx 0{,}58915947$$

$$x_8 = 1{,}4 \qquad y_8 = \frac{2x_8}{x_8^4+2} = \frac{2\cdot 1{,}4}{1{,}4^4+2} = \frac{1750}{3651} \approx 0{,}47932073$$

$$x_9 = 1{,}6 \qquad y_9 = \frac{2x_9}{x_9^4+2} = \frac{2\cdot 1{,}6}{1{,}6^4+2} = \frac{1000}{2673} \approx 0{,}37411149$$

$$x_{10} = 1{,}8 \qquad y_{10} = \frac{2x_{10}}{x_{10}^4+2} = \frac{2\cdot 1{,}8}{1{,}8^4+2} = \frac{2250}{7811} \approx 0{,}28805531$$

$$x_{11} = 2 \qquad y_{11} = \frac{2x_{11}}{x_{11}^4+2} = \frac{2\cdot 2}{2^4+2} = \frac{2}{9} \approx 0{,}22222222$$

Аппроксимация линейной функцией: $\varphi_1(x) = ax + b$

$$SX = \sum_{i=1}^{n} x_i = 0 + 0{,}2 + \cdots + 2 = 11$$

$$SXX = \sum_{i=1}^{n} x_i^2 = 0 + 0{,}2^2 + \cdots + 2^2 = 15{,}4$$

$$SY = \sum_{i=1}^{n} y_i \approx 0 + 0{,}19984 + 0{,}19747 + \cdots + 0{,}22222 = 4{,}4418174$$

$$SXY=\sum_{i=1}^{n}x_iy_i=0+0{,}2\cdot 0{,}19984+\cdots+2\cdot 0{,}22222\approx 4{,}6734755$$

$$\begin{cases} aSX+bn=SY \\ aSXX+bSX=SXY \end{cases}$$

$$\Delta=SX^2-SXX\cdot n=11^2-15{,}4\cdot 11=-48{,}4$$

$$\Delta_a=SX\cdot SY-SXY\cdot n\approx 11\cdot 4{,}4418174-4{,}6734755\cdot 11=-2{,}5482391$$

$$\Delta_b=SX\cdot SXY-SXX\cdot SY\approx 11\cdot 4{,}6734755-15{,}4\cdot 4{,}4418174=-16{,}9957575$$

$$a=\frac{\Delta_a}{\Delta}\approx\frac{2{,}5482391}{48{,}4}\approx 0{,}0526496 \qquad b=\frac{\Delta_b}{\Delta}\approx\frac{16{,}9957575}{48{,}4}\approx 0{,}3511520$$

$$\varphi_1(x)=0{,}0526496x+0{,}3511520$$

$$\delta_1=\sqrt{\frac{\sum_{i=1}^{n}(\varphi_1(x_i)-y_i)^2}{n}}=\sqrt{\frac{0{,}3554^2+\cdots+\left(0{,}07403\cdot 2+0{,}3554-\frac{2}{9}\right)^2}{11}}\approx 0{,}20048$$

Аппроксимация квадратичной функцией: $\varphi_2(x)=ax^2+bx+c$

$$\begin{cases} a\sum_{i=1}^{n}x_i^2+b\sum_{i=1}^{n}x_i+cn=\sum_{i=1}^{n}y_i \\ a\sum_{i=1}^{n}x_i^3+b\sum_{i=1}^{n}x_i^2+c\sum_{i=1}^{n}x_i=\sum_{i=1}^{n}x_iy_i \\ a\sum_{i=1}^{n}x_i^4+b\sum_{i=1}^{n}x_i^3+c\sum_{i=1}^{n}x_i^2=\sum_{i=1}^{n}x_i^2y_i \end{cases}$$

$$\sum_{i=1}^{n}x_i=11 \qquad \sum_{i=1}^{n}x_i^2=15{,}4 \qquad \sum_{i=1}^{n}y_i\approx 4{,}4418174 \qquad \sum_{i=1}^{n}x_iy_i\approx 4{,}6734755$$

$$\sum_{i=1}^{n}x_i^3=0+0{,}2^3+\ \ldots+2^3=24{,}2$$

$$\sum_{i=1}^{n}x_i^4=0+0{,}2^4+\ \ldots+2^4=40{,}5328$$

$$\sum_{i=1}^{n}x_i^2y_i\approx 0{,}2^2\cdot 0{,}19984+\cdots+2^2\cdot\frac{2}{9}\approx 5{,}9334450$$

$$\begin{cases} 15{,}4a+11b+11c=4{,}4418175 \\ 24{,}2a+15{,}4b+11c=4{,}6734755 \\ 40{,}5328a+24{,}2b+15{,}4c=5{,}9334450 \end{cases}$$

$$\Delta \approx \begin{vmatrix} 15{,}4 & 11 & 11 \\ 24{,}2 & 15{,}4 & 11 \\ 40{,}5328 & 24{,}2 & 15{,}4 \end{vmatrix} \approx -66{,}44352$$

$$\Delta_a \approx \begin{vmatrix} 4{,}441818 & 11 & 11 \\ 4{,}673476 & 15{,}4 & 11 \\ 5{,}933445 & 24{,}2 & 15{,}4 \end{vmatrix} \approx 36{,}223344 \qquad a = \frac{\Delta_a}{\Delta} \approx \frac{36{,}223344}{-66{,}44352} \approx -0{,}545175$$

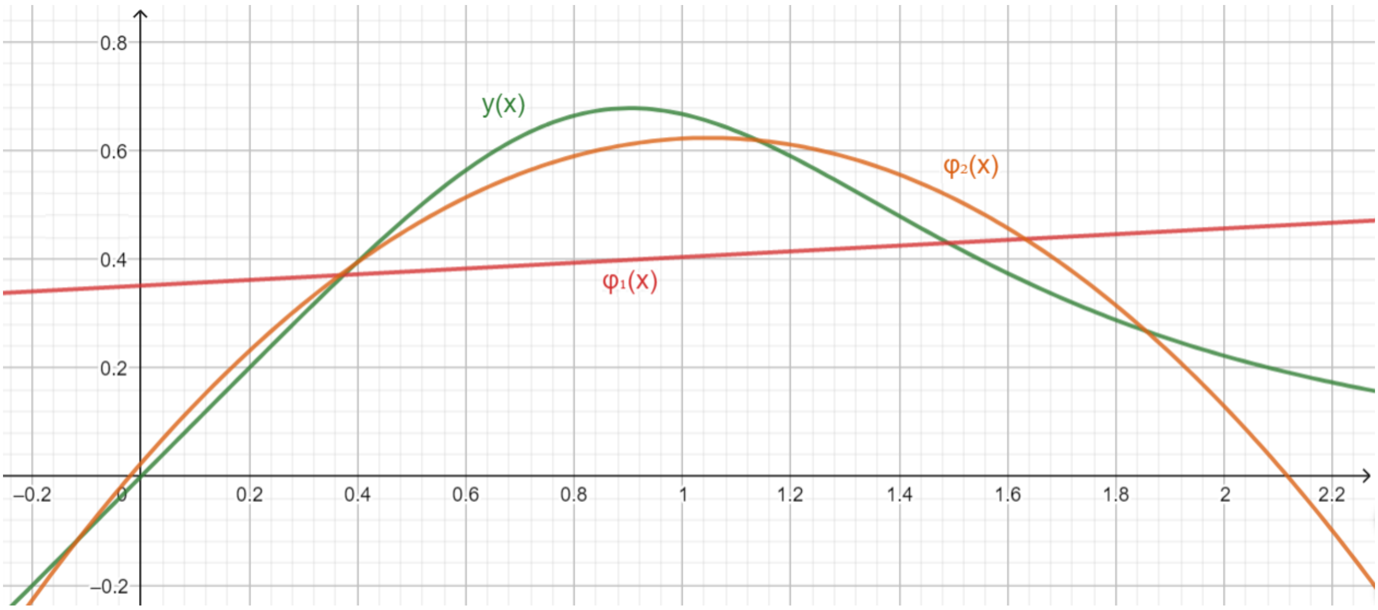
$$\Delta_b \approx \begin{vmatrix} 15{,}4 & 4{,}441818 & 11 \\ 24{,}2 & 4{,}673476 & 11 \\ 40{,}5328 & 5{,}933445 & 15{,}4 \end{vmatrix} \approx -75{,}944909 \qquad b = \frac{\Delta_b}{\Delta} \approx \frac{-75{,}944909}{-66{,}44352} \approx 1{,}142999$$

$$\Delta_c \approx \begin{vmatrix} 15{,}4 & 11 & 4{,}441818 \\ 24{,}2 & 15{,}4 & 4{,}673476 \\ 40{,}5328 & 24{,}2 & 5{,}933445 \end{vmatrix} \approx -1{,}597775 \qquad c = \frac{\Delta_c}{\Delta} \approx \frac{-1{,}597775}{-66{,}44352} \approx 0{,}024047$$

$$\varphi_2(x) = -0{,}545175x^2 + 1{,}142999x + 0{,}024047$$

$$\delta_2 = \sqrt{\frac{\sum_{i=1}^{n}(\varphi_2(x_i) - y_i)^2}{n}} =$$

$$= \sqrt{\frac{0{,}024047^2 + \cdots + \left(-0{,}545175 \cdot 2^2 + 1{,}142999 \cdot 2 + 0{,}024047 - \frac{2}{9}\right)^2}{11}} \approx 0{,}05571$$



Как и ожидалось, $\delta_1 > \delta_2$. А значит квадратичная аппроксимация точнее.

Программная реализация.

```
X = np.array(X)
Y = np.array(Y)
n = len(X)

if n < 3:
  print('Ошибка! Вы задали слишком мало точек. Ожидались данные о хотя бы трёх точках.')
  finish()

if np.any(X <= 0):
  print('Ошибка! Значения x должны быть положительными.')
  finish()

if np.any(Y <= 0):
  print('Ошибка! Значения y должны быть положительными.')
  finish()


SX = np.array([n, X.sum()])
Xi = X.copy()
for i in range(2, 7):
  Xi *= X
  SX = np.append(SX, Xi.sum())

SXY = [Y.sum()]
XYi = Y.copy()
for i in range(1, 4):
  XYi *= X
  SXY.append(XYi.sum())


useFile = input("Желаете ли вы использовать файл для вывода данных? Если да, введите 'y': ")
if useFile == 'y':
  path = input('Введите путь к файлу: ')
  output = open(path, 'w', encoding = 'utf-8')
else:
  output = sys.stdout
  print()


def polynomialApproximation(n, SX, SXY):
  A = np.fromfunction(lambda i, j: SX[(n - j + i).astype(int)], (n + 1, n + 1))
  d = np.array([SXY[i] for i in range(n + 1)])
  return [e for e in np.linalg.solve(A, d)]

def deviation(φ, X, Y):
  D = np.array([φ(x) for x in X]) - Y
  return ((D * D).sum() / n)**0.5

def format1(x):
  return str(round(x, 4))

def format2(x):
```

```
  x = round(x, 4)
  return '+ ' + str(x) if x >= 0 else '- ' + str(-x)


# Линейная аппроксимация
(a1, b1) = polynomialApproximation(1, SX, SXY)
φ1 = lambda x: a1 * x + b1
δ1 = deviation(φ1, X, Y)
label1 = 'φ₁(x) = ' + format1(a1) + ' · x ' + format2(b1)

normX = X - X.sum() / n
normY = Y - Y.sum() / n
r = (normX * normY).sum() / ((normX * normX).sum() * (normY *
normY).sum())**0.5

print('Линейная аппроксимация:', file = output)
print(' ' + label1, file = output)
print(' δ₁ =', δ1, file = output)
print(' r =', r, file = output)

best_φ_label = label1
best_φ = φ1
min_δ = δ1

# Квадратичная аппроксимация
(a2, b2, c2) = polynomialApproximation(2, SX, SXY)
φ2 = lambda x: a2 * x**2 + b2 * x + c2
δ2 = deviation(φ2, X, Y)
label2 = 'φ₂(x) = ' + format1(a2) + ' · x² ' + format2(b2) + ' · x ' +
format2(c2)

print('\nКвадратичная аппроксимация:', file = output)
print(' ' + label2, file = output)
print(' δ₂ =', δ2, file = output)

if δ2 < min_δ:
  best_φ_label = label2
  best_φ = φ2
  min_δ = δ2

# Кубическая аппроксимация
(a3, b3, c3, d3) = polynomialApproximation(3, SX, SXY)
φ3 = lambda x: a3 * x**3 + b3 * x**2 + c3 * x + d3
δ3 = deviation(φ3, X, Y)
label3 = 'φ₃(x) = ' + format1(a3) + ' · x³ ' + format2(b3) + ' · x² ' +
format2(c3) + ' · x ' + format2(d3)

print('\nКубическая аппроксимация:', file = output)
print(' ' + label3, file = output)
print(' δ₃ =', δ3, file = output)

if δ3 < min_δ:
  best_φ_label = label3
  best_φ = φ3
  min_δ = δ3

# Аппроксимация степенной функцией
```

```
lnX = np.log(X)
lnY = np.log(Y)
SlnX = np.array([n, lnX.sum(), (lnX * lnX).sum()])
SlnXlnY = [lnY.sum(), (lnX * lnY).sum()]

(a4, b4) = polynomialApproximation(1, SlnX, SlnXlnY)
(a4, b4) = (np.exp(b4), a4)
φ4 = lambda x: a4 * x**b4
δ4 = deviation(φ4, X, Y)
label4 = 'φ₄(x) = ' + format1(a4) + ' · x^(' + format1(b4) + ')'

print('\nАппроксимация степенной функцией:', file = output)
print(' ' + label4, file = output)
print(' δ₄ =', δ4, file = output)

if δ4 < min_δ:
  best_φ_label = label4
  best_φ = φ4
  min_δ = δ4

# Аппроксимация показательной функцией
SXlnY = [lnY.sum(), (X * lnY).sum()]

(a5, b5) = polynomialApproximation(1, SX, SXlnY)
(a5, b5) = (np.exp(b5), a5)
φ5 = lambda x: a5 * np.exp(b5 * x)
δ5 = deviation(φ5, X, Y)
label5 = 'φ₅(x) = ' + format1(a5) + ' · exp(' + format1(b5) + ' · x)'

print('\nАппроксимация показательной функцией:', file = output)
print(' ' + label5, file = output)
print(' δ₅ =', δ5, file = output)

if δ5 < min_δ:
  best_φ_label = label5
  best_φ = φ5
  min_δ = δ5

# Аппроксимация логарифмической функцией
SlnXY = [Y.sum(), (lnX * Y).sum()]

(a6, b6) = polynomialApproximation(1, SlnX, SlnXY)
φ6 = lambda x: a6 * np.log(x) + b6
δ6 = deviation(φ6, X, Y)
label6 = 'φ₆(x) = ' + format1(a6) + 'ln(x) ' + format2(b6)

print('\nАппроксимация логарифмической функцией:', file = output)
print(' ' + label6, file = output)
print(' δ₆ =', δ6, file = output)

if δ6 < min_δ:
  best_φ_label = label6
  best_φ = φ6
```

Пример работы программы.

```
Желаете ли вы использовать файл с входными данными? Если да, введите 'y': y
Введите путь к файлу: data.txt
Желаете ли вы использовать файл для вывода данных? Если да, введите 'y':

Линейная аппроксимация:
 φ₁(x) = 0.0427 · x + 1.3339
 δ₁ = 0.19134246568663293
 r = 0.13390265196904452

Квадратичная аппроксимация:
 φ₂(x) = -0.5626 · x² + 2.2929 · x - 0.7101
 δ₂ = 0.05296390421642438

Кубическая аппроксимация:
 φ₃(x) = 0.2442 · x³ - 2.0281 · x² + 5.0632 · x - 2.3427
 δ₃ = 0.03217696032045039

Аппроксимация степенной функцией:
 φ₄(x) = 1.2877 · x^(0.1361)
 δ₄ = 0.18755645323068276

Аппроксимация показательной функцией:
 φ₅(x) = 1.2968 · exp(0.0402 · x)
 δ₅ = 0.1923505580484659

Аппроксимация логарифмической функцией:
 φ₆(x) = 0.1625ln(x) + 1.3148
 δ₆ = 0.18571146214752174

Программа завершила работу.
```

Вывод.

Узнал изнутри, как работает расчёт аппроксимирующей функции того или иного вида.