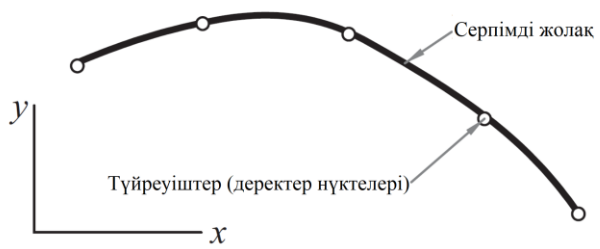
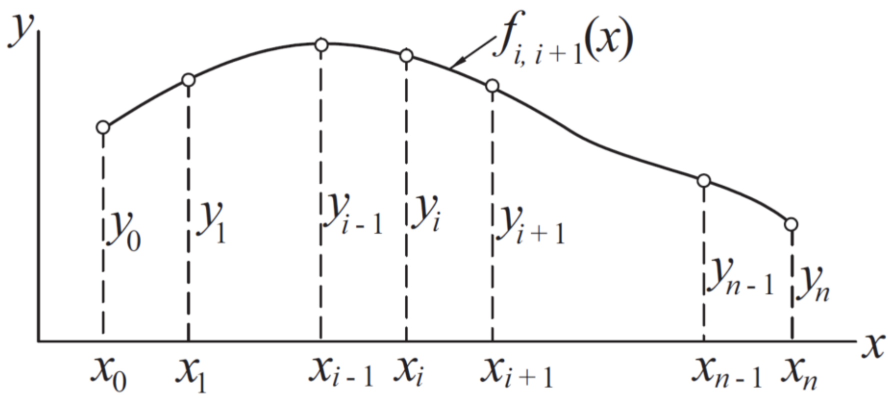
Кубтық сплайн көмегімен интерполяция

Егер деректердің бірнеше көбірек нүктелері бар болса, кубтық сплайнды жалпы интерполянт ретінде қолдану тиімді. Ол деректер нүктелері арасында тербеліске бейімділігі аз болғандықтан, полиномға қарағанда айтарлықтай жақсы.



1-сурет. Кубтық сплайнның механикалық моделі.

Кубтық сплайнның механикалық моделі 1-суретте көрсетілген. Бұл деректер нүктелеріне түйреуіштермен бекітілген жұқа серпімді арқалық. Арқалық түйреуіштер арасында жүктелмегендіктен, сплайн қисығының әрбір сегменті кубтық полином болып табылады — арқалық теориясынан $d^4y/dx^4 = q/(E \cdot I)$, $q = 0$ болғандықтан, $y(x)$ кубтық шама болатынын еске түсіріңіз, мұндағы E – арқалық жасалған материал үшін Юнгтің серпімділік модулі, I – арқалықтың көлденең қимасының инерция моменті. Түйреуіштерде көлбеу және иілу моменті (демек, екінші туынды) үзіліссіз. Екі шеткі түйреуіштерде иілу моменті жоқ, демек, сплайнның екінші туындысы шеткі нүктелерде нөлге тең. Бұл шеттік жағдайлар сәулелік модельде табиғи түрде орын алатындықтан, алынған қисық кубтық сплайн ретінде белгілі. Түйреуіштер (яғни деректер нүктелері) сплайн түйіндері деп аталады.



2-сурет. Кубтық сплайн.

2-суретте $n + 1$ түйінді қамтитын кубтық сплайн көрсетілген. Біз i және $i + 1$ түйіндері арасындағы кесіндіні қамтитын кубтық көпмүше үшін $f_{i,i+1}(x)$ белгілеуін қолданамыз. Назар аударыңыз, сплайн бөлік кубтық қисықтан біріктірілген n кубтан $f_{0,1}(x), f_{1,2}(x), \dots, f_{n-1,n}(x)$ тұрады, олардың барлығының коэффициенттері әртүрлі.

i түйініндегі сплайнның екінші туындысын k_i арқылы белгілей отырып, екінші туындылардың үзіліссіздігі мынаны талап етеді:

$$f''_{i-1,i}(x_i) = f''_{i,i+1}(x_i) = k_i \tag{1}$$

Бұл кезеңде әрбір k белгісіз, тек шеткі жағдайда ғана

$$k_0 = k_n = 0$$

$f_{i,i+1}(x)$ коэффициенттерін есептеудің бастапқы нүктесі $f''_{i,i+1}(x)$ үшін өрнек болып табылады, оның сызықты екенін білеміз. Лагранждың екі нүктелі интерполяциясын қолдана отырып, келесіні жаза аламыз

$$f''_{i,i+1}(x) = k_i l_i(x) + k_{i+1} l_{i+1}(x)$$

мұндағы

$$l_i(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} \qquad l_{i+1}(x) = \frac{x - x_i}{x_{i+1} - x_i}$$

Осыдан,

$$f''_{i,i+1}(x) = \frac{k_i(x - x_{i+1}) - k_{i+1}(x - x_i)}{x_i - x_{i+1}} \tag{2}$$

x бойынша екі рет интегралдасақ, мынаны аламыз

$$f_{i,i+1}(x) = \frac{k_i(x - x_{i+1})^3 - k_{i+1}(x - x_i)^3}{6(x_i - x_{i+1})} + A(x - x_{i+1}) - B(x - x_i) \tag{3}$$

мұндағы A және B интегралдау тұрақтылары. Интегралдаудан туындайтын мүшелер әдетте $Cx + D$ түрінде жазылады. $C = A - B$ және $D = -Ax_{i+1} + Bx_i$ деп қарастыру арқылы біз (3) теңдеудің соңғы екі мүшесін аламыз және оларды келесі есептеулерде қолдануға ыңғайлы. $f_{i,i+1}(x_i) = y_i$ шартын қойып, (3) теңдеуден келесі теңдікті аламыз

$$\frac{k_i(x_i - x_{i+1})^3}{6(x_i - x_{i+1})} + A(x_i - x_{i+1}) = y_i$$

Осыдан,

$$A = \frac{y_i}{(x_i - x_{i+1})} - \frac{k_i}{6}(x_i - x_{i+1}) \tag{4}$$

Сол сияқты $f_{i,i+1}(x_{i+1}) = y_{i+1}$ теңдігінен

$$B = \frac{y_{i+1}}{(x_i - x_{i+1})} - \frac{k_{i+1}}{6}(x_i - x_{i+1}) \tag{5}$$

аламыз. (4) пент (5)-ті (3)-ке қоя отырып, нәтижесінде

$$\begin{aligned} f_{i,i+1}(x) = {} & \frac{k_i}{6}\left[\frac{(x - x_{i+1})^3}{x_i - x_{i+1}} - (x - x_{i+1})(x_i - x_{i+1})\right] \\ & - \frac{k_{i+1}}{6}\left[\frac{(x - x_i)^3}{(x_i - x_{i+1})} - (x - x_i)(x_i - x_{i+1})\right] \\ & + \frac{y_i(x - x_{i+1}) - y_{i+1}(x - x_i)}{(x_i - x_{i+1})} \end{aligned} \tag{6}$$

шығады.

Ішкі түйіндердегі сплайнның k_i екінші туындылары көлбеудің $f'_{i-1,i}(x_i) = f'_{i,i+1}(x_i)$ үзіліссіздік шарттарынан алынған, мұндағы $i = 1, 2, \ldots, n-1$. Кішкентай түрлендіруден кейін мына теңдеулер пайда болады.

$$\begin{aligned} & k_{i-1}(x_{i-1} - x_i) + 2k_i(x_{i-1} - x_{i+1}) + k_{i+1}(x_i - x_{i+1}) \\ & = 6\left(\frac{y_{i-1} - y_i}{x_{i-1} - x_i} - \frac{y_i - y_{i+1}}{x_i - x_{i+1}}\right), \qquad i = 1, 2, \ldots, n-1 \end{aligned} \tag{7}$$

(7) теңдеулер үшдиагональды коэффициентті матрица, оларды алдыңғы сабақтарда сипатталған LU жіктеу әдісі арқылы үнемді шешуге болады.

Егер деректер нүктелері h аралықта біркелкі орналасса, онда $x_{i-1} - x_i = x_i - x_{i+1} = -h$ және (7) теңдеулерді мына түрде ықшамдаймыз

$$k_{i-1} + 4k_i + k_{i+1} = \frac{6}{h^2}(y_{i-1} - 2y_i + y_{i+1}), \qquad i = 1, 2, \ldots, n-1 \tag{8}$$

<u>cubicSpline</u>

Кубтық сплайн интерполяциясының бірінші кезеңі (7) теңдеулерді реттеу болып табылады және оларды белгісіз k үшін шешу ($k_0 = k_n = 0$ екенін ескеріңіз). Бұл тапсырма curvatures функциясы арқылы орындалады. Екінші кезең – (6) теңдеуден x нүктесінде интерполянтты есептеу. Бұл қадамды evalSpline функциясы арқылы x-тің әртүрлі мәндерімен кез келген рет қайталауға болады. evalSpline ішіндегі findSegment функциясы екіге бөлу әдісін қолданып, құрамында x бар сплайн сегментін табады. Ол сегмент нөмірін қайтарады; яғни (6) теңдеудегі i мәнін табады.

```
## module cubicSpline
''' k = curvatures(xData,yData).
    Кубтық сплайнның түйіндеріндегі қисықтықты қайтарады.

    y = evalSpline(xData,yData,k,x).
    Кубтық сплайнды x нүктесінде есептейді.
    k қисықтықтар «curvatures» функциясымен есептелінеді.
'''
import numpy as np

def LUdecomp3(c,d,e):
    n  =  len(d)
    for  k  in  range(1,n):
        lam  =  c[k-1]/d[k-1]
        d[k]  =  d[k]  - lam*e[k-1]
        c[k-1] = lam
    return c,d,e

def LUsolve3(c,d,e,b):
    n  = len(d)
    for  k  in  range(1,n):
        b[k]  =  b[k]  - c[k-1]*b[k-1]
    b[n-1]   =   b[n-1]/d[n-1]
    for  k  in  range(n-2,-1,-1):
        b[k]  =  (b[k]  - e[k]*b[k+1])/d[k]
    return b
def curvatures(xData,yData):
    n = len(xData) - 1
    c = np.zeros(n)
    d = np.ones(n+1)
    e = np.zeros(n)
    k = np.zeros(n+1)
    c[0:n-1] = xData[0:n-1] - xData[1:n]
```

```
    d[1:n] = 2.0*(xData[0:n-1] - xData[2:n+1])
    e[1:n] = xData[1:n] - xData[2:n+1]
    k[1:n] = 6.0*(yData[0:n-1] - yData[1:n])/(xData[0:n-1] - xData[1:n]) \
           - 6.0*(yData[1:n] - yData[2:n+1])/(xData[1:n] - xData[2:n+1])
    LUdecomp3(c,d,e)
    LUsolve3(c,d,e,k)
    return k

def evalSpline(xData,yData,k,x):
    def findSegment(xData,x):
        iLeft = 0
        iRight = len(xData)-1
        while True:
            if (iRight-iLeft) <= 1: return iLeft
            i = (iLeft + iRight)//2
            if x < xData[i]: iRight = i
            else: iLeft = i

    i = findSegment(xData,x)
    h = xData[i] - xData[i+1]
    y = ((x - xData[i+1])**3/h - (x - xData[i+1])*h)*k[i]/6.0 \
        - ((x - xData[i])**3/h - (x - xData[i])*h)*k[i+1]/6.0 \
        + (yData[i]*(x - xData[i+1])- yData[i+1]*(x - xData[i]))/h
    return y
```

МЫСАЛ 1

$x = 1{,}5$ кезінде **у**-ті анықтау үшін кубтық сплайнды пайдаланыңыз. Деректер нүктелері

x	1	2	3	4	5
y	0	1	0	1	0

Шешуі. Бес түйін $h = 1$ нүктесінде бірдей қашықтықта орналасқан. Сплайнның екінші туындысы бірінші және соңғы түйінде нөлге тең болатынын еске түсірсек, бізде $k_0 = k_4 = 0$ болады. Басқа түйіндердегі екінші туындылар теңдеуден алынады. i = 1, 2, 3 мәнін пайдаланғанда мына теңдеулер шығады.

$$0 + 4k_1 + k_2 = 6(0 - 2 \cdot 1 + 0) = -12$$
$$k_1 + 4k_2 + k_3 = 6(1 - 2 \cdot 0 + 1) = 12$$
$$k_2 + 43 + 0 = 6(0 - 2 \cdot 1 + 0) = -12$$

Шешімі $k_1 = k_3 = -30/7, \;\; k_2 = 36/7$.

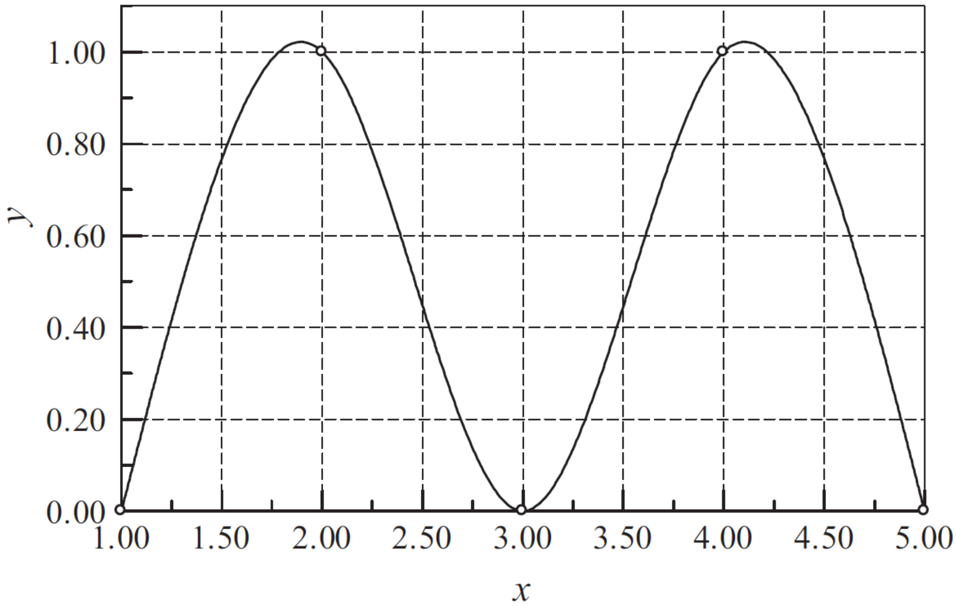
x=1.5 нүктесі 0 және 1 түйіндерінің арасындағы кесіндіде жатыр. Сәйкес интерполянт (6) теңдеуден $i = 0$ орнату арқылы алынады. $x_i - x_{i+1} = -h = -1$ болғанда, (6) теңдеуден келесіні аламыз.

$$f_{0,1}(x) = -\frac{k_0}{6}[(x - x_1)^3 - (x - x_1)] + \frac{k_1}{6}[(x - x_0)^3 - (x - x_0)]$$
$$- [y_0(x - x_1) - y_1(x - x_0)]$$

Осыдан

$$y(1.5) = f_{0,1}(1.5) = 0 + \frac{1}{6}\left(-\frac{30}{7}\right)[(1.5 - 1)^3 - (1.5 - 1)] - [0 - 1(1.5 - 1)] = 0.7679$$

Бұл жағдайда төрт кубтық сегменттен тұратын интерполянттың графигі суретте көрсетілген.



Кубтық сплайнмен берілген деректер нүктелері арасында интерполяция жасайтын бағдарламаны жазу үшін cubicSpline модулін пайдаланыңыз. Бағдарлама x-тің бірден көп мәні үшін интерполянтты есептей алуы керек. Сынақ ретінде жоғарыдағы мысалда көрсетілген деректер нүктелерін пайдаланыңыз және интерполянтты x = 1,5 және x = 4,5 мәндерінде есептеңіз (симметрияға байланысты бұл мәндер тең болуы керек).

```
import numpy as np
from cubicSpline import *
xData = np.array([1,2,3,4,5],float)
yData = np.array([0,1,0,1,0],float)
k = curvatures(xData,yData)
while True:
    try: x = eval(input("\nx ==> "))
    except SyntaxError: break
    print("y =",evalSpline(xData,yData,k,x))
input("Done. Press return to exit")

x ==> 1.5
y = 0.767857142857

x ==> 4.5
y = 0.767857142857
```

```
x ==>
Done. Press return to exit
```

Ең кіші квадраттар әдісімен сәйкестендіру

Шолу

Егер деректер эксперименттерден алынса, олар әдетте өлшеу қателерінен туындаған кездейсоқ шудың айтарлықтай мөлшерін қамтиды. Қисық сызықтарды сәйкестендіру есебі – деректер нүктелеріне «орташа» сәйкес келетін тегіс қисық сызықты табу. Бұл қисық шуды қайталамау үшін қарапайым пішінге ие болуы керек (мысалы, төменгі ретті көпмүше).

$$f(x) = f(x; a_0, a_1, \dots, a_m)$$

$n + 1$ деректер нүктелеріне (x_i, y_i) $\quad i = 0,1, \dots, n$ орнатылатын функция болсын. Бұл белгілеу x-тан тәуелді функциясы $a_0, a_1, \dots, a_m$ $m + 1$ айнымалы параметрлерден тұратын білдіреді мұндағы $m < n$. Әдетте деректер алынған экспериментке байланысты теориядан $f(x)$ формасы алдын-ала анықталады. Сәйкестендіруді реттеудің жалғыз құралы-параметрлер. Мысалы, егер деректер t_i уақыт кезінде шамадан тыс кемитін масса-серіппелі жүйенің y_i ығысуын білдірсе, теория $f(t) = a_0 t e^{-a_1 t}$ таңдауды ұсынады. Осылайша, қисықты сәйкестендіру екі кезеңнен тұрады: $f(x)$ форпмасын таңдау, содан кейін деректерге ең жақсы сәйкестікті қамтамасыз ететін параметрлерді есептеу.

Бұл бізді мына сұраққа әкеледі: «ең жақсы» сәйкестік дегеніміз не? Егер шу y-координатасымен шектелсе, ең көп қолданылатын өлшем – бұл функцияны барынша азайтатын ең кіші квадраттар сәйкестендіру

$$S(a_0, a_1, \dots, a_m) = \sum_{i=0}^{n} [y_i - f(x_i)]^2 \qquad (1)$$

әр a_j-ге қатысты болып табылады. Сондықтан параметрлердің тиімді мәндері мына теңдеулер шешімімен анықталады

$$\frac{\partial S}{\partial a_k} = 0, \qquad k = 0,1, \dots, m \qquad (2)$$

Теңдеудегі $r_i = y_i - f(x_i)$ мүшелері (1) қалдықтар деп аталады; олар деректер нүктелері мен x_i-ге сәйкес келетін функция арасындағы сәйкессіздікті көрсетеді. Осылайша, азайтылатын S функция қалдық квадраттарының қосындысы болып табылады. (2) теңдеулер әдетте a_j-де сызықты емес, сондықтан оларды шешу қиын болуы мүмкін. Көбінесе сәйкестік функциясы берілген $f_j(x)$ функциялардың сызықтық комбинациясы ретінде таңдалады,

$$f(x) = a_0 f_0(x) + a_1 f_1(x) \dots + a_m f_m(x)$$

бұл жағдайда (1) теңдеу сызықты болады. Егер сәйкестік функциясы көпмүшелік болса, онда бізде $f_0(x) = 1,\ f_1(x) = x,\ f_2(x) = x^2$ және т.б.

Сәйкестік қисығы туралы деректердің таралуы сандық түрде анықталған стандартты ауытқумен анықталады

$$\sigma = \sqrt{\frac{S}{n - m}} \qquad (3)$$

Егер $n = m$ болса, онда бізде қисықты сәйкестендіру емес, интерполяция болатынын екенін ескеріңіз. Бұл жағдайда теңдеудегі (3) алымы да, бөлімі де нөлге тең, сондықтан σ анықталмағандық.

Түзу сызықты сәйкестендіру

Түзу сызықты сәйкестендіру

$$f(x) = a + bx \tag{4}$$

деректер теориясында сызықтық регрессия ретінде де белгілі. Бұл жағдайда минимизацияланатын функция мына түрде болады

$$S(a,b) = \sum_{i=0}^{n} [y_i - f(x_i)]^2 = \sum_{i=0}^{n} (y_i - a - bx_i)^2$$

(2) теңдеу түрі келесідей болады

$$\frac{\partial S}{\partial a} = \sum_{i=0}^{n} -2(y_i - a - bx_i) = 2\left[a(n+1) + b\sum_{i=0}^{n} x_i - \sum_{i=0}^{n} y_i\right] = 0$$

$$\frac{\partial S}{\partial b} = \sum_{i=0}^{n} -2(y_i - a - bx_i)x_i = 2\left[a\sum_{i=0}^{n} x_i + b\sum_{i=0}^{n} x_i^2 - \sum_{i=0}^{n} x_i y_i\right] = 0$$

Екі теңдеуді $2(n+1)$-ге бөліп, мүшелерін қайта реттесек, келесіні аламыз

$$a + \bar{x}b = \bar{y}, \qquad \bar{x}a + \left(\frac{1}{n+1}\sum_{i=0}^{n} x_i^2\right)b = \frac{1}{n+1}\sum_{i=0}^{n} x_i y_i$$

мұндағы

$$\bar{x} = \frac{1}{n+1}\sum_{i=0}^{n} x_i\,, \qquad \bar{y} = \frac{1}{n+1}\sum_{i=0}^{n} y_i \tag{5}$$

x және у деректерінің орташа мәндері болып табылады. Параметрлердің шешімі мына түрде табылады

$$a = \frac{\bar{y}\sum x_i^2 - \bar{x}\sum x_i y_i}{\sum x_i^2 - n\bar{x}^2}, \qquad b = \frac{\sum x_i y_i - \bar{x}\sum y_i}{\sum x_i^2 - n\bar{x}^2} \tag{6}$$

Бұл өрнектер дөңгелектеу қателеріне бейім (әр алымдағы, сондай-ақ әрбір бөлгіштегі екі мүше шамамен тең болуы мүмкін). Сондықтан параметрлерді мынадан есептеген дұрыс

$$b = \frac{\sum y_i(x_i - \bar{x})}{\sum x_i(x_i - \bar{x})}, \qquad a = \bar{y} - \bar{x}b \tag{7}$$

бұлар (6) теңдеулерге тең, бірақ дөңгелектеуге азырақ әсер етеді.

Сызықтық форманы сәйкестендіру

Сызықтық форманың ең кіші квадраттарының сәйкестігін қарастырайық

$$f(x) = a_0 f_0(x) + a_1 f_1(x) \ldots + a_m f_m(x) = \sum_{j=0}^{m} a_j f_j(x) \tag{8}$$

мұндағы әрбір $f_j(x)$ x-тен тәуелді алдын ала анықталған функциясы, оларды базистік функция деп атайды. (1) теңдеудегі алмастыру бізге келесіні береді

$$S = \sum_{i=0}^{n}\left[y_i - \sum_{j=0}^{m} a_j f_j(x_i)\right]^2$$

Осыдан (2) теңдеу

$$\frac{\partial S}{\partial a_k} = -2\left\{\sum_{i=0}^{n}\left[y_i - \sum_{j=0}^{m} a_j f_j(x_i)\right] f_k(x_i)\right\} = 0, \qquad k = 0,1,\dots,m$$

түрінде болады. (-2) тұрақтыға бөліп тастап, қосу ретін ауыстырсақ мынаны аламыз

$$\sum_{j=0}^{m}\left[\sum_{i=0}^{n} f_j(x_i) f_k(x_i)\right] a_j = \sum_{i=0}^{n} f_k(x_i)\, y_i, \qquad k = 0,1,\dots,m$$

Матрицалық белгілерде бұл теңдеулер

$$\boldsymbol{Aa} = \boldsymbol{b} \tag{9а}$$

мұндағы

$$A_{kj} = \sum_{i=0}^{n} f_j(x_i) f_k(x_i) \qquad b_k = \sum_{i=0}^{n} f_k(x_i)\, y_i \tag{9ә}$$

(9а) теңдеуін ең кіші квадраттардың нормаль теңдеулері ретінде белгілі оны сызықты алгебралық теңдеулер жүйесінде қарастырылған әдістермен шешуге болады. Коэффициент матрицасы симметриялы екенін ескеріңіз (яғни, $A_{kj} = A_{jk}$).

Көпмүшелікті сәйкестендіру

Әдетте жиі қолданылатын сызықтық форма көпмүше болып табылады. Егер көпмүшенің дәрежесі m болса, бізде $f(x) = \sum_{j=0}^{m} a_j x^j$ болады. Мұнда базистік функциялар

$$f_j(x) = x^j, \qquad j = 0,1,\dots,m \tag{10}$$

болып табылады, сондықтан (9ә) теңдеулер

$$A_{kj} = \sum_{i=0}^{n} x_i^{j+k} \qquad b_k = \sum_{i=0}^{n} x_i^k\, y_i \tag{9ә}$$

немесе

$$\mathbf{A} = \begin{bmatrix} n & \sum x_i & \sum x_i^2 & \dots & \sum x_i^m \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \dots & \sum x_i^{m+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum x_i^{m-1} & \sum x_i^m & \sum x_i^{m+1} & \dots & \sum x_i^{2m} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \vdots \\ \sum x_i^m y_i \end{bmatrix} \tag{11}$$

мұндағы $\sum \quad - \sum_{i=0}^{n}$ дегенді білдіреді. Нормаль теңдеулер m үлкейген сайын біртіндеп нашарлайды. Бақытымызға орай, бұл практикалық тұрғыдан маңызды емес, өйткені қисықты сәйкестендіру кезінде тек төмен ретті көпмүшелер пайдалы. Жоғары ретті көпмүшелер ұсынылмайды, өйткені олар мәліметтерге тән шуды шығаруға бейім.

`polyFit` функциясы

Бұл модульдегі `polyFit` функциясы дәрежелі көпмүшенің коэффициенттері үшін нормаль теңдеулерді тұрғызып, шешеді. Ол көпмүшенің коэффициенттерін тауып береді. Есептеулерді жеңілдету үшін (11) теңдеудегі коэффициент матрицасын құрайтын n мүшелер, $\sum x_i, \sum x_i^2, \dots, \sum x_i^{2m}$ алдымен s векторында сақталады, содан кейін A ішіне енгізіледі. Содан кейін нормаль теңдеулер Гаусстың біртіндеп жою әдісімен шешіледі. Шешім табылғаннан кейін стандартты ауытқу σ `std-Dev` функциясымен есептелінеді `stdDev` ішіндегі полиномды бағалау ендірілген `evalPoly` функциясы арқылы жүзеге асырылады — алгоритмнің түсіндірмесі алдыңғы бөлімдерде қарастырпмыз.

Қолданылған әдебиеттер тізімі

1) Шакенов Қ.Қ. Есептеу математикасы әдістері лекциялар курсы.Алматы,2019. – 193б
2) P. Dechaumphai, N. Wansophark. Numerical Methods in Science and Engineering Theories with MATLAB, Mathematica, Fortran, C and Python Programs. Alpha Science International Ltd. 2022
3) Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков Численные методы: Классический университетский учебник. —М.: Издательство «Бином. Лаб. знаний»,2020. — 636 с.
4) Вабищевич П.Н. Численные методы: Вычислительный практикум. — М.: Книжный дом «ЛИБРОКОМ», 2020. — 320 с.

Интернет ресурстар

1) https://docs.python.org/3/
2) http://math-hse.info/f/2018-19/py-polit/instruction_JN.pdf
3) https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html
4) https://colab.research.google.com/
5) https://planetcalc.ru/search/?tag=2874