

### 3 Интерполяция және қисықты сәйкестіндіру

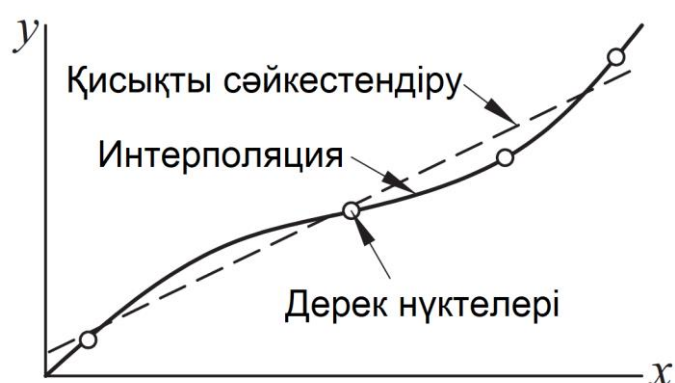
Берілген  $n + 1$  деректер нүктелері  $(x_i, y_i)$ ,  $i = 0, 1, 2, \dots, n$ ,  $y(x)$  мәнін бағалау.

Кіріспе

Дискретті деректер жиыны немесе мына түрдегі кесте

$x_0$	$x_1$	$x_2$	...	$x_n$
$y_0$	$y_1$	$y_2$	...	$y_n$

әдетте техникалық есептеулерге қатысады. Деректер көзі эксперименттік бақылаулар немесе сандық есептеулер болуы мүмкін. Интерполяция мен қисықты сәйкестіндіру арасында айырмашылық бар. Интерполяцияда деректер нүктелері арқылы қисық сызық саламыз. Бұл ретте біз деректер нүктелерінің дәл және анық екендігін жасырын болжам жасаймыз. Керісінше, қисықты сәйкестіндіру әдетте өлшеу қателерінен туындаған шашырау (шуыл) бар деректерге қолданылады. Мұнда біз белгілі бір мағынада деректерді жақындататын тегіс қисық сызығын тапқымыз келеді. Осылайша, қисық міндетті түрде деректер нүктелері арқылы өтпейді. Интерполяция мен қисықты сәйкестіндіру арасындағы айырмашылық 1-суретте көрсетілген.



1-сурет. Интерполяция мен қисықты сәйкестіндіру

### 3.1 Көпмүшелік интерполяция

#### Лагранж әдісі

Интерполянттың қарапайым түрі көпмүше болып табылады. Әрқашан  $n + 1$  нақты деректер нүктелері арқылы өтетін  $n$  дәрежелі жалғыз көпмүшені құруға болады. Бұл көпмүшені алудың бір әдісі – Лагранж формуласы,

	$P_n(x) = \sum_{i=0}^n y_i l_i(x)$	(3.1a)
--	------------------------------------	--------

мұндағы  $n$  индексі көпмүшесінің дәрежесін білдіреді және

	$l_i(x) = \frac{x - x_0}{x_i - x_0} \cdot \frac{x - x_1}{x_i - x_1} \cdot \dots \cdot \frac{x - x_{i-1}}{x_i - x_{i-1}} \cdot \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdot \dots \cdot \frac{x - x_n}{x_i - x_n}$ $= \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \quad i = 0, 1, 2, \dots, n$	(3.1b)
--	---	--------

негізгі функциялар деп аталады.

Мысалы, егер  $n = 1$  болса, интерполянт  $P_1(x) = y_0 l_0(x) + y_1 l_1(x)$  түзу сызығы болады, мұндағы

	$l_0(x) = \frac{x - x_1}{x_0 - x_1} \quad l_1(x) = \frac{x - x_0}{x_1 - x_0}$	
--	---	--

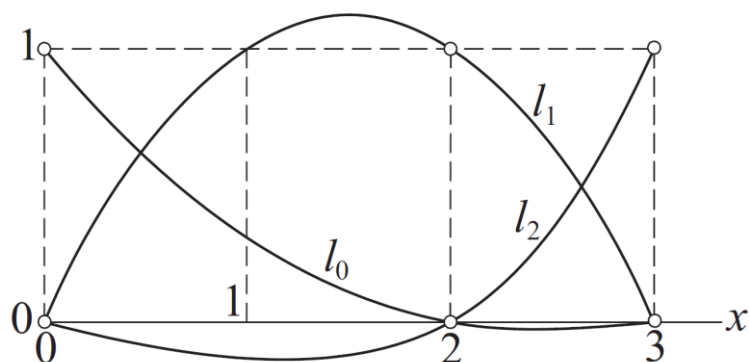
$n = 2$  болғанда интерполяция параболалық сипатта болады:  $P_2(x) = y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x)$ , мұнда енді

	$l_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}$ $l_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}$ $l_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$	
--	--	--

Негізгі функциялар  $n$  дәрежелі көпмүшеліктер болып табылады және мына қасиетке ие

	$l_i(x_j) = \begin{cases} 0 & \text{егер } i \neq j \\ 1 & \text{егер } i = j \end{cases} = \delta_{ij}$	(3.2)
--	--	-------

мұндағы  $\delta_{ij}$  –Кронекер дельтасы. Бұл қасиет  $x_0 = 0, x_1 = 2, x_2 = 3$  болатын үш нүктелі интерполяция ( $n = 2$ ) үшін 2-суретте көрсетілген.



2-сурет. Квадраттық негізгі функциялардың мысалы.

Интерполяциялық көпмүшеліктің деректер нүктелері арқылы өтетінін дәлелдеу үшін  $x = x_j$  (3.1а) теңдеуіне қоямыз, содан кейін (3.2) формуланы пайдаланамыз.

Нәтижесінде

	$P_n(x_j) = \sum_{i=0}^n y_i l_i(x_j) = \sum_{i=0}^n y_i \delta_{ij} = y_j$	
--	---	--

Көпмүшелік интерполяциядағы қателікті көрсетуге болады

	$f(x) - P_n(x_j) = \frac{(x - x_0)(x - x_1) \dots (x - x_n)}{(n + 1)!} f^{(n+1)}(\xi)$	(3.3)
--	--	-------

мұндағы  $\xi - (x_0, x_n)$  интервалында жатыр, оның мәні әзірге белгісіз. Мәліметтер нүктесі  $x$  нүктесінен неғұрлым алыс болса, ол  $x$  қатесіне соғұрлым көп ықпал ететінін ескерген жөн.

### Ньютон әдісі

Лагранж әдісі жалпы тұрғыда қарапайым болғанымен, ол тиімді алгоритм қатарына жатпайды. Неғұрлым жақсы есептеу процедурасы Ньютон әдісімен алынады, мұнда интерполяциялық көпмүше келесі түрде жазылады

$$P_n(x) = a_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})a_n$$

Бұл көпмүше бағалаудың тиімді үрдісін береді. Мысалы, төрт деректер нүктесін ( $n = 3$ ) қарастырайық. Мұнда интерполяциялық көпмүше мына түрде болады

$$\begin{aligned} P_3(x) &= a_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 + (x - x_0)(x - x_1)(x - x_2)a_3 = \\ &= a_0 + (x - x_0)\{a_1 + (x - x_1)[a_2 + (x - x_2)a_3]\} \end{aligned}$$

келесі қайталану қатынастарымен кері бағытта есептелінуі мүмкін:

	$P_0(x) = a_3$ $P_1(x) = a_2 + (x - x_2)P_0(x)$ $P_2(x) = a_1 + (x - x_1)P_1(x)$ $P_3(x) = a_0 + (x - x_0)P_2(x)$	
--	---	--

Кез келген  $n$  үшін біз келесіні аламыз

	$P_0(x) = a_n, \quad P_k(x) = a_{n-k} + (x - x_{n-k})P_{k-1}(x), \quad k = 1, 2, \dots, n$	(3.4)
--	--	-------

Деректер нүктелерінің  $x$ -координаталық массивін `xData` арқылы және көпмүше дәрежесін  $n$  арқылы белгілей отырып,  $P_n(x)$  есептеудің келесі алгоритмі бар:

```

p = a[n]
for k in range(1, n+1):
    p = a[n-k] + (x - xData[n-k]) * p

```

$P_n$  коэффициенттері көпмүшені әрбір деректер нүктесінен өткізу арқылы анықталады:  $y_i = P_n(x_i)$ ,  $i = 1, 2, \dots, n$ . Бұл келесі теңдеулерді береді:

	$y_0 = a_0$ $y_1 = a_0 + (x_1 - x_0)a_1$ $y_2 = a_0 + (x_2 - x_0)a_1 + (x_2 - x_0)(x_2 - x_1)a_2$ $\vdots$ $y_n(x) = a_0 + (x_n - x_0)a_1 + \dots + (x_n - x_0) \dots (x_n - x_{n-1})a_n$	(a)
--	---	-----

Бөлінген айырымдармен танысу

	$\nabla y_i = \frac{y_i - y_0}{x_i - x_0}, \quad i = 1, 2, \dots, n$ $\nabla^2 y_i = \frac{\nabla y_i - \nabla y_1}{x_i - x_1}, \quad i = 2, 3, \dots, n$ $\nabla^3 y_i = \frac{\nabla^2 y_i - \nabla^2 y_2}{x_i - x_2}, \quad i = 3, 4, \dots, n$ $\vdots$ $\nabla^n y_n = \frac{\nabla^{n-1} y_n - \nabla^{n-1} y_{n-1}}{x_n - x_{n-1}}$	(3.5)
--	--	-------

(a) теңдеулерінің шешімі келесі болады

	$a_0 = y_0, a_1 = \nabla y_1, a_2 = \nabla^2 y_2, \dots, a_n = \nabla^n y_n$	(3.6)
--	--	-------

Егер коэффициенттер қолмен есептелетін болса, 1-кестедегі форматпен жұмыс істеу ыңғайлы ( $n = 4$  үшін көрсетілген).

$x_0$	$y_0$				
$x_1$	$y_1$	$\nabla y_1$			
$x_2$	$y_2$	$\nabla y_2$	$\nabla^2 y_2$		
$x_3$	$y_3$	$\nabla y_3$	$\nabla^2 y_3$	$\nabla^3 y_3$	
$x_4$	$y_4$	$\nabla y_4$	$\nabla^2 y_4$	$\nabla^3 y_4$	$\nabla^4 y_4$

1-кесте. Ньютон әдісіне арналған кесте.

Кестедегі диагональды мүшелер ( $y_0, \nabla y_1, \nabla^2 y_2, \nabla^3 y_3, \nabla^4 y_4$ ) көпмүшенің коэффициенттері болып табылады. Егер деректер нүктелері басқа ретпен тізімделсе, кестедегі жазбалар өзгереді, бірақ нәтижесіндегі көпмүше бірдей болады —  $n + 1$  әртүрлі деректер нүктелерін интерполяциялайтын  $n$  дәрежелі көпмүшенің жалғыз екенін ескеріңіз.

Компьютерлік есептеулер келесі алгоритмді қолдана отырып, бір өлшемді массив ішінде жүзеге асырылуы мүмкін (біз  $m = n + 1 =$  дерек нүктелер саны белгісін қолданамыз):

```
a = yData.copy()
for k in range(1,m):
    for i in range(k,m):
        a[i] = (a[i] - a[k-1]) / (xData[i] - xData[k-1])
```

Бастапқыда **a** деректердің **y**-координатасын қамтиды, осылайша ол 1-кестедегі екінші бағанмен бірдей болады. Сыртқы цикл арқылы әрбір өту келесі бағандағы жазбаларды жасайды, олар **a**-ның сәйкес элементтерін қайта жазады. Демек, **a** соңында 1-кестенің диагональды мүшелері (яғни, көпмүшенің коэффициенттері) болады.

### **newtonPoly**

Бұл модульде Ньютон әдісі бойынша интерполяцияға қажетті екі функция бар. **xData** және **yData** деректер нүктесінің массивтерін ескере отырып, **coeffts** функциясы **a** коэффициент массивін қайтарады. Коэффициенттер табылғаннан кейін  $P_n$  интерполяциясын **evalPoly** функциясымен **x**-тің кез келген мәнінде есептеуге болады.

```
## module newtonPoly
```

```

''' p = evalPoly(a, xData, x).
Evaluates Newton's polynomial p at x. The coefficient
vector {a} can be computed by the function 'coeffts'.
a = coeffts(xData, yData).
Computes the coefficients of Newton's polynomial.
p = evalPoly(a, xData, x)
Ньютонаң р көпмүшелігін x-де есептейді.
Кoeffфициент {a} векторын coeffts функциясы
арқылы есептеуге болады.
a = coeffts(xData, yData).
Ньютон көпмүшесінің коoeffфициенттерін есептейді.
'''
def evalPoly(a, xData, x):
    n = len(xData) - 1 # Degree of polynomial
    p = a[n]
    for k in range(1, n+1):
        p = a[n-k] + (x - xData[n-k])*p
    return p
def coeffts(xData, yData):
    m = len(xData) # Number of data points
    a = yData.copy()
    for k in range(1, m):
        a[k:m] = (a[k:m] - a[k-1]) / (xData[k:m] - xData[k-1])
    return a

```

## Невилл әдісі

Ньютонның интерполяция әдісі екі кезеңнен тұрады: коэффициенттерді табу, содан кейін көпмүшенің мәнін есептеу. Интерполяция сол көпмүшені пайдаланып  $x$ -тің әртүрлі мәндерінде қайта-қайта орындалса, ол жақсы жұмыс істейді. Егер бір ғана нүкте интерполяцияланатын болса, Невилл алгоритмі сияқты интерполяцияны бір қадаммен есептейтін әдіс қолайлырақ таңдау болып табылады.

$(x_i, y_i), (x_{i+1}, y_{i+1}), \dots, (x_{i+k}, y_{i+k})$   $k + 1$  деректер нүктелері арқылы өтетін  $k$  дәрежелі көпмүшені  $P_k[x_i, x_{i+1}, \dots, x_{i+k}]$  деп белгілейік. Бір деректер нүктесі үшін біз мына түрде аламыз

	$P_0[x_i] = y_i$	(3.7)
--	------------------	-------

Екі деректер нүктесіне негізделген интерполяция келесідей болады

	$P_1[x_i, x_{i+1}] = \frac{(x - x_{i+1})P_0[x_i] + (x_i - x)P_0[x_{i+1}]}{x_i - x_{i+1}}$	
--	---	--

$P_1[x_i, x_{i+1}]$  екі деректер нүктесі арқылы өтетіні оңай тексеріледі; яғни  $x = x_i$  болғанда  $P_1[x_i, x_{i+1}] = y_i$ , ал  $x = x_{i+1}$  болғанда  $P_1[x_i, x_{i+1}] = y_{i+1}$ .

Үш нүктелі интерполяция

	$P_2[x_i, x_{i+1}, x_{i+2}] = \frac{(x - x_{i+2})P_1[x_i, x_{i+1}] + (x_i - x)P_1[x_{i+1}, x_{i+2}]}{x_i - x_{i+2}}$	
--	--	--

Бұл интерполяция деректер нүктелерімен қиылысатынын көрсету үшін алдымен  $x = x_i$  ауыстырып, мынаны аламыз.

	$P_2[x_i, x_{i+1}, x_{i+2}] = P_1[x_i, x_{i+1}] = y_i$	
--	--	--

Сол сияқты,  $x = x_{i+2}$  нәтиже береді

	$P_2[x_i, x_{i+1}, x_{i+2}] = P_1[x_{i+1}, x_{i+2}] = y_{i+2}$	
--	--	--

Ақырында,  $x = x_{i+1}$  болғанда бізде болады

	$P_1[x_i, x_{i+1}] = P_1[x_{i+1}, x_{i+2}] = y_{i+1}$	
--	---	--

сондай-ақ

	$P_2[x_i, x_{i+1}, x_{i+2}] = \frac{(x_{i+1} - x_{i+2})y_{i+1} + (x_i - x_{i+1})y_{i+1}}{x_i - x_{i+2}} = y_{i+1}$	
--	--	--

Заңдылықты анықтағаннан кейін біз жалпы рекурентті формуланы шығара аламыз:

$P_k[x_i, x_{i+1}, \dots, x_{i+k}]$ $= \frac{(x - x_{i+k})P_{k-1}[x_i, x_{i+1}, \dots, x_{i+k-1}] + (x_i - x)P_{k-1}[x_{i+1}, x_{i+2}, \dots, x_{i+k}]}{x_i - x_{i+k}}$	(3.8)
---	-------

$x$  мәнін ескере отырып, есептеулерді келесі кестелік форматта орындауға болады (төрт деректер нүктесі үшін көрсетілген):

	$k = 0$	$k = 1$	$k = 2$	$k = 3$
$x_0$	$P_0[x_0] = y_0$	$P_1[x_0, x_1]$	$P_2[x_0, x_1, x_2]$	$P_3[x_0, x_1, x_2, x_3]$
$x_1$	$P_0[x_1] = y_1$	$P_1[x_1, x_2]$	$P_2[x_1, x_2, x_3]$	
$x_2$	$P_0[x_2] = y_2$	$P_1[x_2, x_3]$		
$x_3$	$P_0[x_3] = y_3$			

3.2-кесте. Невилл әдісіне арналған кесте.

Мәліметтер нүктелерінің санын  $m$  арқылы белгілей отырып, кестенің элементтерін есептейтін алгоритм

```

y = yData.copy()
for k in range(1,m):
    for i in range(m-k):
        y[i] = ((x - xData[i+k])*y[i] + (xData[i] - x)*y[i+1])/ \
            (xData[i]-xData[i+k])

```

Бұл алгоритм бастапқыда деректердің  $y$  мәндерін қамтитын бір өлшемді  $y$  массивімен жұмыс істейді (3.2 кестедегі екінші баған). Сыртқы цикл арқылы әрбір өту алдыңғы жазбаларды қайта жазатын келесі бағандағы  $y$  элементтерін есептейді. Процедураның соңында  $y$  кестенің диагональды мүшелерін қамтиды. Барлық деректер нүктелері арқылы өтетін интерполянттың мәні ( $x$ -те бағаланады)  $y$ -тің бірінші элементі болып табылады.

**neville**

Келесі функция Невилл әдісін жүзеге асырады; ол  $P_n(x)$  мәнін қайтарады:

```

## module neville
''' p = neville(xData, yData, x) .
Невилл әдісі бойынша көрсетілген деректер нүктелері
арқылы өтетін көпмүшелік интерполянтты p(x) есептейді.
'''
def neville(xData, yData, x):
    m = len(xData) # number of data points

```



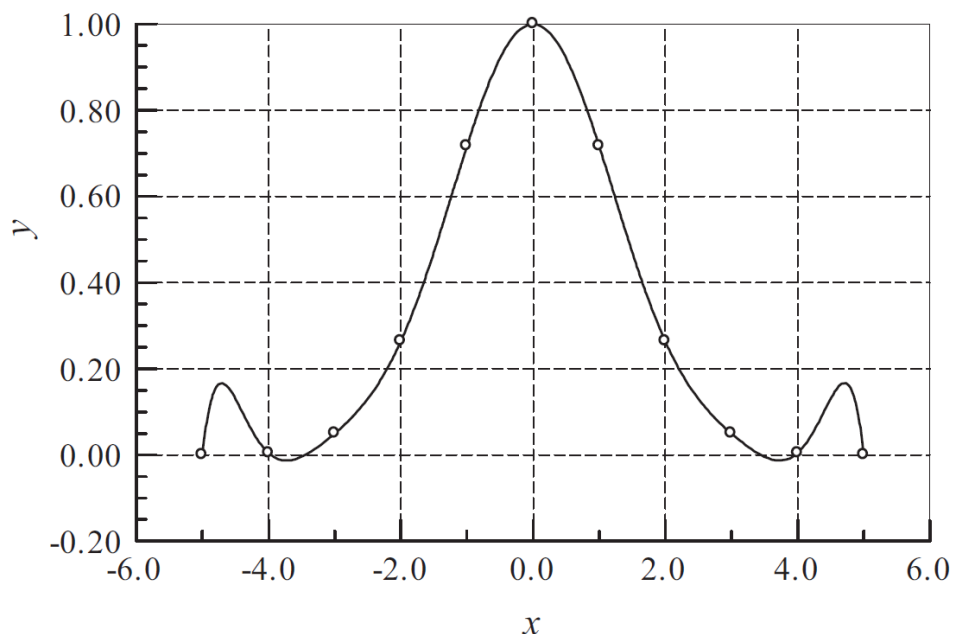
```

y = yData.copy()
for k in range(1,m):
    y[0:m-k] = ((x - xData[k:m])*y[0:m-k] + \
                (xData[0:m-k] - x)*y[1:m-k+1])/ \
                (xData[0:m-k] - xData[k:m])
return y[0]

```

### Көпмүшелік интерполяцияның шектеулері

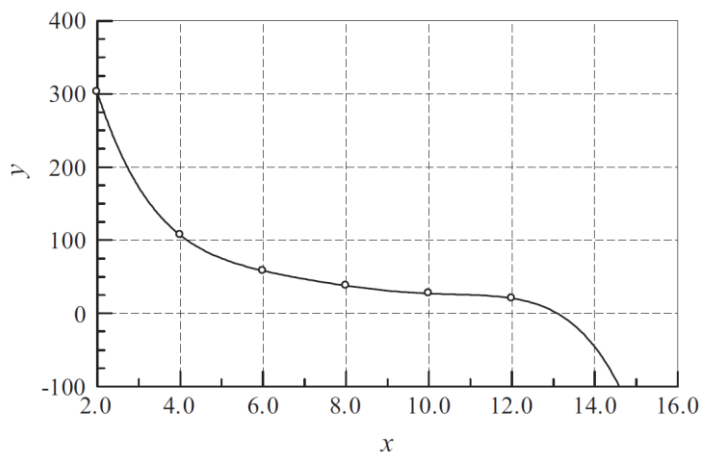
Көпмүшелік интерполяция деректер нүктелерінің ең аз мүмкін болатын санымен жүзеге асырылуы керек. Жақын екі нүктені қолданатын сызықтық интерполяция үшін, егер деректер нүктелері жақын орналасқан болса, жеткілікті. Үштен алтыға дейін көршілес нүктелер көп жағдайда жақсы нәтиже береді. Алты нүктеден артық қиылысатын интерполянтқа күдікпен қарау керек. Себебі, қызықтыратын нүктеден алыс деректер нүктелері интерполянттың дәлдігіне ықпал етпейді. Шын мәнінде, олар зиянды болуы мүмкін.



3.3-сурет. Тербелістерді көрсететін көпмүшелік интерполянт.

Тым көп нүктелерді пайдаланудың қауіптілігі 3.3-суретте көрсетілген. Бірдей аралықтағы 11 деректер нүктесі ашық дөңгелекшелермен көрсетілген. Тұтас сызық – интерполянт, 10 дәрежелі көпмүше, ол барлық нүктелерді қиып өтеді. Суретте көрсетілгендей, мұндай жоғары дәрежелі көпмүше деректер нүктелері арасында шамадан тыс тербеліске бейім. Неғұрлым тегіс нәтиже төрт жақын көрші нүктені қамтитын үшінші дәрежелі интерполянтты пайдалану арқылы алынады.

Көпмүшелі экстраполяция (деректер нүктелерінің аймағынан тыс интерполяциялау) қауіпті. Мысал ретінде 3.4-суретті қарастырайық. Алты деректер нүктесі дөңгелекшелер түрінде көрсетілген. Бесінші дәрежелі интерполяциялық көпмүше тұтас сызықпен берілген. Интерполяция деректер нүктелерінің аймағында жақсы көрінеді, бірақ  $x > 12$  болған кезде айқын бағыттан айтарлықтай ауытқиды. Мысалы,  $x = 14$  кезінде  $y$  экстраполяциялау бұл жағдайда жалған болар еді.

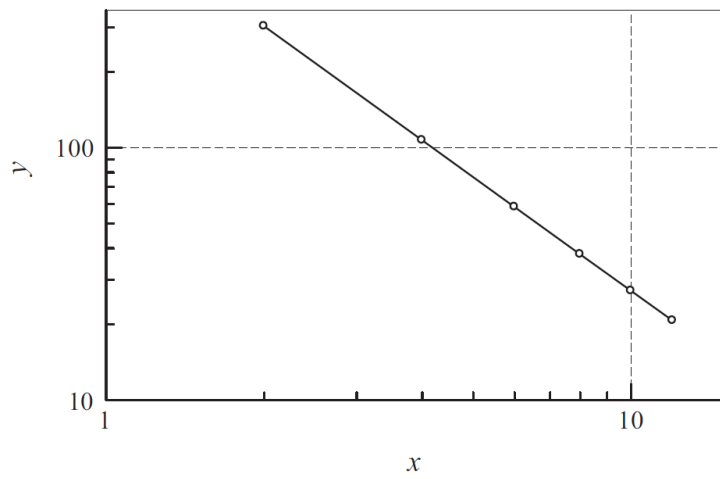


Егер экстраполяцияны болдырмау мүмкін болмаса, келесі шаралар пайдалы болуы мүмкін:

- Деректерді сызыңыз және экстраполяцияланған мән мағынасы бар екенін көрнекі түрде тексеріңіз.
- Ең жақын көрші деректер нүктелеріне негізделген төменгі ретті көпмүшені пайдаланыңыз. Мысалы, сызықтық немесе квадраттық интерполяция 3.4-суреттегі деректер үшін **y(14)** орынды бағалауды береді.

Әдетте  $x$ - $y$  қисығына қарағанда әлдеқайда тегіс және осылайша экстраполяциялау қауіпсізрек болатын  $\log x$  және  $\log y$  сызбасымен жұмыс жасаңыз. Көбінесе бұл түзу сызық сияқты болып көрінеді.

Бұл 3.5-суретте көрсетілген, ол 3.4-суреттегі деректердің логарифмдік сызбасын көрсетеді.



### МЫСАЛ 3.1

Деректер нүктелерін ескере отырып

$x$	0	2	3
$y$	7	11	28

$x = 1$  кезінде  $y$ -ті анықтау үшін Лагранж әдісін қолданыңыз.

**Шешуі:**

$$l_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(1 - 2)(1 - 3)}{(0 - 2)(0 - 3)} = \frac{1}{3}$$

$$l_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(1 - 0)(1 - 3)}{(2 - 0)(2 - 3)} = 1$$

$$l_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(1 - 0)(1 - 2)}{(3 - 0)(3 - 2)} = -\frac{1}{3}$$

$$y = y_0 l_0 + y_1 l_1 + y_2 l_2 = \frac{7}{3} + 11 - \frac{28}{3} = 4$$

### МЫСАЛ 3.2

Деректер нүктелері

$x$	-2	1	4	-1	3	-4
$y$	-1	2	59	4	24	-53

көпмүшенің бойында жатыр. Осы көпмүшенің дәрежесін 3.1-кестеге ұқсас бөлінген айырымдар кестесін құру арқылы анықтаңыз.

**Шешуі:**

$i$	$x_i$	$y_i$	$\nabla y_i$	$\nabla^2 y_i$	$\nabla^3 y_i$	$\nabla^4 y_i$	$\nabla^5 y_i$
0	-2	-1					
1	1	2	1				
2	4	59	10	3			
3	-1	4	5	-2	1		
4	3	24	5	2	1	0	
5	-4	-53	26	-5	1	0	0

Мұнда кестедегі сандарды алуда қолданылатын бірнеше үлгі есептеулер берілген:

$$\nabla y_2 = \frac{y_2 - y_0}{x_2 - x_0} = \frac{59 - (-1)}{4 - (-2)} = 10$$

$$\nabla^2 y_2 = \frac{\nabla y_2 - \nabla y_1}{x_2 - x_1} = \frac{10 - 1}{4 - 1} = 3$$

$$\nabla^3 y_5 = \frac{\nabla^2 y_5 - \nabla^2 y_2}{x_5 - x_2} = \frac{-5 - 3}{-4 - 4} = 1$$

Кестеден Ньютон көпмүшесінің соңғы нөлдік емес коэффициенті (нөлге тең емес соңғы диагональ мүшесі)  $\nabla^3 y_3$  екенін көреміз, бұл кубтық көпмүшенің коэффициенті. Демек, көпмүше үшінші дәрежелі.

### МЫСАЛ 3.3

Деректер нүктелерін ескере отырып

x	4.0	3.9	3.8	3.7
y	-0,06604	-0,02724	0,01282	0,05383

$y(x) = 0$  түбірін Невилл әдісімен анықтаңыз.

**Шешуі:** Бұл  $x$  пен  $y$  рөлдері алмасатын кері интерполяцияның мысалы. Берілген  $x$ -ке  $y$ -ті есептеудің орнына біз берілген  $y$ -ке сәйкес келетін  $x$ -ті табамыз (бұл жағдайда  $y = 0$ ). 3.2-кестені қолдана отырып (әрине,  $x$  пен  $y$  ауыстырылады), біз келесіні аламыз

$i$	$y_i$	$P_0[ ] = x_i$	$P_1[ , ]$	$P_2[ , , ]$	$P_3[ , , , ]$
0	-0.06604	4.0	3.8298	3.8316	3.8317
1	-0.02724	3.9	3.8320	3.8318	
2	0.01282	3.8	3.8313		
3	0.05383	3.7			

Кестеде қолданылатын үлгі есептеулер төменде берілген:

$$P_1[y_0, y_1] = \frac{(y - y_1)P_0[y_0] + (y_0 - y)P_0[y_1]}{y_0 - y_1}$$

$$= \frac{(0 + 0.02724) \cdot 4.0 + (-0.06604 - 0) \cdot 3.9}{-0.06604 + 0.02724} = 3.8598$$

$$P_2[y_1, y_2, y_3] = \frac{(y - y_3)P_1[y_1, y_2] + (y_1 - y)P_1[y_2, y_3]}{y_1 - y_3}$$

$$= \frac{(0 - 0.05383) \cdot 3.8320 + (-0.02724 - 0) \cdot 3.8313}{-0.02724 - 0.05383} = 3.8318$$

Кестедегі барлық  $P$  мәні әртүрлі деректер нүктелерін қамтитын интерполяцияның әртүрлі реттері нәтижесінде пайда болатын түбірдің бағалауы

болып табылады. Мысалы,  $P_1[y_0, y_1]$  – алғашқы екі нүктеге негізделген сызықтық интерполяциядан алынған түбір, ал  $P_2[y_1, y_2, y_3]$  – соңғы үш нүктені пайдаланып квадраттық интерполяцияның нәтижесі. Барлық төрт деректер нүктесі бойынша кубтық интерполяциясынан алынған түбір  $x = P_2[y_0, y_1, y_2, y_3] = 3.8317$ .

### МЫСАЛ 3.4

Кестедегі деректер нүктелері  $f(x) = 4.8 \cos \pi x / 20$  сызбасында жатыр. Бұл деректерді Ньютон әдісімен  $x = 0, 0.5, 1.0, \dots, 8.0$  интерполяциялаңыз және нәтижелерді  $y_i = f(x_i)$  “нақты” мәндерімен салыстырыңыз.

x	0.15	2.30	3.15	4.85	6.25	7.95
y	4.79867	4.49013	4.2243	3.47313	2.66674	1.51909

#### Шешуі:

```
## example3_4
import numpy as np
import math
from newtonPoly import *
xData = np.array([0.15, 2.3, 3.15, 4.85, 6.25, 7.95])
yData=np.array([4.79867, 4.49013, 4.2243, 3.47313, 2.66674, 1.51909])
a = coeffs(xData, yData)
print(" x yInterp yExact")
print("-----")
for x in np.arange(0.0, 8.1, 0.5):
    y = evalPoly(a, xData, x)
    yExact = 4.8*math.cos(math.pi*x/20.0)
    print('{:3.1f} {:9.5f} {:9.5f}'.format(x, y, yExact))
input("\nPress return to exit")
```

#### Нәтижелері

```
x   yInterp  yExact
-----
0.0 4.80003 4.80000
0.5 4.78518 4.78520
1.0 4.74088 4.74090
1.5 4.66736 4.66738
2.0 4.56507 4.56507
2.5 4.43462 4.43462
```

3.0 4.27683 4.27683  
3.5 4.09267 4.09267  
4.0 3.88327 3.88328  
4.5 3.64994 3.64995  
5.0 3.39411 3.39411  
5.5 3.11735 3.11735  
6.0 2.82137 2.82137  
6.5 2.50799 2.50799  
7.0 2.17915 2.17915  
7.5 1.83687 1.83688  
8.0 1.48329 1.48328

### **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](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>