Әл-Фараби атындағы Қазақ ұлттық университеті

Механика-математика факультеті

Көпмүшелік интерполяция

Темирбеков Нурлан Муханович ф-м.ғ.д., профессор

Жоспар

1. Интерполяция және қисықты сәйкестендіру.

2. Лагранж әдісі.

3. Ньютон әдісі.

4. Көпмүшелік интерполяцияның шектеулері.

5. Интерполяция әдісі бойынша есеп шығару.

Мақсаты

Кейбір нүктелерде кесте түрінде берілген функцияны қалпына келтіру бойынша интерполяция әдістерімен танысу және есептеу алгоритмдерін талдау.

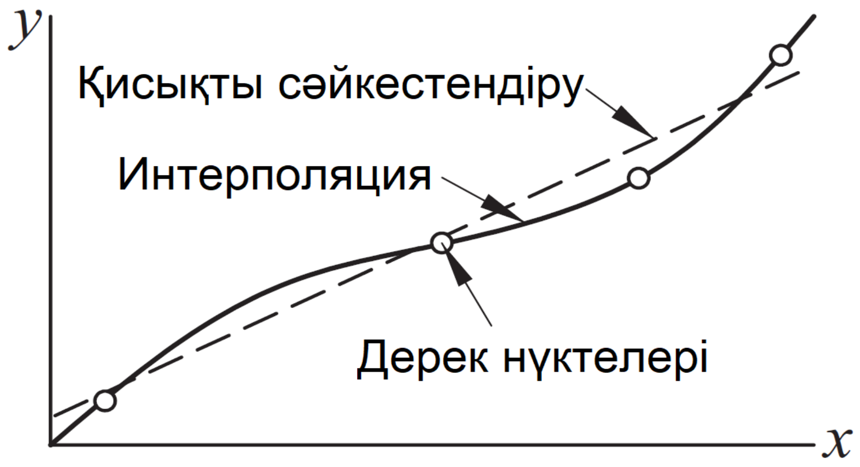
Интерполяция және қисықты сәйкестендіру

1-кесте. Интерполяцияның деректер нүктелері

x_0	x_1	x_2	…	x_n
y_0	y_1	y_2	…	y_n

Интерполяцияда деректер нүктелері арқылы қисық сызық саламыз.

Қисықты сәйкестендіру белгілі бір мағынада деректерді жақындататын тегіс қисық сызығы табылады.



1-сурет.

Интерполяция мен қисықты сәйкестіндіру

Лагранж әдісі

Әрқашан $n+1$ нақты деректер нүктелері арқылы өтетін n дәрежелі жалғыз көпмүшені құруға болады.

Бұл көпмүшені алудың бір әдісі – Лагранж формуласы

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

$$l_i(x) = \frac{x - x_0}{x_i - x_0} \cdot \frac{x - x_1}{x_i - x_1} \cdot \ldots \cdot \frac{x - x_{i-1}}{x_i - x_{i-1}} \cdot \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdot \ldots \cdot \frac{x - x_n}{x_i - x_n}$$ негізгі функциялар деп аталады

$$i = 0,1,2, \ldots, n$$

Мысал 3.1

Деректер нүктелерін ескере отырып

x	0	2	3
y	7	11	28

$x = 1$ кезінде **y**-ті анықтаңыз.

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

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

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

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

Ньютон әдісі

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

Кез келген n үшін $P_0(x) = a_n,\quad P_k(x) = a_{n-k} + (x - x_{n-k})P_{k-1}(x),\quad k = 1,2,\ldots,n$

Бөлінген айырымдар

$$\nabla y_i = \frac{y_i - y_0}{x_i - x_0},\qquad i = 1,2,\ldots,n$$

$$\nabla^2 y_i = \frac{\nabla y_i - \nabla y_1}{x_i - x_1},\qquad i = 2,3,\ldots,n$$

$$\nabla^3 y_i = \frac{\nabla^2 y_i - \nabla^2 y_2}{x_i - x_2},\qquad i = 3,4,\ldots,n$$

$\vdots$

$$\nabla^n y_n = \frac{\nabla^{n-1} y_n - \nabla^{n-1} y_{n-1}}{x_n - x_{n-1}}$$

$$a_0 = y_0, a_1 = \nabla y_1, a_2 = \nabla^2 y_2, \ldots, a_n = \nabla^n y_n$$

Коэффициенттер қолмен есептеуде кестедегі форматпен жұмыс істеу ыңғайлы ($n = 4$ үшін көрсетілген)

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

Кестедегі диагональды мүшелер ($y_0, \nabla y_1, \nabla^2 y_2, \nabla^3 y_3, \nabla^4 y_4$) көпмүшенің коэффициенттері болып табылады.

Ньютон әдісі бойынша newtonPoly модулі

```
## module newtonPoly
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
```

xData және yData деректер нүктесінің массивтері

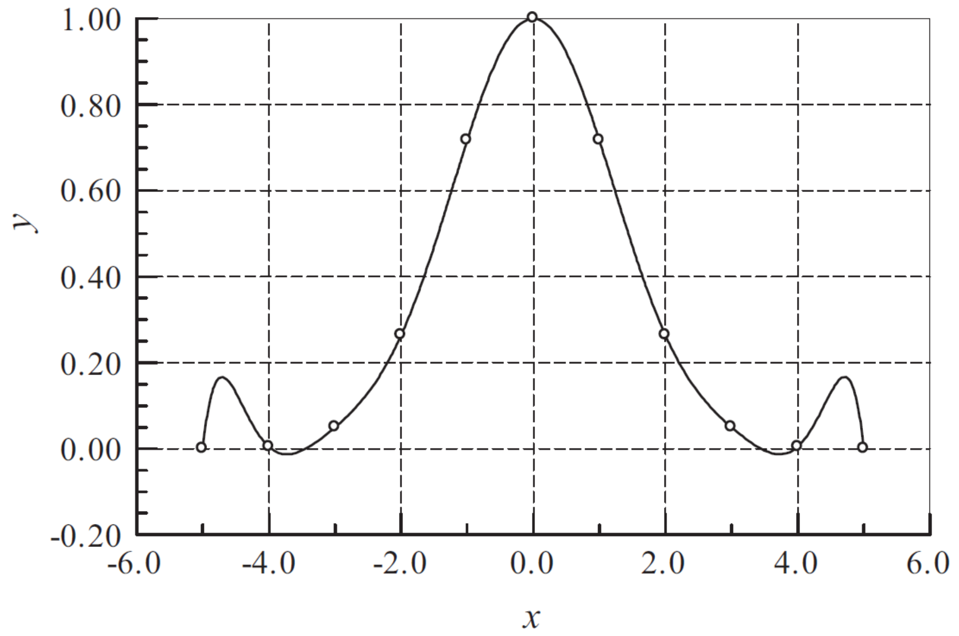
p = evalPoly(a,xData,x)

Ньютонның p көпмүшелігін x-де есептейді.

a = coeffts(xData,yData).

Ньютон көпмүшесінің коэффициенттерін есептейді.

Көпмүшелік интерполяцияның шектеулері



Көпмүшелік интерполяция деректер нүктелерінің ең аз мүмкін болатын санымен жүзеге асырылуы керек.

Интерполяция әдісі бойынша есеп шығару

Кестедегі деректер нүктелері $f(x) = 4.8\cos \pi x/20$ сызбасында жатыр.

Бұл деректерді Ньютон әдісімен $x = 0, 0.5, 1.0, \ldots, 8.0$ интерполяцияланыз 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

Есепті шешу коды

```
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([4.79867,4.49013,4.2243,3.47313,2.6667
4,1.51909])
a = coeffts(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
```

Қорытынды

1. Интерполяция мен қисықты сәйкестендіру есебі.
2. Лагранж әдіс сипаттамасы.
3. Ньютон әдісі сипаттамасы.
4. Көпмүшелік интерполяцияның ерекшеліктері.
5. Python тілінде коды бойынша мысал.

Пайдаланылған әдебиеттер тізімі

1. Jaan Kiusalaas. Numericalmethods in engineering with Python. Cambridge University Press. ISBN 978-1-107-03385
2. Вабищевич П.Н. Численные методы: Вычислительный практикум. — М.: Книжный дом «ЛИБРОКОМ», 2010. — 320 с.
3. Киреев В. И., Пантелеев А. В. Численные методы в примерах и задачах: Учебное пособие. — СПб.: Издательство «Лань», 2015. — 448 с.