5. Чисельні методи наближення функцій. Апроксимація, інтерполяція та екстраполяція

Задача наближення функцій виникає при розв'язанні багатьох практичних і теоретичних задач, а інколи і як самостійна. Так, наближення функцій є важливим допоміжним апаратом при розв'язанні інших задач чисельного аналізу: чисельного інтегрування і диференціювання, розв'язання диференціальних рівнянь, розв'язання систем нелінійних рівнянь, задач оптимізації та ін.

5.1. Постановка задачі. Поняття апроксимації та інтерполяції

Слід розглянути декілька постановок задачі наближення функцій.

Постановка 1. Проста задача, що приводить до наближення функцій, полягає в наступному. Нехай є деяка функція $f(x)$, $f: R^1 \to R^1$, про яку відомо, що в n точках $x_1, x_2, \ldots, x_n$ вона приймає, відповідно, значення $y_1, y_2, \ldots, y_n$, тобто $y_i = f(x_i), i = \overline{1,n}$. Потрібно відновити її значення при інших значеннях $x \in (x_1, x_n)$. Функція $y = f(x)$ може бути як невідомою, тобто її значення тільки вимірюються (так званий "чорний ящик"), а може бути і просто дуже складною для обчислень. Наприклад, вона може використовуватися в яких-небудь фізико-технічних або суто математичних розрахунках, де її доводиться багато разів обчислювати. Слід зазначити, що якщо функція $y = f(x)$ невідома, то набір точок (x_i, y_i), $i = \overline{1,n}$, називають **табличним заданням функції**, оскільки ці значення подають у вигляді таблиці.

У цих випадках функцію $f(x)$ стараються замінити на простішу функцію $g(x;a)$, близьку до $f(x)$, тобто

$$f(x) \approx g(x;a), \tag{5.1}$$

де $a \in R^k$, $a = (a_1, a_2, \ldots, a_k)^T$ – деякі параметри функції g, вид якої відомий. Процес наближення однієї функції іншою ще називають **апроксимацією**, а функцію g при цьому називають **апроксимуючою** для f.

Варто зазначити, що параметри $a \in R^k$ є інструментом для підгонки (наближення) функції g до функції f.

Якщо параметри $a_1, a_2, \ldots, a_k$ визначаються з умови збігу значень функції $f(x)$ і апроксимуючої функції в точках $x_1, x_2, \ldots, x_n$, тобто

$$g(x_i;a) = f(x_i), \ \forall i = \overline{1,n},$$

то такий спосіб наближення називають **інтерполяцією** або **інтерполюванням**.

Слід зазначити, що точки x_i, $i = \overline{1,n}$ називають **вузлами інтерполяції** або ще **полюсами**, тобто "інтерполяція" – це відновлення функції між полюсами.

Постановка 2. У задачах планування експериментів виникає така проблема. Відомо вид гарного наближення функції, наприклад функція $f(x)$ добре наближається поліномом 2-го степеня. В той же час вимірювані значення функції y_i мають великі помилки. Потрібно отримати найкраще в певному значенні наближення при мінімальній кількості вимірювань. Така задача виникає при плануванні експериментів в біології, хімії, фізиці, географії, військовій справі та ін.

Постановка 3. Задача наближення функцій розв'язується і при складанні стандартних процедур обчислення елементарних і спеціальних функцій, наприклад $\cos(x)$, $\sin(x)$ й ін. Так

$$e^x \approx \sum_{k=0}^{n} \frac{x^k}{k!},$$

$$\sin(x) \approx x + \sum_{k=1}^{n} (-1)^k \frac{x^{2k-1}}{(2k-1)!},$$

де n підбирається так, щоб забезпечити достатню точність обчислень.

5.2. Метод найменших квадратів для апроксимації функцій

Найбільш відомим і ефективним з методів розв'язання задачі апроксимації функцій (5.1) є **метод найменших квадратів (МНК).** МНК призначений безпосередньо для знаходження параметрів a апроксимуючої функції $g(x;a)$, вид якої вже обраний. Суть метода полягає в такому.

Вводиться функція від a виду

$$\Phi(a) \equiv \sum_{i=1}^{n} [(y_i - g(x_i; a)]^2, \qquad (5.2)$$

яку можна розглядати як міру відхилення функції $g(x;a)$ (по аргументу x) від функції $f(x)$ по сукупності точок $x_1, x_2, \ldots, x_n$. Тоді параметри a функції $g(x;a)$ можна визначити з умови найменшого відхилення $g(x;a)$ від $f(x)$, тобто параметри a знаходяться як точка, в якій функція $\Phi(a)$ досягає по $a \in R^k$ мінімального значення (точка мінімуму). Нехай a^* – шукані параметри. При цьому величини

$$r_i \equiv y_i - g(x_i; a^*), \; i = \overline{1,n}$$

називаються **залишковими відхиленнями** (або **залишками)**і використовуються для аналізу отриманого розв'язку.

Вид графіка залишків дозволяє оцінити, наскільки правильно спочатку був вибраний вид апроксимуючої функції $g(x;a)$. Якщо в графіку залишків спостерігається деяка функціональна закономірність (наприклад, парабола), то це означає, що цю закономірність потрібно перенести в апроксимуючу функцію. Нормальною є ситуація, коли графік залишків має вид ламаної синусоїди (ще кажуть, має випадковий характер).

Слід зазначити, що метод найменших квадратів зазвичай застосовують для розв'язання задачі наближення функцій у постановці 2 (див. розділ 7.1).

Приклад 5.1. Використовуючи метод найменших квадратів, підібрати найкращу апроксимацію поліномом (визначити його степінь) для функції, заданої таблицею:

x	– 1,5	– 1	– 0,5	0	0,5	1	1,5	2	2,5
y	9,8	6,4	5,2	1,7	2,9	5,6	7,2	16,5	27,5

Розв'язання в математичному пакеті R

Відправними даними задачі є вектор значень аргумента x, вектор значень функції y та кількість точок спостереження n:

```
x = c(-1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5)
y = c(9.8, 6.4, 5.2, 1.7, 2.9, 5.6, 7.2, 16.5, 27.5)
n = 9
```

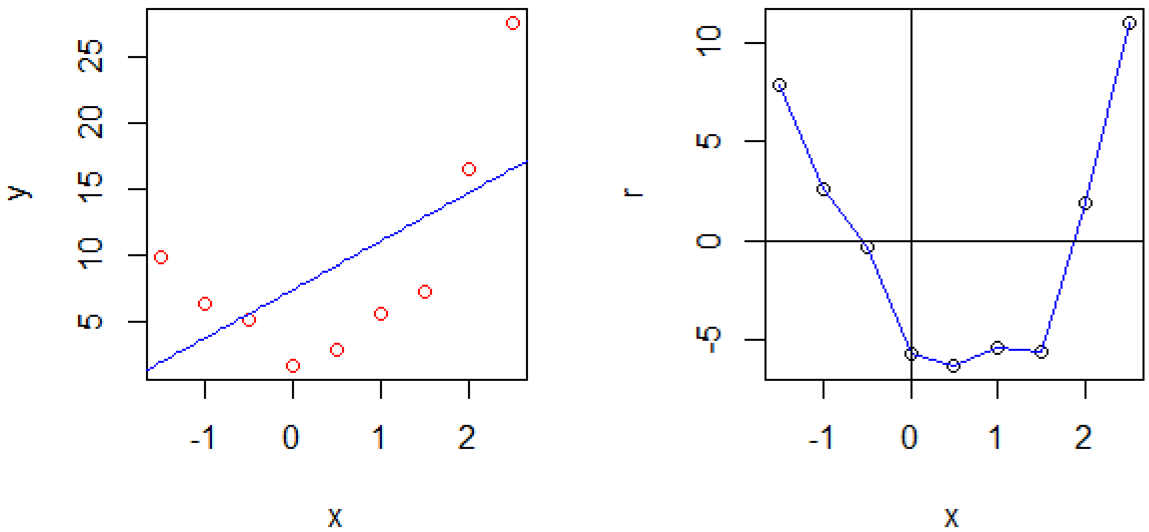
Апроксимація функції методом найменших квадратів із використанням вбудованої процедури *optim* .

1. Апроксимація лінійною функцією (поліномом 1-ї степені):

```
# Задаємо вид апроксимуючої функції
FunModel = function(x,a)
{
    return( a[1]+a[2]*x )
}
# Визначаємо цільову функцію метода МНК
FunMNK = function(a)
{
    S = 0
    for(i in 1:n)
         S = S + (y[i]-FunModel(x[i],a))^2
    return( S )
}
> # Задаємо початкове значення
> # для параметрів а апроксимуючої функції
> a0 = c(0,0)
> res = optim(fn=FunMNK, par=a0)
> a1 = res$par
> a1
[1] 7.383490 3.633125
> FunMNK(a1)
[1] 327.2633
> # Обчислюємо значення апроксимуючої функції в точках х
> ym = FunModel(x,a1)
> # Обчислюємо залишки
> r = y - ym
> a = -2; b = 3; n = 100
> # Обчислюємо n точок на відрізку [a,b]
> h = (b-a)/(n-1)
> x1 = c(1:n)
> for(i in 1:n){ x1[i] = a + h*(i-1) }
> # Обчислюємо значення апроксимуючої функції в точках х
> y1m = FunModel(x1,a1)
```

```
> par(mfrow=c(1, 2))          # задаємо дві графічні підобласті
> plot(x, y, type="p", col="red")# рисуємо перший графік у вигляді точок
> lines(x1, y1m, col="blue")  # додаємо в 1-у графічну підобласть ще лінію
> plot(x, r, type="p")        # рисуємо перший графік у вигляді точок
> lines(x, r, col="blue")     # додаємо в 2-у графічну підобласть ще лінію
> abline(h=0)
> abline(v=0)
```

Графіки функцій (заданої та апроксимуючої) і графік залишків:

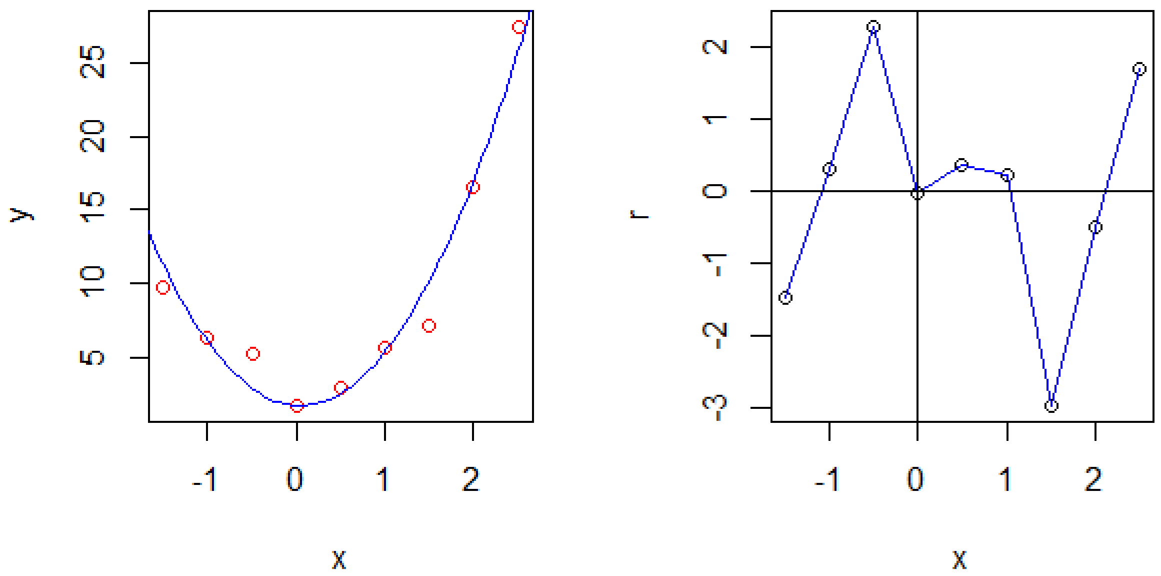


З графіків функцій видно, що апроксимуюча функція виду $y = a_0 + a_1 x$ погано наближається до функції, заданої таблицею. В графіку залишків спостерігається закономірність у вигляді параболи, яку необхідно перенести в апроксимуючу функцію.

2. Апроксимація квадратичною функцією (поліномом 2-ї степені):

```
# Задаємо вид апроксимуючої функції
FunModel = function(x,a)
{
     return( a[1]+a[2]*x+a[3]*x^2 )
}
# Визначаємо цільову функцію метода МНК
FunMNK = function(a)
{
     S = 0
     for(i in 1:n)
          S = S + (y[i]-FunModel(x[i],a))^2
     return( S )
}
> # Задаємо початкове значення
> # для параметрів а апроксимуючої функції
> a0 = c(0,0,0)
> res = optim(fn=FunMNK, par=a0)
> a1 = res$par
> a1
[1]  1.7213365 -0.3640413  3.9971283
> FunMNK(a1)
[1] 19.66321
```

Графіки функцій (заданої та апроксимуючої) і графік залишків:

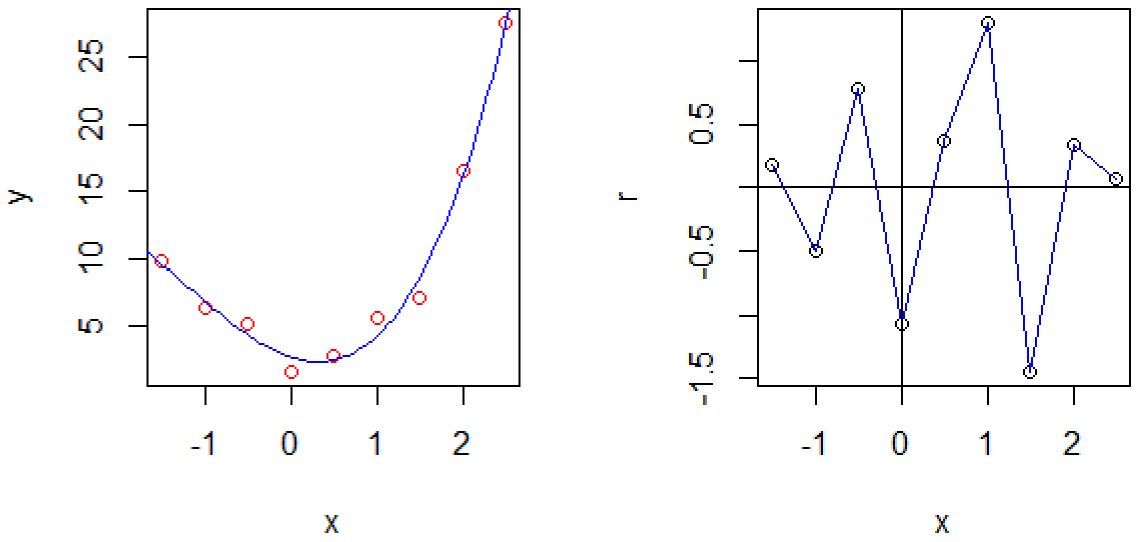


З графіків функцій видно, що апроксимуюча функція виду $y = a_0 + a_1 x + a_2 x^2$ також не дуже гарно наближається до функції, заданої таблицею. В графіку залишків спостерігається закономірність у вигляді кубічної функції, яку необхідно перенести в апроксимуючу функцію.

3. Апроксимація кубічною функцією (поліномом 3-ї степені):

```
# Задаємо вид апроксимуючої функції
FunModel = function(x,a)
{
     return( a[1]+a[2]*x+a[3]*x^2+a[4]*x^3 )
}
# Визначаємо цільову функцію метода МНК
FunMNK = function(a)
{
     S = 0
     for(i in 1:n)
          S = S + (y[i]-FunModel(x[i],a))^2
     return( S )
}
> # Задаємо початкове значення
> # для параметрів a апроксимуючої функції
> a0 = c(0,0,0,0)
> res = optim(fn=FunMNK, par=a0)
> a1 = res$par
> a1
[1]  2.7745381 -2.0843292  2.8265234  0.7812671
> FunMNK(a1)
[1] 6.071312
```

Графіки функцій (заданої та апроксимуючої) і графік залишків:



Як видно з графіків і значень функцій FunkMNK(a1) для різних апроксимуючих функцій, кращою виявилась кубічна функція виду $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3$.

У випадках, коли функція $g(x;a)$ є лінійною по параметрам a точку мінімуму функції $\Phi(a)$ можна знайти аналітично з необхідної умови мінімуму 1-го порядку $\Phi'(a) = 0$ [**Ошибка! Источник ссылки не найден.**; **Ошибка! Источник ссылки не найден.**], тобто

$$\frac{\partial \Phi}{\partial a_j} = 0, \quad \forall j = \overline{1,k}. \tag{5.3}$$

Слід розглянути загальний випадок, коли функція $g(x;a)$ лінійна по параметрам, тобто має вигляд

$$g(a;x) = \sum_{m=1}^{k} a_m \psi_m(x), \tag{5.4}$$

де $\psi_m(x)$, $m = \overline{1,k}$, – деякі відомі функції від x. Тоді з (5.2) – (5.4) отримуємо $\forall j = \overline{1,k}$

$$\frac{\partial \Phi}{\partial a_j} = -2\sum_{i=1}^{n} [y_i - g(x_i;a)]\psi_j(x_i) = -2\sum_{i=1}^{n}\left[y_i - \sum_{m=1}^{k} a_m \psi_m(x_i) \right]\psi_j(x_i) = 0.$$

Звідки для $\forall j = \overline{1,k}$

$$\sum_{m=1}^{k} a_m \left[\sum_{i=1}^{n} \psi_m(x_i)\psi_j(x_i) \right] = \sum_{i=1}^{n} y_i \psi_j(x_i). \tag{5.5}$$

Таким чином, параметри a можуть бути знайдені як розв'язок системи лінійних рівнянь (5.5), яку можна записати в матричному вигляді

$$Ca = b, \tag{5.6}$$

де

$$c_{jm} = \sum_{i=1}^{n} \psi_m(x_i)\psi_j(x_i),\ j = \overline{1,k},\ m = \overline{1,k};$$

$$b_j = \sum_{i=1}^{n} y_i \psi_j(x_i),\ j = \overline{1,k}.$$

З (5.6) за умови, що матриця C має обернену, можна отримати й аналітичний вираз для a, а саме $a = C^{-1} \times b$.

Приклад 5.2. Використовуючи метод найменших квадратів, знайти параметри апроксимуючої функції виду $y = a_0 + a_1 x + a_2 x^2$ для таблично заданої функції шляхом розв'язання лінійної системи рівнянь.

x	– 1,5	– 1	– 0,5	0	0,5	1	1,5	2	2,5
y	9,8	6,4	5,2	1,7	2,9	5,6	7,2	16,5	27,5

Розв'язання в математичному пакеті R

Відправними даними задачі є вектор значень аргумента x, вектор значень функції y, кількість точок спостереження n, кількість параметрів апроксимуючої функції k та векторна функція $Fi(x)$ відомих функцій від x:

```
x = c(-1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5)
y = c(9.8, 6.4, 5.2, 1.7, 2.9, 5.6, 7.2, 16.5, 27.5)
n = 9
k=3
```

```
# Вводимо векторну функцію Fi
Fi = function(x)
{
  z = c(1:3)
  z[1] = 1
  z[2] = x
  z[3] = x^2
  return( z )
}
```

Процедура знаходження параметрів апроксимуючої функції шляхом розв'язання системи лінійних рівнянь може мати вигляд:

```
MNKSolve = function(x, y, n, Fi, k)
{
  # Формуємо матрицю значень векторної функції Fi
  # в точках x[i] по рядкам
  FiX = matrix(nrow=n, ncol=k)
  for(i in 1:n)
  {
    FiX[i,] = Fi(x[i])
  }
  C = matrix(nrow=k, ncol=k)
     for(j in 1:k)
     {
          for(m in 1:k)
          {
               S = 0
                    for(i in 1:n)
                         S = S + FiX[i,j]*FiX[i,m]
               C[j,m] = S
          }
     }
  # Формуємо вектор d
  d = c(1:k)
  for(j in 1:k)
     {
          S = 0
          for(i in 1:n)
          {
               S = S + y[i]*FiX[i,j]
          }
          d[j] = S
     }
  # Розв'язуємо систему
  a = solve(C, d)
  return(a)
}
```

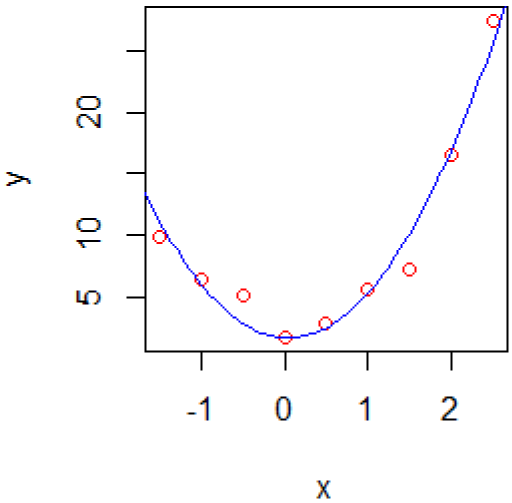
Результат знаходження параметрів апроксимуючої функції з використанням записаної процедури:

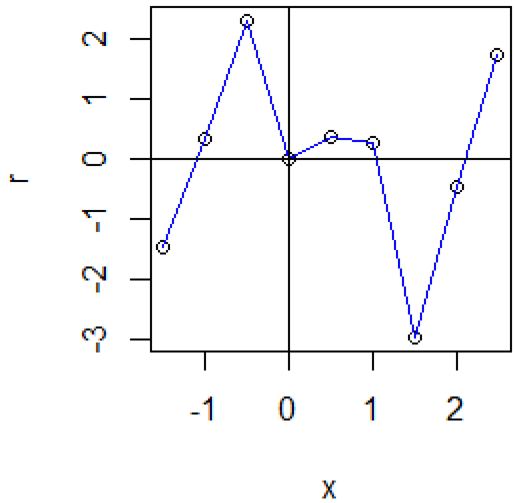
```
> a1 = MNKSolve(x, y, n, Fi, k)
> a1
[1]  1.7203463 -0.3640693  3.9974026
```

Таким чином, апроксимуюча функція має вигляд:

$$y = 1.72 - 0.364x + 3.997x^2 .$$

Графіки функцій (заданої та апроксимуючої) і графік залишків:





Варто розглянути ще випадок, коли апроксимуюча функція $g(x;a)$ є поліномом заданого ($k-1$)-го степеня. Тоді вона в загальному випадку може бути записана у вигляді

$$g(x;a) = b_0 + b_1 x + b_2 x^2 + \cdots + b_{k-1} x^{k-1} = \sum_{i=0}^{k-1} b_i x^i ,$$

де $a \in R^k$, $a_1 = b_0$, $a_2 = b_1$, ..., $a_k = b_{k-1}$, b_j, $j = \overline{0,k-1}$, – коефіцієнти полінома. При цьому функція $g(x;a)$ є лінійною по параметрам a і має вигляд (5.4), де $\psi_1(x) = 1$, $\psi_2(x) = x$, ..., $\psi_k(x) = x^{k-1}$, тобто $\psi_m(x) = x^{m-1}$, $m = \overline{1,k}$. Тоді параметри a можна знайти, розв'язавши систему лінійних рівнянь (5.6), з елементами

$$c_{jm} = \sum_{i=1}^{n} x_i^{m-1} x_i^{j-1} = \sum_{i=1}^{n} x_i^{m+j-2} ,\ j = \overline{1,k},\ m = \overline{1,k}; \tag{5.7}$$

$$b_j = \sum_{i=1}^{n} y_j x_i^{j-1} ,\ j = \overline{1,k}. \tag{5.8}$$

Приклад 5.3. Використовуючи метод найменших квадратів, знайти параметри апроксимуючої функції виду $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ (поліном 3-го степеня) для таблично заданої функції шляхом розв'язання системи лінійних рівнянь.

x	– 1,5	– 1	– 0,5	0	0,5	1	1,5	2	2,5
y	9,8	6,4	5,2	1,7	2,9	5,6	7,2	16,5	27,5

Розв'язання в математичному пакеті R

Відправними даними задачі є вектор значень аргумента x, вектор значень функції y, кількість точок спостереження n та кількість параметрів апроксимуючої функції K:

```
x = c(-1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5)
y = c(9.8, 6.4, 5.2, 1.7, 2.9, 5.6, 7.2, 16.5, 27.5)
n = 9
k=4
```

Процедура знаходження параметрів апроксимуючої функції шля-хом розв'язання системи лінійних рівнянь може бути записана так:

```
MNKSolvePolinom = function(x, y, n, k)
{
  # Формуємо матрицю C
  C = matrix(nrow=k, ncol=k)
     for(j in 1:k)
     {
          for(m in 1:k)
          {
               S = 0
               for(i in 1:n)
               {
                    S = S + x[i]^(m+j-2)
               }
          C[j,m] = S
          }
     }
  # Формуємо вектор d
  d = c(1:k)
  for(j in 1:k)
     {
       S = 0
       for(i in 1:n)
       {
          S = S + y[i]*x[i]^(j-1)
       }
       d[j] = S
     }
```

```
    # Розв'язуємо систему
    a = solve(C, d)
    return(a)
}
```

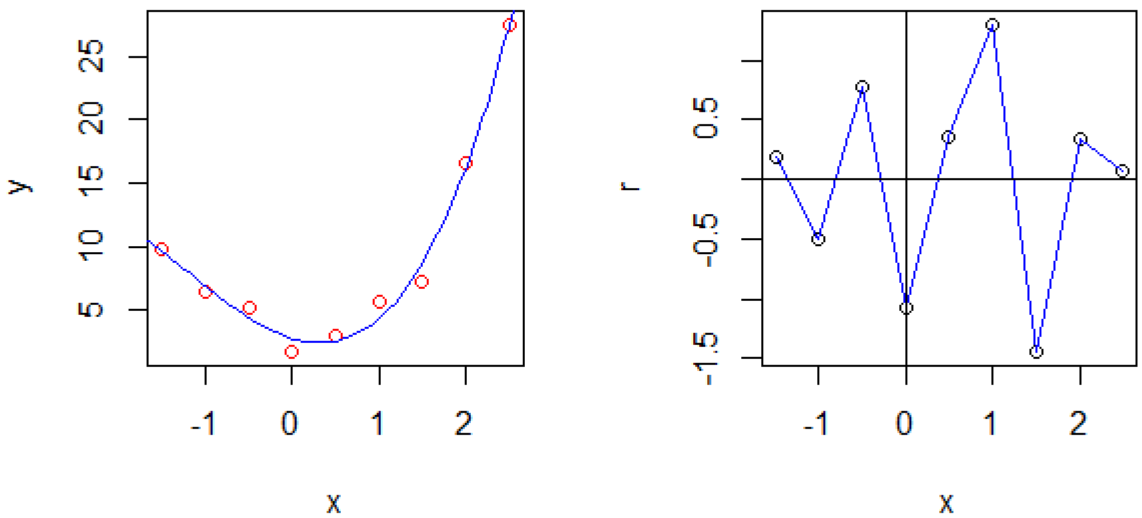
Результат знаходження параметрів апроксимуючої функції з використанням записаної процедури:

```
> # Знаходимо параметри апроксимуючої функції
> a1 = MNKSolvePolinom(x, y, n, k)
> a1
[1]  2.7748918 -2.0825878  2.8256854  0.7811448
```

Таким чином, апроксимуюча функція має вигляд:

$$y = 2.775 - 2.083x + 2.826x^2 + 0.781x^3.$$

Графіки функцій (заданої та апроксимуючої) і графік залишків:



5.3. Інтерполяція лінійна та квадратична

У випадках, коли вид функції $g(x;a)$ наперед невідомий, а значення y_i, $i = \overline{1,n}$ містять малу погрішність, задачу наближення таблично заданої функції $f(x)$ часто розв'язують як задачу інтерполяції.

Кусочно-лінійна інтерполяція

Такий вид інтерполяції полягає в тому, що на кожному окремому інтервалі (x_i, x_{i+1}) функція $f(x)$ наближається лінійною функцією $g_i(x;a_i,b_i) = a_i x + b_i$, $i = \overline{1,n-1}$ (рис. 5.1).

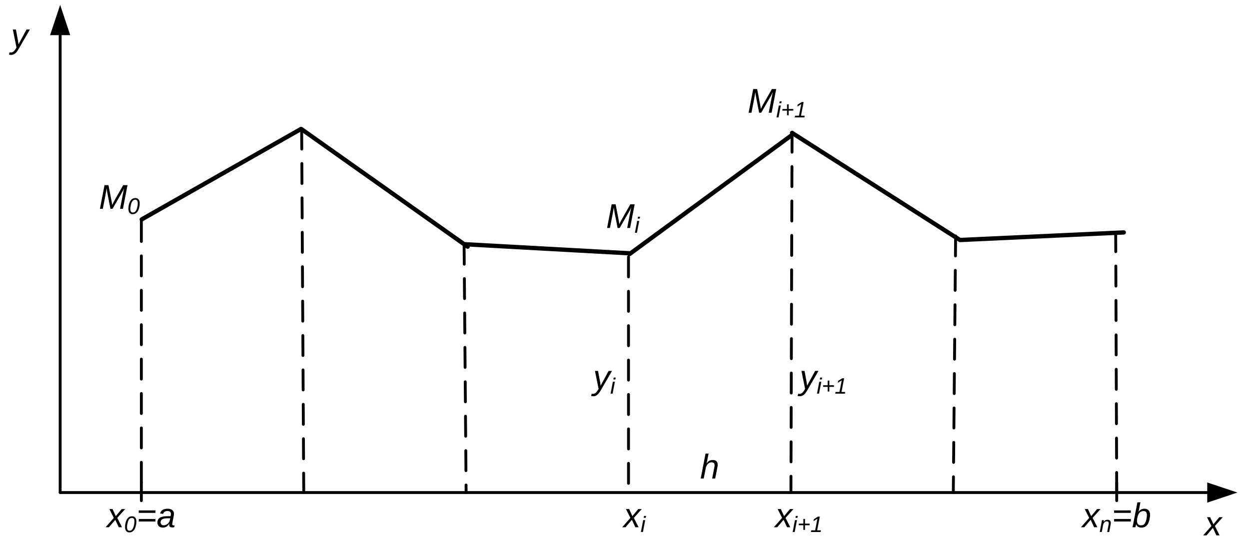


Рис. 5.1. **Графічна інтерпретація кусочно-лінійної інтерполяції**

Оскільки значення функцій $f(x)$ і $g_i(x;a_i,b_i)$ в точках x_i і x_{i+1} повинні співпадати, то

$$g_i(x_i;a_i,b_i) = a_i x_i + b_i = y_i,$$

$$g_i(x_{i+1};a_i,b_i) = a_i x_{+1i} + b_i = y_{i+1}.$$

Тоді, розв'язуючи отриману систему рівнянь відносно a_i, b_i, отримуємо

$$a_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i},$$

$$b_i = y_i - a_i x_i.$$

Звідси випливає, що параметри a_i, b_i функції $g_i(x)$ повністю визначаються значеннями координат точок (x_i, y_i), (x_{i+1}, y_{i+1}). Неважко перевірити, що лінійна функція, яка проходить через 2 точки (x_i, y_i), (x_{i+1}, y_{i+1}), також може бути записана в зручнішому вигляді:

$$g_i(x) = \frac{(x - x_{i+1})}{(x_i - x_{i+1})} \cdot y_i + \frac{(x - x_i)}{(x_{i+1} - x_i)} \cdot y_{i+1}. \tag{5.9}$$

Таким чином, у ході використання кусочно-лінійної інтерполяції для таблично заданої функції, для обчислення наближеного значення функції

$f(x)$ у деякій точці $\bar{x} \in (x_1, x_n)$ діють таким чином. Спочатку знаходять інтервал (x_i, x_{i+1}), якому точка $\bar{x}$ належить, а потім обчислюють наближене значення функції $f(x)$ у точці $\bar{x}$ як $g_i(\bar{x})$, де функція $g_i(x)$ визначена в (5.9).

Варто зазначити, що інтерполююча функція $g(x)$ для функції $f(x)$ буде наче "зшитою" з лінійних функцій $g_i(x)$, кожна з яких розглядається на своєму "кусочку" (x_i, x_{i+1}), звідси і назва даного виду інтерполяції – кусочно-лінійна.

При цьому функція $g(x)$ по суті є наближеним розв'язком задачі наближення функції $f(x)$, тобто істинного розв'язку. Тому виникає питання про точність розв'язку цієї задачі, тобто про погрішність отриманого розв'язку. Слід розглянути випадок, коли значення y_i, $i = \overline{1,n}$ відомі точно, тобто $y_i = f(x_i)$, $i = \overline{1,n}$.

Оцінка погрішності. Нехай функція $f(x)$ двічі неперервно-диференційована на відрізку (x_1, x_n). Тоді для всіх $x \in (x_1, x_n)$

$$|f(x) - g(x)| \le M_2 h^2,$$

де $M_2 = \max\limits_{\xi \in (x_1, x_n)} |f''(\xi)|$, $h = \max\limits_{i=\overline{1,n-1}} |x_{i+1} - x_i|$.

Таким чином, погрішність кусочно-лінійної інтерполяції має порядок $O(h^2)$.

Кусочно-лінійна інтерполяція застосовується тоді, коли не вимагається великої точності наближення. Наприклад, при побудові графіків по точках тощо.

Приклад 5.4. Використовуючи кусочно-лінійну інтерполяцію, обчислити наближене значення функції, заданої таблично, в точці $x = -1.2$.

x	– 1,5	– 1	– 0,5	0	0,5	1	1,5	2	2,5
y	9,8	6,4	7,0	1,7	17,3	5,6	10,8	6,2	27,5

Розв'язання в математичному пакеті R

Відправними даними задачі є вектор значень аргумента x, вектор значень функції y та кількість точок спостереження n:

```
x = c(-1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5)
y = c(9.8, 6.4, 7, 1.7, 17.3, 5.6, 10.8, 6.2, 27.5)
n = 9
```

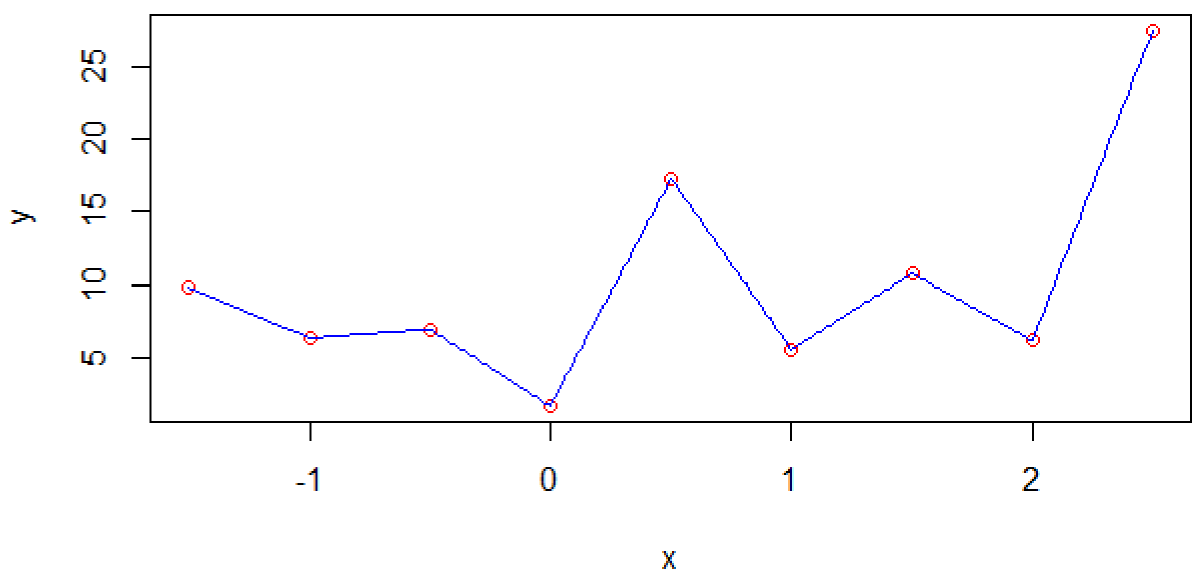
Процедуру, що реалізує кусочно-лінійну інтерполяцію, можна записати так:

```
 # Кусково-лінійна інтерполяція
 LinInterpol = function(x, y, n, xk)
 {
   for (i in 1:(n-1))
   {
      if (xk <= x[i+1])
         break
   }
   yk = (xk-x[i+1])*y[i]/(x[i]-x[i+1])+
        (xk-x[i])*y[i+1]/(x[i+1]-x[i])

   return(yk)
 }

#для побудови двох графіків
x1 = seq(-1.5, 2.5, by=0.01)
for(i in 1:length(x1))
     y1m[i] = LinInterpol(x, y, n, x1[i])
# рисуємо перший графік у вигляді точок
plot(x, y, type="p", col="red")
# додаємо в 1-у графічну підобласть ще лінію
lines(x1, y1m, col="blue")
```

Графіки заданої та інтерполюючої кусочно-лінійної функції:



Обчислення значення функції в заданій точці з використанням записаної процедури:

```
> xk = 2.1
> yk = LinInterpol(x, y, n, xk)
> yk
[1] 10.46
```

Отже, значення функції в точці $x = 2.1$, отримане методом кусочно-лінійної інтерполяції, дорівнює 10.46.

Кусочно-квадратична інтерполяція

Подібно до кусочно-лінійної інтерполяції можна використовувати для наближення на кожному окремому інтервалі (x_i, x_{i+1}) і поліноми вищого порядку, наприклад квадратичну функцію $g(x; a, b, c) = ax^2 + bx + c$. У цьому випадку для знаходження коефіцієнтів a, b, c слід використовувати додаткову точку x_{i+2} (три невідомих, тому необхідно і три рівняння для їх визначення) (рис. 5.2).

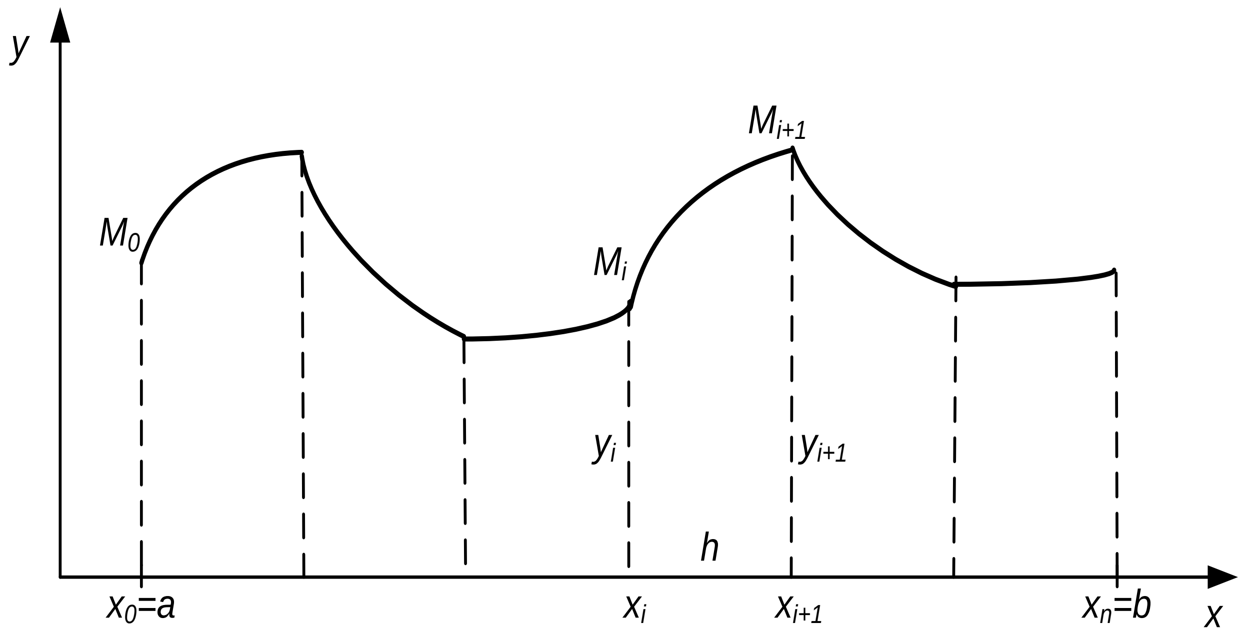


Рис. 5.2. **Графічна інтерпретація кусочно-квадратичної інтерполяції**

Неважко перевірити, що квадратична функція, що проходить через 3 точки (x_i, y_i), (x_{i+1}, y_{i+1}), (x_{i+2}, y_{i+2}), може бути записана у вигляді

$$g_i(x)=\frac{(x-x_{i+1})(x-x_{i+2})}{(x_i-x_{i+1})(x_i-x_{i+2})}\cdot y_i+\frac{(x-x_i)(x-x_{i+2})}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})}\cdot y_{i+1}+$$
$$+\frac{(x-x_i)(x-x_{i+1})}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})}\cdot y_{i+2}. \qquad (5.10)$$

Оцінка погрішності. Нехай функція $f(x)$ тричі неперервно-диференційована на відрізку (x_1, x_n). Тоді для всіх $x \in (x_1, x_n)$

$$|f(x)-g(x)| \le M_3 h^3,$$

де $M_3 = \max\limits_{\xi \in (x_1, x_n)} \left| f^{(3)}(\xi) \right|$, $h = \max\limits_{i=1,n-1} |x_{i+1}-x_i|$.

Таким чином, погрішність кусочно-квадратичної інтерполяції має порядок $O(h^3)$.

Приклад 5.5. Використовуючи кусочно-квадратичну інтерполяцію, обчислити наближене значення функції, заданої таблично, в точці $x = -1.2$.

x	– 1,5	– 1	– 0,5	0	0,5	1	1,5	2	2,5
y	9,8	6,4	7,0	1,7	17,3	5,6	10,8	6,2	27,5

Розв'язання в математичному пакеті R

Відправними даними задачі є вектор значень аргумента x, вектор значень функції y та кількість точок спостереження n:

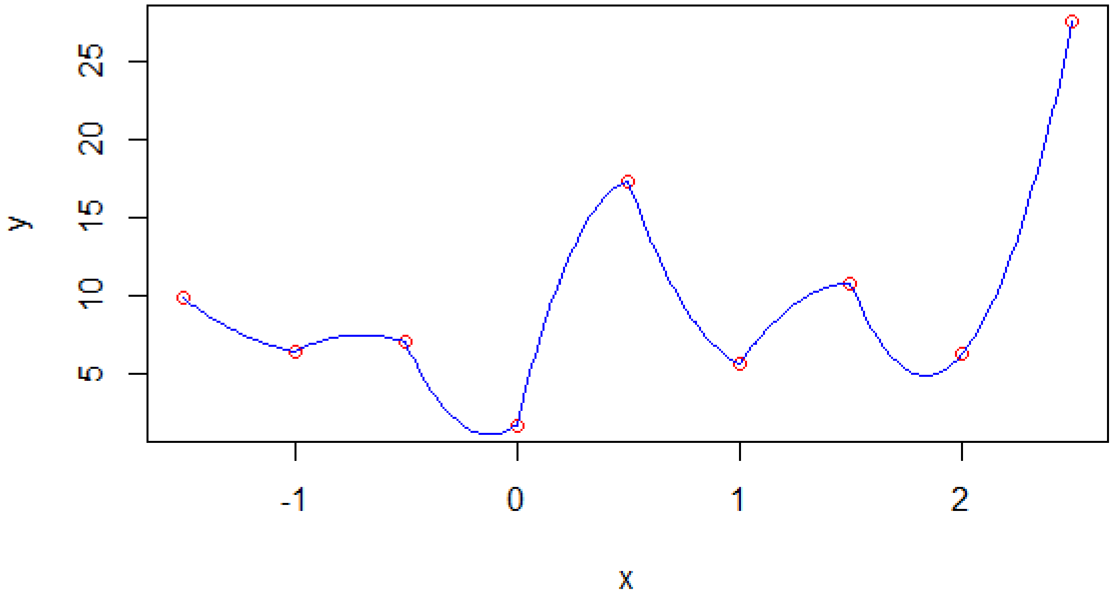
```
x = c(-1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5)
y = c(9.8, 6.4, 7, 1.7, 17.3, 5.6, 10.8, 6.2, 27.5)
n = 9
```

Процедура, що реалізує кусочно-квадратичну інтерполяцію, може бути записана так:

```
# Кусково-квадратична інтерполяція
KwInterpol = function(x, y, n, xk)
{
  for (i in 1:(n-2))
  {
     if (xk <= x[i+1])
        break
  }
  yk = (xk-x[i+1])*(xk-x[i+2])*y[i]/((x[i]-x[i+1])*(x[i]-x[i+2]))+
       (xk-x[i])*(xk-x[i+2])*y[i+1]/((x[i+1]-x[i])*(x[i+1]-x[i+2]))+
       (xk-x[i])*(xk-x[i+1])*y[i+2]/((x[i+2]-x[i])*(x[i+2]-x[i+1]))

  return(yk)
}
```

Графіки заданої та інтерполюючої кусочно-квадратичної функції:



Обчислення значення функції в заданій точці з використанням записаної процедури:

```
> xk = 2.1
> yk = KwInterpol(x, y, n, xk)
> yk
[1] 8.388
```

Отже, значення функції в точці $x = 2.1$, отримане методом кусочно-квадратичної інтерполяції, дорівнює 8.388.

5.4. Інтерполяційний поліном Лагранжа

Якщо відомі значення y_i функції $f(x)$ в n точках x_i, $i = \overline{1,n}$, то поліномом мінімального степеня, що інтерполює функцію $f(x)$,матиме степінь $n-1$. Якщо він записаний у вигляді

$$L_{n-1}(x) = \sum_{i=1}^{n}\left[\prod_{\substack{j=1\\ j\neq i}}^{n}\left(\frac{x-x_j}{x_i-x_j}\right)\right]y_i, \qquad (5.11)$$

то його називають інтерполяційним багаточленом Лагранжа.

Неважко перевірити, що $L_{n-1}(x_i) = f(x_i)$ для всіх $i = \overline{1,n}$.

Погрішність наближення інтерпольованої функції багаточленом Лагранжа сильно залежить від розкиду точок x_i, тобто чим менше інтервал (x_1, x_n), тим точніше $L_{n-1}(x)$ наближає значення функції $f(x)$ в точках $x \in (x_1, x_n)$.

Оцінка погрішності. Нехай функція $f(x)$ n разів неперервно-диференційована на відрізку (x_1, x_n). Тоді [**Ошибка! Источник ссылки не найден.**; **Ошибка! Источник ссылки не найден.**] для всіх $x \in (x_1, x_n)$

$$\left|f(x) - L_{n-1}(x)\right| \le \frac{M_n h^n}{n!},$$

де $M_n = \max\limits_{\xi \in (x_1, x_n)} \left|f^{(n)}(\xi)\right|$, $h = \max\limits_{i=\overline{1,n-1}} \left|x_{i+1} - x_i\right|$.

Таким чином, погрішність інтерполяції за допомогою багаточлена Лагранжа має порядок $O(h^n)$.

Однак між вузлами інтерполяції x_i поліном Лагранжа, як правило, нестійкий до погрішностей обчислень, причому нестійкість зростає зі збільшенням n. Крім того, навіть невеликі погрішності значень y_i можуть сильно змінити поведінку полінома між вузлами. Зазначені ефекти нестійкості виявляються вже при $n \ge 15$, а в практичних задачах, коли n близьке до 100, нестійкість полінома Лагранжа дуже велика.

Приклад 5.6. Використовуючи поліном Лагранжа, обчислити наближене значення таблично заданої функції в точці $x = -1.2$.

x	– 1,5	– 1	– 0,5	0	0,5	1	1,5	2	2,5
y	9,8	6,4	7,0	1,7	17,3	5,6	10,8	6,2	27,5

Розв'язання в математичному пакеті R

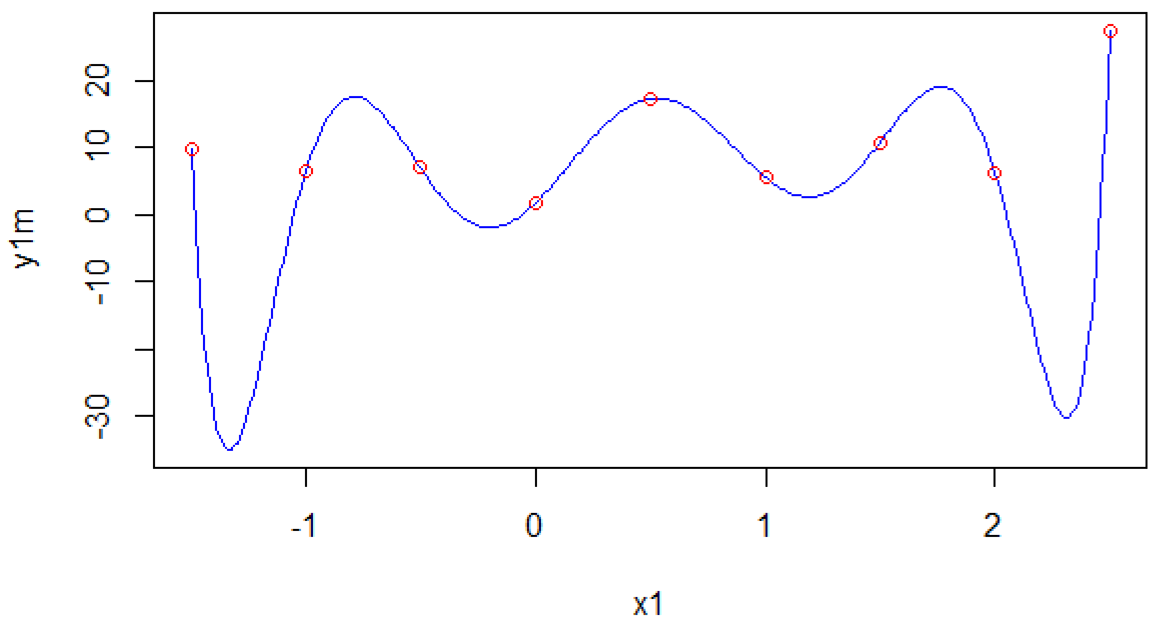
Відправними даними задачі є вектор значень аргумента x, вектор значень функції y та кількість точок спостереження n:

```
x = c(-1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5)
y = c(9.8, 6.4, 7, 1.7, 17.3, 5.6, 10.8, 6.2, 27.5)
n = 9
```

Поліном Лагранжа в загальному вигляді записується так:

```
LagrangeInterpol = function(x, y, n, xk)
{
    yk = 0
    for (i in 1:n)
    {
        P = 1
        for (j in 1:n)
        {
            if (j!=i)
                P = P*((xk-x[j])/(x[i]-x[j]))
        }
        yk = yk + P*y[i]
    }
    return(yk)
}
```

Графіки заданої функції та полінома Лагранжа:



Обчислення значення функції в заданій точці з використанням записаної процедури:

```
> xk =  2.1
> yk = LagrangeInterpol(x, y, n, xk)
> yk
[1] -6.982716
```

Отже, значення функції в точці $x = 2.1$, обчислене за формулою полінома Лагранжа, дорівнює – 6.98.

5.5. Інтерполяційний поліном Ньютона

На відміну від (5.11), поліном мінімального степеня, що інтерполює функцію $f(x)$ в n точках x_i, $i = \overline{1,n}$, може бути записаний в іншому вигляді. Вводяться такі позначення:

розділені різниці 1-го порядку:

$$f(x_1;x_2) = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \text{ , …, } f(x_{n-1};x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}};$$

розділені різниці 2-го порядку:

$$f(x_1;x_2;x_3) = \frac{f(x_2;x_3) - f(x_1;x_2)}{x_3 - x_1}, \dots,$$

$$f(x_{n-2};x_{n-1};x_n) = \frac{f(x_{n-1};x_n) - f(x_{n-2};x_{n-1})}{x_n - x_{n-2}};$$

розділена різниця ($n-1$)-го порядку:

$$f(x_1;x_2;\dots;x_n) = \frac{f(x_2;x_2;\dots x_n) - f(x_1;x_2;\dots x_{n-1})}{x_n - x_1}.$$

Тоді розглядається поліном ($n-1$)-го степеня виду

$$\begin{aligned} H_{n-1}(x) = f(x_1) + (x - x_1)f(x_1;x_2) + \\ + (x - x_1)(x - x_2)f(x_1;x_2;x_3) + \dots + \\ + (x - x_1)(x - x_2)\dots(x - x_{n-1})f(x_1;x_2;\dots;x_n). \end{aligned} \qquad (5.12)$$

Неважко перевірити, що $H_{n-1}(x_i) = f(x_i)\ \forall i = \overline{1,n}$, тобто поліном $H_{n-1}(x)$ є інтерполяційним для функції $f(x)$. Його називають **інтерполяційним поліномом Ньютона**.

Інтерполяційний поліном Ньютона виявляється дуже корисним при розв'язанні інших задач чисельного аналізу, що буде показано далі.

5.6. Сплайн-інтерполяція

Інтерполяція за допомогою поліномів Лагранжа або Ньютона з використанням великої кількості вузлів часто призводить до поганого наближення, що пояснюється накопиченням великої погрішності в процесі обчислень. З іншого боку, кусочно-лінійна і кусочно-квадратична інтерполяція не дозволяють добитися гарного наближення зважаючи на їх теоретичну неточність. Одним із способів добитися гарного наближення в практичних задачах є інтерполяція за допомогою сплайн-функцій.

Визначення. Сплайном (з англ. *spline* – рейка, лінійка) називається кусочно-поліноміальна функція, визначена на відрізку (x_1, x_n) і яка має на цьому відрізку не менше двох неперервних похідних. Сплайн буде позначено через $S(x)$. Інтерполяція за допомогою сплайнів називається **сплайн-інтерполяцією**.

Розглянемо **кубічну сплайн-інтерполяцію** [**Ошибка! Источник ссылки не найден.**], яка найчастіше застосовується на практиці, тобто на кожному інтервалі (x_i, x_{i+1}), $i = \overline{1, n-1}$функція $S(x)$ задається поліномом третьої степені:

$$S_i(x) = y_i + c_{1i}(x - x_i) + c_{2i}(x - x_i)^2 + c_{3i}(x - x_i)^3. \tag{5.13}$$

Очевидно, $S(x_i) = S_i(x_i) = y_i$, $i = \overline{1, n-1}$.

Слід зазначити, що у випадку застосування кубічного сплайна він повинен мати дві неперервні похідні на відрізку (x_1, x_n).

Для того щоб побудувати кубічний сплайн $S(x)$, необхідно знайти всі коефіцієнти c_{1i}, c_{2i}, c_{3i} $(i = \overline{1, n-1})$ поліномів $S_i(x)$, $i = \overline{1, n-1}$. Кількість цих невідомих коефіцієнтів $3(n-1)$.

Коефіцієнти c_{1i}, c_{2i}, c_{3i} $(i = \overline{1, n-1})$ підбираються так, щоб на межах інтервалів (x_i, x_{i+1}) забезпечити неперервність, як функції $S(x)$, так і її першої $S'(x)$ і другої $S''(x)$ похідних. Варто зазначити, що згідно з (5.13) для всіх $i = \overline{1, n-1}$

$$S_i'(x) = c_{1i} + 2c_{2i}(x - x_i) + 3c_{3i}(x - x_i)^2,$$

$$S_i''(x) = 2c_{2i} + 6c_{3i}(x - x_i).$$

Таким чином, повинні виконуватися умови

$$
\begin{gathered}
S_i(x_{i+1}) = S_{i+1}(x_{i+1}) = y_{i+1},\ i = \overline{1, n-2};\ S_{n-1}(x_n) = y_n;\\
S'_i(x_{i+1}) = S'_{i+1}(x_{i+1}) = c_{1,i+1},\ i = \overline{1, n-2};\\
S''_i(x_{i+1}) = S''_{i+1}(x_{i+1}) = c_{2,i+1},\ i = \overline{1, n-2}.
\end{gathered}
\tag{5.14}
$$

Можна переписати (5.14) в більш наочному вигляді

$$
\begin{cases}
y_i + c_{1i}(x_{i+1} - x_i) + c_{2i}(x_{i+1} - x_i)^2 + c_{3i}(x_{i+1} - x_i)^3 = y_{i+1}, i = \overline{1, n-1},\\
c_{1i} + 2c_{2i}(x_{i+1} - x_i) + 3c_{3i}(x_{i+1} - x_i)^2 = c_{1,i+1}, i = \overline{1, n-2},\\
2c_{2i} + 6c_{3i}(x_{i+1} - x_i) = 2c_{2,i+1}, i = \overline{1, n-2}.
\end{cases}
\quad . \tag{5.15}
$$

Як видно, система (5.15) визначає $3(n-1)-2$ лінійних рівняння для пошуку $3(n-1)$ невідомих c_{1i}, c_{2i}, c_{3i} $(i = \overline{1, n-1})$. Тому необхідно довизначити ще два рівняння. Для цього задають так звані **граничні умови** – значення першої або другої похідної функції $f(x)$ на межах інтервалу (x_1, x_n). Якщо значення однієї з похідних на межах відомі, то задавши їх, можна отримати дуже точну інтерполяційну схему. Якщо ж ці значення невідомі, то можна задати другу похідну на межах рівною нулю і отримати також достатньо гарні результати. Необхідно розглянути простий випадок, коли

$$
f''(x_1) = 0, f''(x_n) = 0,
$$

тобто

$$
\begin{cases}
S''_1(x_1) = 2c_{21} = 0,\\
S''_{n-1}(x_n) = 2c_{2,n-1} + 6c_{3,n-1}(x_n - x_{n-1}) = 0.
\end{cases}
\tag{5.16}
$$

Таким чином, для обчислення коефіцієнтів c_{1i}, c_{2i}, c_{3i}, $i = \overline{1, n-1}$, необхідно розв'язати систему лінійних алгебраїчних рівнянь (5.15) – (5.16).

Оцінка погрішності. Нехай функція $f(x)$ 4 рази неперервно-диференційована на відрізку (x_1, x_n). Тоді [**Ошибка! Источник ссылки не найден.**; **Ошибка! Источник ссылки не найден.**] для всіх $x \in (x_1, x_n)$

$$|f(x) - S(x)| \le \frac{M_4 h^4}{8},$$

де $M_4 = \max\limits_{\xi \in (x_1, x_n)} \left| f^{(4)}(\xi) \right|$, $h = \max\limits_{i = \overline{1, n-1}} |x_{i+1} - x_i|$

Таким чином, погрішність інтерполяції за допомогою кубічного сплайна має порядок $O(h^4)$. Перевагою кубічних сплайнів перед іншими наведеними способами інтерполяції є: по-перше, їх стійкість до процесу обчислень, і, по-друге, достатньо висока точність.

При використовуванні сплайн-інтерполяції для таблично заданої функції діють таким чином. Спочатку один раз визначають всі коефіцієнти c_{1i}, c_{2i}, c_{3i}, $i = \overline{1, n-1}$, а потім вже для обчислення наближеного значення функції $f(x)$ у деякій точці $\overline{x} \in (x_1, x_n)$ спочатку знаходять інтервал (x_i, x_{i+1}), якому точка $\overline{x}$ належить, а потім обчислюють наближене значення функції $f(x)$ в точці $\overline{x}$ як $S_i(\overline{x})$, де функція $S_i(x)$ визначена в (5.13).

5.7. Поняття екстраполяції функцій

На відміну від задачі інтерполяції (з англ. *inter* – між), тобто відновлення функції між полюсами, задача екстраполяції (з англ. *extra* – додатково) полягає в відновленні функції за межами полюсів. Тобто, якщо для деякої функції $f(x)$, $f: R^1 \to R^1$ відомо, що в n точках $x_1, x_2, \ldots, x_n$ вона приймає, відповідно, значення $y_1, y_2, \ldots, y_n$, тобто $y_i = f(x_i)$, $i = \overline{1, n}$, то потрібно відновити її значення при $x > x_n$ або $x < x_1$.

Така задача виникає, наприклад, при прогнозуванні в часі значень деякого показника (економічного або фізичного) на основі відомих його значень у вже минулих моментах часу. В цьому випадку x – це час, y – значення показника. Відомий набір значень $y_1, y_2, \ldots, y_n$ називається **часовим рядом**. Прикладами часових рядів є: курси деяких акцій, поквартальний обсяг виробництва, річний ВВП країни, добовий обсяг продажів, погодинний обсяг водоспоживання міста, середньорічна температура на планеті Земля та ін.

У процесі розв’язання задачі екстраполяції застосовують методи апроксимації (див. розділ 7.2), оскільки значення y_i, як правило, містять в собі значні випадкові шуми і тому методи інтерполяції дають велику похибку наближення. Для прогнозування часових рядів, окрім апроксимації (прямі методи), застосовують і спеціальні статистичні методи (адаптивні або стохастичні) [**Ошибка! Источник ссылки не найден.**].

5.8. Висновки

1. Наближення функцій є важливим допоміжним апаратом при розв'язанні інших задач чисельного аналізу.
2. Найбільш відомим і ефективним методом розв'язання задачі апроксимації функцій є метод найменших квадратів.
3. Одним із способів добитися гарного наближення в практичних задачах є інтерполяція за допомогою сплайн-функцій.

5.9. Контрольні запитання та завдання

1. Сформулюйте постановку задачі наближення функцій. Які постановки задачі наближення функцій ви ще знаєте?
2. Чим відрізняються задачі апроксимації, інтерполяції та екстраполяції функцій?
3. У чому полягає метод найменших квадратів для апроксимації функцій?
4. У чому полягає лінійна інтерполяція функцій? Коли її слід застосовувати?
5. У чому полягає квадратична інтерполяція функцій? Коли її слід застосовувати?
6. Що називається інтерполяційним поліномом Лагранжа? Коли його слід застосовувати?
7. Що називають сплайном?
8. У чому полягає лінійна сплайн-інтерполяція функції? Коли її слід застосовувати?
9. Використовуючи метод найменших квадратів, підберіть найкращу апроксимацію для функції, заданої таблично

x	1,6	1,8	2	2,2	2,4	2,6
y	6,6	6,7	6,2	5,8	4,3	3,4

серед функцій виду:

1) $y = a_0 + a_1 x$;

2) $y = a_0 + a_1 x + a_2 x^2$;

3) $y = a_0 + a_1 \sin(a_2 x)$.

Побудуйте графіки апроксимуючих функцій і графіки залишків.

10. Побудуйте кусочно-лінійну та кусочно-квадратичну інтерполяцію для функції, заданої таблицею в завданні 8.