17장 다항식 보간법

17.1 보간법의 소개
17.2 Newton 보간다항식
17.3 Lagrange 보간다항식

17장 다항식 보간법

White (1999) 보고에 의한 1기압에서 온도(T)에 따른 밀도 (ρ), 점성계수 (μ)와 동점성계수 (v)

T (°C)	ρ (kg/m³)	μ (N·s/m²)	v (m²/s)
-40	1.52	1.51×10^{-5}	0.99×10^{-5}
0	1.29	1.71×10^{-5}	1.33×10^{-5}
20	1.20	1.80×10^{-5}	1.50×10^{-5}
50	1.09	1.95×10^{-5}	1.79×10^{-5}
100	0.946	2.17×10^{-5}	2.30×10^{-5}
150	0.835	2.38×10^{-5}	2.85×10^{-5}
200	0.746	2.57×10^{-5}	3.45×10^{-5}
250	0.675	2.75×10^{-5}	4.08×10^{-5}
300	0.616	2.93×10^{-5}	4.75×10^{-5}
400	0.525	3.25×10^{-5}	6.20×10^{-5}
500	0.457	3.55×10^{-5}	7.77×10^{-5}

17장 다항식 보간법

어떻게 원하는 온도에서의 밀도, 점성계수,
그리고 동점성계수를 구할 수 있을까?

→가장 간단한 방법은 인접한 두 점을 잇는 직선을 구한 후, 그 직선 식을 이용하여 원하는 온도에서의 매개변수의 값을 얻는 것이다.

→"선형보간법" (많은 경우에 매우 적절함)이라고 알려진 방법

→데이터가 상당히 큰 곡률을 가지면 오차가 발생

17.1 보간법의 소개

06/02

■ 다항식 보간법

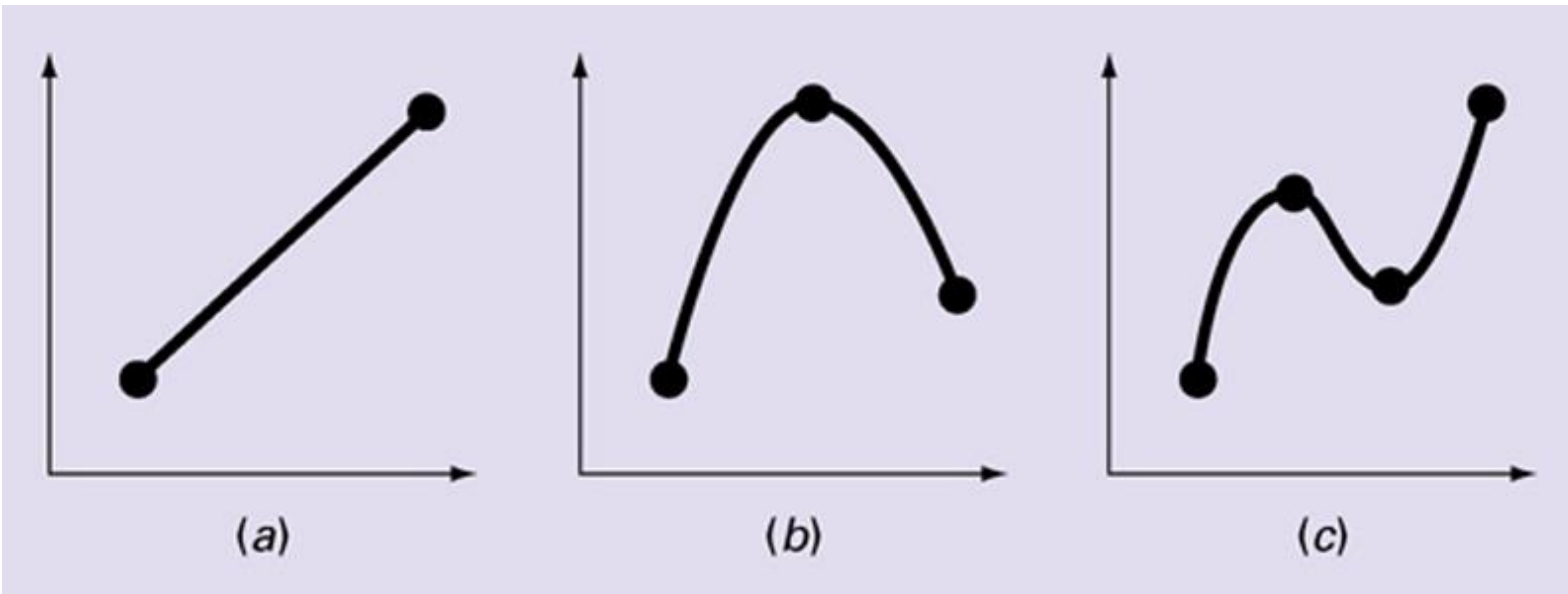
- 정확한 데이터 점들 사이에 위치한 값을 추정
- *n* 개의 데이터 점을 지나는 *유일한* ($n-1$)차 다항식으로 값을 추정

$$f(x) = a_1 + a_2 x + a_3 x^2 + \Lambda + a_n x^{n-1}$$

← 두 점을 지나는 *유일한* 직선 (1차 다항식)

← 세 점을 지나는 *유일한* 포물선 (2차 다항식)

17.1 보간법의 소개

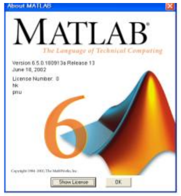


ex) 12개 데이터 ~> 11차 함수

보간다항식의 예:

(a) 두 점을 잇는 1차식, (b)세 점을 잇는 포물선, (c) 네 점을 잇는 3차식

17.1 보간법의 소개



유의사항: MATLAB은 다항식을 다음과 같이 내림차순으로 표현한다.

$$f(x) = p_1 x^{n-1} + p_2 x^{n-2} + \Lambda + p_{n-1} x + p_n$$

- **다항식 계수의 결정**
  - n 개의 대수 방정식으로 n 개의 계수를 동시에 결정

    → 명료한 방법

예제 17.1

Q. 표 17.1의 아래쪽에 기재된 세 개의 밀도값을 지나는 포물선 $f(x) = p_1 x^2 + p_2 x + p_3$ 의 계수를 구하라.

풀이)

(1페이지)

$$\begin{cases} x_1 = 300 \\ x_2 = 400 \\ x_3 = 500 \end{cases} \qquad \begin{aligned} f(x_1) &= 0.616 \\ f(x_2) &= 0.525 \\ f(x_3) &= 0.457 \end{aligned}$$

* 데이터 3개 → ∴ 2차 함수

$$\begin{aligned} 0.616 &= p_1(300)^2 + p_2(300) + p_3 \\ 0.525 &= p_1(400)^2 + p_2(400) + p_3 \\ 0.457 &= p_1(500)^2 + p_2(500) + p_3 \end{aligned}$$

$Ap = b$

$p = A \backslash b$

$= \mathrm{inv}(A) * b$

$$\underbrace{\begin{bmatrix} 90{,}000 & 300 & 1 \\ 160{,}000 & 400 & 1 \\ 250{,}000 & 500 & 1 \end{bmatrix}}_{A} \underbrace{\begin{Bmatrix} p_1 \\ p_2 \\ p_3 \end{Bmatrix}}_{p} = \underbrace{\begin{Bmatrix} 0.616 \\ 0.525 \\ 0.457 \end{Bmatrix}}_{b}$$

예제 17.1

Command Window

File Edit View Web Window Help

```
>> format long
>> A = [90000 300 1; 160000 400 1; 250000 500 1];
>> b = [0.616 0.525 0.457]';
>> p = A\b

p =        0.00000115000000
          -0.00171500000000
           1.02700000000000
```

*

short 하면 0 이나옴

p_1 & p_2 & p_3

Ready

예제 17.1

따라서 2차 식은

$$f(x) = \overset{p_1}{\underline{0.00000115}}x^2 - \overset{p_2}{\underline{0.001715}}x + \overset{p_3}{\underline{1.027}}$$

350°C에서의 밀도를 계산하면

$$f(350) = 0.00000115(350)^2 - 0.001715(350) + 1.027 = 0.567625$$

17.1 보간법의 소개

예제 17.1에 나타나는 계수행렬은 특정한 형태의 구조를 가진다.

$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix} \begin{Bmatrix} p_1 \\ p_2 \\ p_3 \end{Bmatrix} = \begin{Bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \end{Bmatrix}$$

→ Vandermonde matrix
매우 불량한 조건의 행렬
→ 반올림오차에 매우 민감

다른 방법으로 컴퓨터 실행에 적합한 *Newton*과 *Lagrange* 다항식을 다룬다.

\# det(A)

MATLAB 함수: polyfit과 polyval

- 데이터 점의 수 > 계수의 수 → 다항식 *회귀분석*
- 데이터 점의 수 = 계수의 수 → 다항식 *보간*

17.2 Newton 보간다항식

▪ 선형보간법

닮은꼴 삼각형에서

$$\frac{f_1(x)-f(x_1)}{x-x_1}=\frac{f(x_2)-f(x_1)}{x_2-x_1}$$

따라서 ***Newton 선형보간공식***은 다음과 같다.

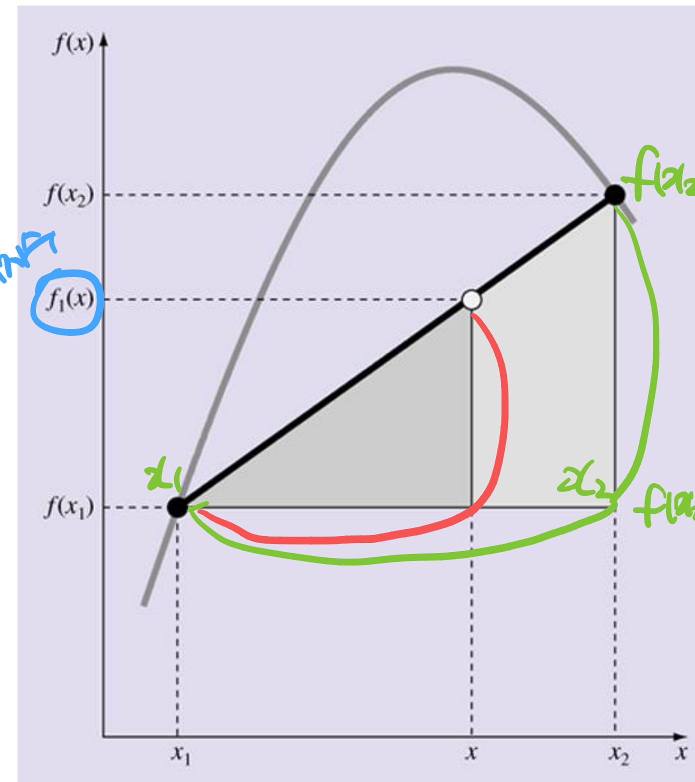
* 기울기 구하기

$$f_1(x)=f(x_1)+\frac{f(x_2)-f(x_1)}{x_2-x_1}(x-x_1)$$

1차 도함수의 유한제차분근사값

여기서 $f_1(x)$ = 1차 보간다항식

- 간격이 작을수록 → 보다 나은 근사값



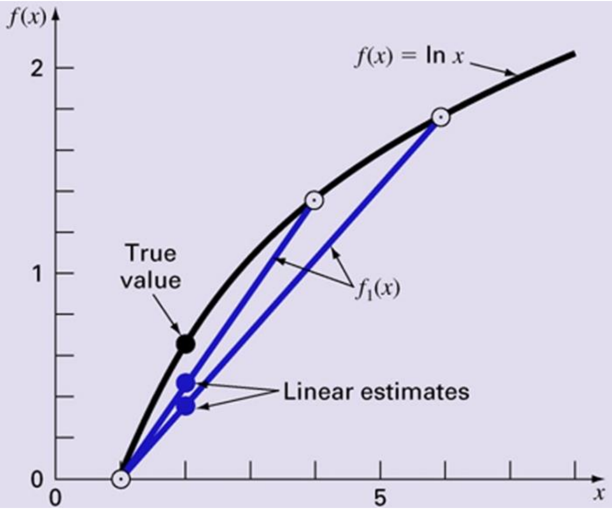
선형보간법의 도식적 표현

예제 17.2 (선형보간법)

$= \log(2) = \log_e(2)$: 자연 log

$\neq \log_{10}(2)$: 상용 log

Q. 선형보간법을 이용하여 자연로그 ln2의 값을 추정하라.
먼저 보간법으로 $\ln 1 = 0$과 $\ln 6 = 1.791759$사이에서 추정하고, 그 다음에 더 작은 간격인 $\ln 1$과 $\ln 4 (= 1.386294)$ 사이에서 추정하라. 참고로 $\ln 2 = 0.6931472$이다.



풀이) x_1 = 1과 x_2 = 6에 대하여

$$f_1(2) = 0 + \frac{\overset{f(x_2)}{1.791759} - \overset{f(x_1)}{0}}{\underset{x_2}{6} - \underset{x_1}{1}}(2-1) = 0.3583519$$

x_1 = 1과 x_2 = 4에 대하여

$$f_1(2) = 0 + \frac{1.386294 - 0}{4-1}(2-1) = 0.4620981$$

백분율 상대오차는 각각 48.3%와 33.3%이다.

17.2 Newton 보간다항식

- Newton 보간다항식의 일반적인 형태

  n 개의 데이터 점에 $(n-1)$차 다항식을 접합시키는 것으로 일반화할 수 있다.

$$f_{n-1}(x) = b_1 + b_2(x - x_1) + \Lambda + b_n(x - x_1)(x - x_2)\Lambda\ (x - x_{n-1})$$

→ b들을 구해야...

  n 개의 데이터 점 $[x_1, f(x_1)], [x_2, f(x_2)], \Lambda, [x_n, f(x_n)]$을 이용하여 계수 b_1, b_2, Λ, b_n을 계산한다.

$$b_1 = f(x_1)$$

$$b_2 = f[x_2, x_1]$$

$$b_3 = f[x_3, x_2, x_1]$$

$$\mathrm{M}$$

$$b_n = f[x_n, x_{n-1}, \mathrm{K}, x_2, x_1]$$

17 Newton 보간다항식

- 2차 보간법
  - 세 개의 데이터 점
  - 2차 다항식 = 포물선

$$f_2(x) = b_1 + b_2(x - x_1) + b_3(x - x_1)(x - x_2)$$ → **Taylor 급수전개와 유사함**

$$b_1 = f(x_1)$$ ≈ 절편

$$b_2 = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$ ≈ 기울기

→ **두 점 x_1 과 x_2를 잇는 직선의 기울기**

$$b_3 = \frac{\dfrac{f(x_3) - f(x_2)}{x_3 - x_2} - \dfrac{f(x_2) - f(x_1)}{x_2 - x_1}}{x_3 - x_1}$$

→ **2차 곡률**

→ **2차 도함수의 유한제차분근사와 매우 유사 식 (4.27) 참조**

17.2 Newton 보간다항식

여기서 괄호로 표시된 함수는 유한제차분을 나타낸다.

- 1차 유한제차분 $$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j}$$
- 2차 유한제차분 $$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k}$$
  $(k<i)$
- $(n-1)$차 유한제차분

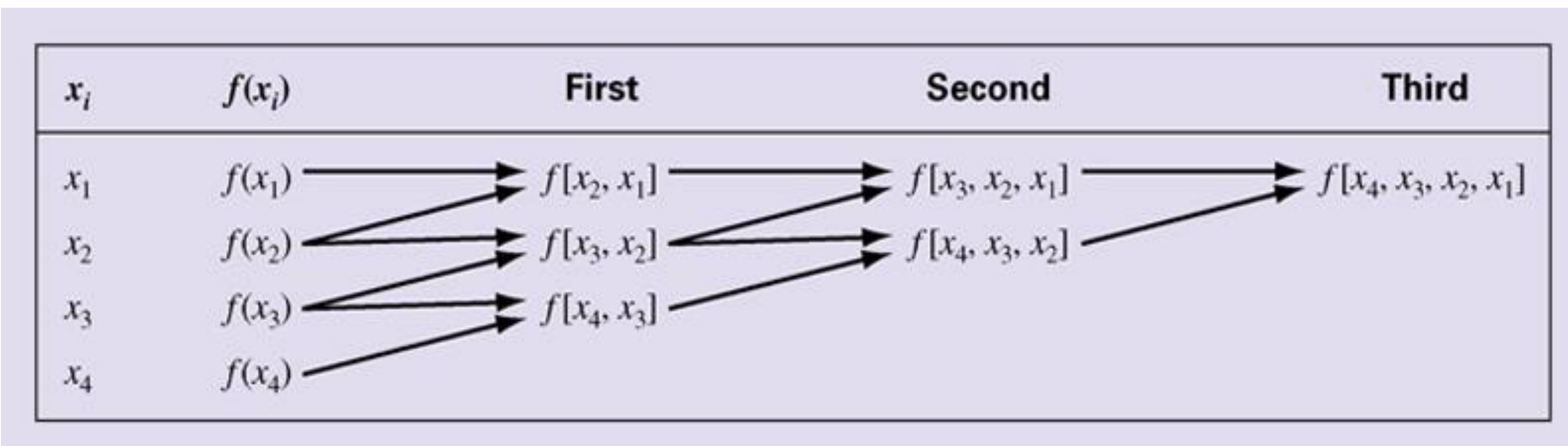
$$f[x_n, x_{n-1}, \mathrm{K}\ , x_2, x_1] = \frac{f[x_n, x_{n-1}, \mathrm{K}\ , x_2] - f[x_{n-1}, x_{n-2}, \mathrm{K}\ , x_1]}{x_n - x_1}$$

다음과 같은 일반적인 Newton 보간다항식을 얻을 수 있다.

$$f_{n-1}(x) = f(x_1) + (x - x_1) f[x_2, x_1] + (x - x_1)(x - x_2) f[x_3, x_2, x_1]$$
$$+ \Lambda\ + (x - x_1)(x - x_2) \Lambda\ (x - x_{n-1}) f[x_n, x_{n-1}, \mathrm{K}\ , x_2, x_1]$$

17.2 Newton 보간다항식

- 데이터 점들이 등간격일 필요가 없고, 수평축의 좌표값이 올림차순일 필요도 없다.
- 고차 차분이 저차 차분의 차이에 의해 계산되어지는 순환적인 시스템이다.



17.2 Newton 보간다항식

[Newton 보간다항식을 실행하는 M-파일]

Untitled

File Edit View Text Debug Breakpoints Web Window Help

Stack: Base

```
function yint = Newint(x,y,xx)
% Newint(x,y,xx)
%    Newton interpolation uses an (n-1)-order Newton interpolating
%    polynomial based on n data points (x,y) to determine yint at xx.
% input
%   x = independent variable
%   y = dependent variable
%   xx = value of independent variable for interpolation
% output
%   yint = interpolated value of dependent variable
```

Ready

17.2 Newton 보간다항식

[Newton 보간다항식을 실행하는 M-파일]

```
% compute a difference table
n = length(x);
if length(y) ~=n, error ('x and y must be same length'); end
b = zeros(n,n);
% assign dependent variables to the first column of b
b(:,1) = y(:);  % (:) ensures that y is a column vector
for j = 2:n
    for i=1:n-j+1
        b(i,j) = (b(i+1,j-1)-b(i,j-1))/(x(i+j-1)-x(i));
    end
end
```

3

b(1,2)
b(2,2)
b(3,2)
b(1,3)
b(2,3)

17.2 Newton 보간다항식

[Newton 보간다항식을 실행하는 M-파일]

Untitled

File Edit View Text Debug Breakpoints Web Window Help

Stack: Base

```
% use the finite divided differences to interpolate
xt = 1;
yint = b(1,1);
for j = 1:n-1
   xt = xt.*(xx-x(j));
   yint = yint + b(1,j+1)*xt;
end
```

ok

$b_1 + b_2(x-x_1) + b_3(x-x_2x_1)$

Ready

17.2 Newton 보간다항식

Command Window

File Edit View Web Window Help

\>> format long

\>> x =[1 4 6 5]';

\>> y = log(x);

\>> Newint(~~x~~,y,2) % ln 2의 값을 추정

ans =

0.62876857890841

Ready

* 4개의 데이터를 가지고 구함 ⇒ 3차 함수

yint = Newint(x, y, 2) 이런식으로 변수를 지정!!

17.2 Newton 보간다항식

점점 커져야 함

```
>> xx=[1 2 3 4 5 6]';    % 6개 점에서의 값을 추정
>> format short
>> Newint(x,y,xx)
ans =
         0
    0.6288
    1.0751
    1.3863
    1.6094
    1.7918
>> Newint(x,y,2)         % ln 2의 값을 추정
ans =
    0.6288
```

17.3 Lagrange 보간다항식

weighting factor

직선으로 연결하고자 하는 두 값의 가중평균으로 선형 보간다항식을 만들어 보자.

$$f(x) = L_1 f(x_1) + L_2 f(x_2)$$

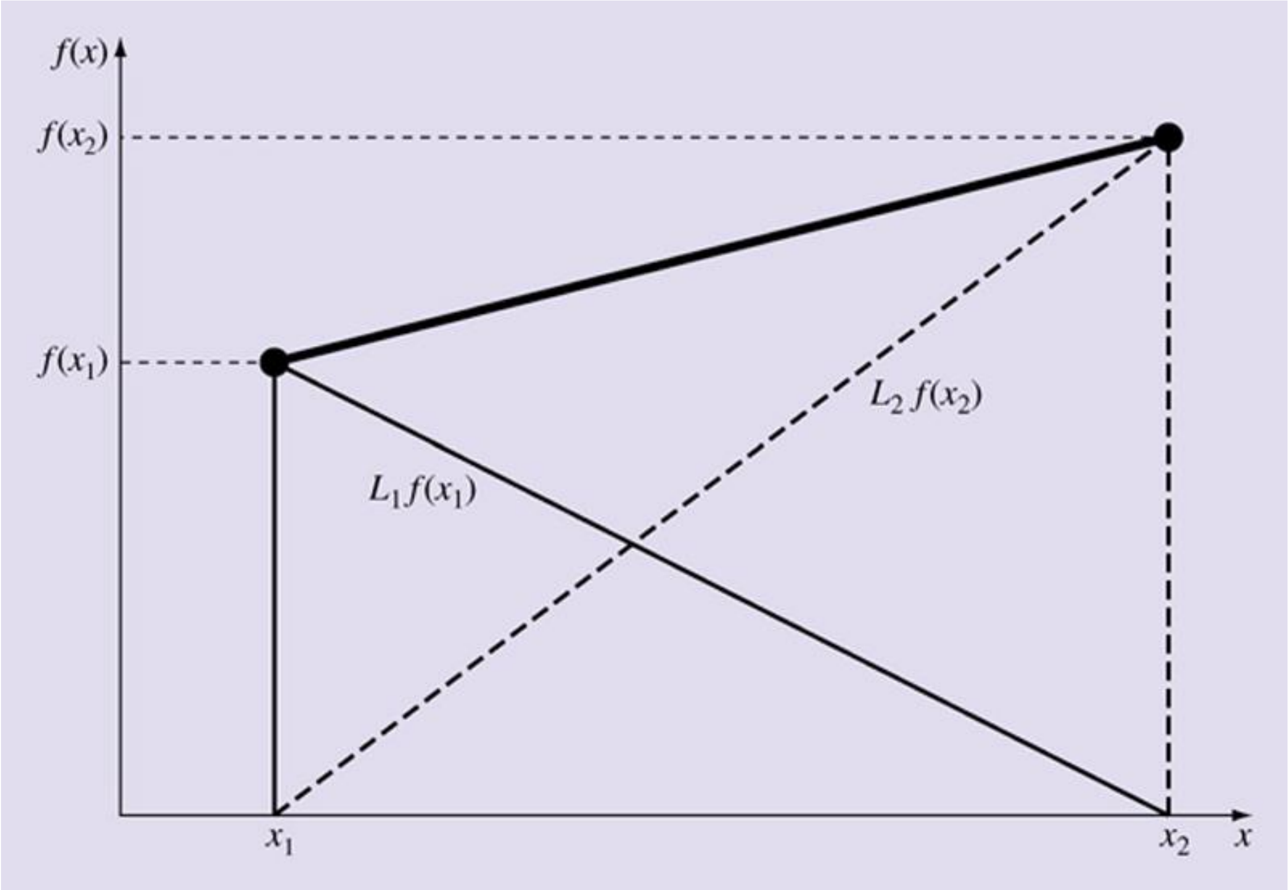
여기서 L은 직선에 대한 가중계수이다.

- L_1은 x_1에서 1이며, x_2에서는 0이다. → $L_1 = \dfrac{x - x_2}{x_1 - x_2}$
- L_2는 x_2에서 1이며, x_1에서는 0이다. → $L_2 = \dfrac{x - x_1}{x_2 - x_1}$

따라서

$$f_1(x) = \frac{x - x_2}{x_1 - x_2} f(x_1) + \frac{x - x_1}{x_2 - x_1} f(x_2)$$: *선형 Lagrange 보간다항식*

17.3 Lagrange 보간다항식



Lagrange 보간다항식의 근본 원리에 대한 시각적 표현으로 1차식의 경우를 나타낸다. 두 항의 합은 두 점을 연결하는 유일한 직선이다.

17.3 Lagrange 보간다항식

세 점에 대해 확장 → 2차 Lagrange 보간다항식

2차 2차 2차

$$f_2(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} f(x_1) + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} f(x_2) + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} f(x_3)$$

* 규칙성 파악

- 각 포물선은 세 점 중 한 점을 지나고 나머지 점에서는 0 이다.
- 세 포물선의 합은 세 점을 지나는 유일한 포물선이다.

17.3 Lagrange 보간다항식

고차 Lagrange 다항식으로 일반화하면

$$f_{n-1}(x) = \sum_{i=1}^{n} L_i(x) f(x_i)$$

여기서

$$L_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

∵ 분모 ≠ 0

ex) i=1 이면, j=1 이 X

j=2 부터

n = 데이터 점의 수

17.3 Lagrange 보간다항식

[Lagrange 보간다항식을 실행하는 M-파일]

```
function yint = Lagrange(x,y,xx)
% Lagrange(x,y,xx)
%    Lagrange interpolation uses an (n-1)-order Lagrange
interpolating
%    polynomial based on n data points (x,y) to determine yint
at xx.
% input
%   x = independent variable
%   y = dependent variable
%   xx = value of independent variable for interpolation
% output
%   yint = interpolated value of dependent variable
```

17.3 Lagrange 보간다항식

[Lagrange 보간다항식을 실행하는 M-파일]

Untitled

File Edit View Text Debug Breakpoints Web Window Help

```
n = length(x);
if length(y) ~=n, error ('x and y must be same length'); end
s = 0;
for i = 1:n
   product = y(i);
   for j = 1:n
      if i ~= j
         product = product .* (xx-x(j))/(x(i)-x(j));
      end
   end
   s = s + product;
end
yint = s;
```

Ready

ok =3

$y(1) = f(x_1)$

$y(2) = f(x_2)$

$\vdots$

$\Rightarrow 2, 3$

-: 분모≠0

$f(x_i)$

17.3 Lagrange 보간다항식

Command Window

File Edit View Web Window Help

```
>> format long;
   T = [-40 0 20 50]';
   d = [1.52 1.29 1.2 1.09]';
   density = Lagrange(T,d,15)
density =
   1.22112847222222
```

Ready

T(f)

T= [0 20]';

d= [1.29 1.2]';

얘는~ linear Reg임

(-: 2개데이터)

예제 17.7 (고차 다항식보간법의 위험성)

18장에서 해결

Q. 1901년에 Carl Runge는 아래의 간단한 함수로 고차 다항식보간법의 위험성을 보였다.

$$f(x) = \frac{1}{1+25x^2}$$

이 함수는 Runge 함수라고 불린다. 구간 [-1, 1]에서 등간격으로 5개와 11개로 데이터가 주어질 때 polyfit 함수와 polyval 함수를 사용하여 4차와 10차 다항식으로 접합시켜 예측한 값을 실제 값과 그림으로 비교하라.

예제 17.7 (고차 다항식보간법의 위험성)

Command Window

File Edit View Web Window Help

①

```
>> x = linspace(-1,1,5);  y = 1./(1+25*x.^2);
   xx = linspace(-1,1);
>> p = polyfit(x,y,4);  y4 = polyval(p,xx);
   yr = 1./(1+25*xx.^2);
   plot(x,y,'o',xx,y4,xx,yr,'--')      % 4차 보간다항식 결과 그래프
```

②

```
>> x = linspace(-1,1,11);          y = 1./(1+25*x.^2);
   p = polyfit(x,y,10);             y10 = polyval(p,xx);
   plot(x,y,'o',xx,y10,xx,yr,'--')  % 10차 보간다항식 결과 그래프
```

5개

시작 끝

100개 (* 개수 X 면, default 100개)

4차

내장함수

→ 4차이므로 데이터 5개 ☆

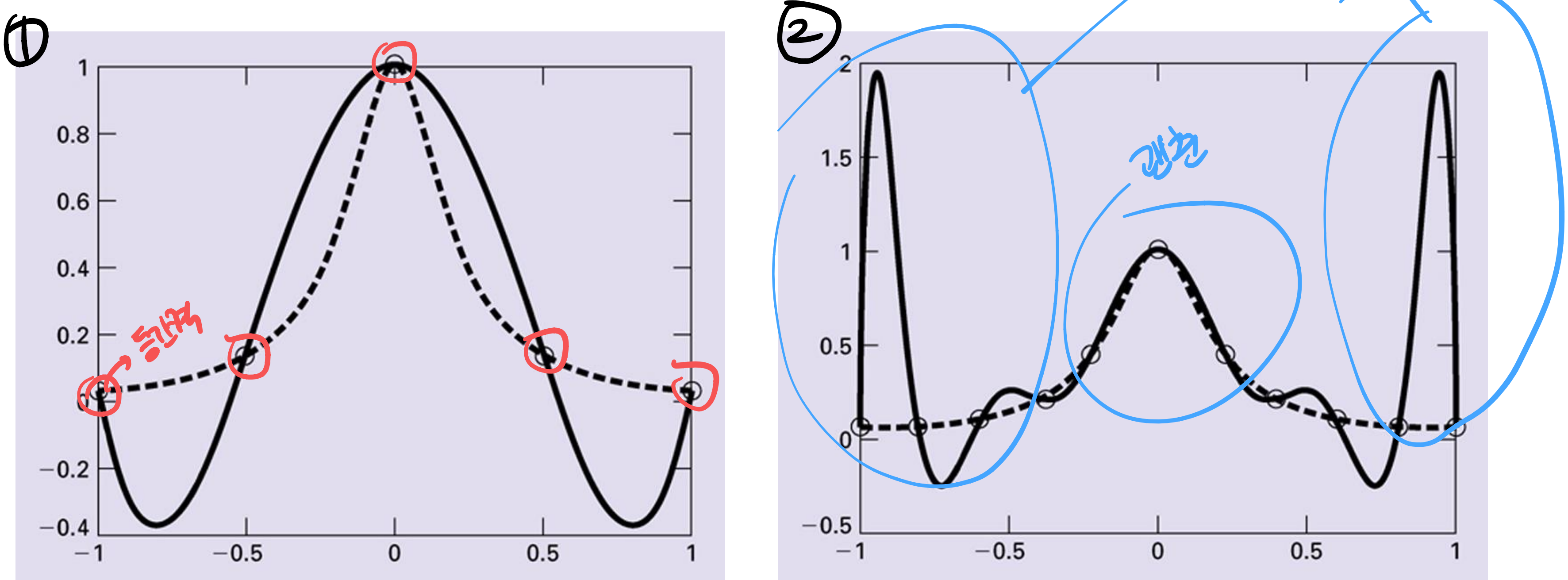
10차

10차이므로 데이터 11개

$p = P(1)x^4 + P(2)x^3 + P(3)x^2 + P(4)x$

Ready

예제 17.7 (고차 다항식보간법의 위험성)



- 접합은 특히 구간의 양 끝에서 더욱 나빠지게 된다.
- 고차 다항식이 필요한 상황도 있지만 보통의 경우에는 고차 다항식의 사용을 피해야 한다. 대부분의 공학과 과학 문제에서 이 장에서 설명한 형태의 저차 다항식은 데이터의 곡선 형상을 진동 없이 표현하는데 효과적으로 사용할 수 있다.