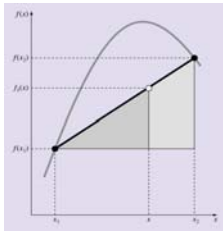
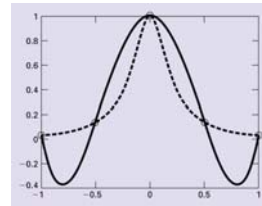


Chap. 17 다항식 보간법 (Interpolation)



- ❖ 다항식 보간의 개요
- ❖ Newton 보간다항식
- ❖ Lagrange 보간다항식
- ❖ 외삽법과 진동



[Review] 곡선맞춤 (Curve Fitting)

□ 곡선맞춤

- 불연속적인 데이터 사이에 있는 점에서의 값 추정시 필요
- 주어진 데이터를 가장 적절히 표현할 수 있는 함수식 계산
- 곡선맞춤 종류: 최소제곱 회귀분석 vs. 보간법

□ 최소제곱 회귀분석 (Least-Square Regression)

- 데이터가 상당한 크기의 오차를 포함하거나 데이터가 산재하는 경우
- 데이터의 일반 경향을 나타낼 수 있는 단일 곡선을 유도하는 방법
- 유도된 곡선이 모든 점(데이터)를 포함할 수는 없음

□ 보간법 (Interpolation)

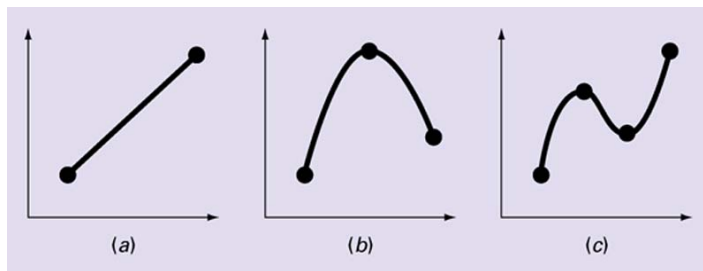
- 데이터가 매우 정확하게 알려져 있는 경우
- 각 데이터를 직접 통과하는 하나의 곡선 또는 일련의 곡선을 만드는 방법

□ 다항식 보간법의 개요

- n개의 데이터가 있는 경우 (n-1)차의 다항식 보간 가능

$$f(x) = a_1 + a_2x + a_3x^2 + \dots + a_nx^{n-1} \quad : \text{일반적인 표현 형태}$$

$$f(x) = p_1x^{n-1} + p_2x^{n-2} + \dots + p_{n-1}x + p_n \quad : \text{Matlab에서의 사용 형태}$$



□ 연립방정식을 사용한 보간다항식의 결정

- 3개의 데이터 point $(x_1, y_1), (x_2, y_2), (x_3, y_3)$

- 2차 보간다항식
$$p(x) = p_1x^2 + p_2x + p_3$$

- 연립방정식 구성
$$p(x_1) = p_1x_1^2 + p_2x_1 + p_3 = y_1$$

$$p(x_2) = p_1x_2^2 + p_2x_2 + p_3 = y_2$$

$$p(x_3) = p_1x_3^2 + p_2x_3 + p_3 = y_3$$

- 행렬식 풀어 계수 결정
$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Ill-conditioned matrix

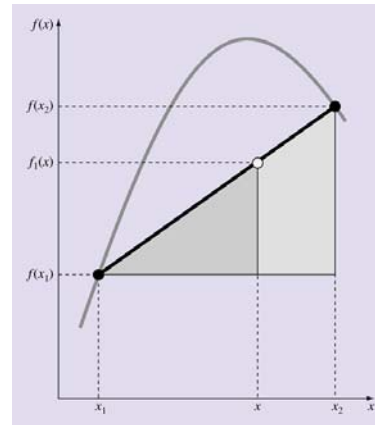
P. 434 예제 17.1

Newton 보간다항식

□ 선형보간법

- 두개의 점 $(x_1, f(x_1)), (x_2, f(x_2))$ 를 사용하여 선형보간한 함수를 $f_1(x)$ 라 하면

P. 437 예제 17.2

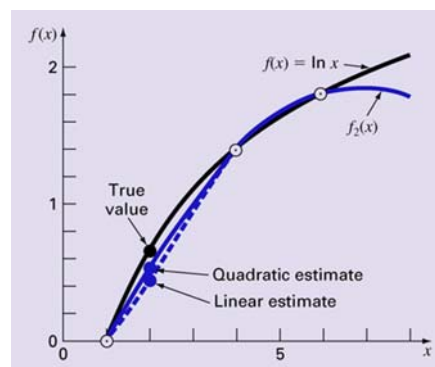


Newton 보간다항식

□ 2차보간법

- 3개의 점 $(x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3))$ 를 사용하여 2차보간 : $f_2(x)$

P. 439 예제 17.3



Newton 보간다항식

□ 일반적인 형태

- n개의 점 $(x_1, f(x_1)), (x_2, f(x_2)), \dots, (x_n, f(x_n))$

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

$$b_1 = f(x_1)$$

$$b_2 = f[x_2, x_1] = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \quad : 1차 유한제차분$$

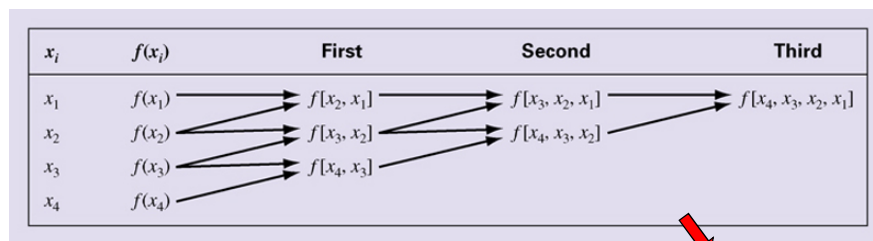
: 2차 유한제차분

: (n-1)차 유한제차분

Newton 보간다항식

□ 일반적인 형태 (cont'd)

P. 441 예제 17.4



유한제차분의 순환적(recursive) 특성

$b(i, j)$

Newton 보간다항식 – M File

```

function yint = Newtint(x,y,xx)
% yint = Newtint(x,y,xx):
% Newton interpolation. Uses an (n - 1)-order
% Newton interpolating polynomial based on n
% data points (x, y) to determine a value of the
% dependent variable (yint) at a given value
% of the independent variable, xx.
% input:
% x = independent variable
% y = dependent variable
% xx = value of independent variable at which
% interpolation is calculated
% output:
% yint = interpolated value of dependent variable

% compute the finite divided differences in the
% form of 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(:); % 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

% 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
    
```

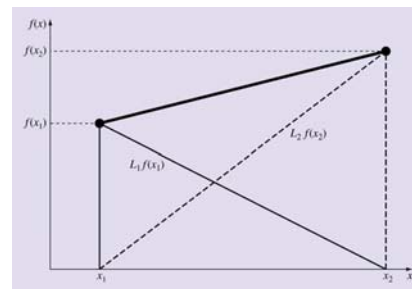
Lagrange 보간다항식

□ 선형보간법

- 두개의 점 $(x_1, f(x_1)), (x_2, f(x_2))$ 를 사용하여 선형보간한 함수를 $f_1(x)$ 라 하면

$$n = 1 \quad (x_1, f(x_1)) \quad (x_2, f(x_2))$$

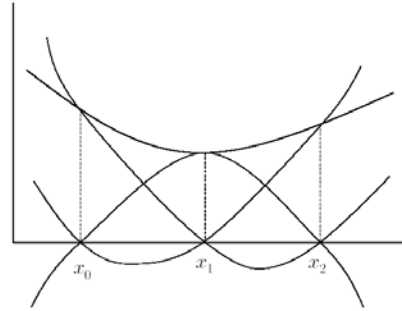
L : 가중함수 (Weight function)



□ 2차보간법

- 3개의 점 $(x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3))$ 를 사용하여 2차보간 : $f_2(x)$

P. 445 예제 17.5



□ 일반적인 형태

- n개의 점 $(x_1, y_1), \dots, (x_n, y_n)$



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

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

```
function yint = Lagrange(x,y,xx)

% yint = Lagrange(x,y,xx):
% Lagrange interpolation. Uses an (n - 1)-order Lagrange
% interpolating polynomial based on n data points (x, y)
% to determine a value of the dependent variable (yint)
% at a given value of the independent variable, xx.
% input:
% x = independent variable
% y = dependent variable
% xx = value of independent variable at which the
% interpolation is calculated
% output:
% yint = interpolated value of dependent variable

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

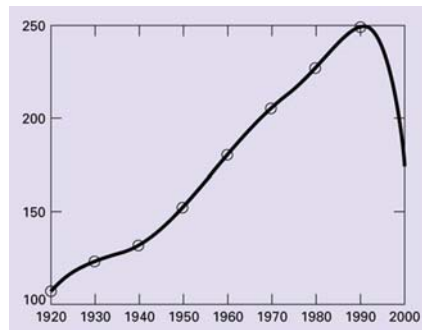
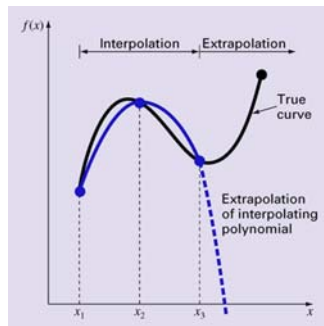
yint = s;
```

외삽법 (Extrapolation)

□ 내삽법과 외삽법

- 내삽법 (Interpolation): x_1, x_2, \dots, x_n 의 범위 내에 있는 $f(x)$ 값 계산
- 외삽법 (Extrapolation): x_1, x_2, \dots, x_n 의 범위 밖에 있는 $f(x)$ 값 계산

P. 448 예제 17.6

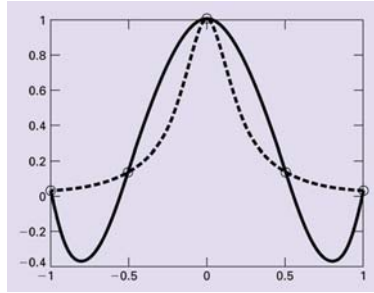


□ 고차 다항식보간의 위험성

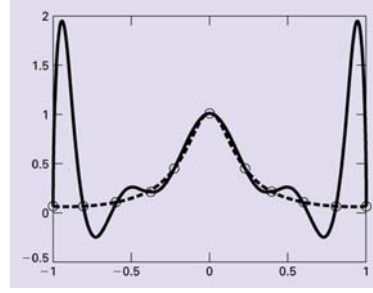
- 다항식 차수가 높아지면 진동 발생 위험
- 고차다항식의 경우 반올림 오차에 민감하여 불량한 조건이 되기 쉬움

P. 450 예제 17.7

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



Runge 함수: 5개 점, 4차 다항식 접합



Runge 함수: 11개 점, 10차 다항식 접합