과제 7

- LearnUs에 올린 초신성 광도곡선 데이터를 이용한다.
  - 16개의 (JD, B-mag)
- JD의 최소값과 최대값 사이를 균등하게 나누어 100개의 JD 값을 생성한다.
- 직접 구현한 1차 선형 보간(linear interpolation) 함수를 사용하여 각 JD에 대응하는 B-mag 값을 계산하라
  - 표준 라이브러리(`numpy`, `scipy` 등)의 보간 함수 사용 금지
- 원래의 관측 데이터와 보간된 데이터를 하나의 그래프에 함께 출력하라.
- 토론
  - 보간 결과가 물리적으로 의미 있다고 생각하는가?
  - 보간의 한계는 무엇이며, 어떤 경우에 보간이 오히려 잘못된 결론을 이끌 수 있는가?
- 선택과제 (가산점 30%):
  - 관측값에 잡음(noise)이 있다고 가정하고, 보간 결과에 어떤 영향을 미치는지 시뮬레이션하고 분석하라.

Filter	JD	Mag	dMag
B	2454007.60200	17.583	0.018
B	2454009.66300	17.011	0.017
B	2454012.60600	16.501	0.016
B	2454019.60500	16.059	0.014
B	2454021.57900	16.064	0.015
B	2454025.59900	16.127	0.013
B	2454031.57700	16.460	0.015
B	2454034.59600	16.715	0.014
B	2454046.59400	17.693	0.050
B	2454052.63300	18.312	0.026
B	2454054.58600	18.521	0.026
B	2454055.59800	18.575	0.034
B	2454056.59900	18.573	0.031
B	2454062.57900	18.825	0.032
B	2454066.57500	19.003	0.063
B	2454071.56100	19.157	0.158

천문학에서 일반 달력 사용시 발생하는 문제점

Julian Date

Astronomers refer to a Julian date as the number of days since the beginning of the Julian Period. (**January 1, 4713** BC).

JDN = (1461 × (Y + 4800 + (M – 14)/12))/4 +(367 × (M – 2 – 12 × ((M – 14)/12)))/12 – (3 × ((Y + 4900 + (M - 14)/12)/100))/4 + D – 32075

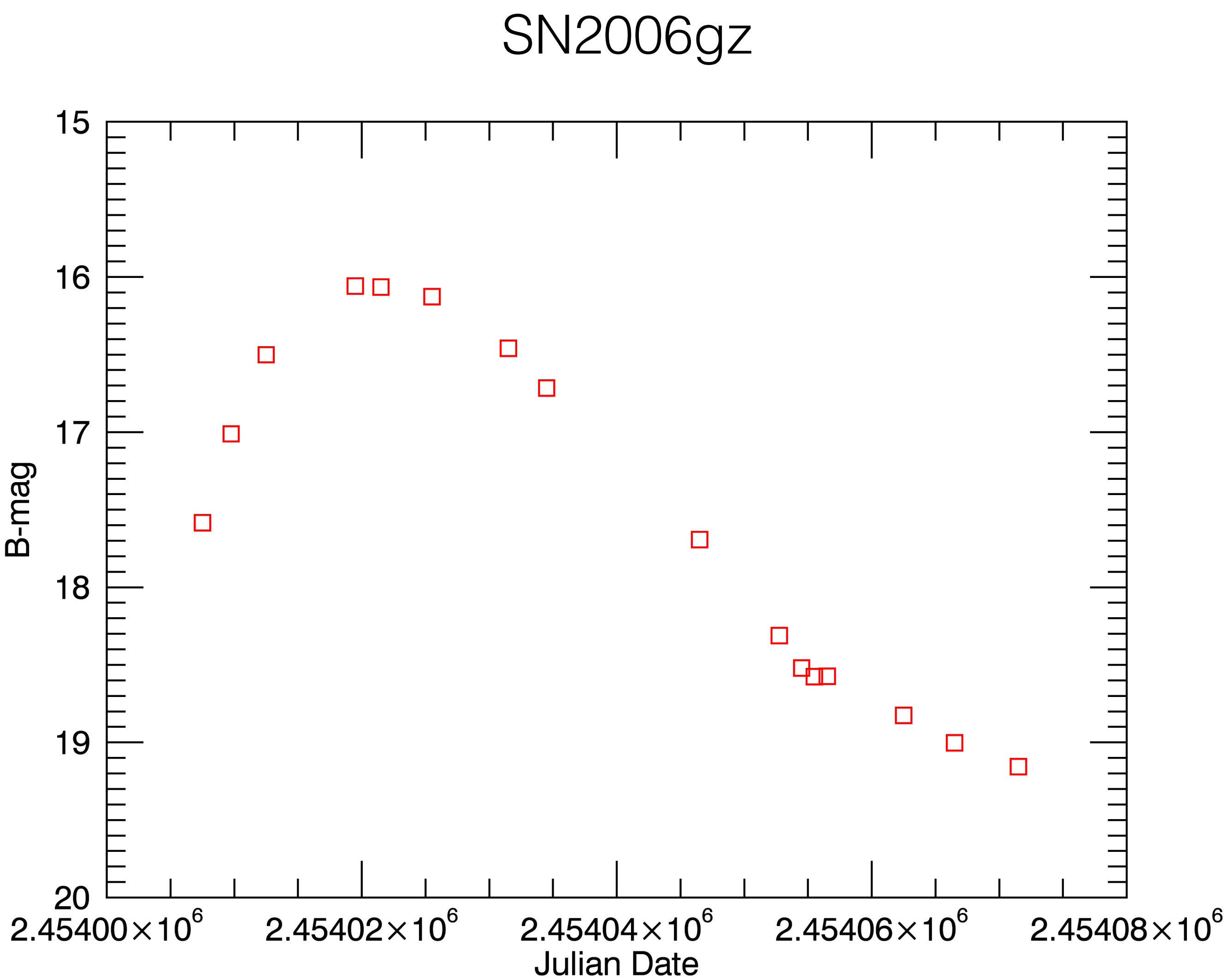
$$\mathrm{JDN} = \left\lfloor \frac{1461 \cdot (Y + 4800 + \lfloor \frac{M-14}{12} \rfloor)}{4} \right\rfloor + \left\lfloor \frac{367 \cdot (M - 2 - 12 \cdot \lfloor \frac{M-14}{12} \rfloor)}{12} \right\rfloor - \left\lfloor \frac{3 \cdot \left\lfloor \frac{Y+4900+\lfloor \frac{M-14}{12} \rfloor}{100} \right\rfloor}{4} \right\rfloor + D - 32075$$

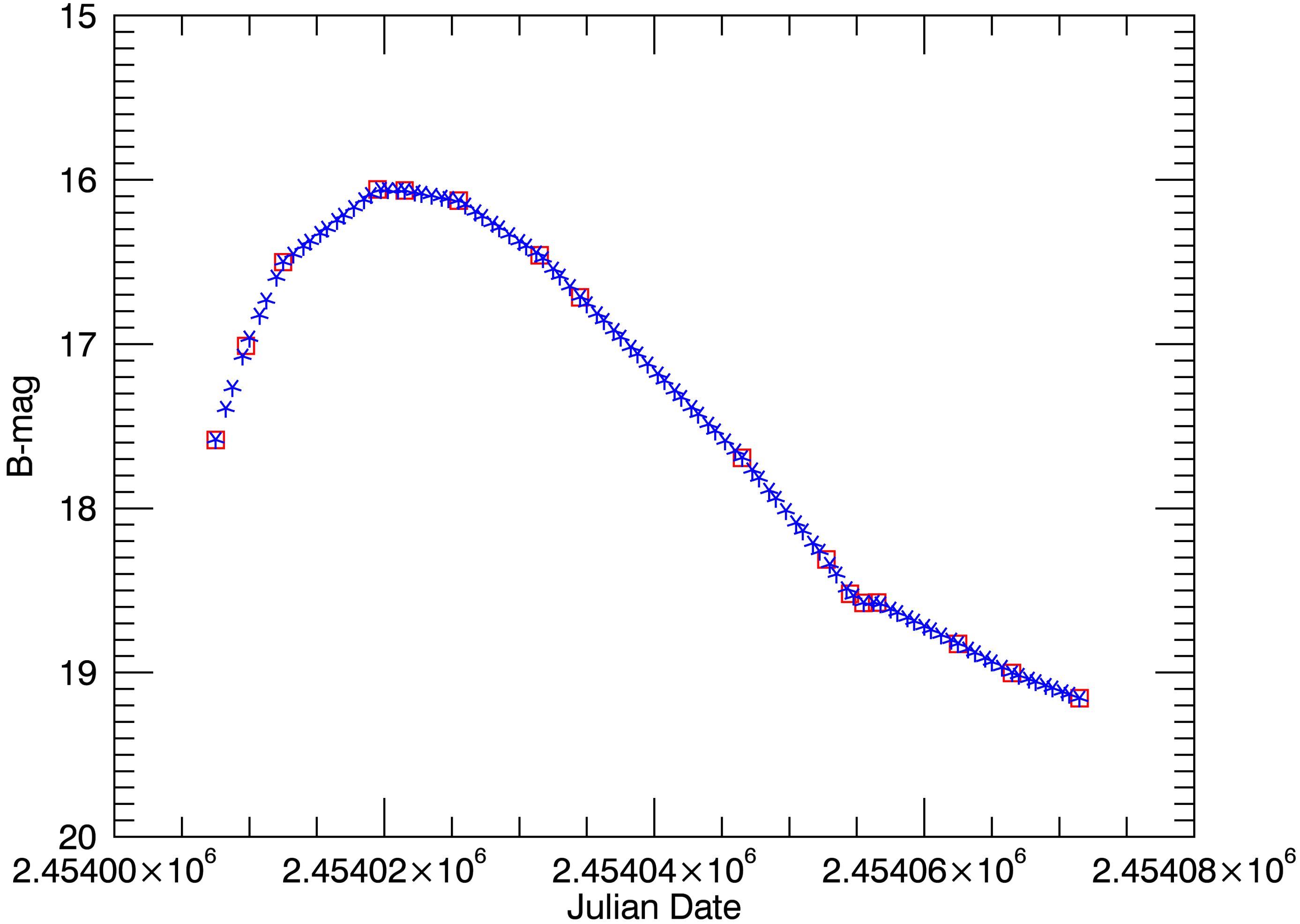
-연도(Y), 월(M), 일(D)는 정수여야 함, []는 정수만 취함

$$JD = JDN + \frac{\text{hour} - 12}{24} + \frac{\text{minute}}{1440} + \frac{\text{second}}{86400} - 0.5$$

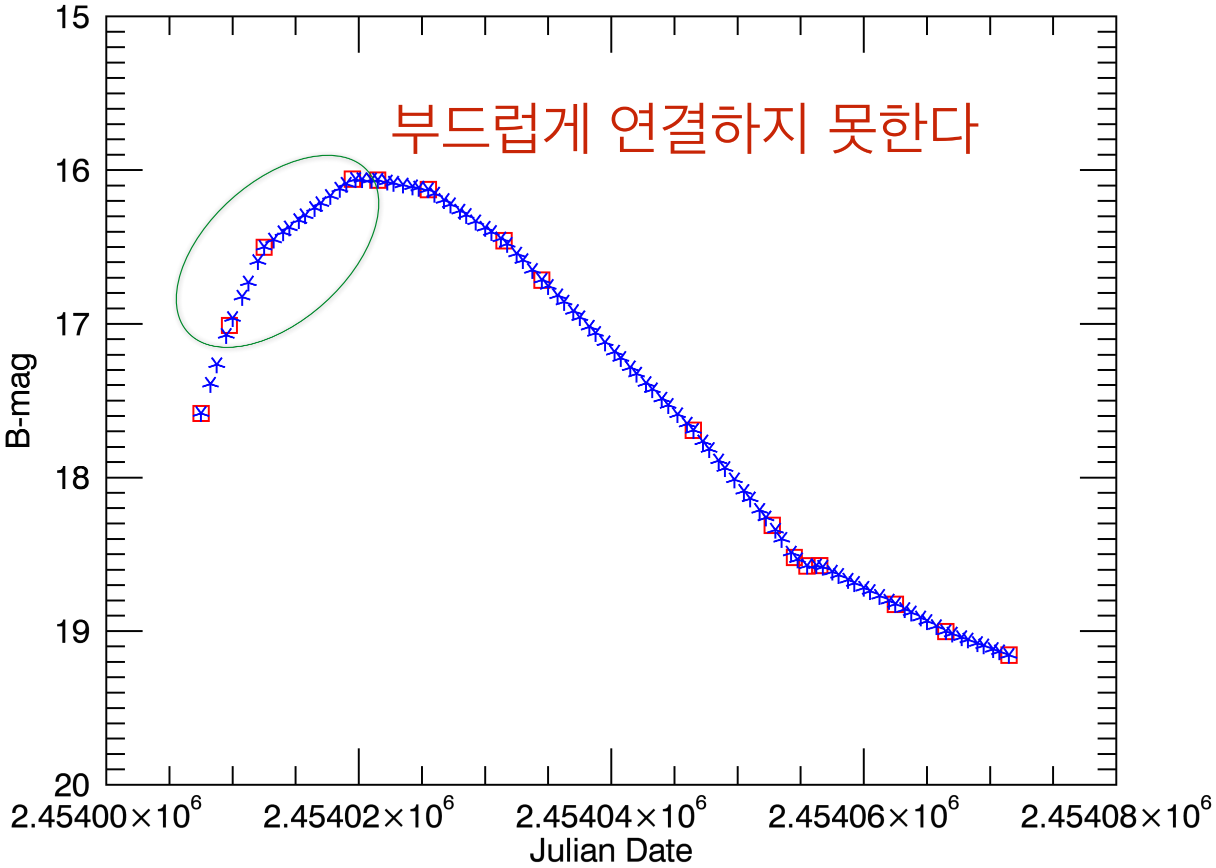
Julian day number 0 assigned to the day starting at noon on Monday, January 1, 4713 BC.

예, January 1, 2000, at 18:00:00 UT corresponds to *JD* = 2451545.25


SN2006gz
B-mag
15
16
17
18
19
20
2.45400×10^6
2.45402×10^6
2.45404×10^6
2.45406×10^6
2.45408×10^6
Julian Date

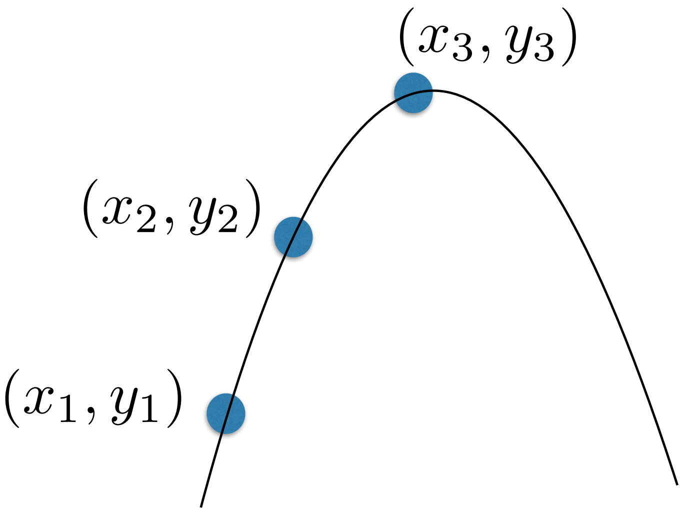

15
16
17
18
19
20
B-mag
2.45400×10^6
2.45402×10^6
2.45404×10^6
2.45406×10^6
2.45408×10^6
Julian Date

Linear Interpolation의 약점



Polynomial Interpolation

기본원리: N개의 점이 주어지면 그 점들을 지나는 N-1차 곡선은 유일하게 결정된다.



$$y = ax^2 + bx + c$$

$$\begin{cases} ax_1^2 + bx_1 + c = y_1 \\ ax_2^2 + bx_2 + c = y_2 \\ ax_3^2 + bx_3 + c = y_3 \end{cases}$$

$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$y = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} y_1 + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} y_2 + \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} y_3$$

Lagrange's Classical Formula

$$P(x) = \frac{(x-x_2)(x-x_3)...(x-x_N)}{(x_1-x_2)(x_1-x_3)...(x_1-x_N)}y_1 + \frac{(x-x_1)(x-x_3)...(x-x_N)}{(x_2-x_1)(x_2-x_3)...(x_2-x_N)}y_2$$
$$+ \cdots + \frac{(x-x_1)(x-x_2)...(x-x_{N-1})}{(x_N-x_1)(x_N-x_2)...(x_N-x_{N-1})}y_N$$

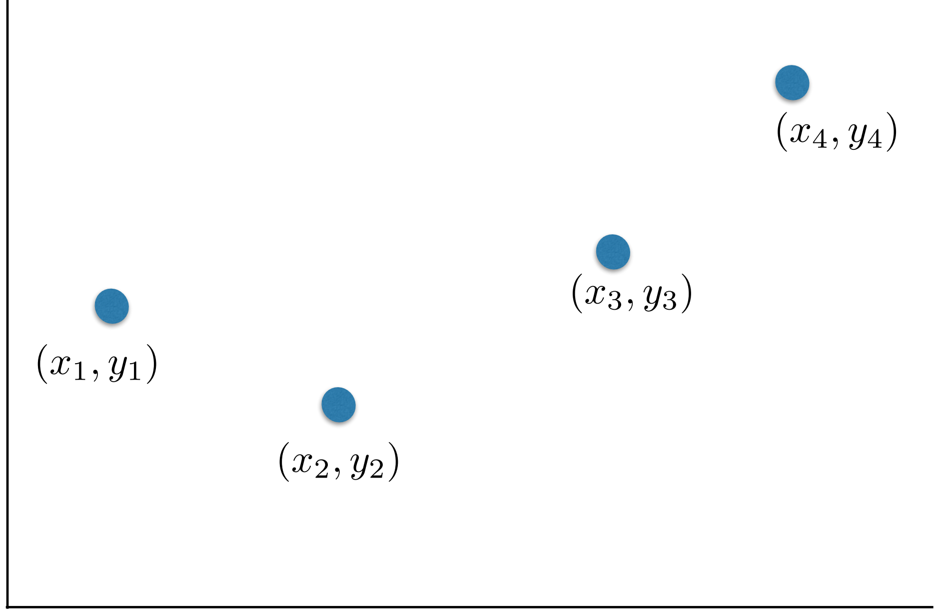
coding이 불편하다 ㅠㅠ

-여러개의 기저함수 합으로 구성됨

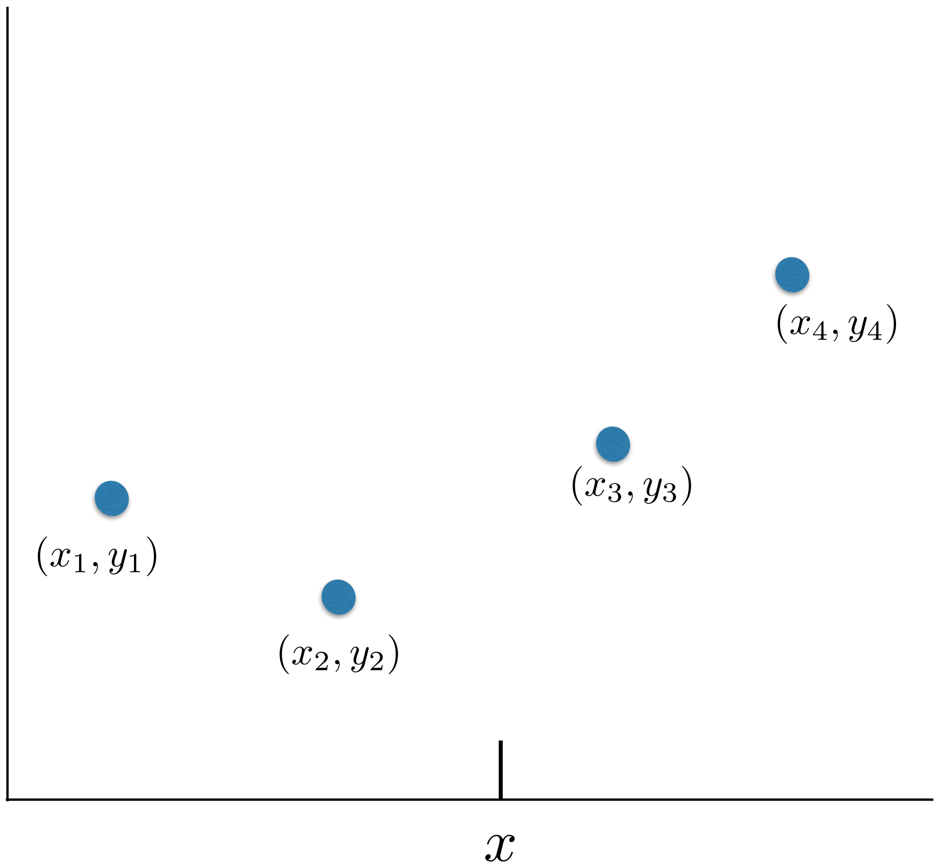
-분모가 작을경우 오차 증폭

Neville's Algorithm

$$P_{i(i+1)\ldots(i+m)} = \frac{(x - x_{i+m})P_{i(i+1)\ldots(i+m-1)} + (x_i - x)P_{(i+1)(i+2)\ldots(i+m)}}{x_i - x_{i+m}}$$



Neville's Algorithm



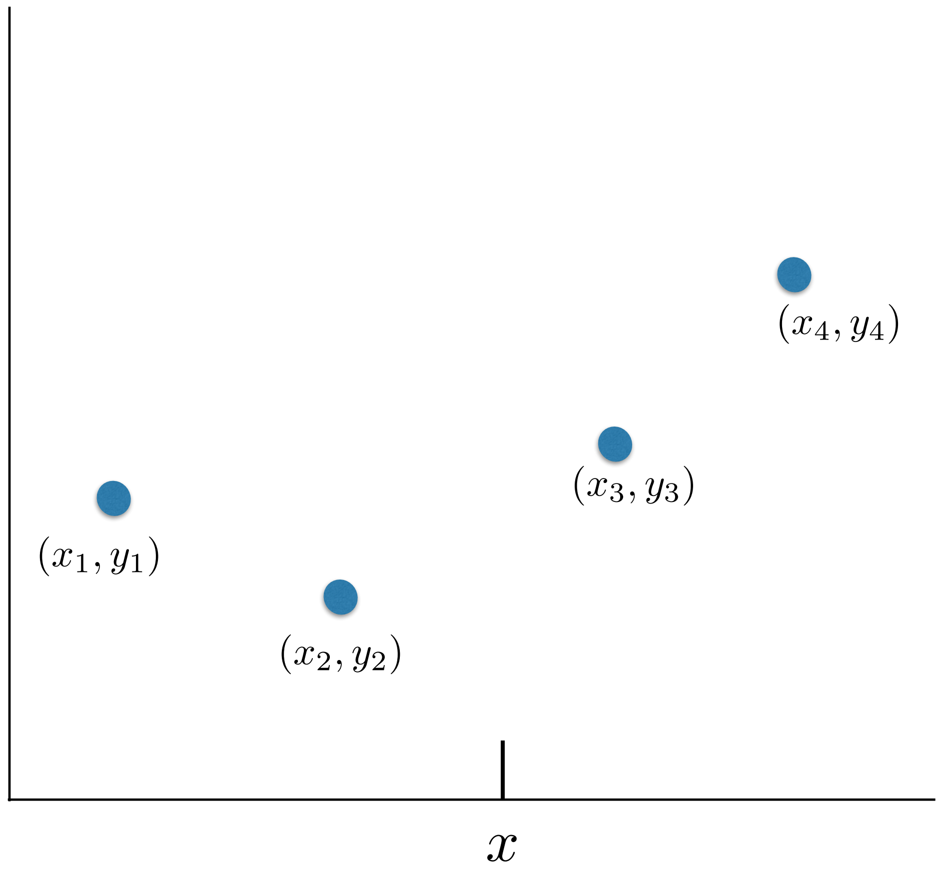
Neville's Algorithm

P_1

P_2

P_3

P_4



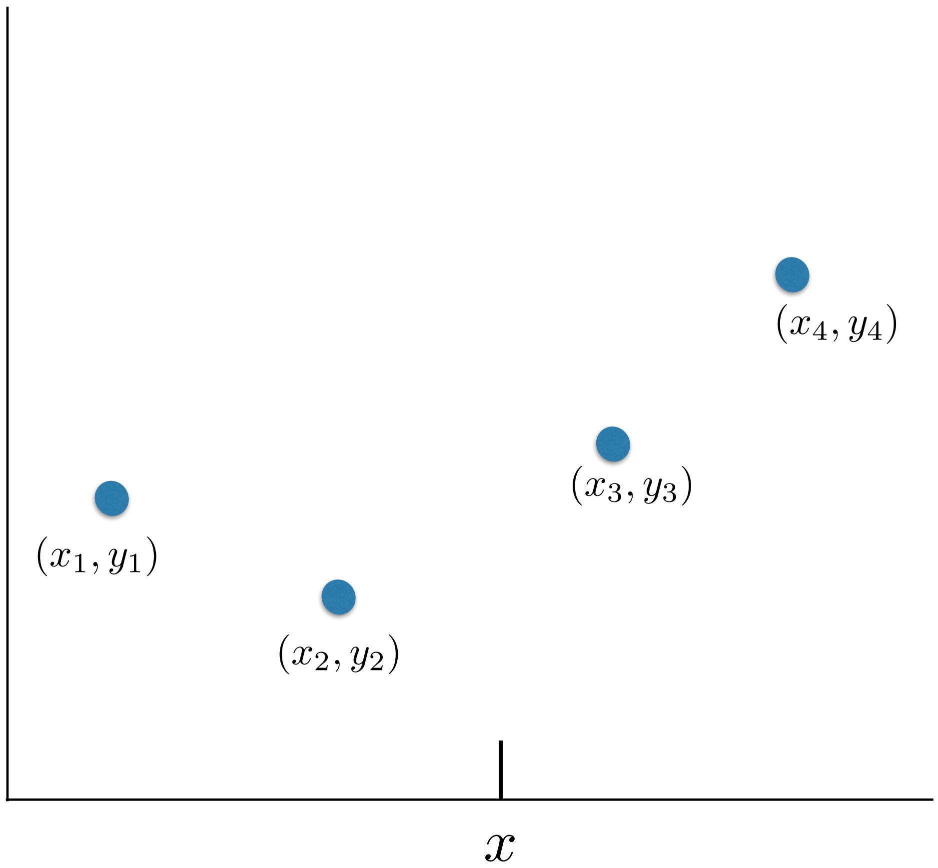
Neville's Algorithm

P_1

P_2

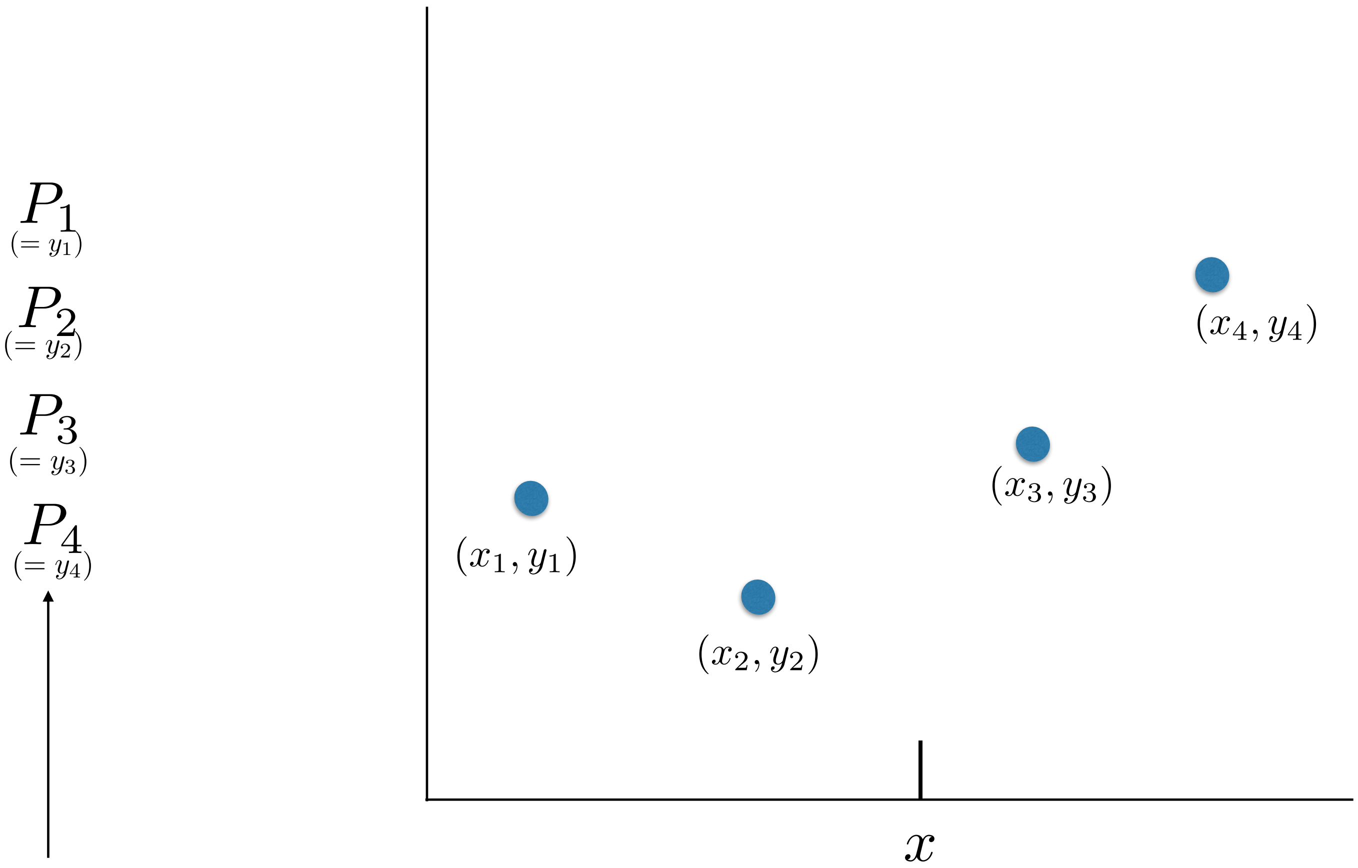
P_3

P_4



1개의 포인트를 이용한 y 값 추정

Neville's Algorithm



1개의 포인트를 이용한 y 값 추정

Neville's Algorithm

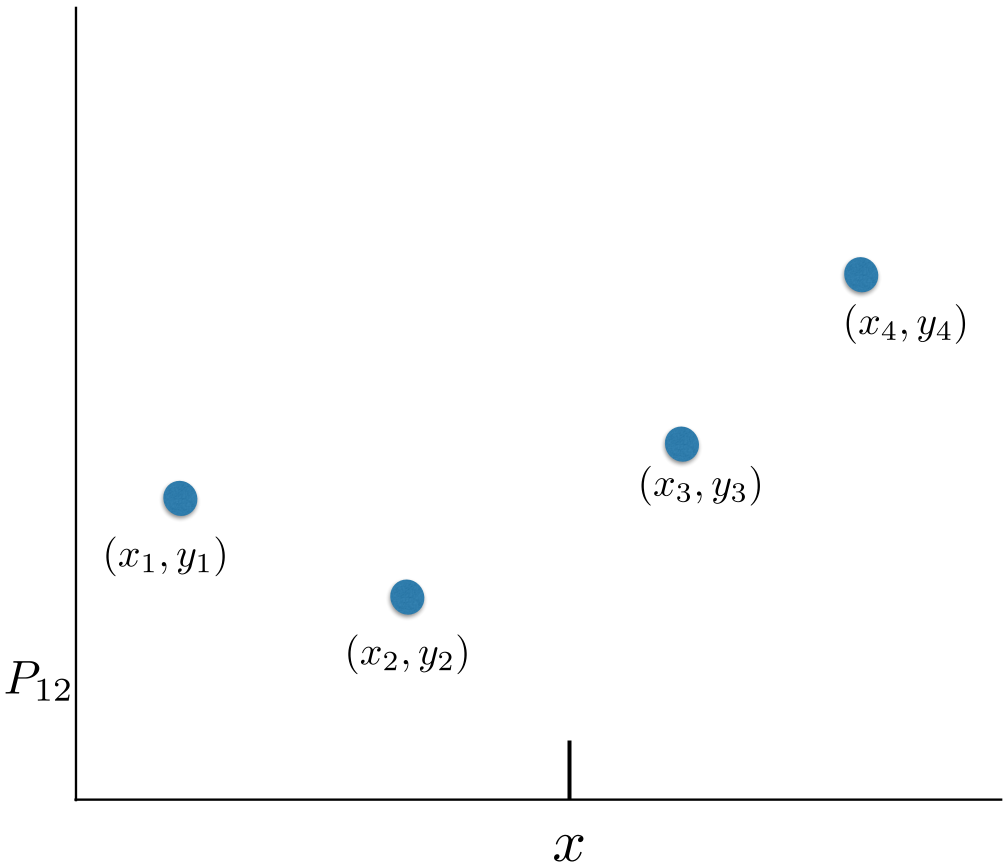
P_1
$(= y_1)$

P_{12}

P_2
$(= y_2)$

P_3
$(= y_3)$

P_4
$(= y_4)$



1개의 포인트를 이용한 y 값 추정

Neville’s Algorithm

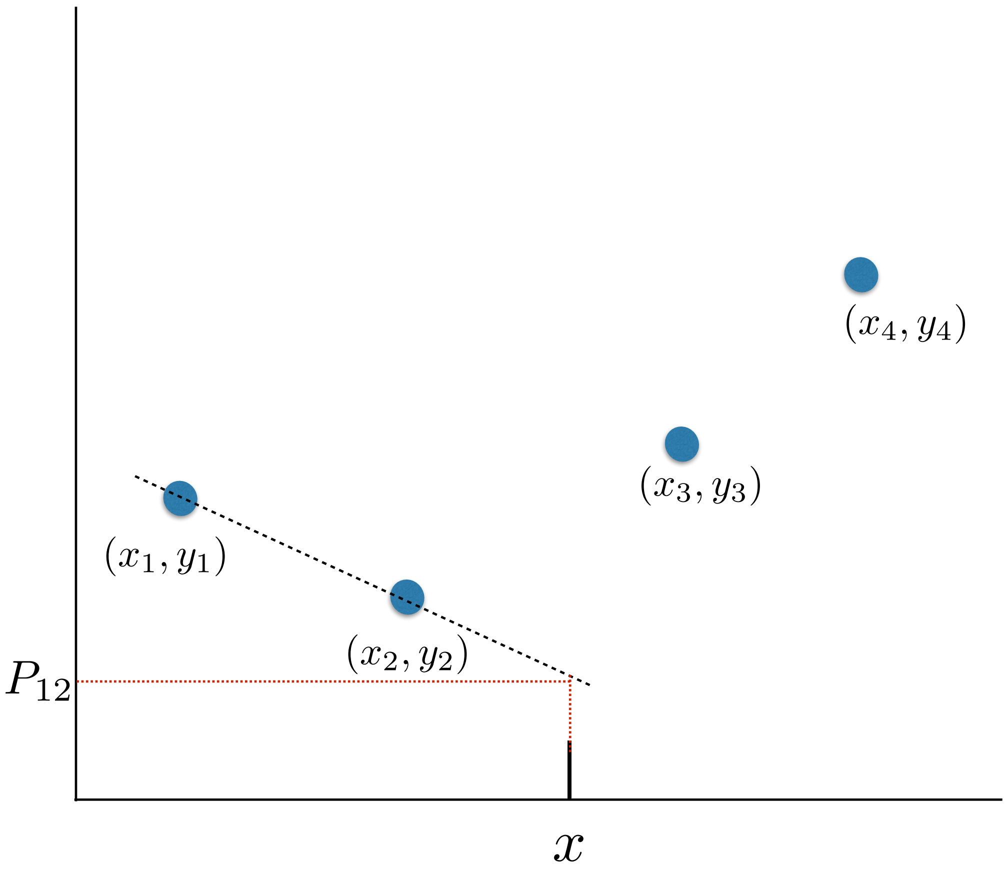
P_1
$(= y_1)$

P_{12}

P_2
$(= y_2)$

P_3
$(= y_3)$

P_4
$(= y_4)$

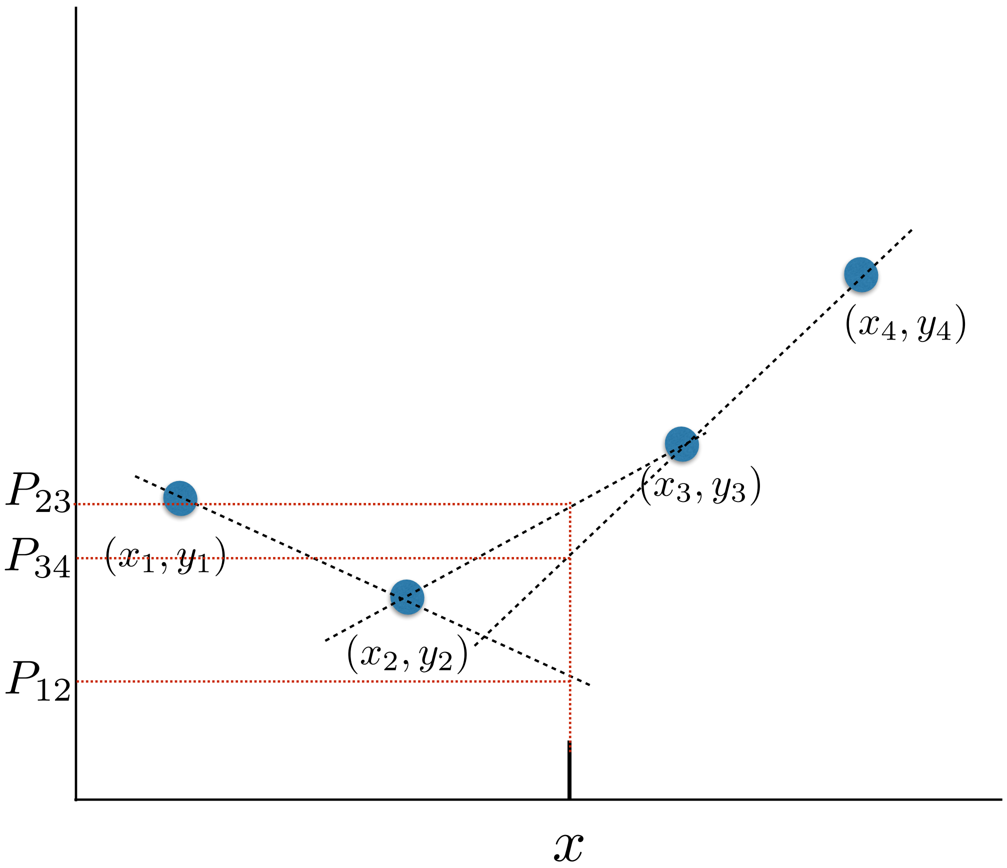


1개의 포인트를 이용한 y 값 추정

Neville's Algorithm

2개의 포인트를
이용한 y 값 추정

$$
\begin{array}{cc}
P_1 \; (= y_1) & \\
 & P_{12} \\
P_2 \; (= y_2) & \\
 & P_{23} \\
P_3 \; (= y_3) & \\
 & P_{34} \\
P_4 \; (= y_4) &
\end{array}
$$



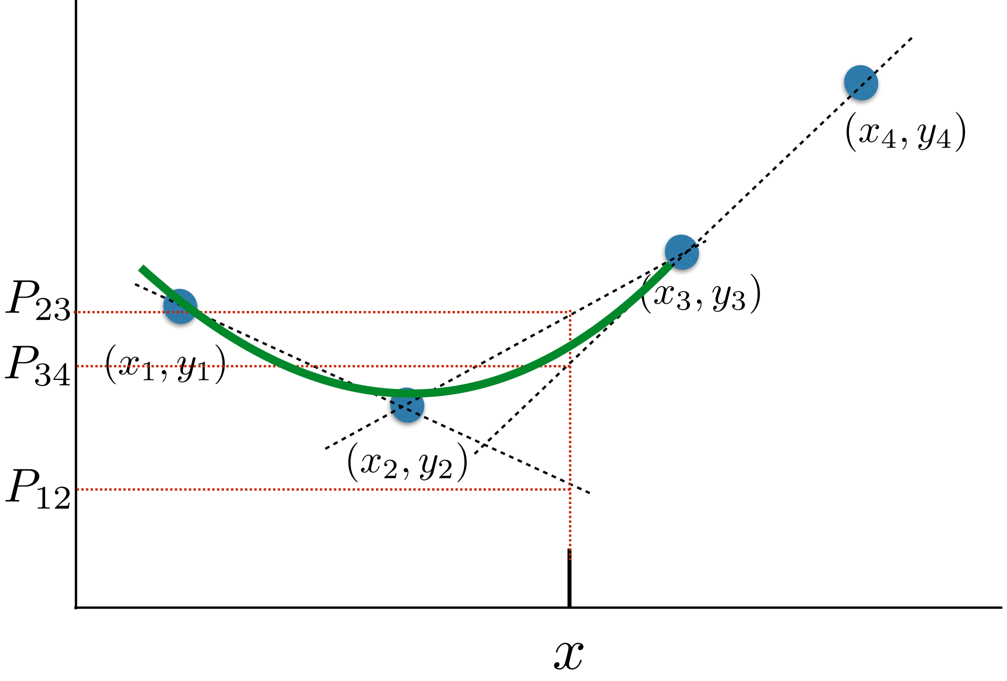
1개의 포인트를 이용한 y 값 추정

2개의 포인트를
이용한 y 값 추정

Neville's Algorithm

$$P_{123} = \frac{(x - x_3)P_{12} + (x_1 - x)P_{23}}{x_1 - x_3}$$

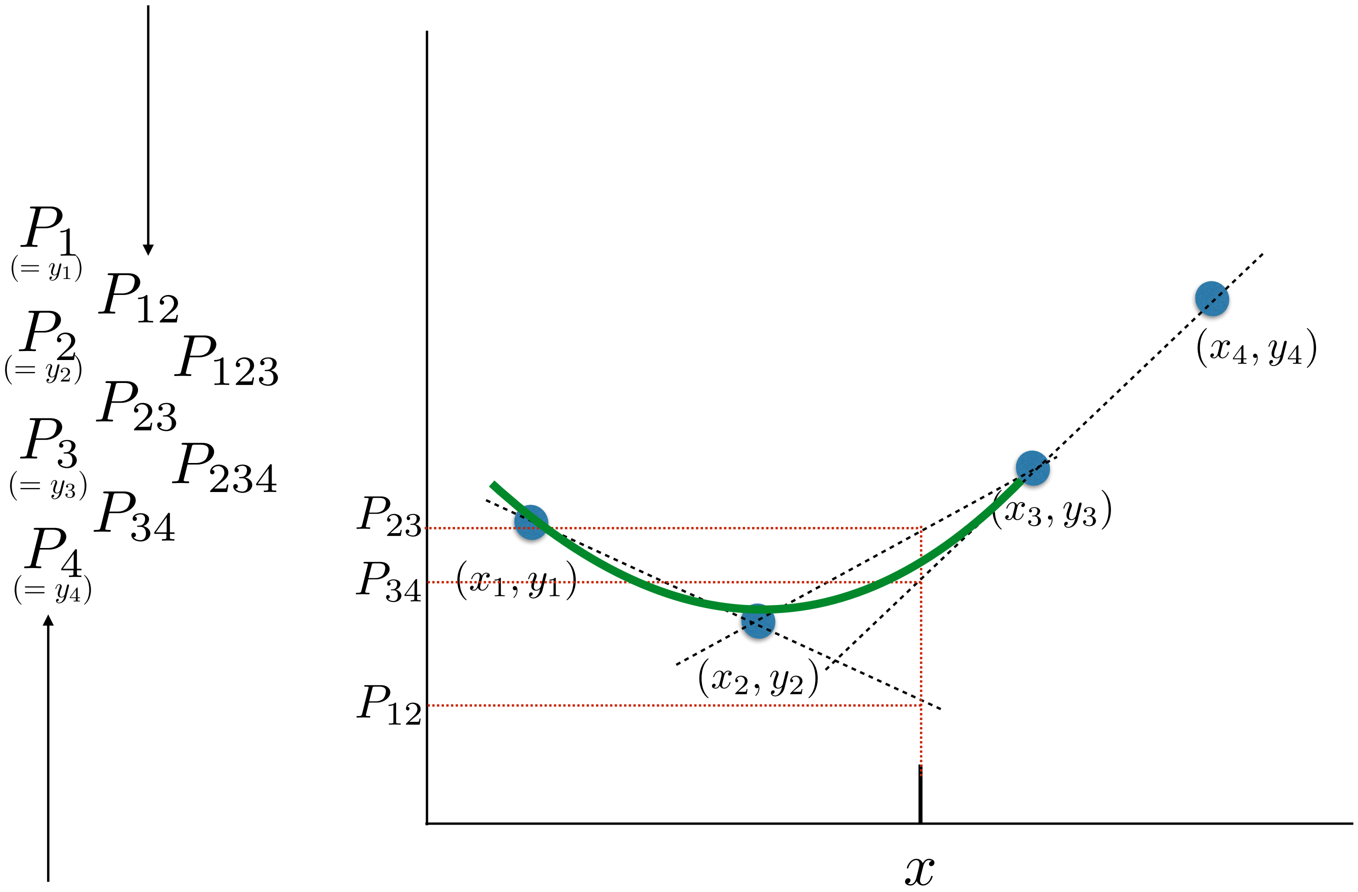
$$\begin{array}{lll}
\underset{(=y_1)}{P_1} & & \\
 & P_{12} & \\
\underset{(=y_2)}{P_2} & & P_{123} \\
 & P_{23} & \\
\underset{(=y_3)}{P_3} & & \\
 & P_{34} & \\
\underset{(=y_4)}{P_4} & &
\end{array}$$



1개의 포인트를 이용한 y 값 추정

Neville's Algorithm

2개의 포인트를
이용한 y 값 추정

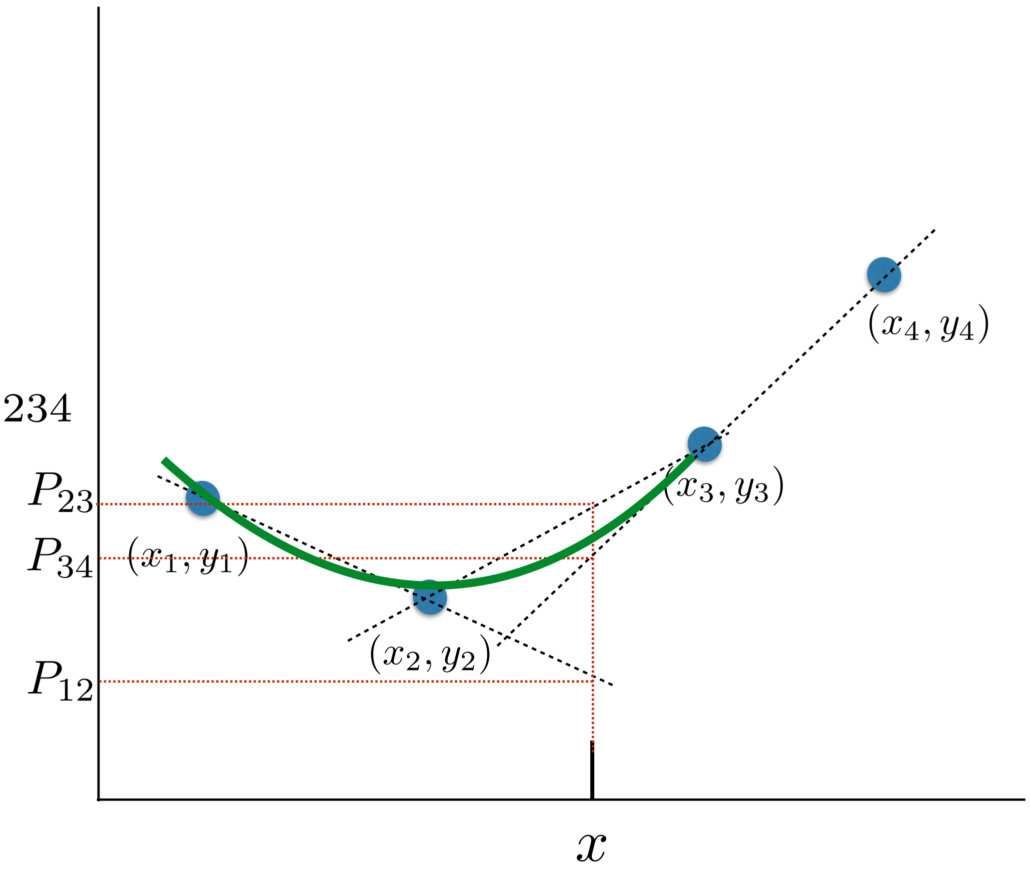


1개의 포인트를 이용한 y 값 추정

Neville's Algorithm

2개의 포인트를 이용한 y 값 추정

$$
\begin{array}{cccc}
P_1 \ (= y_1) & & & \\
 & P_{12} & & \\
P_2 \ (= y_2) & & P_{123} & \\
 & P_{23} & & P_{1234} \\
P_3 \ (= y_3) & & P_{234} & \\
 & P_{34} & & \\
P_4 \ (= y_4) & & &
\end{array}
$$

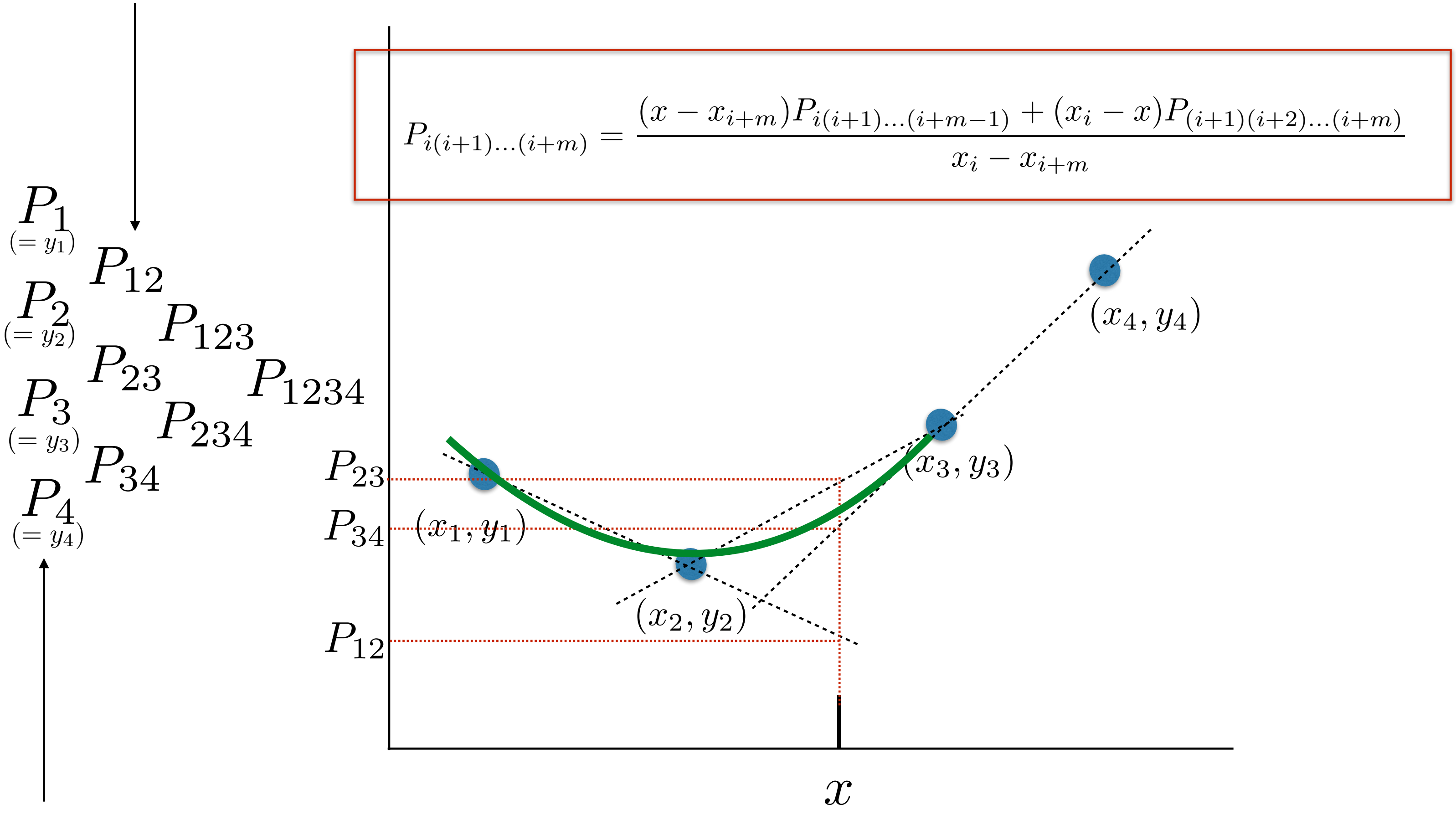


1개의 포인트를 이용한 y 값 추정

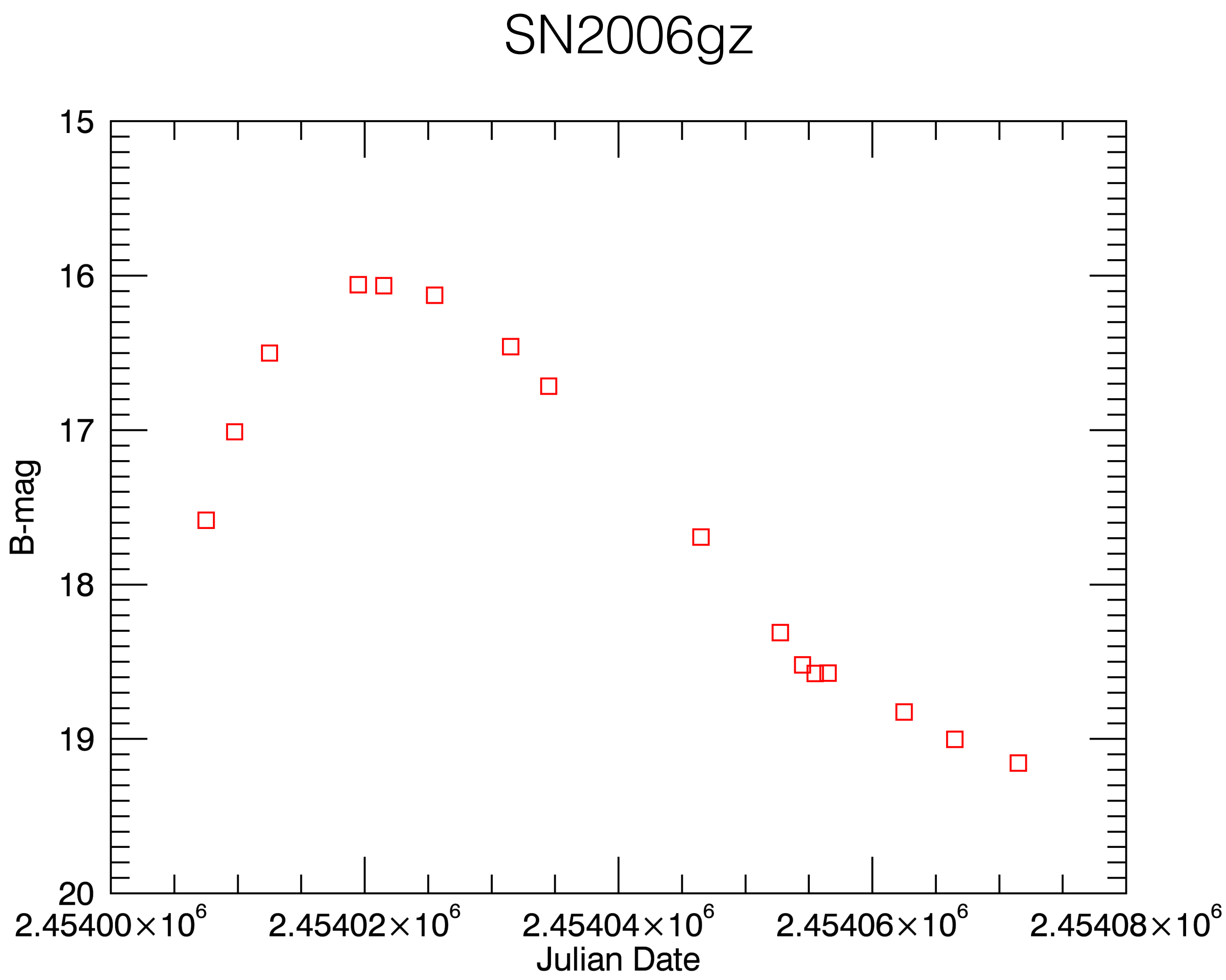
Neville's Algorithm

2개의 포인트를
이용한 y 값 추정

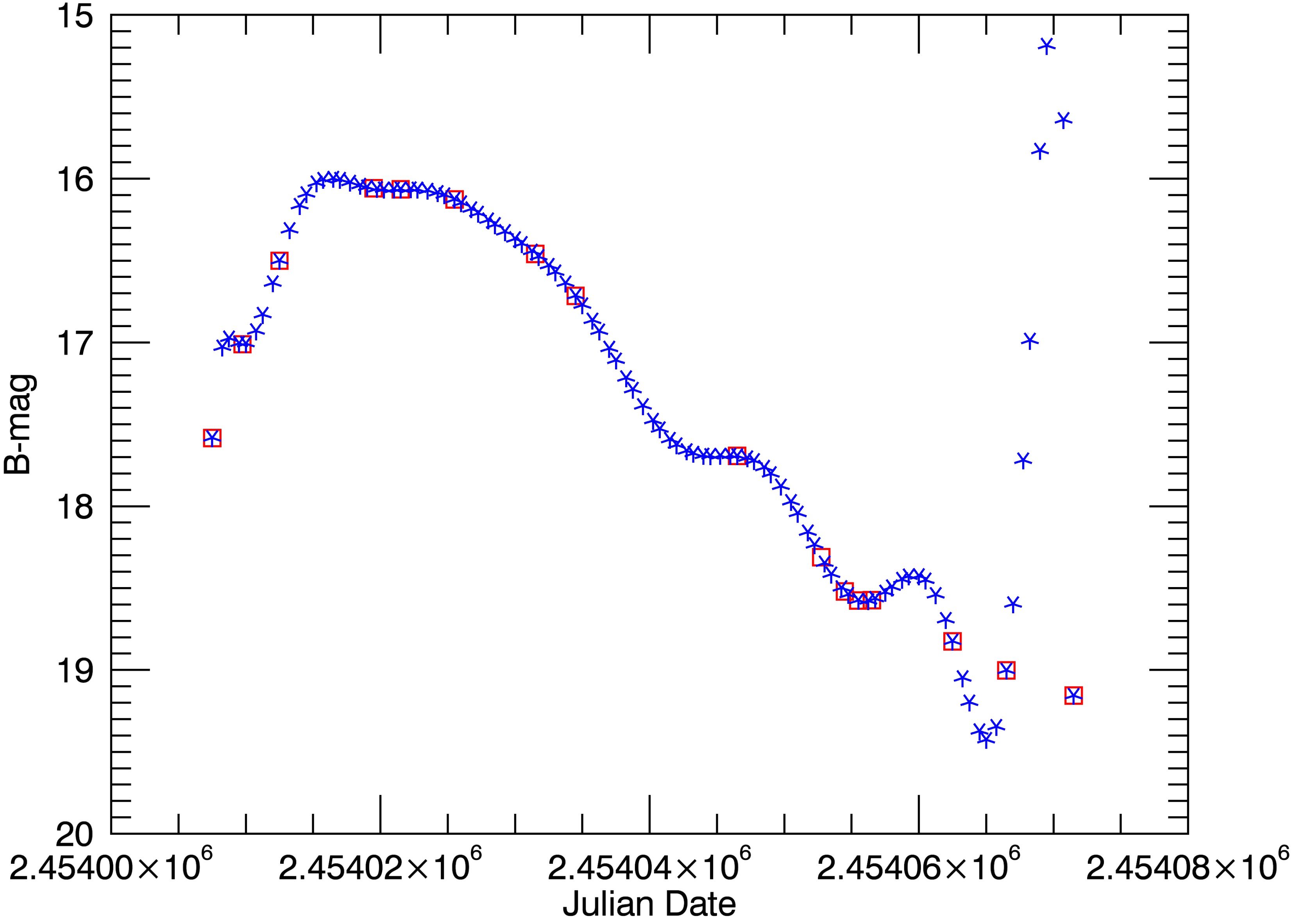
$$P_{i(i+1)\dots(i+m)} = \frac{(x - x_{i+m})P_{i(i+1)\dots(i+m-1)} + (x_i - x)P_{(i+1)(i+2)\dots(i+m)}}{x_i - x_{i+m}}$$



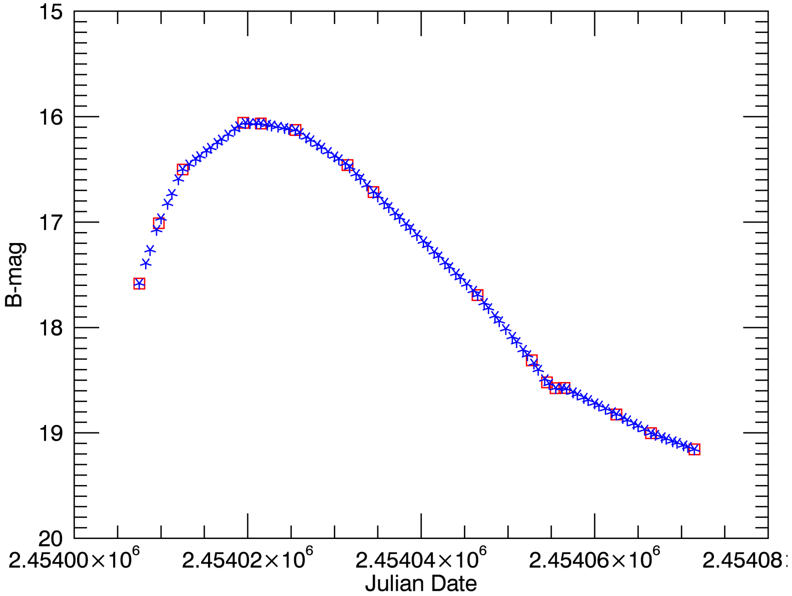
1개의 포인트를 이용한 y 값 추정

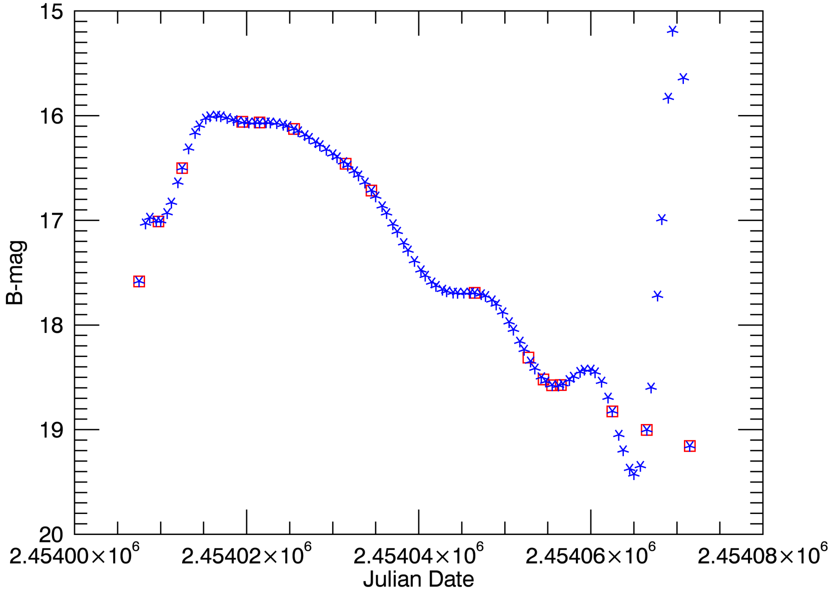


16차 함수를 이용한 내삽.   결과를 기대하시라!



15
16
17
18
19
20
B-mag
2.45400×10^6
2.45402×10^6
2.45404×10^6
2.45406×10^6
2.45408×10^6
Julian Date





다항식이 고차일때는 사용하지 않는것이 좋다.

과제 8

- LearnUs에 올려진 초신성 광도곡선 자료를 활용

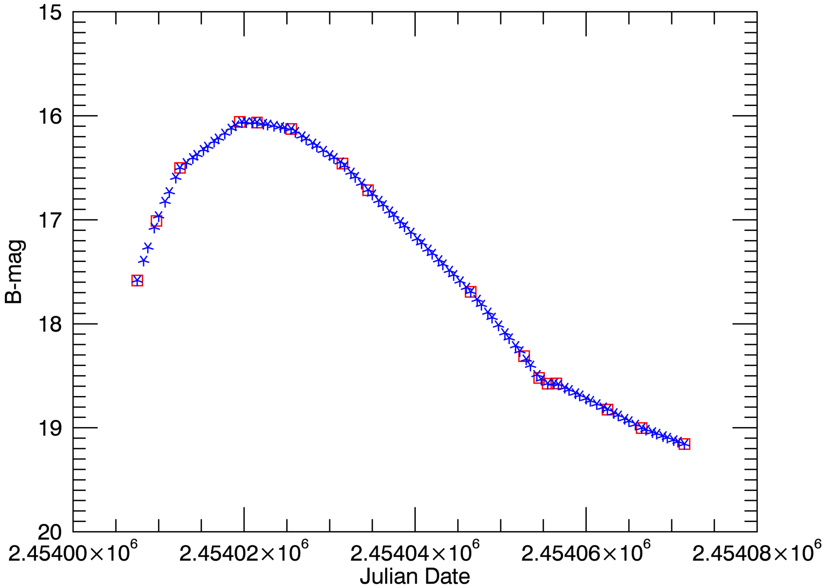
- Polynomial interpolation(Lagrange formula 사용)을 이용하여 초신성 광도곡선의 내삽을 수행

- Neville's algorithm를 사용. 만약 구현이 힘들 경우 이미 존재하는 Library를 불러서 사용가능 (예, *Numerical Recipe*의 **polint** 또는 *Numerical Methods in Engineering with Python*의 **newtonPoly**)(직접 구현할 경우 10% 가산점)
  - 보간 차수를 다양하게 설정 (3점, 5점, 7개, 16개 모든 점)

- 결과를 그래프로 중첩 표시

- 선택과제 (가산점 20%): 관측값에 가우시안 노이즈 sigma=0.1 mag 를 추가한 후 같은 방식으로 보간을 수행하고 **노이즈에 대한 민감도를 분석**하라.

Cubic Spline Interpolation



Analyze the weakness.

What are the first derivatives between data points?

Constant

What are the first derivatives on the data points?

not defined

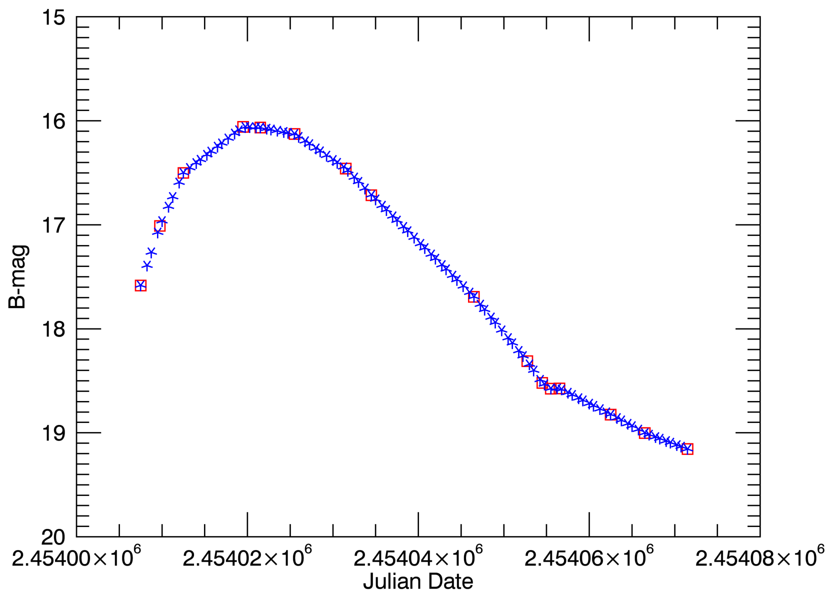
What are the second derivatives between data points?

zero

What are the second derivatives on the data points?

not defined

Cubic Spline Interpolation



Analyze the weakness.

What are the first derivatives between data points?

Constant -> Smoothly Varying

What are the first derivatives on the data points?

not defined -> Defined and Continuous

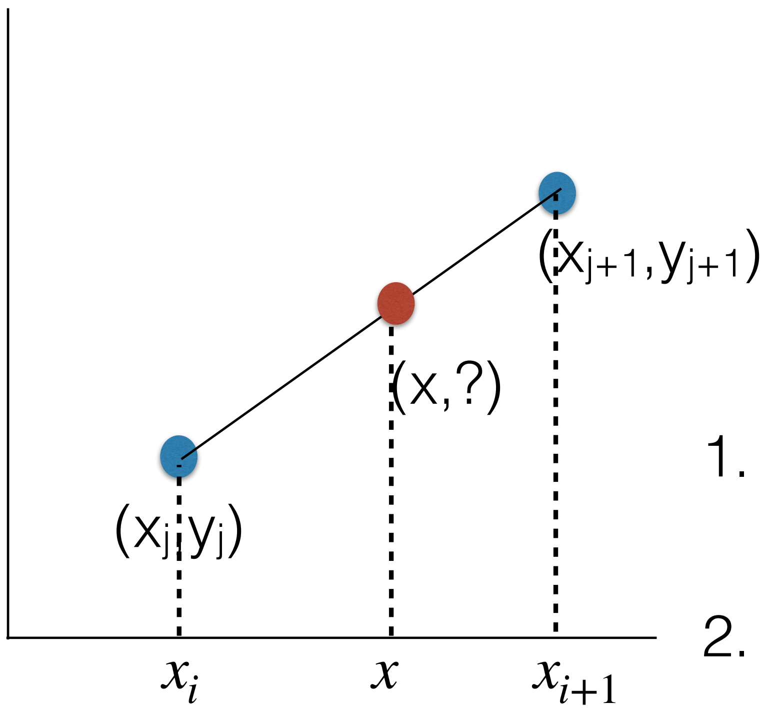
What are the second derivatives between data points?

zero -> Non-zero (constant)

What are the second derivatives on the data points?

not defined -> Defined and Continuous

MODIFYING LINEAR INTERPOLATION



$$y = y_j + \frac{y_{j+1} - y_j}{x_{j+1} - x_j}(x - x_j)$$

$$y = Ay_i + By_{j+1}$$

$$A = \frac{x_{j+1} - x}{x_{j+1} - x_j} \qquad B = \frac{x - x_j}{x_{j+1} - x_j}$$

Let $y = Ay_j + By_{j+1} + Cy_j'' + Dy_{j+1}''$

C and *D* are the second order polynomials to be determined.

1. At $x = x_j$, $y'' = y_j''$ & $y = y_j$

   At $x = x_{j+1}$, $y'' = y_{j+1}''$ & $y = y_{j+1}$

2. y'' varies linearly from y_j'' to y_{j+1}''

$$C \equiv \frac{1}{6}(A^3 - A)(x_{j+1} - x_j)^2 \qquad D \equiv \frac{1}{6}(B^3 - B)(x_{j+1} - x_j)^2$$

Let's Verify it.

$$y = Ay_j + By_{j+1} + Cy''_j + Dy''_{j+1}$$

$$A = \frac{x_{j+1} - x}{x_{j+1} - x_j} \quad B = \frac{x - x_j}{x_{j+1} - x_j} \quad C \equiv \frac{1}{6}(A^3 - A)(x_{j+1} - x_j)^2 \quad D \equiv \frac{1}{6}(B^3 - B)(x_{j+1} - x_j)^2$$

Let's take the first derivative.

$$\frac{dy}{dx} = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{3A^2 - 1}{6}(x_{j+1} - x_j)y''_j + \frac{3B^2 - 1}{6}(x_{j+1} - x_j)y''_{j+1}$$

Let's take the second derivative.

$$\frac{d^2y}{dx^2} = Ay''_j + By''_{j+1}$$

At $x = x_j$, $y'' = y''_j$ & $y = y_j$

At $x = x_{j+1}$, $y'' = y''_{j+1}$ & $y = y_{j+1}$

y'' varies linearly from y''_j to y''_{j+1}

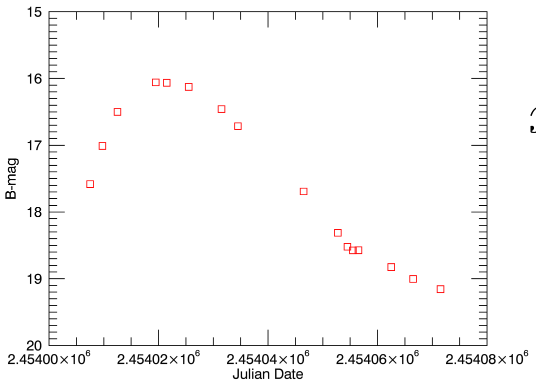
Are you ready?

We don't know what y''_j and y''_{j+1} are.

Cubic Spline Interpolation

The first derivative must be continuous.

$$\frac{dy}{dx} = \frac{y_{j+1}-y_j}{x_{j+1}-x_j} - \frac{3A^2-1}{6}(x_{j+1}-x_j)y''_j + \frac{3B^2-1}{6}(x_{j+1}-x_j)y''_{j+1}$$



(x_{j-1}, x_j) 구간 $\frac{dy}{dx}$ 와 (x_j, x_{j+1}) 구간 $\frac{dy}{dx}$ 가

$x = x_j$ 에서 일치

$$\frac{x_j - x_{j-1}}{6} y''_{j-1} + \frac{x_{j+1}-x_{j-1}}{3} y''_j + \frac{x_{j+1}-x_j}{6} y''_{j+1} = \frac{y_{j+1}-y_j}{x_{j+1}-x_j} - \frac{y_j - y_{j-1}}{x_j - x_{j-1}}$$

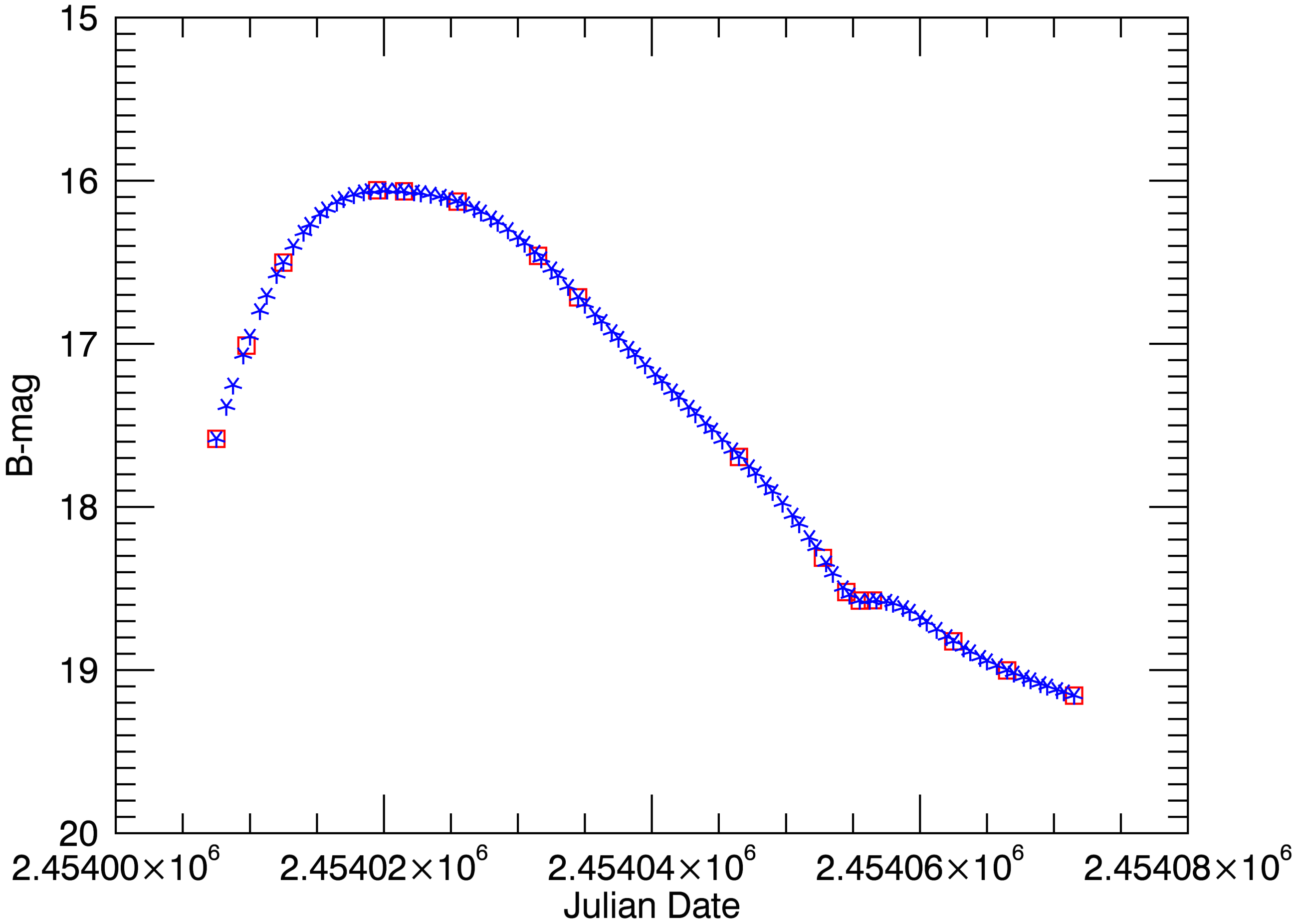
$$\text{for } j = 2, \ldots, N-1$$

How many equations? $N-2$

How many unknown y''_j's? N

Set $y''_1 = y''_N = 0.$ (natural spline)

Spline Interpolation 결과



1. Is the first derivative smooth?

2. Is the second derivative continuous?