CPE 332
Computer Engineering Mathematics II

Chapter 12

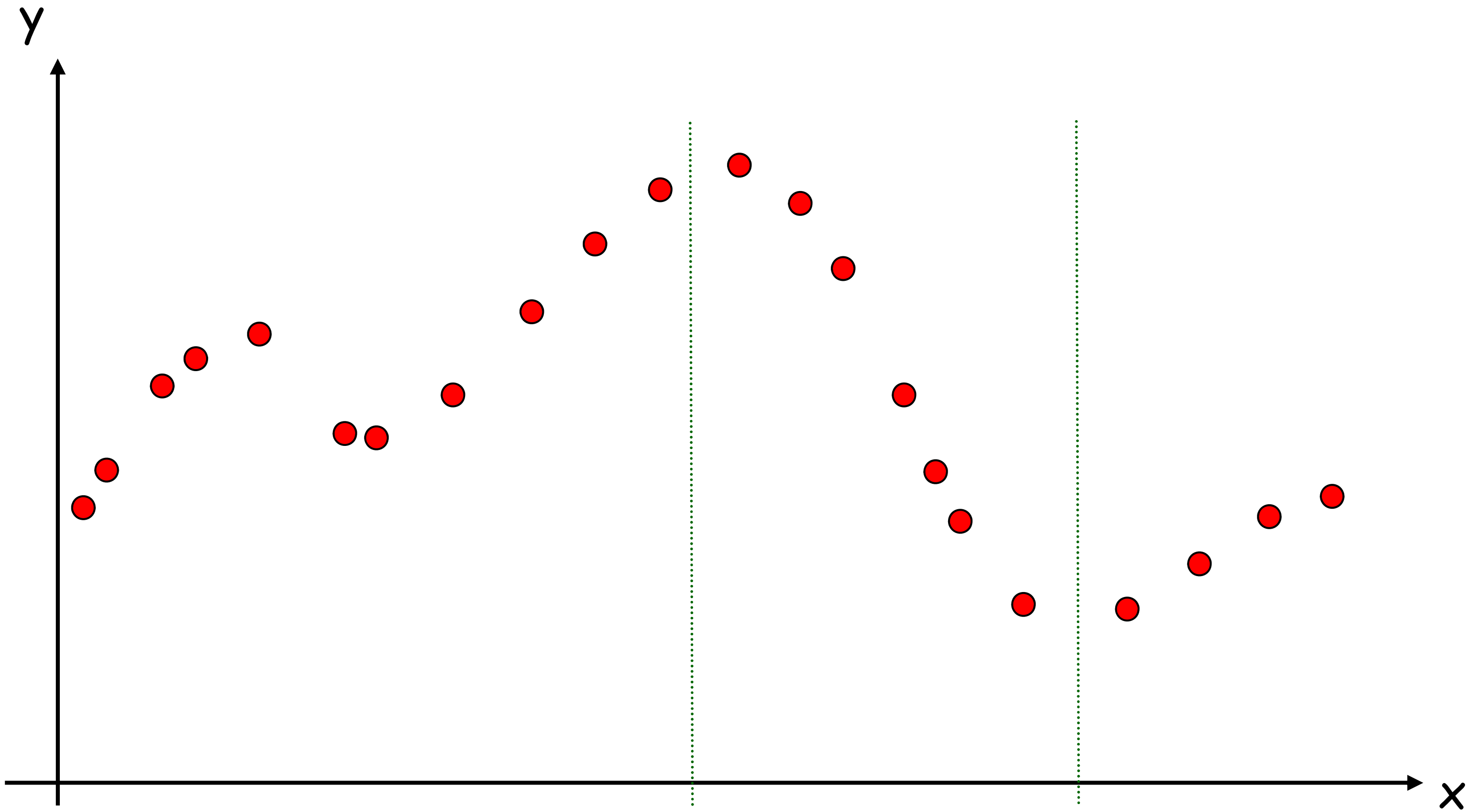
Curve Fitting

(Last)

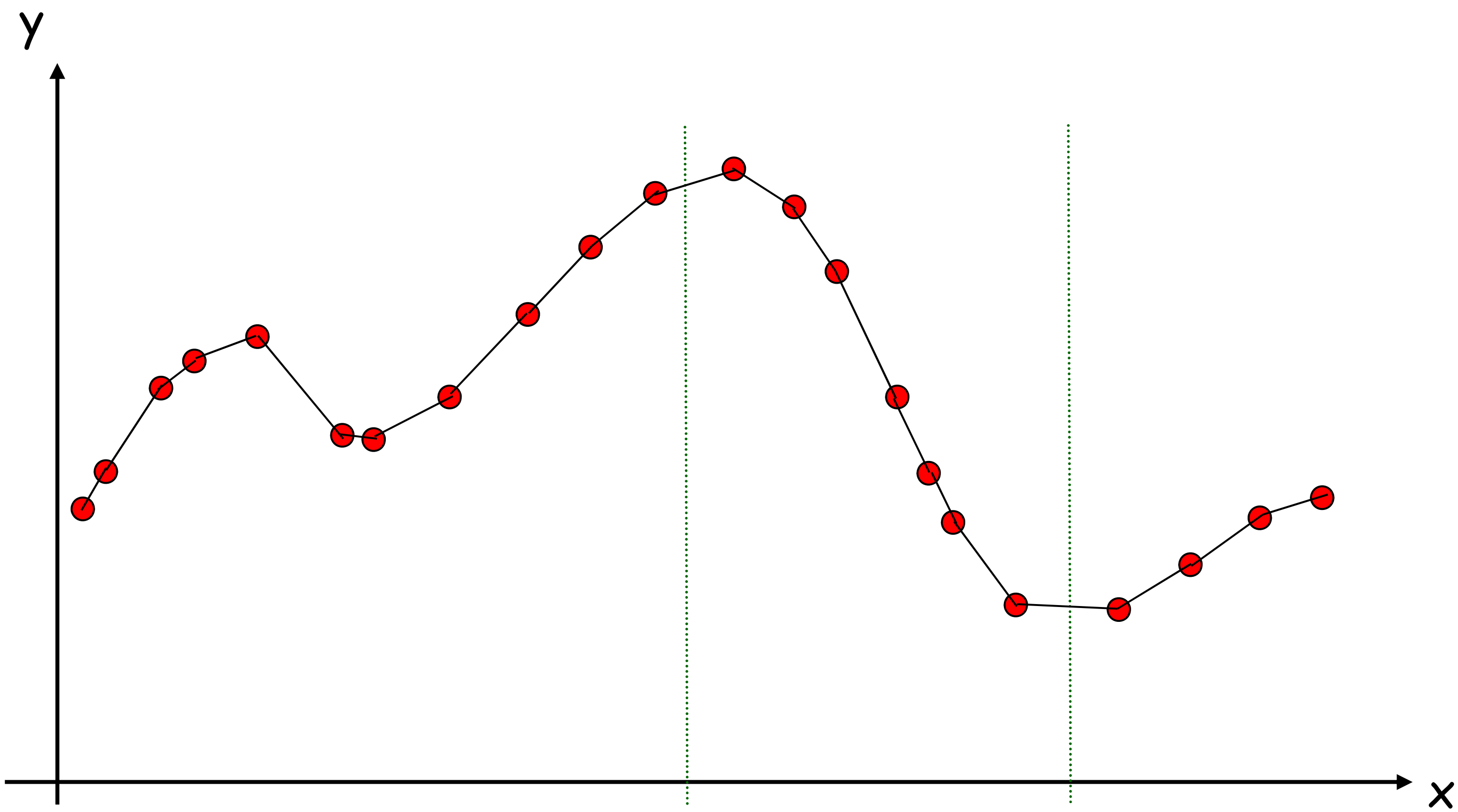
Curve Fitting

- บ่อยๆครั้ง เราได้ข้อมูลจากการวัดในสนาม ซึ่งมีลักษณะเป็นจุด และเราจะต้องการหา Function ทางคณิตศาสตร์เพื่อเชื่อมต่อจุดเหล่านี้
- Function ง่ายๆที่ใช้กันคือ Polynomial ที่ Degree ต่างๆ
- Function ที่สลับซับซ้อนขึ้นได้แก่ Cosine Function
- กรรมวิธีเหล่านี้ เพื่อที่จะหาค่าของ Function ที่อยู่ระหว่างจุดที่เราวัด ซึ่งเราเรียก Interpolation

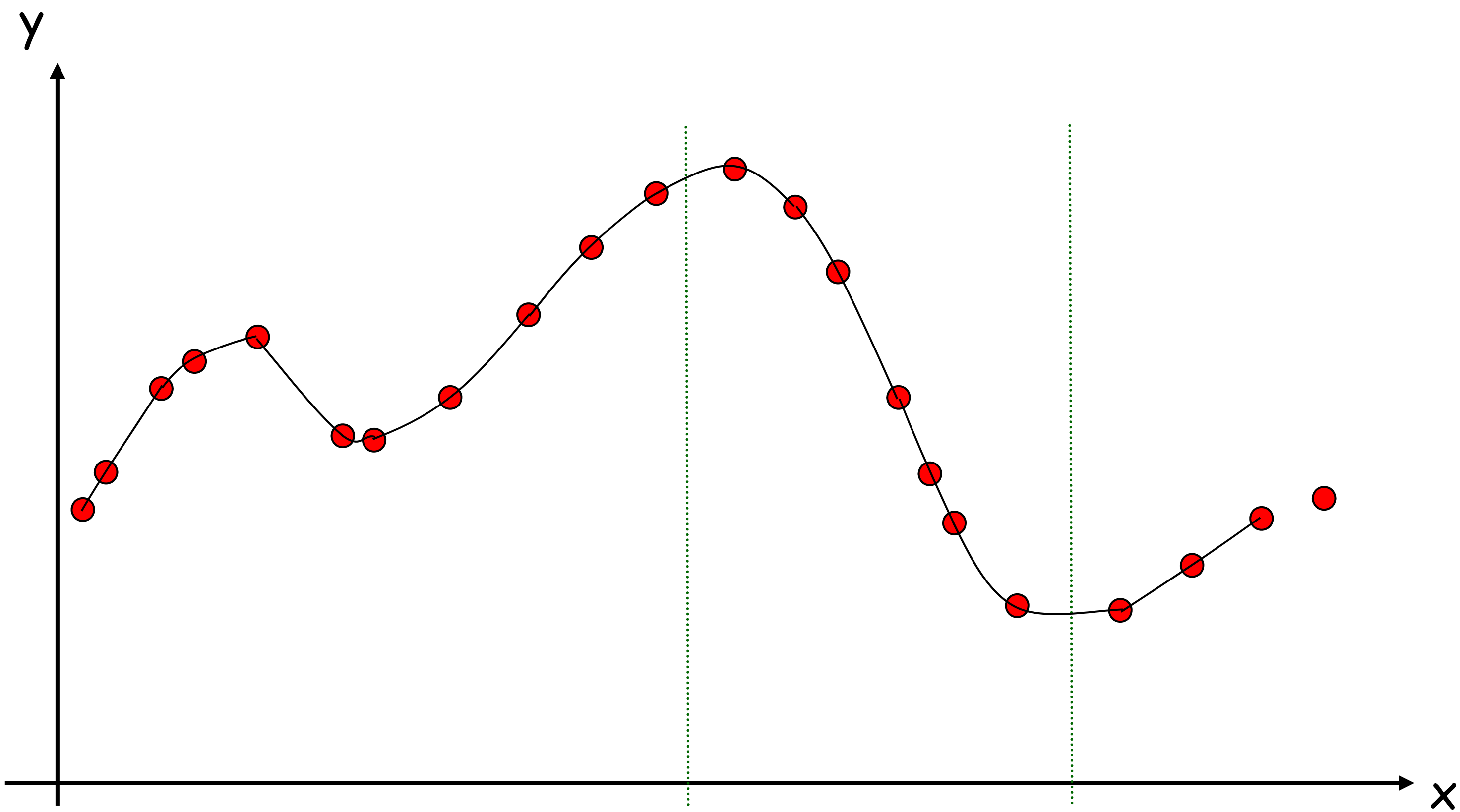
Interpolation



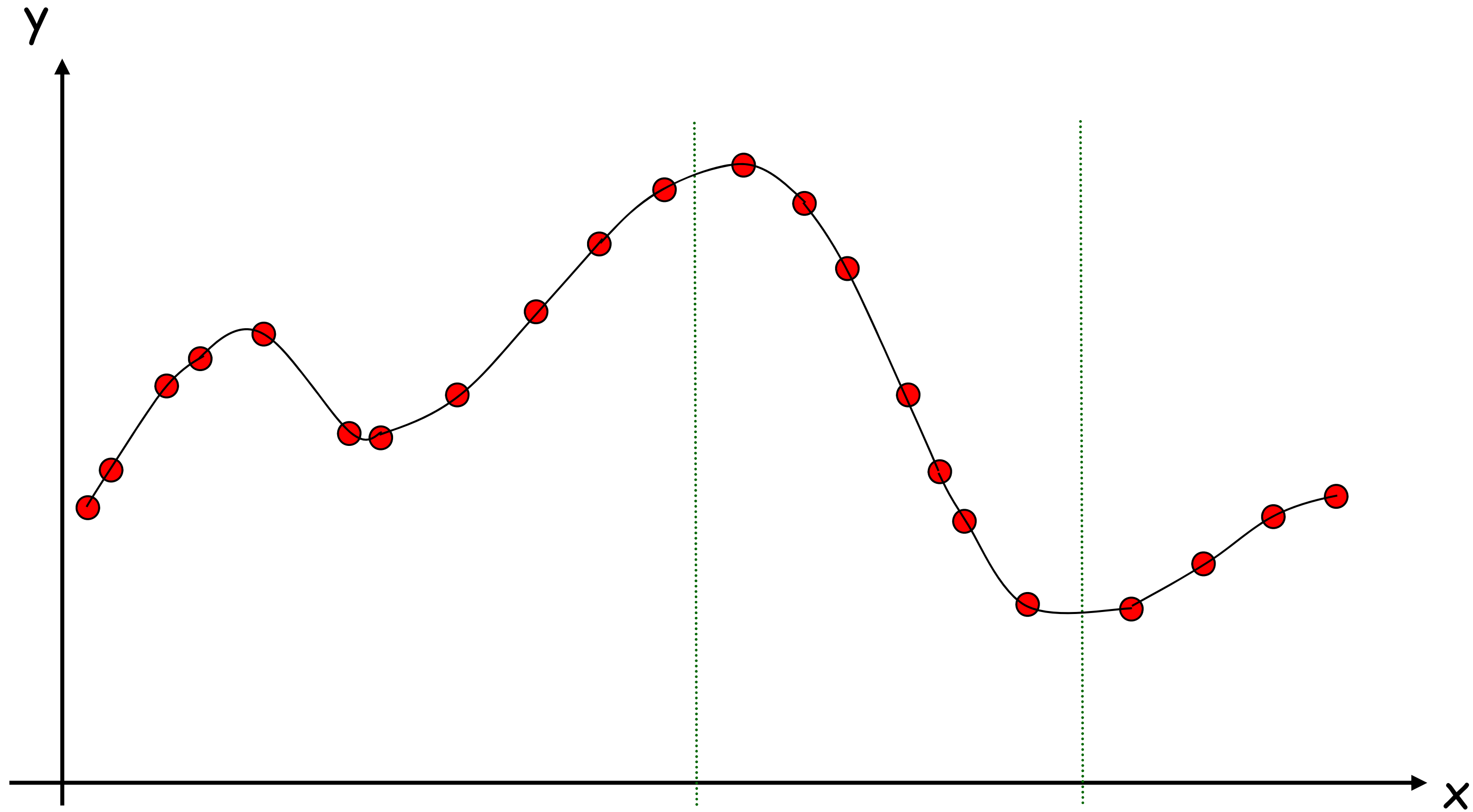
Linear Interpolation



Second Degree Piecewise Interpolation



Third Degree Piecewise Interpolation



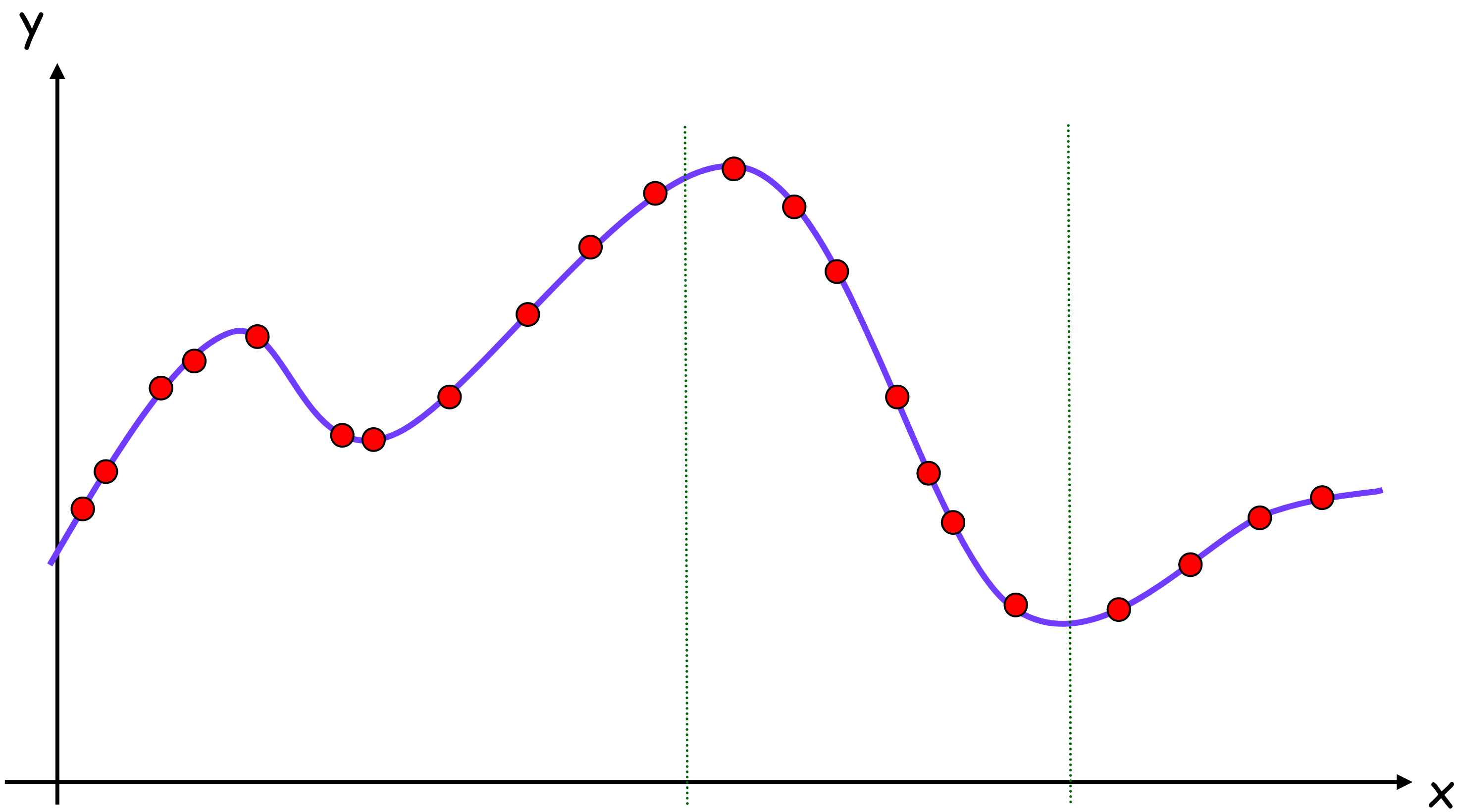
Higher Order Degree

- ไม่นิยมใช้ เพราะจะเกิดการ Oscillate ได้
- Function ที่รอยต่อของ Polynomial จะไม่ต่อเนื่อง
- N Degree จะใช้ N+1 จุดสำหรับหาค่าของ Function

Spline Interpolation

- เป็นการใช้ Polynomial Degree ต่ำๆ ที่นิยมคือ Third Degree หรือ Cube
  - เรียก Cubic Spline
- Cubic Spline จะหา Polynomial จากทีละ 2 จุด(ไม่ใช่ 4) แต่บังคับให้รอยต่อระหว่างแต่ละ Polynomial มีความต่อเนื่อง โดยกำหนดให้ Polynomial ที่ต่อกันมี First Derivative และ Second Derivative เท่ากัน
- 2 จุดได้ 2 สมการ บวกกับอีกสองสมการของ First และ Second Derivative เป็น 4 สมการ 4 Unknown
- ที่ปลายและหัวจะสมมุติ First Derivative จากสมการเส้นตรง และสมมุติ Second Derivative เท่ากับศูนย์ วิธีนี้เราเรียก Natural Spline
- เป็นวิธีที่นิยมมากกว่า

Spline Interpolation



การคำนวณ Cubic Spline Interpolation (1)

- พิจารณาจากจุดของตัวอย่าง $(x_i, y_i); i = 0, \ldots, n-1$
- เราต้องการลาก Curve ผ่านจุดเหล่านี้ โดยใช้ Third Degree Polynomial

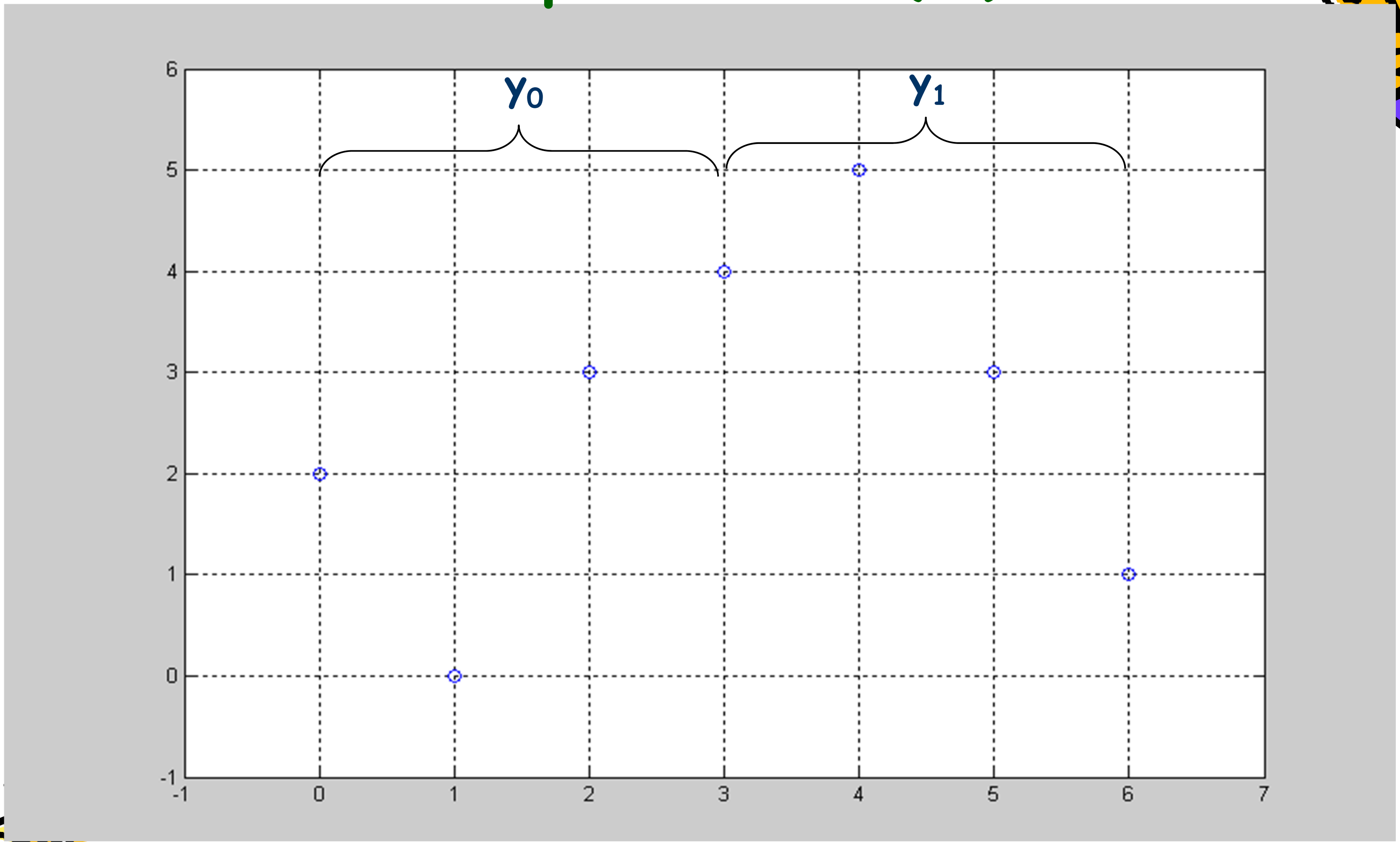
$$y = f_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

- มี 4 Unknown ที่ต้องหา ดังนั้นต้องสร้าง 4 สมการ และใช้ 4 จุดของ Data ในการแก้
- ผลคือ 3rd Degree Polynomial จะลากผ่านทีละ 4 จุด ดังนั้นจะมีรอยต่อทุกๆ 4 จุด ที่ไม่ราบเรียบ เนื่องจากแต่ละ 4 จุดจะใช้ Polynomial คนละตัวกัน
- Polynomial ที่ต่อกันจะใช้จุดร่วมกันที่รอยต่อ

การคำนวณ Cubic Spline Interpolation (2)

- Example: กำหนด Data 7 จุดดังนี้
  - Y_i = [2 0 3 4 5 3 1]; X_i = [0 1 2 3 4 5 6]
- จงคำนวณ Third Degree Polynomial ที่จะ Fit Data เหล่านี้

การคำนวณ Cubic Spline Interpolation (3)



เราต้องใช้ Cubic Polynomial 2 ตัวแรก Fit 4 จุดแรก ตัวที่ 2 Fit 4 จุดถัดไป โดยจุดที่ 4 เป็นจุดร่วม

Cubic Spline Int.(4)

- Polynomial ตัวแรก คำนวณได้ดังนี้

$$2 = a_0$$

$$0 = a_0 + a_1 + a_2 + a_3$$

$$3 = a_0 + 2a_1 + 4a_2 + 8a_3$$

$$4 = a_0 + 3a_1 + 9a_2 + 27a_3$$

- แก้สมการได้

$$y_0 = 2 - 6.8333x + 6x^2 - 1.1667x^3$$

Cubic Spline Int.(5)

- Polynomial ตัวที่สอง คำนวณได้ดังนี้

$$4 = a_0 + 3a_1 + 9a_2 + 27a_3$$

$$5 = a_0 + 4a_1 + 16a_2 + 64a_3$$

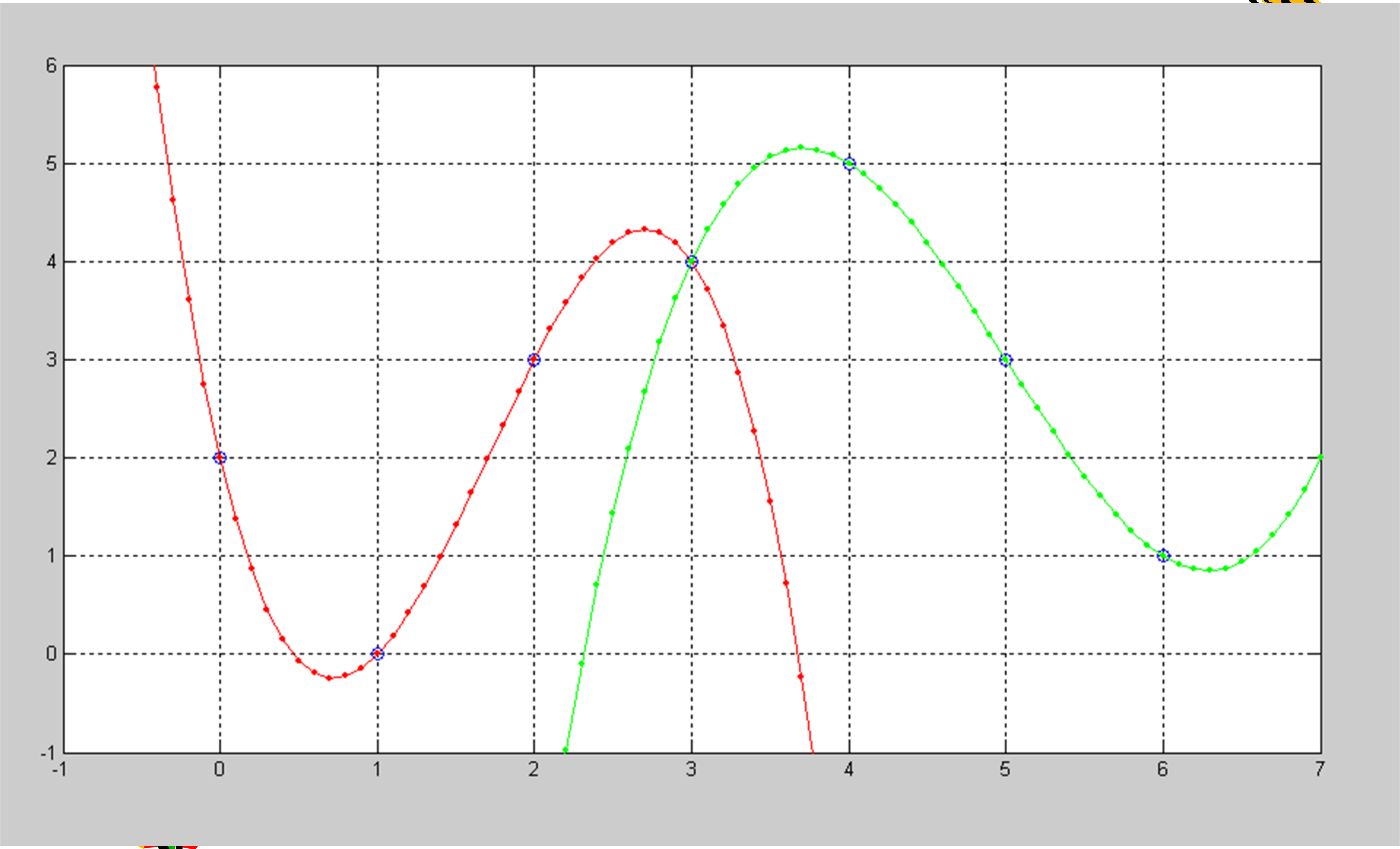
$$3 = a_0 + 5a_1 + 25a_2 + 125a_3$$

$$1 = a_0 + 6a_1 + 36a_2 + 216a_3$$

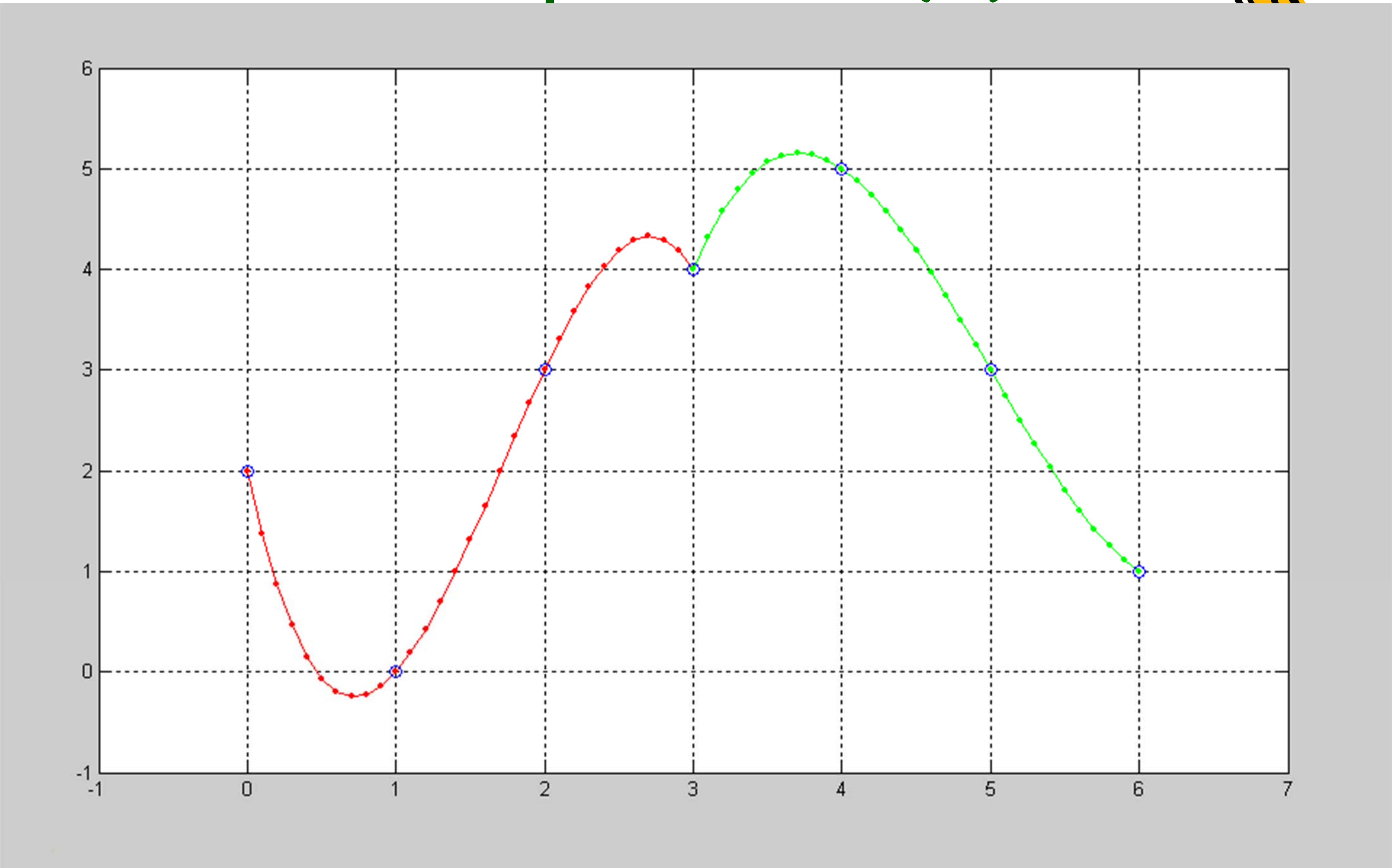
- แก้สมการได้

$$y_1 = -47 + 35x - 7.5x^2 + 0.5x^3$$

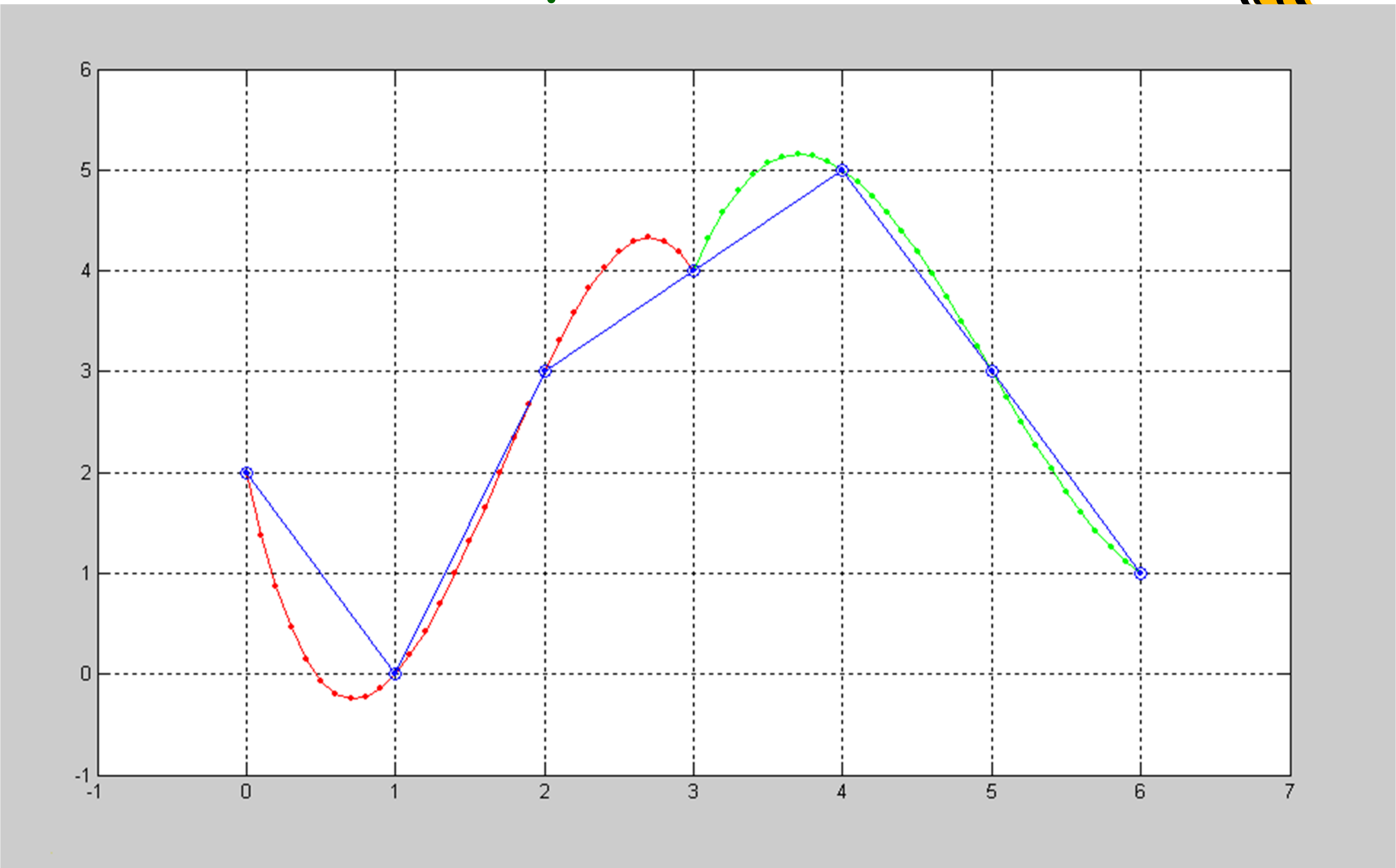
Cubic Spline Int.(6)



Cubic Spline Int.(7)



Cubic Spline Int.(8)



Cubic Spline Int.(9)

- สิ่งที่เราต้องการคือทำให้รอยต่อราบเรียบ
  - โดยการกำหนดให้ค่า First Derivative ที่ปลายของ Polynomial ตัวแรก เท่ากับค่า First Derivative ที่จุดต้นของ Polynomial ตัวที่ 2
  - อีกนัยหนึ่ง คือบังคับให้ค่า First Derivative ที่จุดร่วมของสอง Polynomial ที่มาต่อกัน มีค่าเท่ากัน
  - ในกรณีเราสามารถสร้าง 4 สมการ เพื่อหา 4 Coefficient ได้จากเพียงค่าของ Function ที่สองจุด และค่าของ First Derivative ที่สองจุดนั้น (4 ค่า รวม)
    - กล่าวคือ แต่ละช่วงของ Data (ระหว่างสองจุด) เราจะคำนวณ Third Degree Polynomial หนึ่งตัว

Cubic Spline Int.(10)

- พิจารณาจาก Data n จุดของ x_i i=1,..,n คือ y_i; i=1,..,n เราต้องการหา Cubic Polynomial, $S_i(x)$ ทั้งหมด n-1 function เพื่อทำการ Interpolate ค่าระหว่างจุด ในแต่ละช่วงของสองจุดที่ติดกัน

$$\mathbf{S}(x) = \begin{cases} s_1(x);\ x_1 \le x < x_2 \\ s_2(x);\ x_2 \le x < x_3 \\ \vdots \\ s_{n-1}(x);\ x_{n-1} \le x < x_n \end{cases}$$

$$s_i(x) = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i;\ i = 1,2,\ldots,n-1$$

Cubic Spline Int.(11)

- สมการ Polynomial ที่ใช้เป็นแค่การเปลี่ยน Variable โดยเลื่อน (Delay) สมการจากจุด x_i ให้มาอยู่ที่ตำแหน่ง ศูนย์
- ค่า First และ Second Derivative ของแต่ละ Polynomial สามารถแสดงได้ดังนี้

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

$$s_i'(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i$$

$$s_i''(x) = 6a_i(x - x_i) + 2b_i$$

Cubic Spline Int.(12)

- ถ้าจุด x_i เป็นจุดร่วมระหว่างสอง Polynomial กล่าวคือ

$$s_i(x_i) = s_{i-1}(x_i)$$

- และที่จุด x_i เราจะได้ค่าของ Function คือ

$$y_i = s_i(x_i) = a_i(x_i - x_i)^3 + b_i(x_i - x_i)^2 + c_i(x_i - x_i) + d_i$$

$$y_i = d_i$$

- นอกจากนี้ เราได้(สมมุติว่าแต่ละจุดสุ่มตัวอย่างด้วยระยะห่างเท่ากัน)

$$y_i = s_{i-1}(x_i) = a_{i-1}(x_i - x_{i-1})^3 + b_{i-1}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1}$$

$$y_i = d_i = a_{i-1}h^3 + b_{i-1}h^2 + c_{i-1}h + d_{i-1}; \quad h = x_i - x_{i-1}; i = 2,3,\ldots,n$$

Cubic Spline Int.(13)

- เราสร้างสมการจากการกำหนด First Derivative ของจุดต่อให้เท่ากัน

$$s_i'(x_i) = s_{i-1}'(x_i)$$

- ดังนั้น เราได้

$$s_i'(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i$$

$$s_i'(x_i) = c_i = s_{i-1}'(x_i) = 3a_{i-1}h^2 + 2b_{i-1}h + c_{i-1}; i = 2,3,\ldots,n-1$$

- นอกจากนี้แล้ว ค่า Second Derivative จะต้องต่อเนื่องกันตลอดช่วง

$$s_i''(x_{i+1}) = s_{i+1}''(x_{i+1}); i = 1,2,\ldots,n-1$$

Cubic Spline Int.(14)

- แต่ค่า $s_i''(x) = 6a_i(x - x_i) + 2b_i$
- ดังนั้น

$$s_i''(x_i) = 2b_i$$

$$s_{i+1}''(x_{i+1}) = 2b_{i+1}$$

- และ

$$s_i''(x_{i+1}) = 6a_i(x_{i+1} - x_i) + 2b_i = 6a_i h + 2b_i$$

- จาก

$$s_i''(x_{i+1}) = s_{i+1}''(x_{i+1}); i = 1,2,\ldots,n-1$$

- เราได้

$$2b_{i+1} = 6a_i h + 2b_i$$

Cubic Spline Int.(15)

- เพื่อให้สมการดูง่าย เรากำหนด

$$M_i = s_i''(x_i)$$

- ดังนั้น

$$b_i = M_i / 2$$

- และที่ได้ก่อนหน้านี้ $d_i = y_i$
- ส่วน a_i และ c_i สามารถเขียนใหม่ได้เป็น

$$2b_{i+1} = 6a_i h + 2b_i \Rightarrow a_i = \frac{M_{i+1} - M_i}{6h}$$

$$d_i = a_{i-1}h^3 + b_{i-1}h^2 + c_{i-1}h + d_{i-1}$$

$$d_{i+1} = a_i h^3 + b_i h^2 + c_i h + d_i \Rightarrow c_i = \frac{y_{i+1} - y_i}{h} - \left(\frac{M_{i+1} + 2M_i}{6}\right)h$$

Cubic Spline Int.(16)

- เราสรุปสูตรในการคำนวณดังนี้

$$a_i = \frac{M_{i+1} - M_i}{6h}$$

$$b_i = \frac{M_i}{2}$$

$$c_i = \frac{y_{i+1} - y_i}{h} - \left(\frac{M_{i+1} + 2M_i}{6} \right) h$$

$$d_i = y_i$$

- เมื่อเขียนในรูปของ Matrix เราได้

Cubic Spline Int.(17)

- จาก

$$s_i'(x_{i+1}) = 3a_i h^2 + 2b_i h + c_i = s_{i+1}'(x_{i+1}) = c_{i+1}$$

$$3\left[\frac{M_{i+1} - M_i}{6h}\right]h^2 + 2\left[\frac{M_i}{2}\right]h + \frac{y_{i+1} - y_i}{h} - \left(\frac{M_{i+1} + 2M_i}{6}\right)h = \frac{y_{i+2} - y_{i+1}}{h} - \left(\frac{M_{i+2} + 2M_{i+1}}{6}\right)h$$

$$\frac{h}{6}(M_i + 4M_{i+1} + M_{i+2}) = \frac{y_i - 2y_{i+1} + y_{i+2}}{h}$$

$$M_i + 4M_{i+1} + M_{i+2} = \frac{6}{h}\left(\frac{y_i - 2y_{i+1} + y_{i+2}}{h}\right); i = 1,2,3,.., n-2$$

- สามารถเขียนเป็นสมการ Matrix ได้ดังนี้

Cubic Spline Int.(18)

- $$\begin{bmatrix} 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{n-3} \\ M_{n-2} \\ M_{n-1} \\ M_n \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_1 - 2y_2 + y_3 \\ y_2 - 2y_3 + y_4 \\ y_3 - 2y_4 + y_5 \\ \vdots \\ y_{n-4} - 2y_{n-3} + y_{n-2} \\ y_{n-3} - 2y_{n-2} + y_{n-1} \\ y_{n-2} - 2y_{n-1} + y_n \end{bmatrix}$$
- ระบบประกอบไปด้วย n-2 สมการและ n unknown ดังนั้นเราต้องการอีก สอง Condition เพื่อที่สามารถจะแก้สมการได้
  - สอง Condition สามารถเลือกได้หลายแบบ ทำให้เกิด Variation ของ Cubic Spline Interpolation หลายวิธี

Cubic Spline Int.(19)

- ในที่นี้จะกล่าวถึง Spline สามแบบ ตามสอง Condition ที่กำหนดเพิ่มเติม
  - Natural Spline
    - โดยการกำหนด $M_1=M_n=0$
  - Parabolic Runout Spline
    - โดยการกำหนด $M_1=M_2$ และ $M_n=M_{n-1}$
  - Cubic Runout Spline
    - โดยการใช้ $M_1=2M_2-M_3$ และ $M_n=2M_{n-1}-M_{n-2}$
- ในที่นี้จะกล่าวรายละเอียดเฉพาะ Natural Spline

Cubic Spline Int.(20)

- Natural Spline
  - เมื่อให้ $M_1=M_n=0$ Matrix จะลดรูปเหลือ n-2 สมการ และ n-2 Unknown ดังนี้

$$
\begin{bmatrix}
4 & 1 & 0 & \cdots & 0 & 0 & 0 \\
1 & 4 & 1 & \cdots & 0 & 0 & 0 \\
0 & 1 & 4 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 4 & 1 & 0 \\
0 & 0 & 0 & \cdots & 1 & 4 & 1 \\
0 & 0 & 0 & \cdots & 0 & 1 & 4
\end{bmatrix}
\begin{bmatrix}
M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{n-3} \\ M_{n-2} \\ M_{n-1}
\end{bmatrix}
= \frac{6}{h^2}
\begin{bmatrix}
y_1 - 2y_2 + y_3 \\
y_2 - 2y_3 + y_4 \\
y_3 - 2y_4 + y_5 \\
\vdots \\
y_{n-4} - 2y_{n-3} + y_{n-2} \\
y_{n-3} - 2y_{n-2} + y_{n-1} \\
y_{n-2} - 2y_{n-1} + y_n
\end{bmatrix}
$$

  - เมื่อเราแก้สมการได้ค่า M_i เราสามารถคำนวณค่า Coefficient ของแต่ละ Polynomial ได้

Cubic Spline Int.(21)

- สังเกตุว่า Coefficient Matrix ของระบบมีค่าในส่วน Diagonal ที่เป็นค่าคงที่ ซึ่ง Matrix นี้เป็น Matrix พิเศษ เรียก Toeplitz Matrix
  - Toeplitz Matrix สามารถแก้ปัญหาได้ โดยใช้ $O(n^2)$ Computation Time แทนที่จะเป็น $O(n^3)$
    - Algorithm ที่ใช้คือ Levinson Algorithm
  - Toeplitz Matrix สามารถทำ LU Decomposition โดยใช้ $O(n^2)$ เช่นเดียวกัน
- รายละเอียดของคุณสมบัติพิเศษของ Matrix นี้ และ Levinson Algorithm อยู่นอกขอบเขตของวิชานี้ ผู้สนใจสามารถศึกษาเพิ่มเติมได้จากตำรา ด้าน Linear Algebra และ Numerical Method

Example: Natural Spline

- จากตัวอย่างเดิม เราจะใช้ Spline มาทำการ Fit Data ในกรณีนี้ n=7, h = 1, $M_0=M_7=0$
  - y_i = [2 0 3 4 5 3 1]; x_i = [0 1 2 3 4 5 6]
- เราแก้สมการ Matrix ดังนี้

$$\begin{bmatrix} 4 & 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \end{bmatrix} = 6 \begin{bmatrix} 2-2\cdot 0+3 \\ 0-2\cdot 3+4 \\ 3-2\cdot 4+5 \\ 4-2\cdot 5+3 \\ 5-2\cdot 3+1 \end{bmatrix} = 6 \begin{bmatrix} 5 \\ -2 \\ 0 \\ -3 \\ 0 \end{bmatrix}$$

- และได้คำตอบคือ

$$\mathbf{M} = [0, 8.9923, -5.9692, 2.8846, -5.5692, 1.3923, 0]^T$$

$$\mathbf{M} = [0, 8.9923, -5.9692, 2.8846, -5.5692, 1.3923, 0]^T$$

- ดังนั้น Coefficient ของ Cubic ทั้ง 6 จะเป็น

$$a_i = \frac{M_{i+1} - M_i}{6h} = [1.4987, -2.4936, 1.4756, -1.4090, 1.1603, -.2321]$$

$$b_i = M_i / 2 = [0, 4.4962, -2.9846, 1.4423, -2.7846, 0.6962]$$

$$c_i = \frac{y_{i+1} - y_i}{h} - \left(\frac{M_{i+1} + 2M_i}{6} \right) h$$

$$= [-3.4987, 0.9974, 2.5090, 0.9667, -0.3756, -2.4641]$$

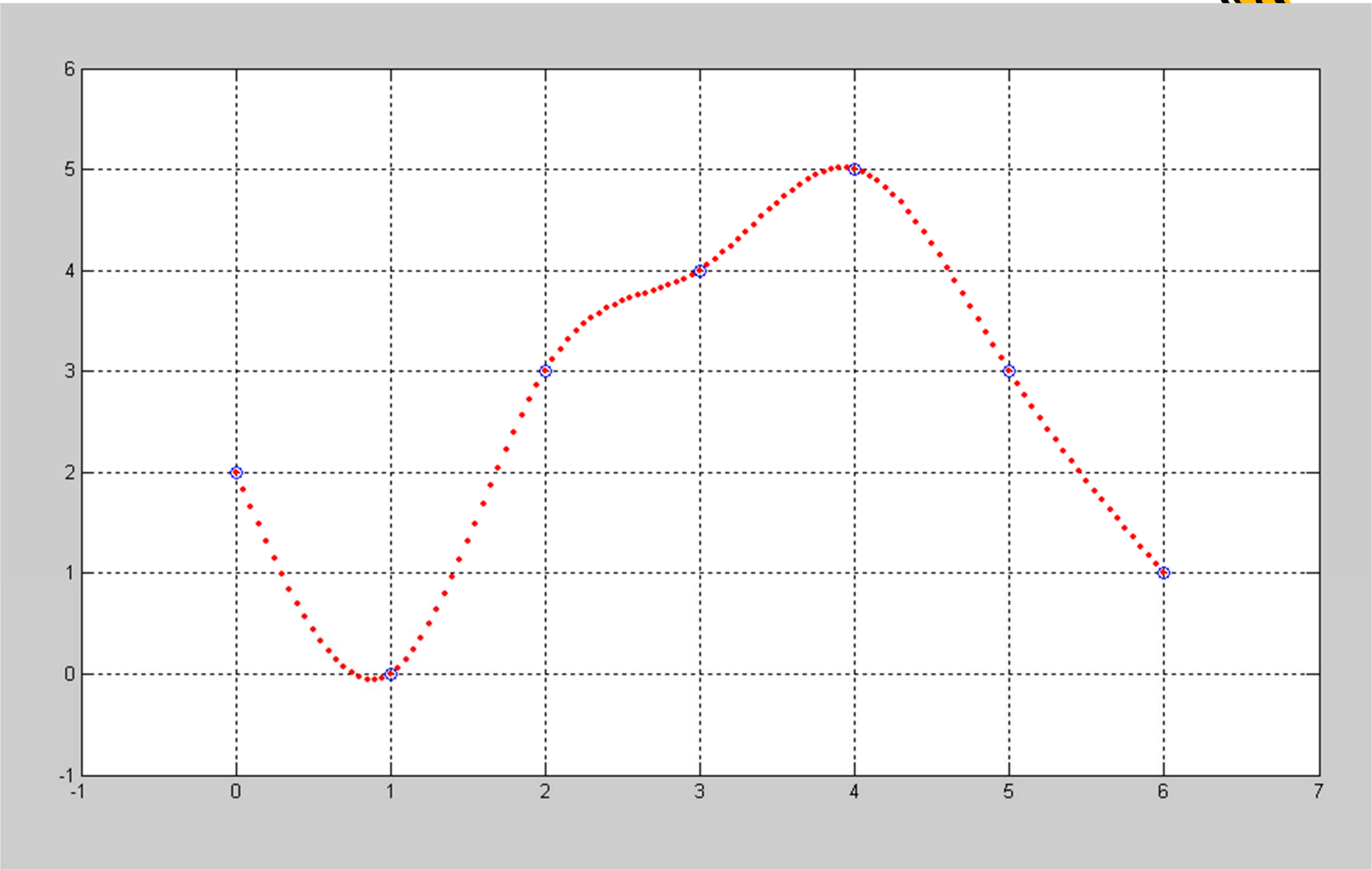
$$d_i = y_i = [2, 0, 3, 4, 5, 3]$$

Polynomial Coefficient Matrix

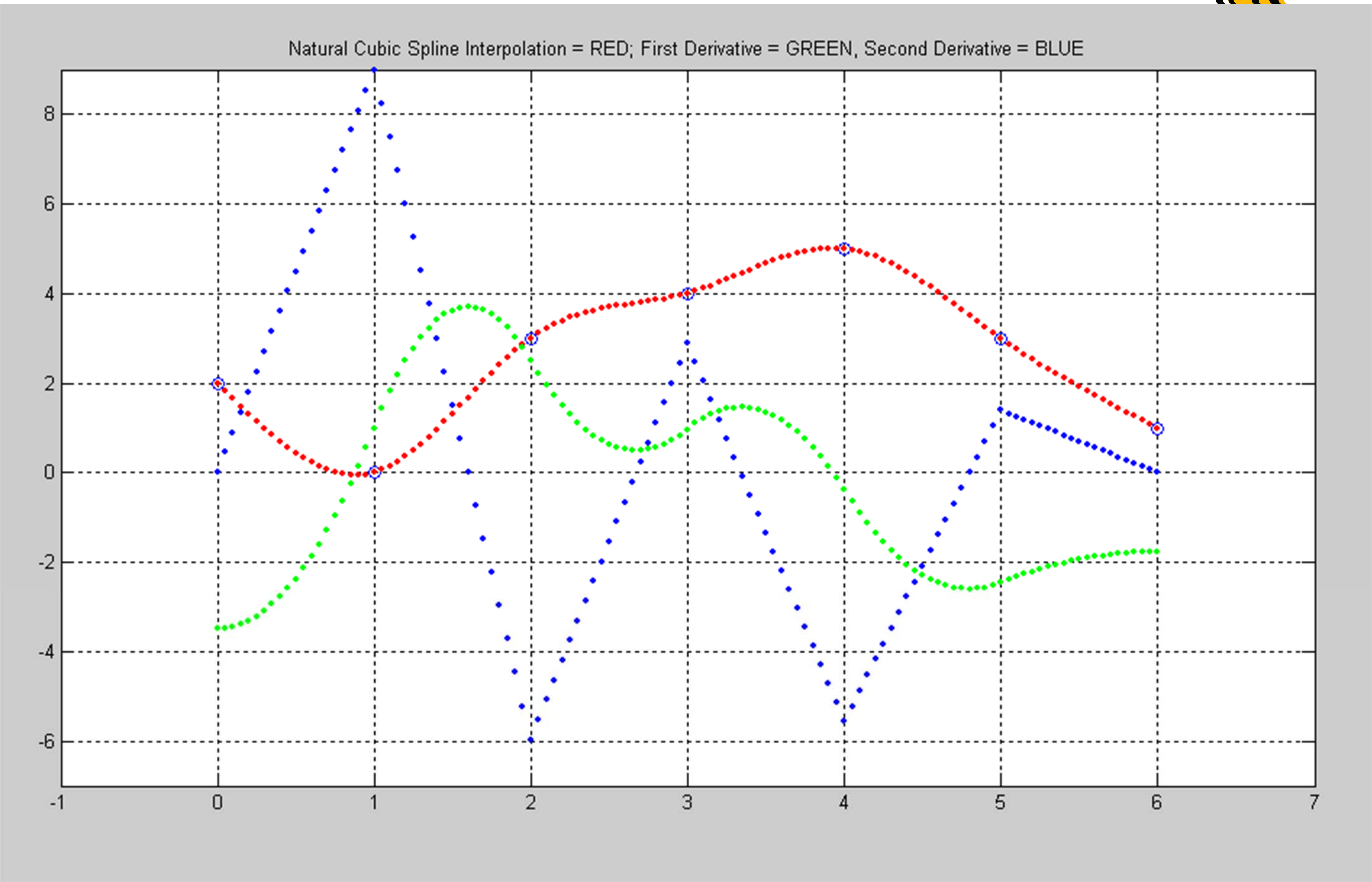
- `>> p=[a b c d]`
- `p =`
- `   1.4987        0  -3.4987   2.0000`
- `  -2.4936   4.4962   0.9974        0`
- `   1.4756  -2.9846   2.5090   3.0000`
- `  -1.4090   1.4423   0.9667   4.0000`
- `   1.1603  -2.7846  -0.3756   5.0000`
- `  -0.2321   0.6962  -2.4641   3.0000`

- **ต่อไปเราจะลอง Plot แต่ละ Polynomial ในแต่ละช่วง ทั้งหมด 6 Polynomial และ 6 ช่วง**

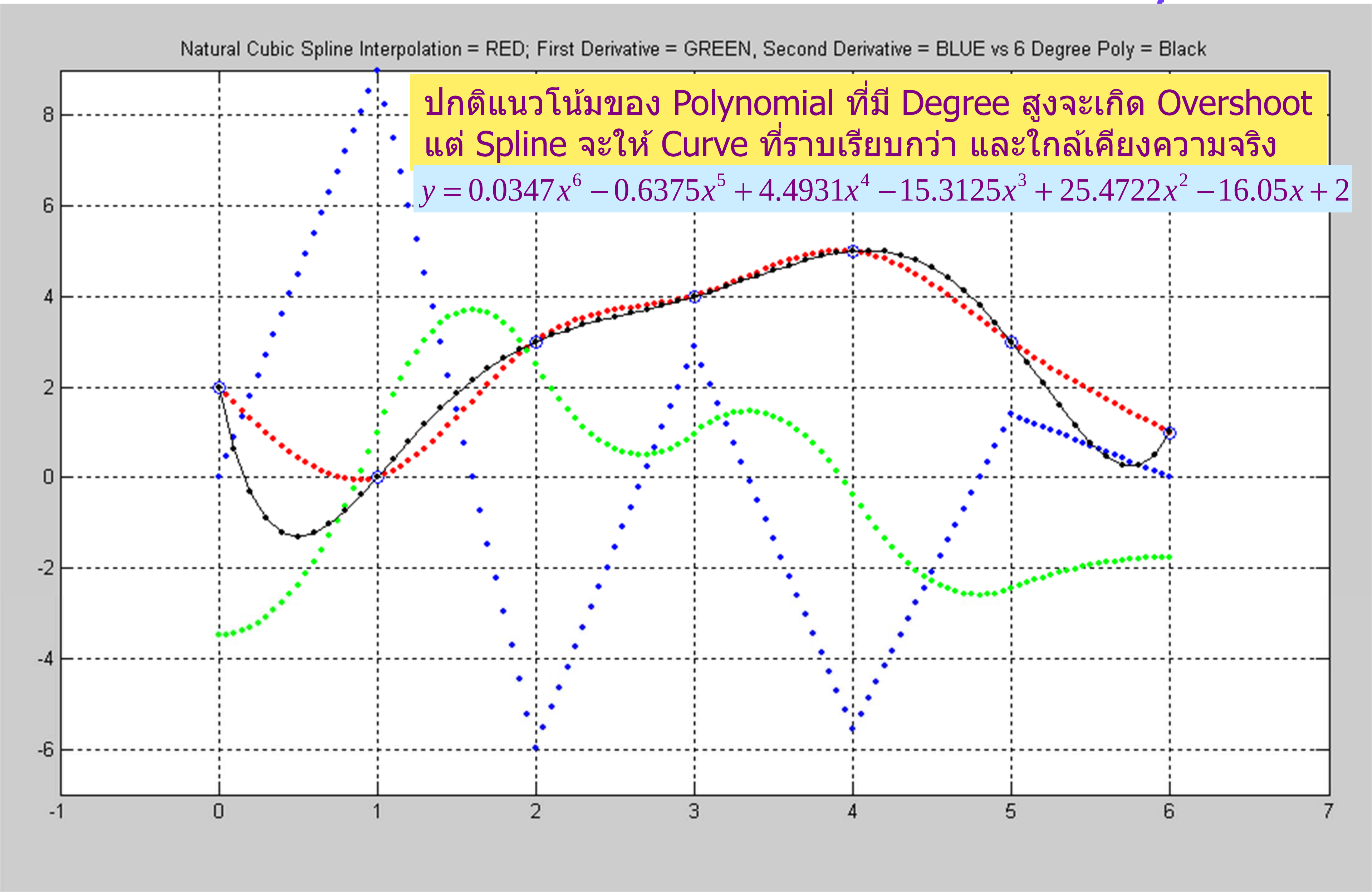
Natural Spline Interpolation



Natural Spline Interpolation = R;
First Der = G; Second Der = B



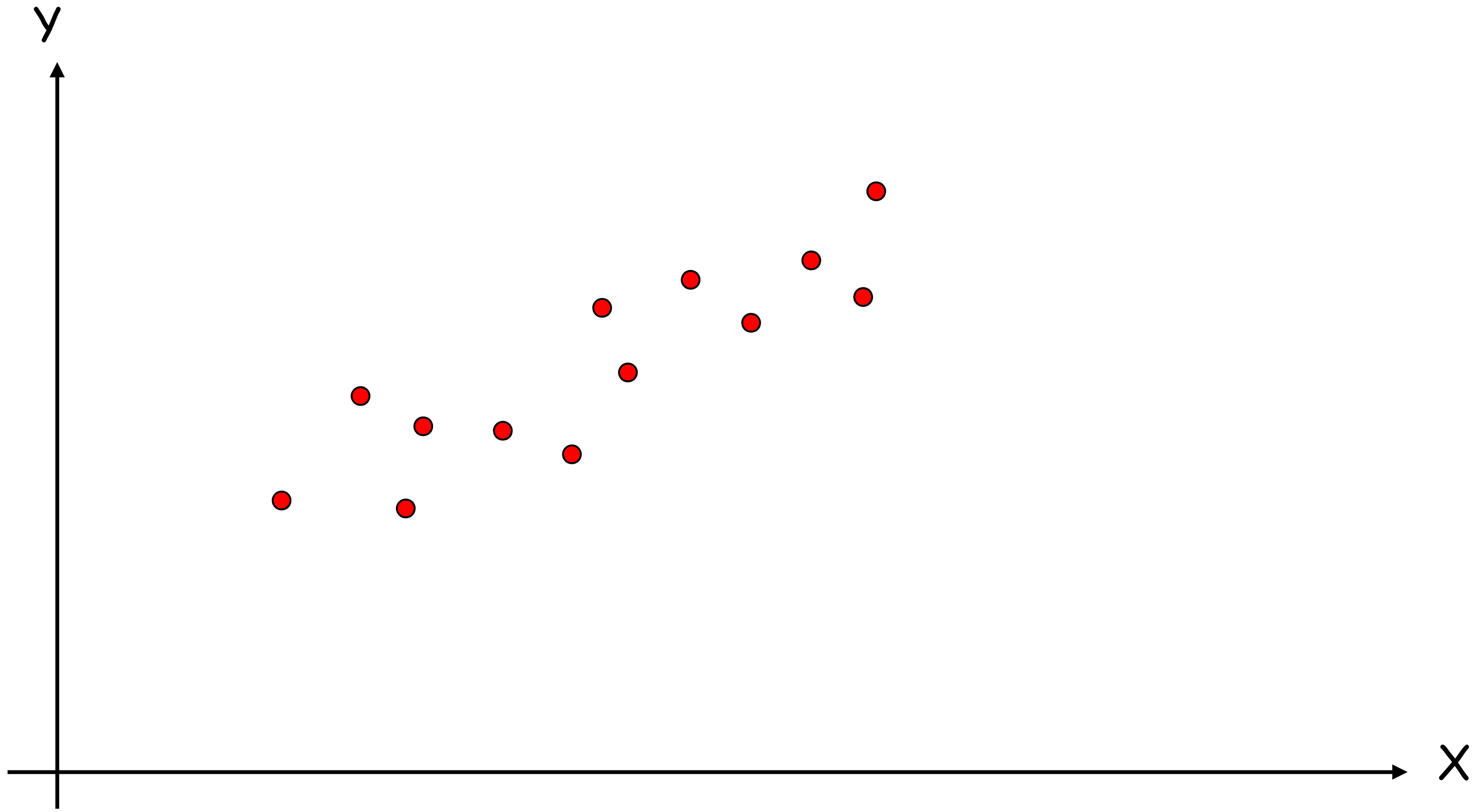
Natural Spline Interpolation = R; First D = G; Second D = B vs 6^{th} D Poly = K



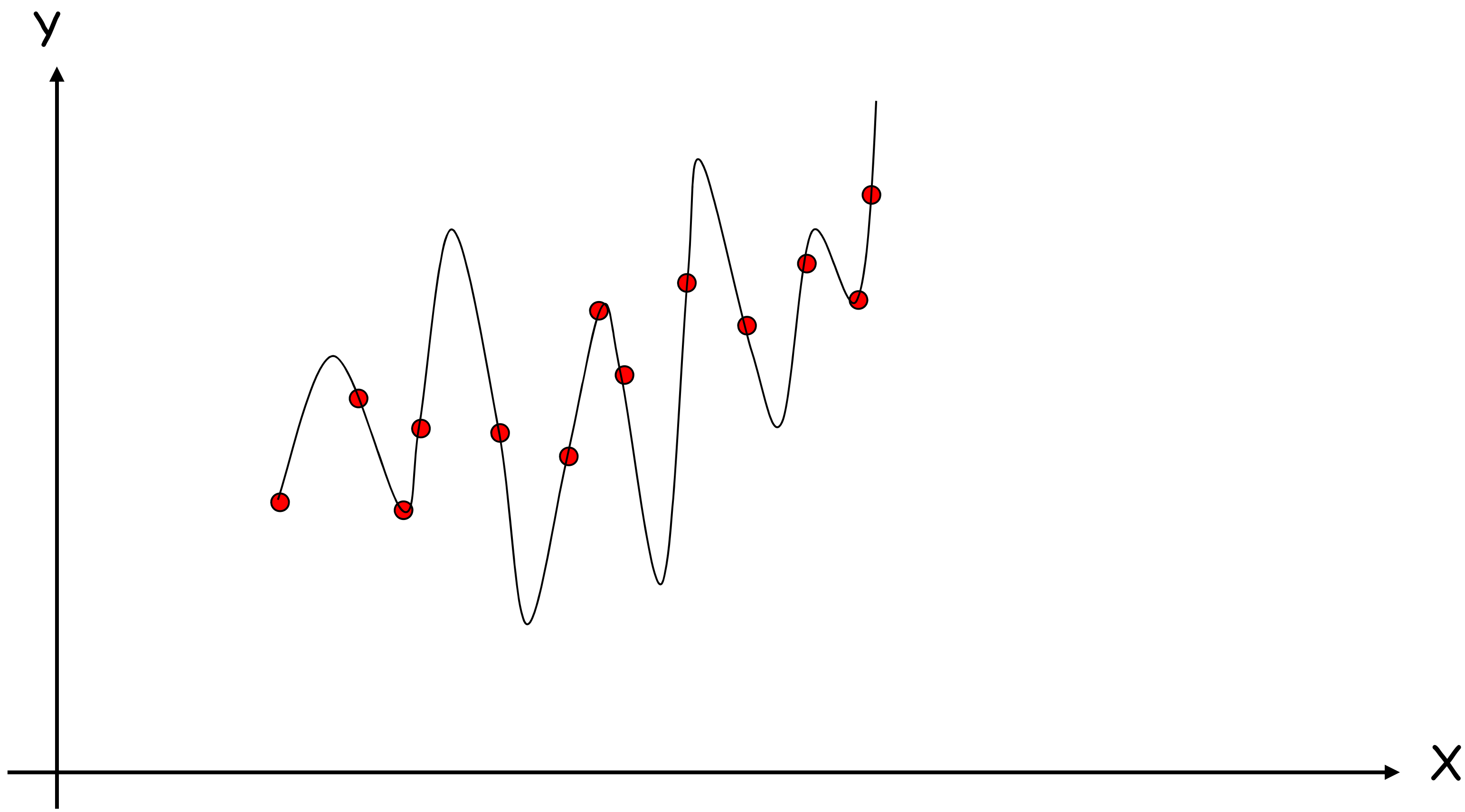
Regression

- บางครั้ง Data ที่เราเก็บได้ กระจาย และการทำ Interpolation ด้วย Polynomial จะไม่เหมาะสม
- การกระจายของ Data อาจจะเกิดจากความไม่แน่นอนในการวัด หรือเป็นลักษณะ Random
- อาจจะเกิดจาก Noise ในการวัด
  - การ Fit ด้วย Polynomial หรือแม้แต่ Spline จะให้คำตอบที่ไม่ตรงกับความจริง

Regression: Data with Noise



Polynomial Fit Noise Data



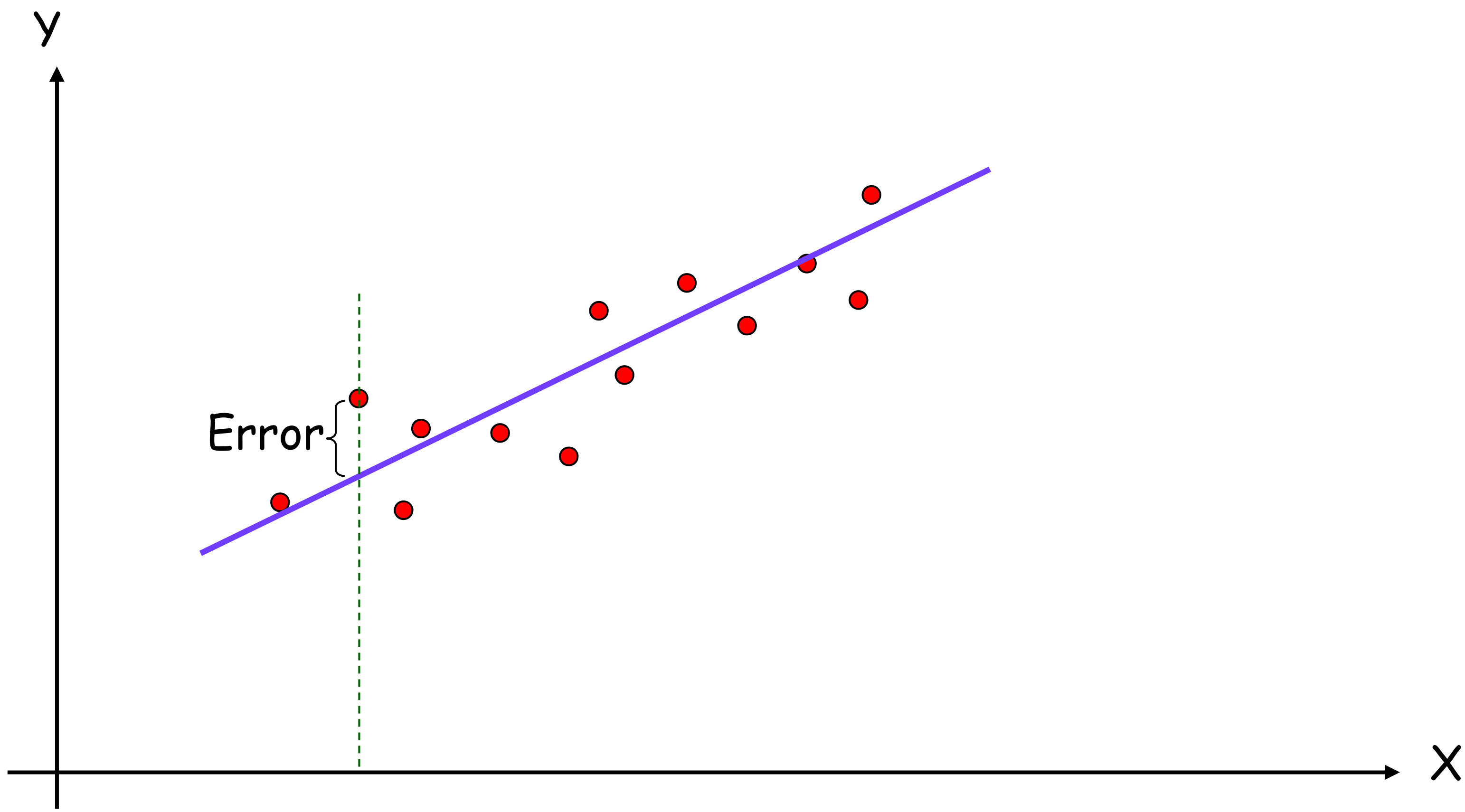
Ordinary Simple Linear Least-Squares Regression

- ในการนี้ การประมาณค่าโดยใช้ Function ง่ายๆ เช่น เส้นตรง หรือ Exponential จะเหมาะสมกว่า
  - โดยเฉพาะถ้าเรารู้พฤติกรรมของระบบอยู่บ้าง และคาดหวังว่า Data ที่ได้ควรจะมีลักษณะอย่างไร
- การหาสมการที่จะ Fit ที่สุด โดยมีค่า Error ต่ำสุด เราเรียกเป็นการทำ Regression
- กรณีของสมการเส้นตรง (สำหรับตัวแปรเดียว)เราเรียก Linear Regression
  - ที่ถูกควรเรียก Simple Linear Regression ใช้สำหรับตัวแปรเดียว จะเป็นสมการเส้นตรงในสองมิติ
    - เนื่องจากมี Multiple Linear Regression ด้วยสำหรับกรณีหลายตัวแปร ซึ่งผลจะเป็น Linear Equation ในหลายมิติ
  - สมการเส้นตรงคือ $y=ax+b$ (บางตำราใช้ $y = a + bx$ ซึ่งในกรณีนี้ต้องสลับค่า a และ b ในการคำนวณที่จะกล่าวต่อไป)
  - Parameter 2 ตัวที่ต้องหาคือ a = slope และ b = y-intercept

Ordinary Simple Linear Least Square Regression

- สมการที่ Fit ที่สุดคือให้ Error รวมต่ำสุด เรามักจะใช้วิธีของ Least-Square เรียก Ordinary Least Square Regression (OLS) (หรือ Linear Least Square)
  - ผลรวมของ Error ยกกำลังสองของแต่ละจุดของ Data เทียบกับเส้นตรงที่ใช้ มีค่าต่ำสุด
- Linear Regression ใช้กันมากในทางสถิติ เพื่อหาความสัมพันธ์แบบเส้นตรงของ Random Variable สองตัว
- Linear Regression ประเภทอื่นๆก็มีใช้กันอยู่ แต่ OLS จะนิยมที่สุด

Linear Least-Squares Regression



Ordinary Linear Least Square Regression (OLS)

- ค่า a และ b หาได้จากสมการ Summation ของ Error ยกกำลังสอง ของทุกจุด
  - ค่าที่ต่ำสุดหาได้จากการหาค่า Derivative ของสมการ และตั้งให้เท่ากับ 0 (จุด Minima)
- ทั้งสอง Unknown คือ a และ b หาได้จากสมการ Derivative สองสมการที่ตั้งให้เท่ากับ 0 ดังนี้
  - df(a,b)/da = 0
  - df(a,b)/db = 0
  - f(a,b) คือ Sum of Square Error Function
- อีกค่าที่ใช้ในทางสถิติคือค่า r = correlation coefficient เป็นตัวบอกว่าเส้นตรงที่ใช้มัน Fit ได้ดีเพียงไร, r จะมีค่าระหว่าง [-1,1]

Linear Least-Squares Regression

$$e_i = y_i - ax_i - b;\ i = 1,2,\ldots,n$$

$$f(a,b) = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - ax_i - b)^2$$

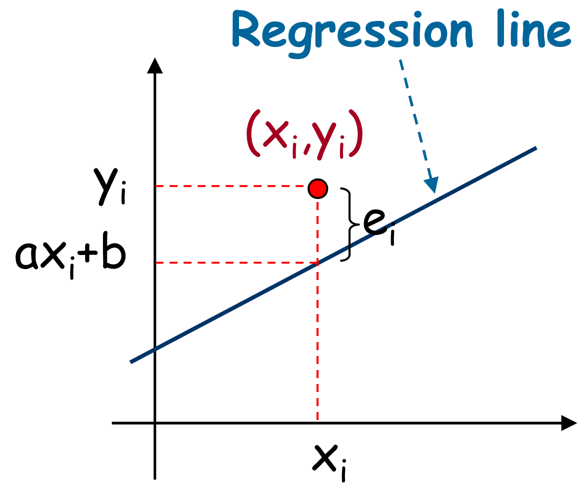
$$\frac{df}{da} = -2\sum_{i=1}^{n} [(y_i - ax_i - b)x_i] = 0$$

$$\frac{df}{db} = -2\sum_{i=1}^{n} (y_i - ax_i - b) = 0$$

$$a = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}$$

$$b = \overline{y} - a\overline{x}$$

$$r = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$



สมมุติเราได้ Data Pair (x_i,y_i); $i = 1,2,\ldots,n$ ทั้งหมด n จุด

Variable X และ Y อาจจะเป็นได้ทั้ง Continuous และ Discrete

$$-2\sum_{i=1}^{n}[(y_i - ax_i - b)x_i] = 0$$

$$\sum_{i=1}^{n} y_i x_i - a\sum_{i=1}^{n} x_i^2 - b\sum_{i=1}^{n} x_i = 0 \qquad (1)$$

replace b in (1)

$$\sum_{i=1}^{n} y_i x_i - a\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} y_i - a\frac{1}{n}\sum_{i=1}^{n} x_i\right)\sum_{i=1}^{n} x_i = 0$$

$$\sum_{i=1}^{n} y_i x_i - \frac{1}{n}\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i = a\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right)$$

$$n\sum_{i=1}^{n} y_i x_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i = a\left(n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right)$$

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$-2\sum_{i=1}^{n}[(y_i - ax_i - b)] = 0$$

$$\sum_{i=1}^{n} y_i - a\sum_{i=1}^{n} x_i - nb = 0$$

$$b = \frac{1}{n}\sum_{i=1}^{n} y_i - a\frac{1}{n}\sum_{i=1}^{n} x_i = \overline{Y} - a\overline{X}$$

R, r : Correlation Coefficient

ส่วน Correlation Coefficient คือ
Pearson Product-Moment Correlation Coefficient
นิยามว่าเท่ากับอัตราส่วนของค่า Covariance ต่อผลคูณของค่า Standard Deviation ของทั้งสอง Variable (ของตัวอย่าง) และ สามารถแสดงได้ว่ามีค่า

$$r = \frac{C_{XY}}{S_X S_Y}$$

$$r = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

r = 1 หรือ -1 แสดงว่ามีความสัมพันธ์แบบเส้นตรงอย่างสมบูรณ์ (ในเชิงบวก หรือในเชิงลบ)
ถ้า r เป็นศูนย์แสดงว่าไม่มีความสัมพันธ์กันเลย

Nonlinear Regression

- ในกรณีที่เส้นตรงไม่เหมาะสมที่จะนำมาใช้ เราสามารถจะใช้ Function อื่นมาทำ Regression เช่น
  - Exponential Model
  - Power Equation
  - Saturation-Growth-Rate Equation
  - Polynomial Regression
- รายละเอียดอยู่นอกเหนือวิชานี้

Example

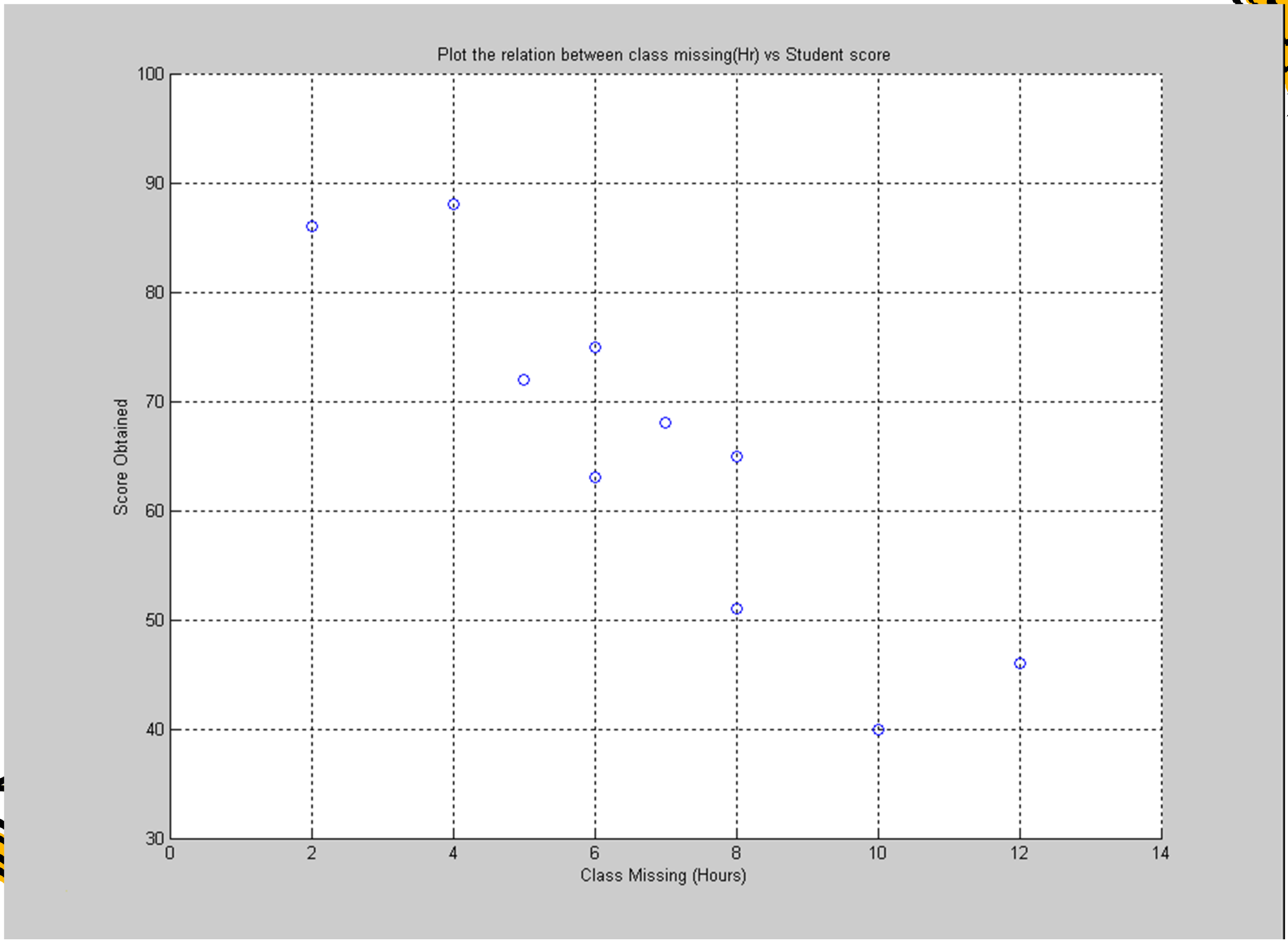
- 1. จากการเก็บข้อมูลตัวอย่าง วิชา CPE 332 ระหว่าง คะแนน ที่นักศึกษาได้ กับจำนวนรวมเป็นชั่วโมงที่นักศึกษาขาดเรียน หรือมาสาย ได้ข้อมูลดังตารางข้างล่าง

Data	Hour	Grade
1	5	72
2	8	51
3	2	86
4	6	75
5	4	88

Data	Hour	Grade
6	10	40
7	7	68
8	6	63
9	12	46
10	8	65

- จงใช้ OLS Regression แสดงสัมพันธ์ระหว่างตัวแปรทั้งสอง ทำการ Plot Scatter Diagram และเส้นตรงที่ได้ จากนั้นให้หาค่า Correlation Coefficient

Example



Plot the relation between class missing(Hr) vs Student score
Score Obtained
Class Missing (Hours)
100
90
80
70
60
50
40
30
0
2
4
6
8
10
12
14

Example

$$n = 10; \quad \sum_{i=1}^{10} x_i = 68; \quad \overline{X} = \frac{1}{10}\sum_{i=1}^{10} x_i = 6.8$$

$$\sum_{i=1}^{10} y_i = 654; \quad \overline{Y} = \frac{1}{10}\sum_{i=1}^{10} y_i = 65.4$$

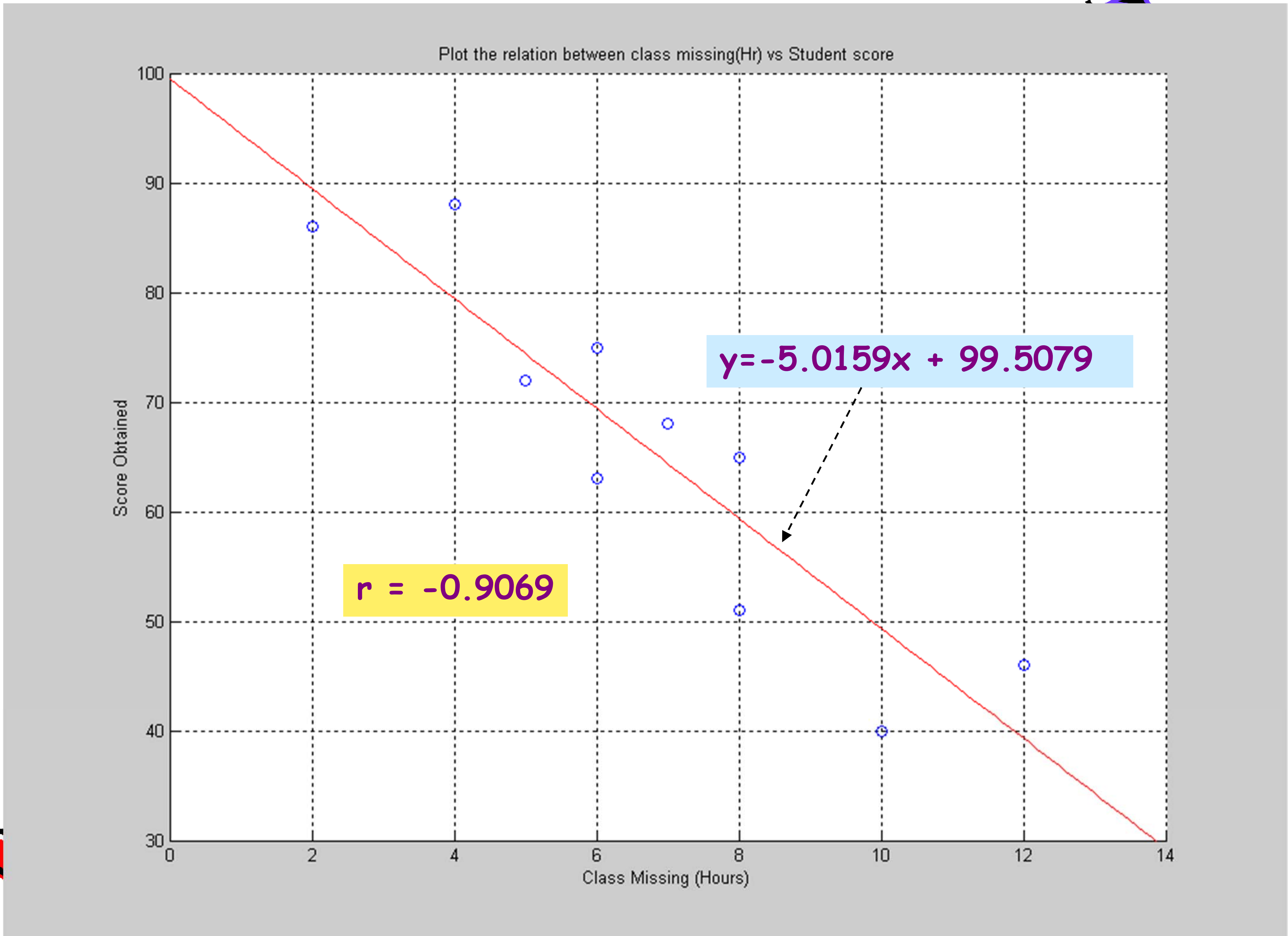
$$\sum_{i=1}^{10} x_i y_i = 4068; \quad \sum_{i=1}^{10} x_i^2 = 538; \quad \sum_{i=1}^{10} y_i^2 = 45084;$$

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} = \frac{10\times 4068 - 68\times 654}{10\times 538 - 68^2} = -5.0159$$

$$b = \overline{Y} - a\overline{X} = 65.4 - (-5.0159)\times 6.8 = 99.5079$$

$$r = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

$$= \frac{10\times 4068 - 68\times 654}{\sqrt{10\times 538 - 68^2}\sqrt{10\times 45084 - 654^2}} = -0.9069$$


Plot the relation between class missing(Hr) vs Student score
Score Obtained
Class Missing (Hours)
100
90
80
70
60
50
40
30
0
2
4
6
8
10
12
14
y=-5.0159x + 99.5079
r = -0.9069

End of Chapter 12

- Download
- Homework 12 ส่งก่อนวันพุธหน้าที่ 3 ธ.ค. ก่อน 12.00 น. (ส่งที่สาขา ใส่กล่อง ห้อง 5-310)

Course Ends

- Prepare For Exam
  - ข้อสอบมี 9 ข้อ ทำทุกข้อ (นำเครื่องคิดเลขมาด้วย)
  - เรื่องก่อน Midterm จะออก 3 ข้อ จาก 6 บทแรก + 6 ข้อหลัง MT
    - 1. Function Approximation (1 ข้อ)
      - Taylor Series/McLauren Series
    - 2. Roots of Function (1 ข้อ)
      - Bisection
      - Newton-Ralphson
    - 3. Linear Equations (1 ข้อ) เรื่องใดเรื่องหนึ่ง
      - Gauss Elimination
      - Gauss Jordan (Including Matrix Inverse)
      - Gauss Seidel
      - LU Decomposition (Crout Decomposition)

Course Ends

- Prepare For Exam
  - 4. Numerical Integration (1 ข้อ) ไม่ออก Finite Difference
    - Trapezoidal Rule
    - Simpson 1/3 Rule
    - Richardson Extrapolation
  - 5. ODE (1 ข้อ)
    - Classical Forth Order RK Method เรื่องเดียว
  - 6. Curve Fitting (1 ข้อ)
    - Natural Spline
    - OLS Regression

Formulas
ดูใน MT+ต่อไปนี้

Linear Regression (OLS):

$$a = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}$$

$$b = \bar{y} - a\bar{x}$$

$$r = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

Natural Spline:

$$\begin{bmatrix} 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{n-3} \\ M_{n-2} \\ M_{n-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_1 - 2y_2 + y_3 \\ y_2 - 2y_3 + y_4 \\ y_3 - 2y_4 + y_5 \\ \vdots \\ y_{n-4} - 2y_{n-3} + y_{n-2} \\ y_{n-3} - 2y_{n-2} + y_{n-1} \\ y_{n-2} - 2y_{n-1} + y_n \end{bmatrix}$$

$$a_i = \frac{M_{i+1} - M_i}{6h};\ b_i = \frac{M_i}{2}$$

$$c_i = \frac{y_{i+1} - y_i}{h} - \left(\frac{M_{i+1} + 2M_i}{6}\right)h$$

$$d_i = y_i$$

Numerical Methods:

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)} \text{ (False-Position Method)}$$

$$x_{i+1} = g(x_i) \quad \text{(Simple One-Point Iteration)}$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \text{ (Newton-Ralphson Formula)}$$

$$x_{i+1} = x_i - \frac{f(x_i)[x_{i-1} - x_i]}{f(x_{i-1}) - f(x_i)}$$

(Secant Method Formula)

LU Decomposition:

$$l_{i1} = a_{i1}, \quad for\ i = 1,2,\ldots,n$$

$$u_{ij} = \frac{a_{1j}}{l_{11}}, \quad for\ j = 2,3,\ldots,n$$

$For\ j = 2,3,\ldots,n-1$

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}, \quad for\ i = j, j+1,\ldots,n$$

$$u_{jk} = \frac{a_{jk} - \sum_{i=1}^{j-1} l_{ji}u_{ik}}{l_{jj}}, \quad for\ k = j+1, j+2,\ldots,n$$

$$l_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk}u_{kn}$$

Simpson's 1/3 Rule:

$$I \cong \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)]$$

Simpson's 3/8 Rule:

$$I \cong \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]$$

Rhomberg Integration:

$$I_{j,k} \approx \frac{4^{k-1}I_{j+1,k-1} - I_{j,k-1}}{4^{k-1} - 1}$$

ODE

Euler's Method: $y_{i+1} = y_i + f(x_i, y_i)h$

Heun's Method:

$$y_{i+1}^0 = y_i + f(x_i, y_i)h$$

$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)}{2}h$$

Polygon Method: $y_{i+1} = y_i + f(x_{i+1/2}, y_{i+1/2})h$

Ralston Method:

$$\boxed{\begin{aligned} y_{i+1} &= y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)h \\ k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + \tfrac{3}{4}h, y_i + \tfrac{3}{4}hk_1) \end{aligned}}$$

Classical Fourth Order Runge-Kutta Method:

$$y_{i+1} = y_i + \left[\tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\right]h$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}hk_1)$$

$$k_3 = f(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}hk_2)$$

$$k_4 = f(x_i + h, y_i + hk_3)$$

Taylor Series Expansion:

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \cdots$$

$$= \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n = \sum_{n=0}^{k} \frac{f^{(n)}(a)}{n!}(x-a)^n + R_k$$

Maclaurin Series:

$$f(x) = f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + \cdots$$

$$= \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!}x^n$$

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots;\ \forall x$$

$$\log(1-x) = -\sum_{n=1}^{\infty} \frac{x^n}{n};\ -1 \le x < 1$$

$$\log(1+x) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^n}{n};\ -1 < x \le 1$$

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots;\ \forall x$$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots;\ \forall x$$

END OF CPE 332 T1-57

- Minimum 40% เพื่อที่จะผ่านวิชานี้
- A จะต้องได้ 80% ขึ้นไป