CPE 332
Computer Engineering Mathematics II

Chapter 1 Vector

Web Site

- http://cpe.rsu.ac.th/ut
  - Download Material, Course Notes
  - Download Slides
  - Download HW Solutions
  - Grading
  - Announcements
  - Resources

Today Topics

- Period 1
  - Course Outlines
  - Course Web Site
  - Part I Chapter 1 Vector (Review)
  - Breaks
  - Part I Chapter 1 Vector (Review)
- Assignment:
  - ยังไม่มีการบ้าน
  - **Download MATLAB Tutorial 1-5 และลองทำ Exercise ดู**
- Next Week ต่อ Vector และ Chapter 2 เรื่อง Matrix

CPE 332

Computer Engineering Mathematic II

PART I: Linear Algebra

(Chapter 1-3)

CPE 332 T1-57 Wk1

CHAPTER I

VECTORS

Definition of Vector

1.1 Vectors: นิยามและคำจำกัดความ

Vector เป็นปริมาณที่มีทั้งขนาด(Magnitude) และทิศทาง Direction ผิดกับ Scalar ที่มีแต่ขนาดอย่างเดียว ดังนั้นการบวก-ลบ และการคูณ(Product) ของ Vector จะต่างจาก Scalar มองอีกมุมมองหนึ่งเสมือนกับว่า Vector มีมากกว่า 1 มิติ ในขณะที่ Scalar มีแค่มิติเดียว

การแสดง Vector แสดงได้ด้วยเส้นที่มีลูกศรกำหนดทิศทางใน Space ที่มีมากกว่า 1 มิติ

Notation: $\mathbf{F} = \overline{OP}, \overrightarrow{OP}$

Magnitude: $|\mathbf{F}| = F = \left|\overrightarrow{OP}\right| = OP$

ถ้าเราให้ $\hat{\mathbf{f}}$ เป็น Vector ที่มีขนาดเท่ากับหนึ่ง และมีทิศทางเดียวกันกับ $\mathbf{F}$ เราเรียก $\hat{\mathbf{f}}$ ว่าเป็น Unit Vector ของ $\mathbf{F}$ ดังนั้นเราสามารถเขียน Vector ได้ในรูปของ Magnitude ของมันคูณกับ Unit Vector

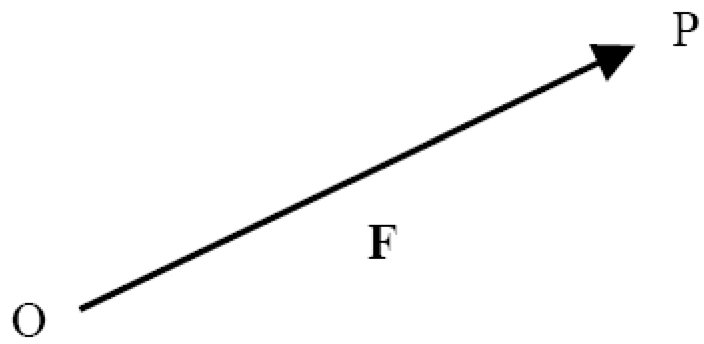
Unit Vector: $\mathbf{F} = |\mathbf{F}|\hat{\mathbf{f}} = F\hat{\mathbf{f}}$

Vector จะเท่ากันก็ต่อเมื่อมีขนาด และทิศทางเดียวกัน กล่าวคือ Vector มีขนาดเท่ากัน และขนานกัน

ถ้าให้ $\mathbf{F}$ เป็น Vector เราจะได้ $-\mathbf{F}$ เป็น Vector ที่มีขนาดเท่ากันแต่มีทิศทางตรงกันข้าม

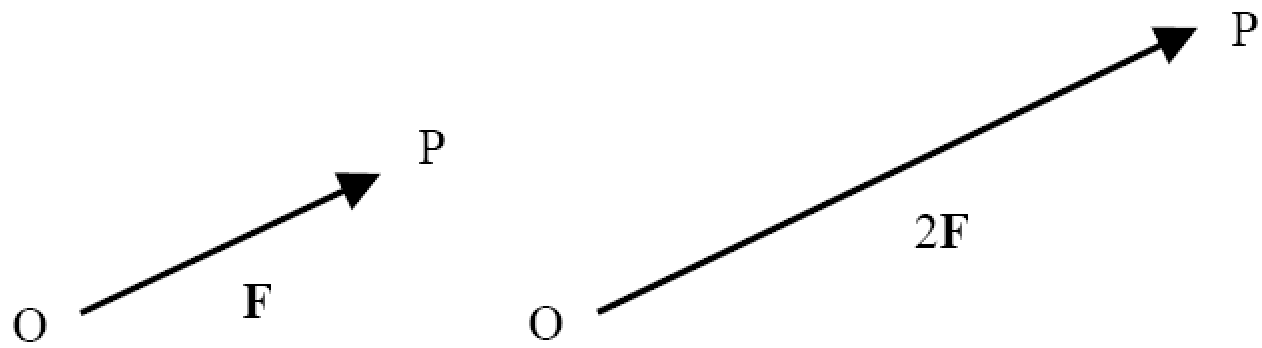
Definition of Vector

สำหรับการแสดงด้วยรูปภาพ เราจะแสดงทั้งขนาดและทิศทางด้วยเส้นที่มีลูกศร โดยขนาดแสดงด้วยความยาวของเส้น และทิศทางตามลูกศรนั้น



รูปที่ 1.1 แสดง Vector

เมื่อเราคูณ Vector ด้วยค่าของ Scalar หมายถึงเราคูณความยาวของมัน หรือคูณ Magnitude ของมันด้วยค่า Scalar นั้น ในกรณีนี้ ทิศทางของ Vector ไม่เปลี่ยนแปลง



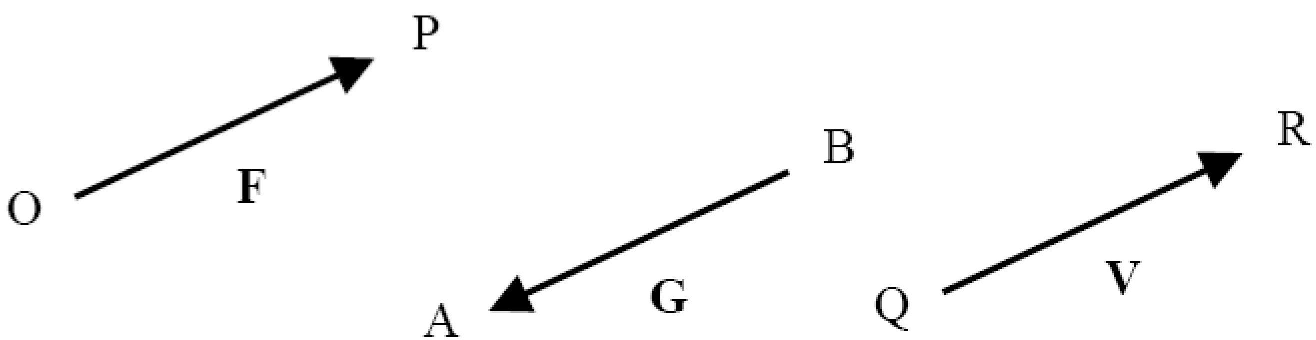
รูปที่ 1.2 แสดงการคูณ Vector ด้วย Scalar

ถ้าเราให้ $\hat{\mathbf{f}}$ เป็น Vector ที่มีขนาด (Magnitude) เท่ากับ “หนึ่ง” และมีทิศทางเดียวกันกับ $\mathbf{F}$ เราเรียก $\hat{\mathbf{f}}$ ว่าเป็น Unit Vector ของ $\mathbf{F}$ ดังนั้นเราสามารถเขียน Vector ได้ในรูปของ Magnitude ของมันคูณกับ Unit Vector

Unit Vector: $\mathbf{F} = |\mathbf{F}|\hat{\mathbf{f}} = F\hat{\mathbf{f}}$

Vector จะเท่ากันก็ต่อเมื่อมีขนาด และทิศทางเดียวกัน กล่าวคือ Vector มีขนาดเท่ากัน และขนานกัน

ถ้าให้ $\mathbf{F}$ เป็น Vector เราจะได้ $-\mathbf{F}$ เป็น Vector ที่มีขนาดเท่ากันแต่มีทิศทางตรงกันข้าม จากรูป เราได้ $\mathbf{F} = \mathbf{V} = -\mathbf{G}$ หรือ $\overrightarrow{OP} = \overrightarrow{QR} = -\overrightarrow{BA}$



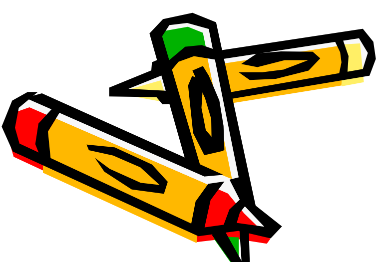
รูปที่ 1.3 แสดง Vector ที่เท่ากัน

Notes

- เนื่องจาก Vector มีทั้งขนาดและทิศทาง เราสามารถเขียน Vector เป็นสองส่วน
  – ส่วนขนาดแทนที่ด้วย Scalar
  – ส่วนทิศทาง จะแทนที่ด้วย Unit Vector ที่มีทิศทางเดียวกับ Vector เดิม

$$\vec{\mathbf{F}} = \mathbf{F} = \left|\mathbf{F}\right| \hat{\mathbf{f}} = F\,\hat{\mathbf{f}} = F_1\mathbf{i} + F_2\mathbf{j} + F_3\mathbf{k} = F(\cos\alpha\,\mathbf{i} + \cos\beta\,\mathbf{j} + \cos\gamma\,\mathbf{k})$$

- การกำหนดทิศทาง อาจจะกำหนดเป็น Component ในแกน Coordinate (x,y,z); อาจจะกำหนดเป็นมุมที่กระทำกับแกน Coordinate
- อาจจะกำหนดเป็น Ratio ที่กระทำกับแกนก็ได้
- จะกล่าวต่อไปภายหลัง
  – เราจะเน้นที่สองอันแรก คือกำหนดเป็น Component i,j,k ในแกน x,y,z
  – หรือกำหนดในรูป Cosine ของมุม
  – ทั้งสองอันนี้จะเกี่ยวข้องกับ Unit Vector



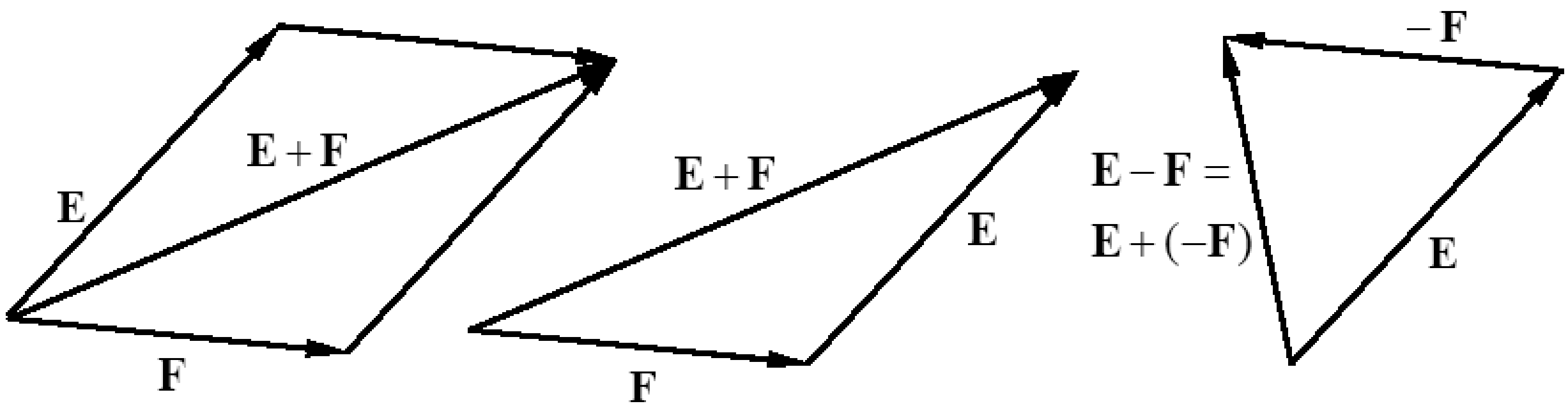
Vector Operations

- เนื่องจาก Vector ประกอบด้วยทั้งขนาดและทิศทาง
  - พีชคณิต เช่น บวก ลบ คูณ หาร จะไม่เหมือนกับ Scalar เนื่องจากต้องนำทิศทางมาประกอบการคำนวณด้วย
  - การ บวก-ลบ ของ Vector จะได้ Vector ใหม่ที่ขนาดและทิศทางต่างจากเดิม
  - การคูณ เราจะไม่ใช้คำว่า ‘Multiplication’ แต่จะใช้คำว่า ‘Product’ แบ่งเป็นสองประเภท
    - Scalar Product (Dot Product; ●) จะได้ Scalar
    - Vector Product (Cross Product; X) จะได้ Vector ที่ตั้งฉากกับ Vector เดิมทั้งสอง

Addition and Substraction

1.2 การบวกและลบของ Vector: Triangle Law of Addition

การบวกกันของ Vector สามารถกระทำได้โดยใช้ Parallelogram law of addition (รูปแรกข้างล่าง) นอกจากนี้เรายังสามารถใช้ Triangle Law of Addition (รูปกลางข้างล่าง) สังเกตว่า การบวกจะเป็น Commutative กล่าวคือ $\mathbf{E}+\mathbf{F}=\mathbf{F}+\mathbf{E}$ ส่วนการลบกันกระทำโดย $\mathbf{E}-\mathbf{F}=\mathbf{E}+(-\mathbf{F})$ (รูปขวา ข้างล่าง)

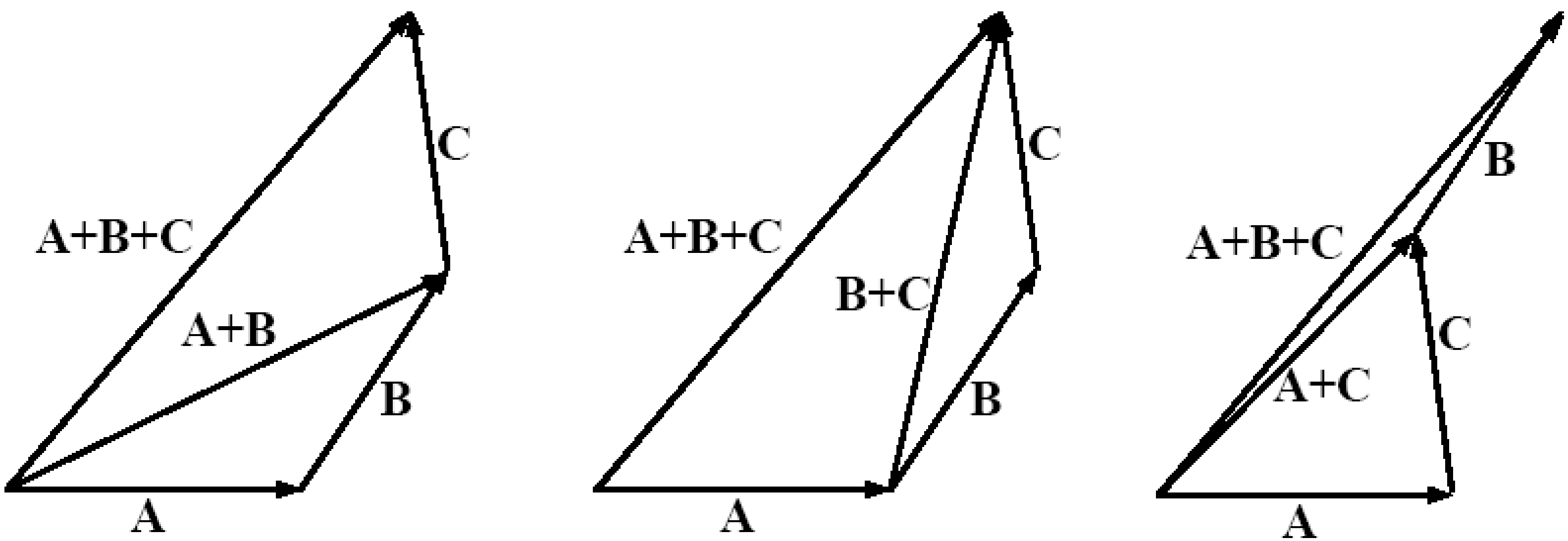


รูปที่ 1.4 แสดงการบวกและลบของ Vector

นอกจากนี้แล้วการบวก-ลบกันยังเป็น Associative ด้วย กล่าวคือ $(\mathbf{A}+\mathbf{B})+\mathbf{C}=\mathbf{A}+(\mathbf{B}+\mathbf{C})$ และเมื่อรวมกับ Commutative Law จะเท่ากับ $(\mathbf{A}+\mathbf{C})+\mathbf{B}$

เราสามารถบวก Vector มากกว่าสองตัว ซึ่งจากหลักการที่กล่าวข้างต้น เราสามารถทำได้โดยการบวกทีละสองตัวไหนก่อนก็ได้ ผลลัพธ์จะได้เหมือนกัน รูปข้างล่างแสดงการบวก Vector $\mathbf{A}+\mathbf{B}+\mathbf{C}$ โดยแสดงวิธีทำ 3 วิธี ตามลำดับจากซ้ายไปขวาคือ $(\mathbf{A}+\mathbf{B})+\mathbf{C}$, $\mathbf{A}+(\mathbf{B}+\mathbf{C})$ และ $(\mathbf{A}+\mathbf{C})+\mathbf{B}$

ถ้าให้ α และ β เป็น Scalar เราจะได้ $\alpha(\mathbf{A}+\mathbf{B})=\alpha\mathbf{A}+\alpha\mathbf{B}$ และ $(\alpha+\beta)\mathbf{A}=\alpha\mathbf{A}+\beta\mathbf{A}$ นอกจากนี้แล้วการบวกและลบ Vector ยังเป็นไปตามกฎของ Algebra ทั่วๆไป และเราสามารถแก้สมการ Linear ของ Vector โดยใช้วิธีการเดียวกันกับการแก้สมการ Linear Algebra

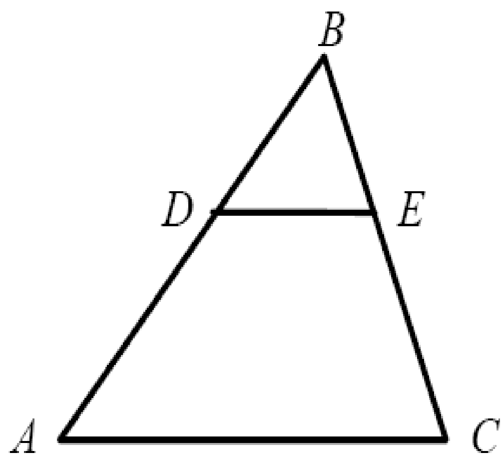


รูปที่ 1.5 แสดงการบวกของ Vector สามตัว

ในการลบ Vector ที่เท่ากัน เราจะได้ Null Vector หรือ Zero Vector ซึ่งเป็น Vector ที่มีขนาดเป็น 0 และไม่มีทิศทาง

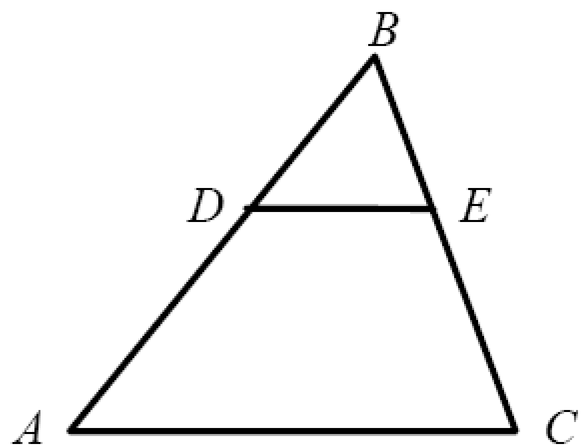
การประยุกต์ใช้ใน Plane Geometry

Example 1.1 จากสามเหลี่ยม ABC และ DBE ดังรูปที่ 1.6 ข้างล่าง ถ้ากำหนดให้ $DB=(2/5)AB$, และ $BE=(2/5)BC$ จงพิสูจน์ว่า $DE \,//\, AC$ และ $DE=(2/5)AC$ กล่าวคือ ABC และ DBE เป็นสามเหลี่ยมเสมือน (Similar Triangle)



รูปที่ 1.6 สามเหลี่ยมสำหรับคำถามตัวอย่างที่ 1.1

Example 1.1 จากสามเหลี่ยม ABC และ DBE ดังรูปที่ 1.6 ข้างล่าง ถ้ากำหนดให้ $DB=(2/5)AB$, และ $BE=(2/5)BC$ จงพิสูจน์ว่า $DE\,//\,AC$ และ $DE=(2/5)AC$ กล่าวคือ ABC และ DBE เป็นสามเหลี่ยมเสมือน (Similar Triangle)



รูปที่ 1.6 สามเหลี่ยมสำหรับคำถามตัวอย่างที่ 1.1

คำตอบ เราแสดงแต่ละด้านของสามเหลี่ยมโดยใช้ Vector ซึ่งโจทย์กำหนด $\overrightarrow{DB}=\frac{2}{5}\overrightarrow{AB}$ และ $\overrightarrow{BE}=\frac{2}{5}\overrightarrow{BC}$ ซึ่งการจะแสดงว่า $DE\,//\,AC$ และ $DE=(2/5)AC$ จะเท่ากับแสดงว่า $\overrightarrow{DE}=\frac{2}{5}\overrightarrow{AC}$ นั่นเอง

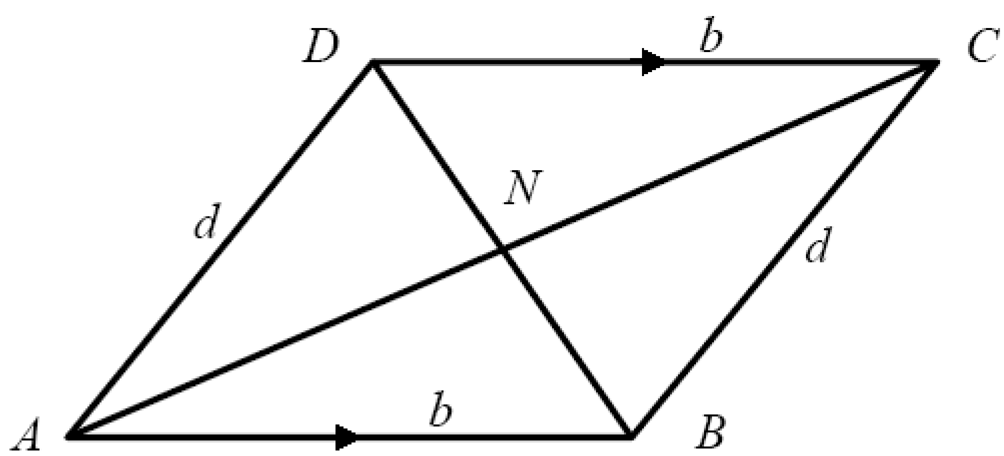
จาก Vector เราได้ $\overrightarrow{AB}+\overrightarrow{BC}=\overrightarrow{AC}$ และจากสามเหลี่ยรูปเล็ก $\overrightarrow{DB}+\overrightarrow{BE}=\overrightarrow{DE}$

ดังนั้นจาก

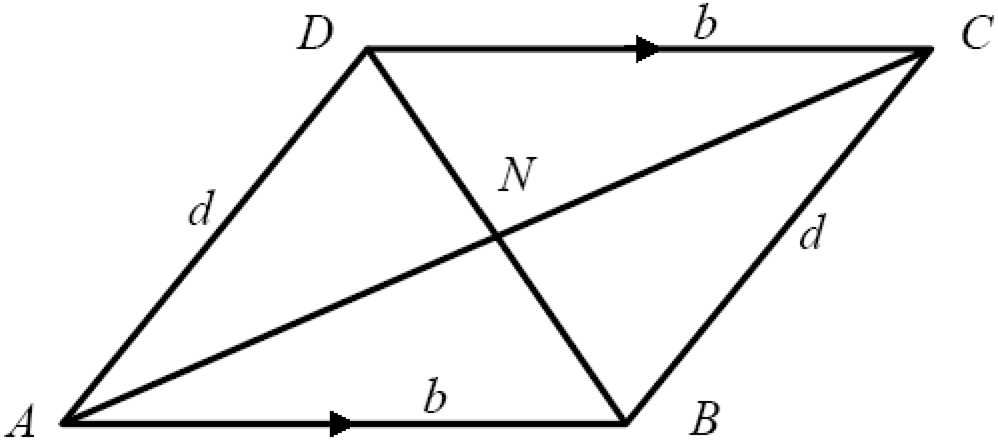
$$\overrightarrow{DE}=\overrightarrow{DB}+\overrightarrow{BE}=\frac{2}{5}\overrightarrow{AB}+\frac{2}{5}\overrightarrow{BC}=\frac{2}{5}(\overrightarrow{AB}+\overrightarrow{BC})=\frac{2}{5}\overrightarrow{AC}$$

Example 1.2 จากสี่เหลี่ยมขนมเปียกปูน (Parallelogram) จงพิสูจน์ว่าเส้นทะแยงมุมจะแบ่งครึ่งซึ่งกันและกัน

คำตอบ พิจารณา Parallelogram ดังรูปที่ 1.7 ข้างล่าง ซึ่งเราจะเขียนแต่ละด้านในรูป Vector



รูปที่ 1.7 Parallelogram



รูปที่ 1.7 Parallelogram

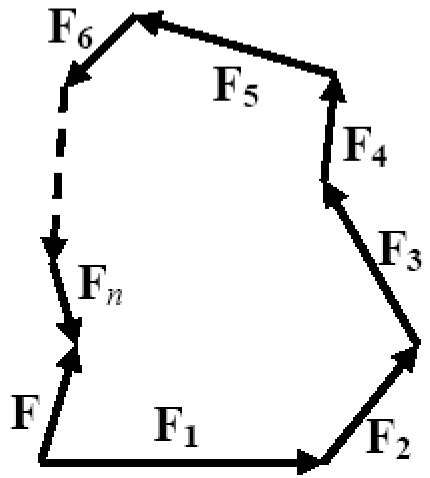
ให้ $\overrightarrow{AB} = \mathbf{b}$ และ $\overrightarrow{AD} = \mathbf{d}$ เราต้องการจะแสดงให้เห็นว่า $\overrightarrow{BN} = \frac{1}{2}\overrightarrow{BD}$ และ $\overrightarrow{AN} = \frac{1}{2}\overrightarrow{AC}$

จากรูป เราได้ $\overrightarrow{AC} = \mathbf{b} + \mathbf{d}$ และ $\overrightarrow{BD} = \mathbf{d} - \mathbf{b}$ นอกจากนี้ $\overrightarrow{BN} = \alpha\overrightarrow{BD}$ และ $\overrightarrow{AN} = \beta\overrightarrow{AC}$ โดยที่ α และ β เป็นค่า Constant ดังนั้นเราสามารถเขียน

$\overrightarrow{BN} = \alpha\overrightarrow{BD} = \alpha(\mathbf{d} - \mathbf{b}) = \alpha\,\mathbf{d} - \alpha\,\mathbf{b}$ และ $\overrightarrow{AN} = \beta\overrightarrow{AC} = \beta(\mathbf{b} + \mathbf{d}) = \beta\mathbf{b} + \beta\mathbf{d}$

จาก $\overrightarrow{AB} + \overrightarrow{BN} = \overrightarrow{AN}$ เราได้ $\mathbf{b} + \alpha\,\mathbf{d} - \alpha\,\mathbf{b} = \beta\,\mathbf{b} + \beta\,\mathbf{d}$ เมื่อจับคู่ที่เหมือนกัน เราได้ $\alpha = \beta$ และ $1 - \alpha = \beta$ เมื่อแก้สมการ เราจะได้ $\alpha = \beta = \frac{1}{2}$

Component Vector

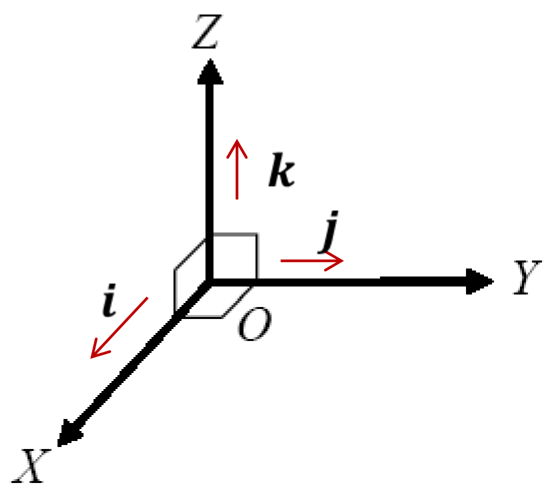


รูปที่ 1.11 Component Vectors ของ F

Component Vector in Cartesian Coordinate

1.4 Vector in Cartesian Coordinate

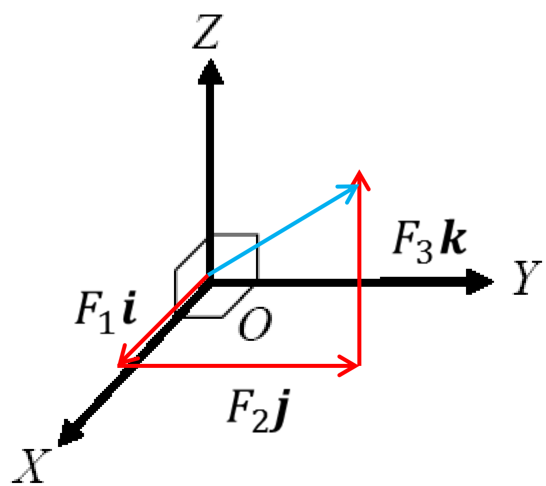
ในกรณีของ Vector ใน $\mathbf{R}^3$ กล่าวคือใน 3-Dimension เราใช้ Right-Handed System ในการเขียน Coordinate ของแกน x, y, z กล่าวคือเป็นลักษณะตามเข็มนาฬิกา โดยที่แต่ละแกนมี Unit Vector เป็น $\mathbf{i}, \mathbf{j}, \mathbf{k}$



รูปที่ 1.12 Cartesian Coordinate

Component Vector in Cartesian Coordinate

- พิจารณา Vector F ใน R^3 หรือ 3 มิติ ประกอบด้วย F_1, F_2, F_3 เราสามารถเขียน
  - $\boldsymbol{F} = F_1\boldsymbol{i} + F_2\boldsymbol{j} + F_3\boldsymbol{k}$

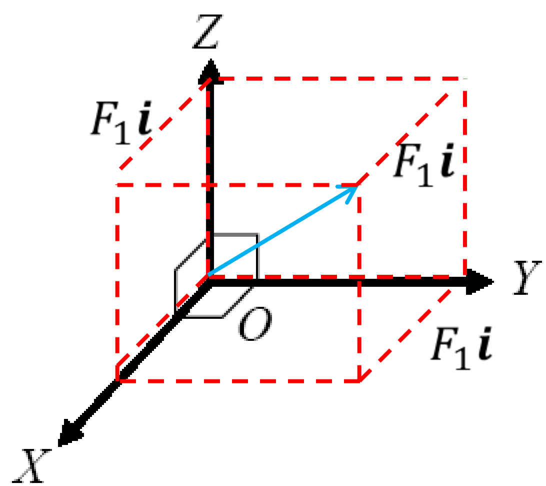


รูปที่ 1.12 Cartesian Coordinate

Component Vector in Cartesian Coordinate

- $\boldsymbol{F} = F_1\boldsymbol{i} + F_2\boldsymbol{j} + F_3\boldsymbol{k}$

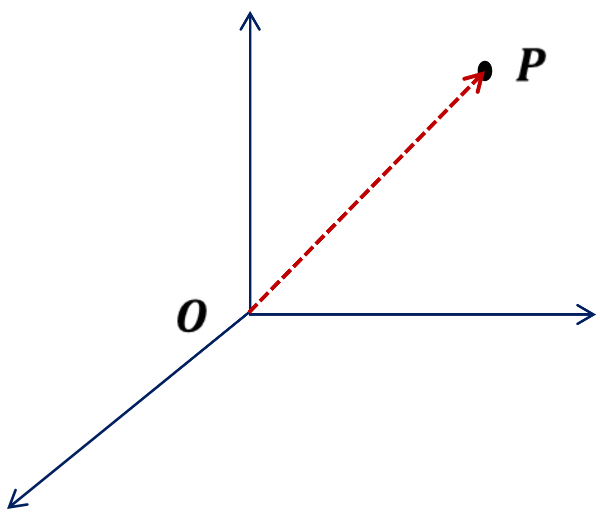
- **กล่าวคือ F มีส่วนประกอบบนสามแกน**
- **มาตรฐานการเขียน Component Vector**



รูปที่ 1.12 Cartesian Coordinate

Position Vector

- จุดใน Space สามารถแสดงได้โดยใช้ Vector เริ่มจาก Origin
  - อาจเรียก Location Vector หรือ Radius Vector
  - จุด P แสดงได้โดยใช้ Vector OP
  - และสามารถแสดงได้โดยใช้ Component Vector



Position Vector และ Addition-Subtraction using Component Vector

$$\mathbf{A} = A_1 i + A_2 j + A_3 k\,,\mathbf{B} = B_1 i + B_2 j + B_3 k \;\; \mathbf{A} \pm \mathbf{B} = (A_1 \pm B_1) i + (A_2 \pm B_2) j + (A_3 \pm B_3)$$

สังเกตว่า A_n, B_n เป็นค่า Scalar กล่าวคือเป็น Magnitude ของ Component สำหรับแต่ละแกน โดยที่จุด Origin ของ Coordinate อยู่ที่จุดเริ่มต้นของ Vector ในกรณีนี้เราเรียก Vector ที่เริ่มจากจุดตั้งต้น(หรืออ้างอิงจาก Origin) ว่าเป็น Position Vector

Position Vector: $\mathbf{r} =$ position vector of $\mathbf{P}(x, y, z)$ $\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}\,,\ |\mathbf{r}| = \sqrt{x^2 + y^2 + z^2}$

ถ้า $\mathbf{A} = A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k}$ และ $\mathbf{B} = B_1\mathbf{i} + B_2\mathbf{j} + B_3\mathbf{k}$ เราได้

$$\mathbf{A} \pm \mathbf{B} = (A_1 \pm B_1)\mathbf{i} + (A_2 \pm B_2)\mathbf{j} + (A_3 \pm B_3)\mathbf{k}$$

Distance Between

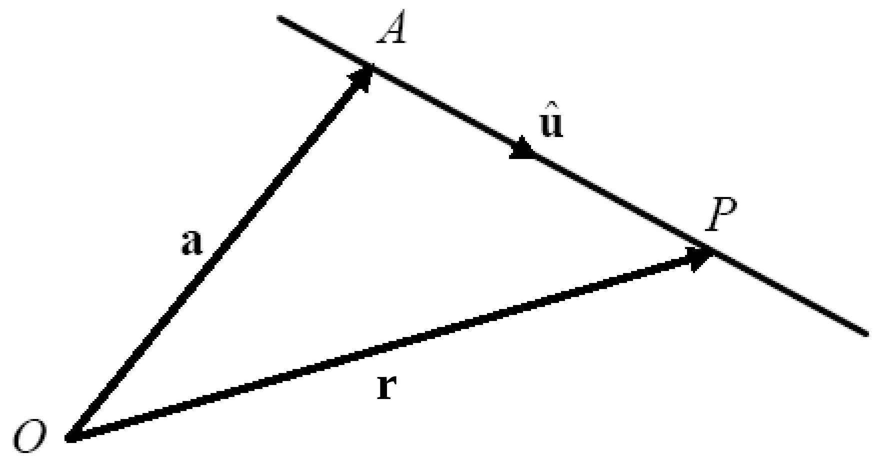
$$\mathbf{P}_1(x_1, y_1, z_1), \mathbf{P}_2(x_2, y_2, z_2), d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

สรุป

- การเขียน Vector ในลักษณะ Component จะสามารถบวกและลบกันได้ง่าย โดยการบวกลบแต่ละ Component บนแกนเดียวกัน
  - Vector Product สามารถคำนวณได้เช่นกัน
- จุดใน Space สามารถแทนด้วย Vector เริ่มจากจุด Origin เรียก Position Vector
  - Vector ที่เกิดจากสองจุดใน Space สามารถคำนวณได้จาก Position Vector นี้

Example 1.3 การหาสมการเส้นตรง เมื่อกำหนดจุดบนเส้นตรงและทิศทางของเส้นตรงนั้น

คำตอบ พิจารณาจากรูปที่ 1.8 สมมุติว่าจุด O เป็น Origin และจุด A เป็นจุดบนเส้นตรงที่ต้องการ โดยให้ $\overrightarrow{OA} = \mathbf{a}$ ทิศทางของเส้นตรงถูกกำหนดโดย Unit Vector $\hat{\mathbf{u}}$ เราสามารถหาสมการเส้นตรงได้ดังนี้

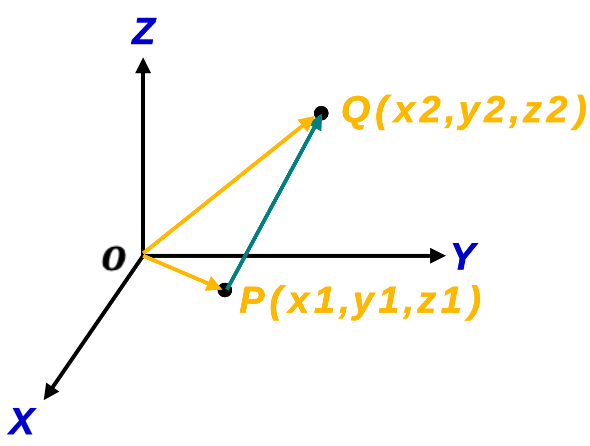


รูปที่ 1.8 สมการเส้นตรงเมื่อกำหนดจุดและทิศทาง

กำหนดจุด P ที่จะวิ่งไปตามเส้นตรง เกิด Vector $\overrightarrow{OP} = \mathbf{r}$ ซึ่งจะเปลี่ยนไปเมื่อ P เปลี่ยนตำแหน่ง และจะเท่ากับ $\mathbf{a}$ ถ้า P มาอยู่ที่จุด A ดังนั้นถ้าให้ t เป็นค่า Scalar มีค่าเท่ากับขนาด $\overrightarrow{AP}$ เราได้ $\overrightarrow{AP} = t\hat{\mathbf{u}}$ และจาก $\overrightarrow{OP} = \overrightarrow{OA} + \overrightarrow{AP}$ เราได้ $\mathbf{r} = \mathbf{a} + t\hat{\mathbf{u}}$ และเป็นสมการเส้นตรงที่ต้องการ ซึ่งแสดงความสัมพันธ์ระหว่าง t และ $\mathbf{r}$

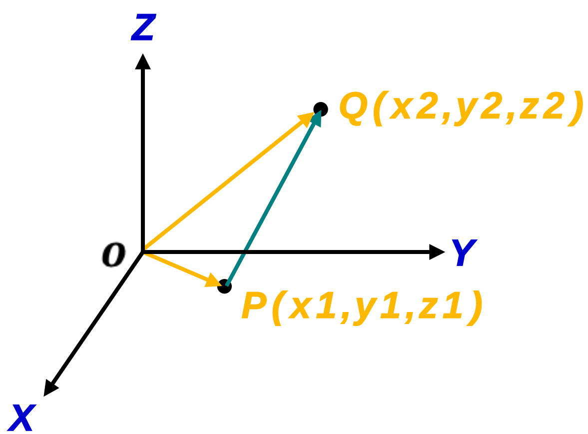
Any vectors in Cartesian Coordinates

- Given 2 Points, P(x1,y1,z1) and Q(x2,y2,z2)
  - We have $\overrightarrow{OP}+\overrightarrow{PQ}=\overrightarrow{OQ}$
  - Then $\overrightarrow{PQ} = \overrightarrow{OQ} - \overrightarrow{OP}$
    - $\overrightarrow{PQ} = x2\mathbf{i}+y2\mathbf{j}+z2\mathbf{k} - x1\mathbf{i}+y1\mathbf{j}+z1\mathbf{k}$
    - $\overrightarrow{PQ} = (x2-x1)\mathbf{i}+(y2-y1)\mathbf{j}+(z2-z1)\mathbf{k}$



Any vectors in Cartesian Coordinates

- Given 2 Points, P(x1,y1,z1) and Q(x2,y2,z2)
  - PQ =(x2-x1)**i**+(y2-y1)**j**+(z2-z1)**k**
  - Also magnitude or length of vector is the distance between those 2 points (Euclidian Distance)
    - PQ = $\sqrt{(x2-x1)^2+(y2-y1)^2+(z2-z1)^2}$



Direction Cosine/Ratio

- Vector สามารถเขียนเป็นสองส่วนประกอบ

$$\mathbf{A} = |\mathbf{A}|\hat{\mathbf{a}}$$

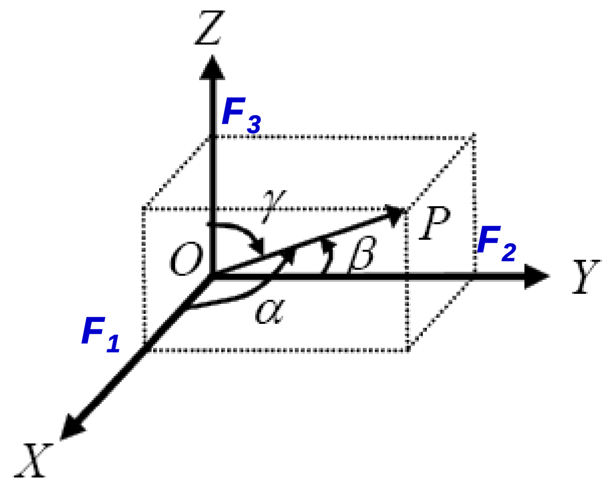
  - ขนาด สามารถหาได้ง่าย กรณี Position Vector
  - ทิศทาง คือ Unit Vector ที่มีทิศทางเดียวกันกับ Vector นั้น
    - ทิศทาง สามารถแตกเป็น Component Vector บนแต่ละแกนได้ด้วย
    - ทิศทางสามารถกำหนดด้วยมุมที่ทำกับแต่ละแกนได้ด้วย
    - ทั้งสองแบบนี้ สัมพันธ์กันทางตรีโกณมิติ โดยการกำหนดด้วยค่า Cosine ของมุม เรียก Direction Cosine

Direction Cosine

- Position vector OP
  - Magnitude equal to OP = $\sqrt{x^2+y^2+z^2}$
  - Direction: $\cos\alpha\,\mathbf{i}+\cos\beta\,\mathbf{j}+\cos\gamma\,\mathbf{k}$
    - **Called Direction Cosine**

$$\mathbf{F}=\overrightarrow{OP}=F_1\mathbf{i}+F_2\mathbf{j}+F_3\mathbf{k}=OP\cos\alpha\mathbf{i}+OP\cos\beta\mathbf{j}+OP\cos\gamma\mathbf{k}=OP(\cos\alpha\mathbf{i}+\cos\beta\mathbf{j}+\cos\gamma\mathbf{k})$$

We have
$\cos\alpha = F_1/OP$
$\cos\beta = F_2/OP$
$\cos\gamma = F_3/OP$



Direction Cosine and Direction Ratio

สมมุติ Vector $\mathbf{F} = \overrightarrow{OP} = F_1\mathbf{i} + F_2\mathbf{j} + F_3\mathbf{k}$ ทิศทางของ Vector สามารถกำหนดได้จาก Direction ของทั้งสาม Component ในแกน x, y, และ z นอกจากนี้ยังสามารถกำหนดได้จากมุมที่ $\overrightarrow{OP}$ กระทำกับ OX, OY, OZ สมมุติว่าเป็นมุม α, β, γ ตามลำดับ ดังนั้นจาก

$$\mathbf{F} = \overrightarrow{OP} = F_1\mathbf{i} + F_2\mathbf{j} + F_3\mathbf{k} = OP\cos\alpha\mathbf{i} + OP\cos\beta\mathbf{j} + OP\cos\gamma\mathbf{k} = OP(\cos\alpha\mathbf{i} + \cos\beta\mathbf{j} + \cos\gamma\mathbf{k})$$

ค่า $\cos\alpha, \cos\beta, \cos\gamma$ เรียก Direction Cosine ของ Vector $\mathbf{F}$ สังเกตว่า $\cos\alpha\mathbf{i} + \cos\beta\mathbf{j} + \cos\gamma\mathbf{k}$ จะต้องเป็น Unit Vector ดังนั้นเราได้ $\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1$ นอกจากนั้นแล้ว $\cos\alpha = F_1/F$, $\cos\beta = F_2/F$, และ $\cos\gamma = F_3/F$

Direction Cosine and Direction Ratio

บางครั้งเราต้องการเขียนให้อยู่ในรูปอัตราส่วนของจำนวนเต็มมากกว่ารูป Cosine ในกรณีนี้เราเขียนในรูปของตัวแปร l,m,n สามตัวที่ทำให้

$$\frac{l}{\cos\alpha}=\frac{m}{\cos\beta}=\frac{n}{\cos\gamma}$$

ค่านี้เราเรียก Direction Ratio ในความเป็นจริงแล้ว $[F_1,F_2,F_3]$ ที่จริงก็คือ Direction Ratio ของ $\mathbf{F}$

Example

- Given points $P_1(2,-4,5)$ and $P_2(1,3,-2)$, find the vector P_1P_2 and its magnitude and direction
  - $OP_1 = 2\mathbf{i}-4\mathbf{j}+5\mathbf{k}$ and $OP_2 = \mathbf{i}+3\mathbf{j}-2\mathbf{k}$
  - $P_1P_2=OP_2-OP_1=-\mathbf{i}+7\mathbf{j}-7\mathbf{k}$
  - $P_1P_2 = \sqrt{1+49+49}=\sqrt{99}$
  - $\text{Cos } \alpha = -1/\sqrt{99}$ then $\alpha = 95.8$ degree
  - $\text{Cos } \beta = 7/\sqrt{99}$ then $\beta = 45.3$ degree
  - $\text{Cos } \gamma = -7/\sqrt{99}$ then $\gamma = 134.7$ degree

Direction Cosine and Direction Ratio

การเปลี่ยน Direction Ratio เป็น Direction Cosine ทำได้โดยให้

$$\frac{l}{\cos\alpha} = \frac{m}{\cos\beta} = \frac{n}{\cos\gamma} = \lambda$$

ดังนั้น $\cos\alpha = l/\lambda, \cos\beta = m/\lambda, \cos\lambda = n/\lambda$ แต่เนื่องจาก $\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1$ เราได้

$l^2 + m^2 + n^2 = \lambda^2$ และ Direction Cosine จะเป็น

$$\frac{l}{\sqrt{l^2+m^2+n^2}}, \frac{m}{\sqrt{l^2+m^2+n^2}}, \frac{n}{\sqrt{l^2+m^2+n^2}}$$

EX. จงหามุมที่กระทำกับ Coordinate สำหรับเส้นที่ลากจากจุด $P(1,1,1)$ และ $Q(3,-2,4)$

Next Week

- Vector Product
  - – Scalar Product(Dot)
  - – Vector Product(Cross)
- Chapter II: MATRICES
- HW II

CPE 332
Computer Engineering Mathematics II

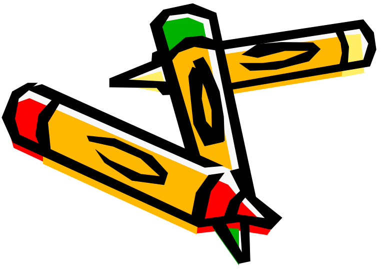
Week 2

Chapter 1 Vector (cont.)

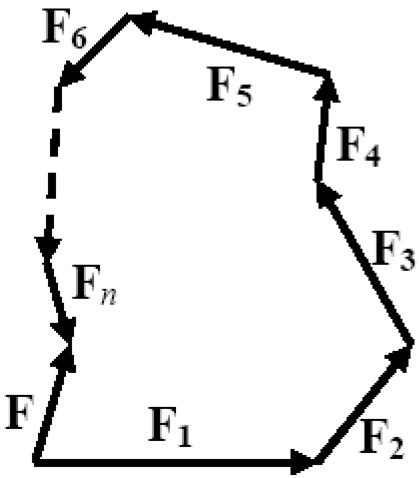
Chapter 2 Matrix

Today Topics

- Chapter 1 Cont.
- Break
- Chapter 2: Matrix
- Download Homework 1: Chapter 1
  - Due Next Week



Component Vector

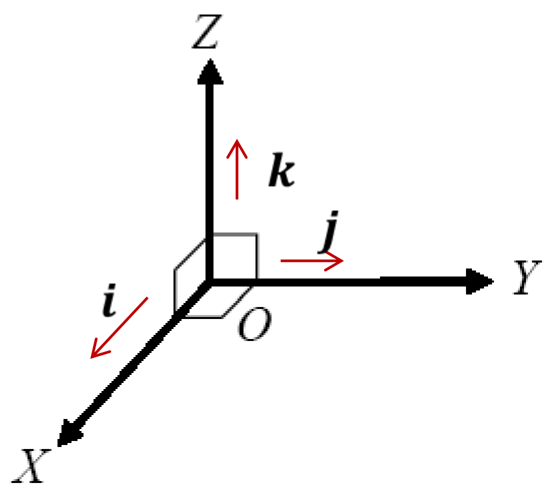


รูปที่ 1.11 Component Vectors ของ F

Component Vector in Cartesian Coordinate

1.4 Vector in Cartesian Coordinate

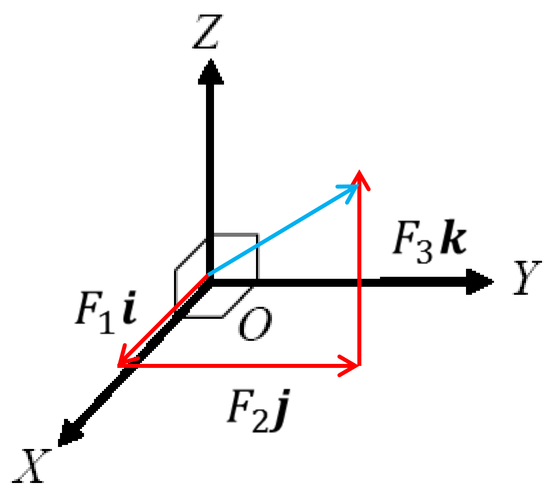
ในกรณีของ Vector ใน $\mathbf{R}^3$ กล่าวคือใน 3-Dimension เราใช้ Right-Handed System ในการเขียน Coordinate ของแกน x, y, z กล่าวคือเป็นลักษณะตามเข็มนาฬิกา โดยที่แต่ละแกนมี Unit Vector เป็น $\mathbf{i}, \mathbf{j}, \mathbf{k}$



รูปที่ 1.12 Cartesian Coordinate

Component Vector in Cartesian Coordinate

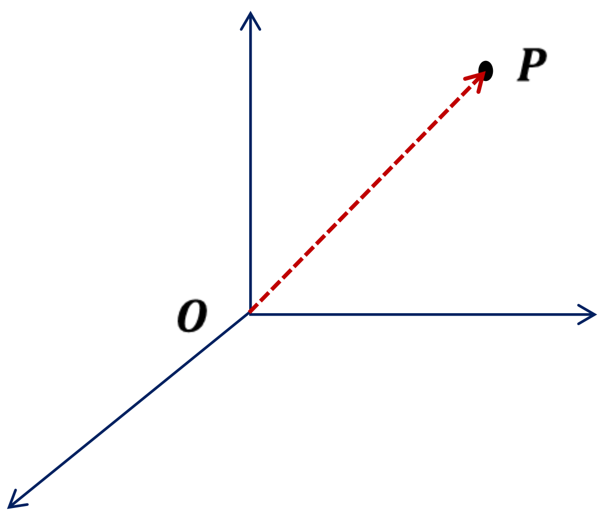
- พิจารณา Vector F ใน R^3 หรือ 3 มิติ ประกอบด้วย F_1,F_2,F_3 เราสามารถเขียน
  - $\boldsymbol{F} = F_1\boldsymbol{i} + F_2\boldsymbol{j} + F_3\boldsymbol{k}$



รูปที่ 1.12 Cartesian Coordinate

Position Vector

- จุดใน Space สามารถแสดงได้โดยใช้ Vector เริ่มจาก Origin
  - อาจเรียก Location Vector หรือ Radius Vector
  - จุด P แสดงได้โดยใช้ Vector OP
  - และสามารถแสดงได้โดยใช้ Component Vector



Position Vector และ Addition-Subtraction using Component Vector

$$\mathbf{A} = A_1 i + A_2 j + A_3 k \,, \mathbf{B} = B_1 i + B_2 j + B_3 k \quad \mathbf{A} \pm \mathbf{B} = (A_1 \pm B_1) i + (A_2 \pm B_2) j + (A_3 \pm B_3)$$

สังเกตว่า A_n, B_n เป็นค่า Scalar กล่าวคือเป็น Magnitude ของ Component สำหรับแต่ละแกน โดยที่จุด Origin ของ Coordinate อยู่ที่จุดเริ่มต้นของ Vector ในกรณีนี้เราเรียก Vector ที่เริ่มจากจุดตั้งต้น(หรืออ้างอิงจาก Origin) ว่าเป็น Position Vector

Position Vector: $\mathbf{r} =$ position vector of $\mathbf{P}(x, y, z)$ $\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$, $|\mathbf{r}| = \sqrt{x^2 + y^2 + z^2}$

ถ้า $\mathbf{A} = A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k}$ และ $\mathbf{B} = B_1\mathbf{i} + B_2\mathbf{j} + B_3\mathbf{k}$ เราได้

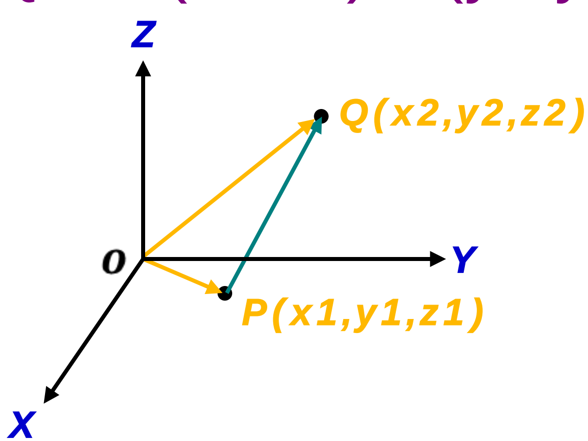
$$\mathbf{A} \pm \mathbf{B} = (A_1 \pm B_1)\mathbf{i} + (A_2 \pm B_2)\mathbf{j} + (A_3 \pm B_3)\mathbf{k}$$

Distance Between

$$\mathbf{P}_1(x_1, y_1, z_1), \mathbf{P}_2(x_2, y_2, z_2), d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Any vectors in Cartesian Coordinates

- Given 2 Points, P(x1,y1,z1) and Q(x2,y2,z2)
  - $\overline{PQ} = (x2\text{-}x1)\mathbf{i} + (y2\text{-}y1)\mathbf{j} + (z2\text{-}z1)\mathbf{k}$
  - Also magnitude or length of vector is the distance between those 2 points (Euclidian Distance)
    - $PQ = \sqrt{(x2\text{-}x1)^2 + (y2\text{-}y1)^2 + (z2\text{-}z1)^2}$



Direction Cosine/Ratio

- Vector สามารถเขียนเป็นสองส่วนประกอบ

$$\mathbf{A} = \left|\mathbf{A}\right|\hat{\mathbf{a}}$$

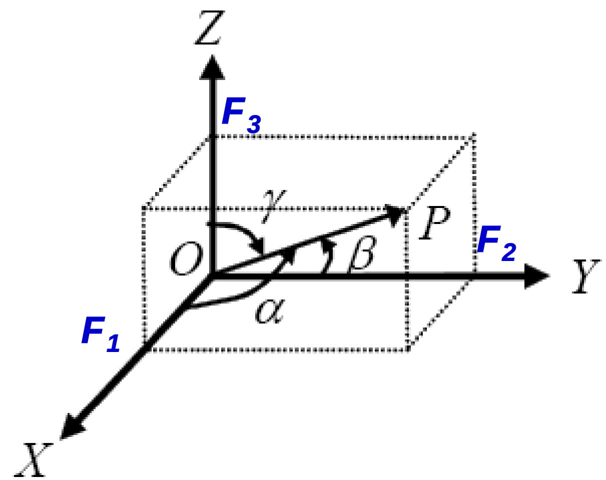
  - ขนาด สามารถหาได้ง่าย กรณี Position Vector
  - ทิศทาง คือ Unit Vector ที่มีทิศทางเดียวกันกับ Vector นั้น
    - ทิศทาง สามารถแตกเป็น Component Vector บนแต่ละแกนได้ด้วย
    - ทิศทางสามารถกำหนดด้วยมุมที่ทำกับแต่ละแกนได้ด้วย
    - ทั้งสองแบบนี้ สัมพันธ์กันทางตรีโกณมิติ โดยการกำหนดด้วยค่า Cosine ของมุม เรียก Direction Cosine

Direction Cosine

- Position vector OP
  - Magnitude equal to OP = $\sqrt{x^2+y^2+z^2}$
  - Direction: $\cos\alpha\,\mathbf{i}+\cos\beta\,\mathbf{j}+\cos\gamma\,\mathbf{k}$
    - **Called Direction Cosine**

$$\mathbf{F}=\overrightarrow{OP}=F_1\mathbf{i}+F_2\mathbf{j}+F_3\mathbf{k}=OP\cos\alpha\mathbf{i}+OP\cos\beta\mathbf{j}+OP\cos\gamma\mathbf{k}=OP(\cos\alpha\mathbf{i}+\cos\beta\mathbf{j}+\cos\gamma\mathbf{k})$$

We have
$\cos\alpha=F_1/OP$
$\cos\beta=F_2/OP$
$\cos\gamma=F_3/OP$



Products of Vectors

- Vector Product
  - – Scalar Product(DOT)
  - – Vector Product(Cross)

Scalar Product(DOT)

Scalar Product: $\mathbf{A} \bullet \mathbf{B} = AB\cos\theta$, จะเป็น Scalar Quantity จะเป็น Commutative $\mathbf{A} \bullet \mathbf{B} = \mathbf{B} \bullet \mathbf{A}$

Scalar Product (DOT)

คุณสมบัติ

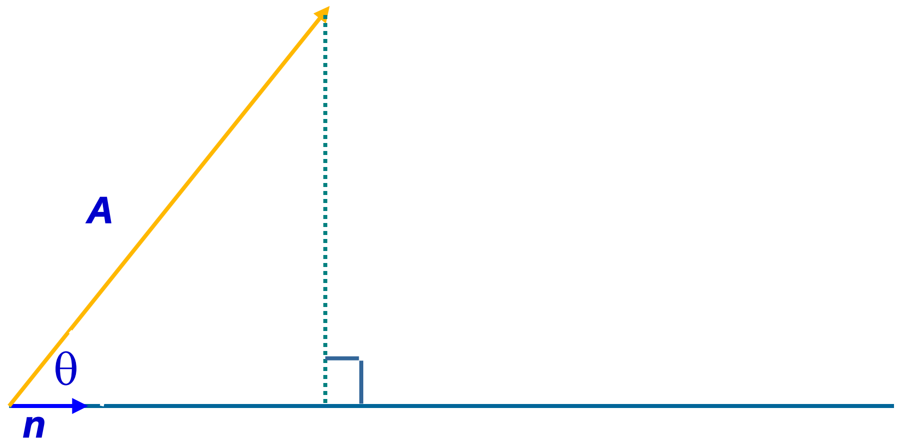
1. ถ้าสอง Vector ตั้งฉากกัน, Perpendicular: $\theta = 90^o$, $\mathbf{A} \bullet \mathbf{B} = 0$
2. ถ้าสอง Vector ขนานกัน, $\mathbf{A} \bullet \mathbf{B} = AB$, กรณีพิเศษ $\mathbf{A} \bullet \mathbf{A} = \mathbf{A}^2 = |\mathbf{A}|^2 = A^2$ = Sq. Magnitude
3. $\mathbf{A} \bullet \mathbf{B} = A(B\cos\theta) = (A\cos\theta)B$ = A คูณ Component ของ $\mathbf{B}$ ในทิศทางของ $\mathbf{A}$ = B คูณ Component ของ $\mathbf{A}$ ในทิศทางของ $\mathbf{B}$ และถ้าให้ $\mathbf{n}$ เป็น Unit Vector ในทิศทางหนึ่ง เราจะได้ $\mathbf{A} \bullet \mathbf{n} = A\cos\theta$ เป็น Component ของ $\mathbf{A}$ ในทิศทางนั้น
4. $\mathbf{A} \bullet (\mathbf{B} + \mathbf{C}) = \mathbf{A} \bullet \mathbf{B} + \mathbf{A} \bullet \mathbf{C}$
5. $\mathbf{i} \bullet \mathbf{j} = \mathbf{j} \bullet \mathbf{k} = \mathbf{k} \bullet \mathbf{i} = \mathbf{0}$ และ $\mathbf{i} \bullet \mathbf{i} = \mathbf{j} \bullet \mathbf{j} = \mathbf{k} \bullet \mathbf{k} = \mathbf{1}$ ดังนั้นถ้า

   $\mathbf{A} = A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k}$ และ $\mathbf{B} = B_1\mathbf{i} + B_2\mathbf{j} + B_3\mathbf{k}$ เราได้

   $\mathbf{A} \bullet \mathbf{B} = A_1B_1 + A_2B_2 + A_3B_3$
6. $\cos\theta = \dfrac{\mathbf{A} \bullet \mathbf{B}}{|\mathbf{A}||\mathbf{B}|}$

Ex. จงหามุมระหว่างสอง Vector $\mathbf{A} = \mathbf{i} - \mathbf{j} - \mathbf{k}, \mathbf{B} = \mathbf{2i} + \mathbf{j} + \mathbf{2k}$

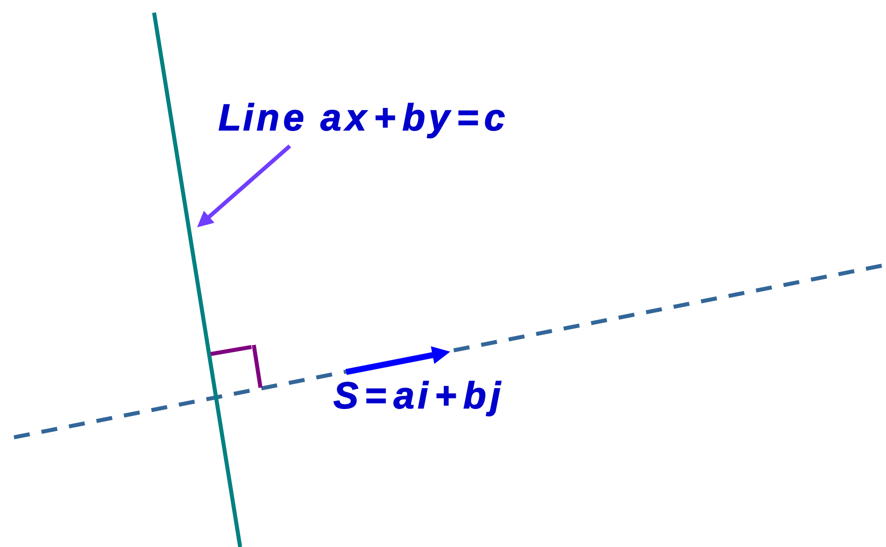
Scalar(Dot) Product



$$\mathbf{A} \bullet \mathbf{n} = A\cos\theta$$

Scalar(Dot) Product

- **A•(B+C)=A•B+A•C**
- Let **A** = a_1**i**+a_2**j**+a_3**k**, **B** = b_1**i**+b_2**j**+b_3**k**
  - We have **A•B =** $a_1b_1+a_2b_2+a_3b_3$
- Also $\cos\theta = \dfrac{\mathbf{A}\bullet\mathbf{B}}{|\mathbf{A}||\mathbf{B}|}$
- Given **S**=a**i**+b**j**, the equation of line perpendicular to this vector is in the form
  - **ax+by=c**



DOT Product

Geometric Application:

1. พิสูจน์ Pythagorus's Theorem $c^2 = a^2 + b^2$
2. พิสูจน์ Cosine Rule $c^2 = a^2 + b^2 - 2ab\cos C$
3. หาสมการเส้นตรงจาก $P(x, y)$ ตั้งฉากกับเส้นที่กำหนด $s = a\mathbf{i} + b\mathbf{j}$ และผ่านจุด $A'(x', y')$ เราได้สมการ $ax + by = c = ax' + by'$
4. มุมระหว่างสองเส้นที่ตัดกัน
5. ระยะทางจากจุด $P(x, y)$ ไปยังเส้นที่กำหนด $a\mathbf{i} + b\mathbf{j} = c$
6. สมการของ Plane โดยกำหนด Normal Direction **a** และ Plane ผ่านจุด $B(\overrightarrow{OB} = \mathbf{b})$ ถ้าให้ P เป็นจุดใดๆบน Plane และ $\overrightarrow{OP} = \mathbf{r}$ เราได้ BP อยู่บน Plane และตั้งฉากกับ **a** เราได้ $(\mathbf{r} - \mathbf{b}) \bullet \mathbf{a} = \mathbf{0}$ ดังนั้น

   ถ้า $\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k},\ \mathbf{b} = b_1\mathbf{i} + b_2\mathbf{j} + b_3\mathbf{k},\quad \mathbf{a} = a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}$ เราสามารถหาสมการของ Plane ได้เป็น $a_1x + a_2y + a_3z = a_1b_1 + a_2b_2 + a_3b_3 = c$ ซึ่งมี Coefficient เท่ากับ Component ใน **a**

Example

- Find the angle between the vector
  - **A**=**i**-**j**-**k** and **B** = 2**i**+**j**+2**k**

- We calculate **A•B** = 1.2-1.1-1.2=-1
- Also A = √(1+1+1)=√3
- Also B = √(4+1+4)=3
- Then Cos θ = -1/3√3
  - θ = 101.1 degrees

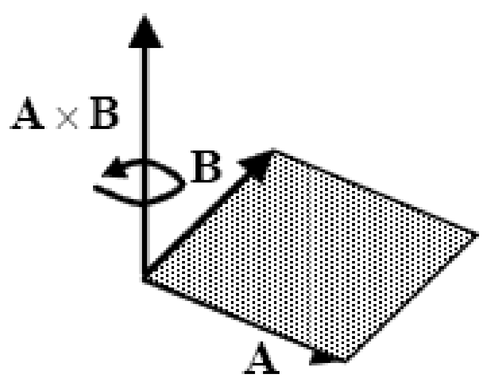
Vector Product (Cross)

Vector Product: $\mathbf{A} \times \mathbf{B}$,

(i) $\mathbf{A} \times \mathbf{B}$ จะตั้งฉากกับทั้ง $\mathbf{A}$ และ $\mathbf{B}$,

(ii) Magnitude: $\left|\mathbf{A} \times \mathbf{B}\right| = AB \sin\theta$,

(iii) มุมที่กระทำ และทิศทางของ Vector ที่ได้จะเป็นไปตาม Right Hand Rotation



Cross Product

คุณสมบัติของ Vector Product:

1. ถ้า $\hat{\mathbf{n}}$ เป็น unit vector ในทิศทาง $\mathbf{A}\times\mathbf{B}$ เราได้ $\mathbf{A}\times\mathbf{B} = AB\sin\theta\,\hat{\mathbf{n}}$ และ $\mathbf{B}\times\mathbf{A} = AB\sin\theta(-\hat{\mathbf{n}})$ ดังนั้นเราสรุปว่า $\mathbf{A}\times\mathbf{B} = -\mathbf{B}\times\mathbf{A}$
2. ถ้าทั้งสอง Vector ขนานกัน $\mathbf{A}\times\mathbf{B} = 0$ ถ้า Vector แต่ละตัวไม่เท่ากับศูนย์
3. ถ้าทั้งสอง Vector ตั้งฉากกัน $\mathbf{A}\times\mathbf{B} = AB\,\hat{\mathbf{n}}$
4. $\mathbf{i}\times\mathbf{i} = \mathbf{j}\times\mathbf{j} = \mathbf{k}\times\mathbf{k} = 0, \mathbf{i}\times\mathbf{j} = \mathbf{k}, \mathbf{j}\times\mathbf{k} = \mathbf{i}, \mathbf{k}\times\mathbf{i} = \mathbf{j}, \mathbf{j}\times\mathbf{i} = -\mathbf{k}, \ldots$
5. $\mathbf{A}\times(\mathbf{B}+\mathbf{C}) = \mathbf{A}\times\mathbf{B}+\mathbf{A}\times\mathbf{C}$
6. $\mathbf{A} = A_1\mathbf{i}+A_2\mathbf{j}+A_3\mathbf{k}$, $\mathbf{B} = B_1\mathbf{i}+B_2\mathbf{j}+B_3\mathbf{k}$ ดังนั้น

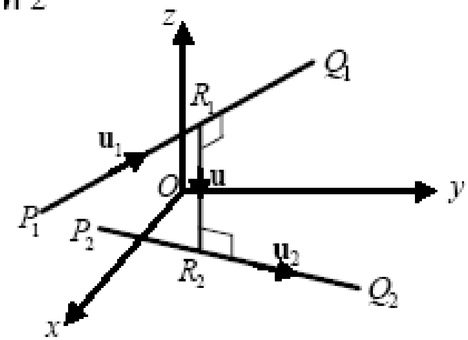
   $\mathbf{A}\times\mathbf{B} = (A_2B_3 - A_3B_2)\mathbf{i} - (A_1B_3 - A_3B_1)\mathbf{j} + (A_1B_2 - A_2B_1)\mathbf{k}$ เขียนอีกอย่างได้

$$\mathbf{A}\times\mathbf{B} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 \end{vmatrix}$$

7. จาก Magnitude $|\mathbf{A}\times\mathbf{B}| = AB\sin\theta$ ซึ่งก็คือพื้นที่ของ Parallelogram ที่มีด้านประกอบด้วยสอง Vector ดังกล่าว

Example 1.17 กำหนดเส้นตรง 2 เส้นในระนาบที่แตกต่างกัน หาระยะห่างระหว่างเส้นตรงทั้งสอง (หาความยาวของเส้นที่ลากตั้งฉากกับเส้นตรงทั้งสอง)

คำตอบ จากรูป 1.23 กำหนดเส้นตรงทั้งสองผ่านจุด P_1Q_1 และ P_2Q_2 และเส้นตั้งฉากกับเส้นตรงทั้งสองผ่านจุด R_1 บนเส้นตรงเส้นแรก และผ่านจุด R_2 บนเส้นตรงเส้นที่ 2



รูปที่ 1.23 ระยะห่างระหว่างสองเส้น

ให้ Unit Vector บนเส้น P_1Q_1, P_2Q_2 และ R_1R_2 เป็น $\mathbf{u}_1$, $\mathbf{u}_2$, และ $\mathbf{u}$ ตามลำดับ และให้ Position Vector ของ P_1 และ P_2 เป็น $\mathbf{r}_1$ และ $\mathbf{r}_2$ ตามลำดับ ดังนั้นเราได้

$$\overrightarrow{OR_1} = \mathbf{r}_1 + P_1R_1\mathbf{u}_1 = \mathbf{r}_2 + P_2R_2\mathbf{u}_2 - R_1R_2\mathbf{u}$$

ทำ Scalar Product กับ $\mathbf{u}$ ทั้งสมการ เราได้

$$\mathbf{r}_1 \bullet \mathbf{u} + P_1R_1\mathbf{u}_1 \bullet \mathbf{u} = \mathbf{r}_2 \bullet \mathbf{u} + P_2R_2\mathbf{u}_2 \bullet \mathbf{u} - R_1R_2\mathbf{u} \bullet \mathbf{u}$$

แต่ $\mathbf{u} \bullet \mathbf{u} = 1$, และ $\mathbf{u}_1 \bullet \mathbf{u} = \mathbf{u}_2 \bullet \mathbf{u} = 0$ ดังนั้น เมื่อเรียงสมการใหม่ จะได้

$$R_1R_2 = (\mathbf{r}_2 - \mathbf{r}_1) \bullet \mathbf{u}$$

เนื่องจาก $\mathbf{u}$ ตั้งฉากกับทั้ง $\mathbf{u}_1$ และ $\mathbf{u}_2$ ดังนั้นมันจะต้องอยู่ในทิศทางของ $\mathbf{u}_1 \times \mathbf{u}_2$ ดังนั้นเราได้

$$R_1R_2 = \left| \frac{(\mathbf{r}_2 - \mathbf{r}_1) \bullet (\mathbf{u}_1 \times \mathbf{u}_2)}{|\mathbf{u}_1 \times \mathbf{u}_2|} \right|$$

3 Vector Products

- $(\mathbf{B} \bullet \mathbf{C})\mathbf{A} = k\mathbf{A}$

- $\mathbf{A} \bullet (\mathbf{B} \times \mathbf{C}) = \mathbf{B} \bullet (\mathbf{C} \times \mathbf{A}) = \mathbf{C} \bullet (\mathbf{A} \times \mathbf{B})$ เท่ากับปริมาตร Parallelepiped และสามารถเขียนได้เป็น

$$\mathbf{A} \bullet (\mathbf{B} \times \mathbf{C}) = \begin{vmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \\ C_1 & C_2 & C_3 \end{vmatrix}$$

- $\mathbf{A} \times (\mathbf{B} \times \mathbf{C})$ เรียก Vector Triple Product เราสามารถพิสูจน์ได้ว่า $\mathbf{A} \times (\mathbf{B} \times \mathbf{C}) = (\mathbf{A} \bullet \mathbf{C})\mathbf{B} - (\mathbf{A} \bullet \mathbf{B})\mathbf{C}$

Examples

- Let A=2i+3j-k, B=i+j+2k
  - A●B = 2+3-2 = 3
  - A×B = (6+1)i-(4+1)j+(2-3)k=7i-5j-k
  - A×B is orthogonal to both A and B
    - Test : A●(A×B) = (2i+3j-k)●(7i-5j-k) = 14-15+1=0
    - Test : B●(A×B) = (i+j+2k)●(7i-5j-k) = 7-5-2=0

Plane Equation in 3D

- ใน 2D สมการเส้นตรงจะมี general Form
  - Ax+By=C
- ใน 3D สมการของ Plane จะมี General Form
  - Ax+By+Cz=D
  - D เป็นค่าคงที่ ทุกๆสมการในรูปเดียวกัน แต่ค่า D ต่างกัน จะเป็นระนาบที่ขนานกัน
    - 3x-2y+5z = 3 จะขนานกับ 3x-2y+5z = 6

Example 1

- กำหนดสมการของ Plane 2x+3y+2z=5 จงหา unit vector ที่ตั้งฉากกับ Plane นี้
  - กำหนด 3 จุด คือ A, B, C ดังนี้
  - A: x=0,y=0,ดังนั้น z=5/2 → A(0,0,2.5)
  - B: x=1,y=0, ดังนั้น z=(5-2)/2 → B(1,0,1.5)
  - C: x=0,y=1, ดังนั้น z=(5-3)/2 → C(0,1,1)
  - Vector $\overrightarrow{AB} \times \overrightarrow{AC}$ จะได้ Vector ที่ตั้งฉากกับ Plane

$$\overrightarrow{AB} = \mathbf{i} - \mathbf{k}, \quad \overrightarrow{AC} = \mathbf{j} - 1.5\mathbf{k}$$

$$\overrightarrow{AB} \times \overrightarrow{AC} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & 0 & -1 \\ 0 & 1 & -1.5 \end{vmatrix} = \mathbf{i} + 1.5\mathbf{j} + \mathbf{k}$$

$$\hat{\mathbf{u}} = \frac{1}{\sqrt{4.25}}\left[\mathbf{i} + 1.5\mathbf{j} + \mathbf{k}\right]$$

- สังเกตุว่าทุกๆ Vector ที่เป็น multiple ของ 2i+3j+2k จะตั้งฉากกับ Plane 2x+3y+2z=k เสมอ โดยที่ k เป็นค่าคงที่ใดๆ

Example 2

- จงหาสมการของ Plane ที่ตั้งฉากกับ Vector 3i-2j-k และกำหนดให้จุด (1,1,2) อยู่บน Plane นั้น
  - จากตัวอย่างก่อน เราได้สมการของ Plane เป็น 3x-2y-z= k
  - เราหาค่า k โดยแทนค่าจุด (1,1,2) ลงในสมการดังนี้ 3(1)-2(1)-(2)=-1=k
  - ดังนั้นสมการที่ต้องการจะเป็น 3x-2y-z+1=0

CPE 332

Computer Engineering Mathematic II

PART I: Linear Algebra

(Chapter 1-3)

Definition of Matrix

$n \times m$ Matrix $\mathbf{A}$ เราเรียก Matrix $\mathbf{A}$ มี orders $n \times m$ ซึ่งประกอบด้วย Array ของตัวเลข a_{ij} เขียนในรูป

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & & & \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

Row Matrix, Column Matrix

Row Matrix หรือ Row Vector มี 1 Row เช่น $\mathbf{B} = [b_1 \; b_2 \; \cdots \; b_n]$ เป็น Row Vector ขนาด $1 \times n$

Column Matrix หรือ Column vector มี 1 Column เช่น Column Vector ขนาด $n \times 1$

$$\mathbf{C} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

สังเกตว่าตัวแปรสำหรับ Matrix เรานิยมใช้ตัว Capital และ Boldface

Basic Operations

Addition&Substraction:

ทำได้สำหรับ Matrix ที่มี order เท่ากัน มีคุณสมบัติเป็น Commutative และ Associative คือ $\mathbf{A}+\mathbf{B}=\mathbf{B}+\mathbf{A}$ และ

$$\mathbf{A}+(\mathbf{B}+\mathbf{C})=(\mathbf{A}+\mathbf{B})+\mathbf{C}$$

Matrix Multiplication

Multiplication:

การคูณกันของ Matrix เราเขียน $\mathbf{A} \bullet \mathbf{B}$ (**A dot B**) หรือ $\mathbf{AB}$, และถ้า $\mathbf{C} = \mathbf{AB}$ เราได้

$$\mathbf{C}_{ij} = \sum_{l=1}^{n} a_{il} b_{lj}$$

การคูณจะต้องสอดคล้องกับ Inner Rule คือจำนวนของ column ของ $\mathbf{A}$ จะต้องเท่ากับจำนวนของ row ของ $\mathbf{B}$, ถ้า $\mathbf{A} = m \times n, \mathbf{B} = n \times p$ เราจะได้ $\mathbf{C} = m \times p$ Matrix การคูณของ Matrix จะไม่ Commutative $(\mathbf{AB} \neq \mathbf{BA})$ แต่จะ Associative และDistributive กล่าวคือ $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$ และ $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$

Square Matrix

<u>**Square Matrix/Matrix Power:**</u>

ถ้า Matrix มี order $\mathbf{A} = n \times n$ เราเรียก Matrix $\mathbf{A}$ ว่าเป็น square matrix, สังเกตว่ากำลังของ Matrix จะเป็นไปได้สำหรับ Square Matrix เท่านั้น เช่น $\mathbf{A} \bullet \mathbf{A} = \mathbf{A}^2, \mathbf{A} \bullet \mathbf{A} \bullet \mathbf{A} = \mathbf{A}^3, \ldots$

Matrix Transpose

<u>Transpose:</u>

Transpose เป็นการแลกเปลี่ยน column และ row ของ Matrix, ถ้า $\mathbf{A} = (a_{nm}) \rightarrow \mathbf{A}^T = (a_{mn})$ ค่าในวงเล็บแสดง Array ของ Elements ของ Matrix หรือเราเขียนได้เป็น

$$\mathbf{A}^T = \mathbf{A}' = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ \vdots & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

พึงสังเกตว่า $(\mathbf{A}+\mathbf{B})^{\mathbf{T}} = \mathbf{A}^{\mathbf{T}} + \mathbf{B}^{\mathbf{T}}$, $(\mathbf{AB})^{\mathbf{T}} = \mathbf{B}^{\mathbf{T}}\mathbf{A}^{\mathbf{T}}$, และ $(\mathbf{A}^{\mathbf{T}})^{\mathbf{T}} = \mathbf{A}$

Types of Matrix

<u>**Symmetric/Skew-Symmetric:**</u>

ถ้า Square Matrix ใดมีคุณสมบัติ $\mathbf{A} = \mathbf{A}^{\mathbf{T}}$ เราเรียก Matrix นั้นว่าเป็น Symmetric, ในกรณีที่ถ้า $\mathbf{A}^{\mathbf{T}} = -\mathbf{A}$ เราเรียก Matrix นั้นว่าเป็น Skew-Symmetric

Real square Matrix(Matrix ที่ประกอบด้วย Elements เป็นค่าจริงทั้งหมด) ใดๆ สามารถเขียนได้ในรูปผลบวกของ Real Symmetric กับ Real Skew-Symmetric Matrix

Types of Matrix

<u>**Complex Conjugate:**</u>

Complex Conjugate ของ Matrix $\mathbf{A}$ เขียนเป็น $\mathbf{A}^{*}$ ซึ่งถ้า $\mathbf{A}=(a_{nm})$ เราได้ $\mathbf{A}^{*}=(a_{nm}^{*})$ เช่นถ้า Element a_{ij} ของ Matrix $\mathbf{A}$ เป็น $x+jy$, เราได้ Element b_{ij} ของ Matrix $\mathbf{B}=\mathbf{A}^{*}$ เป็น a_{ij}^{*} หรือ $x-jy$

Types of Matrix

2.3.3 Hermitian/Skew-Hermitian:

Matrix $\mathbf{A}$ ถ้ามีคุณสมบัติโดยที่ $\mathbf{A} = \mathbf{A}^{*T}$ เราเรียก Matrix $\mathbf{A}$ ว่าเป็น ***Hermitian Matrix***, แต่ถ้าเกิด $\mathbf{A} = -\mathbf{A}^{*T}$ เราจะเรียก Matrix $\mathbf{A}$ ว่าเป็น ***Skew-Hermitian Matrix***

ค่า $\mathbf{A}^{*T}$ เขียนอีกอย่างหนึ่งเป็น $\mathbf{A}^{H}$ และสำหรับ Hermitian Matrix เราได้คุณสมบัติ $\mathbf{A} = \mathbf{A}^{H}$

ในกรณีที่ Matrix มีแต่ค่าจริง ค่า Conjugate ของ Matrix จะไม่เปลี่ยน และ Hermitian Matrix/Skew-Hermitian Matrix จะมีความหมายเดียวกันกับ Symmetric/Skew-Symmetric Matrix

Types of Matrix

<u>Diagonal & Trace</u>:

Square Matrix $\mathbf{A}$ ที่มีแต่ค่า (a_{ij}) , $i = j$ และ Elements อื่นเป็นศูนย์เมื่อ $i \neq j$ เราเรียกว่าเป็น Diagonal Matrix,

$\mathrm{diag}(\mathbf{A})$ หมายถึงเฉพาะ Elements ใน Diagonal ของ $\mathbf{A}$

$\mathrm{trace}(\mathbf{A})$ หมายถึงผลรวมของ Elements ใน Diagonal ของ $\mathbf{A}$ ($\mathbf{A}$ ไม่จำเป็นต้องเป็น Diagonal Matrix)

Tridiagonal Matrix หมายถึง Square Matrix ที่มี Element อื่นๆเป็นศูนย์ นอกจาก Diagonal Element และ Element ที่อยู่บนและล่างของ Diagonal

Types of Matrix

<u>**Identity Matrix**</u>:

I เป็น Square Diagonal Matrix ที่มี Elements ใน Diagonal เป็น “หนึ่ง” ทั้งหมด, เราเรียก **I** เป็น Identity Matrix และมักจะใช้เครื่องหมาย “I” แทน สังเกตว่า $\mathbf{AI} = \mathbf{IA} = \mathbf{A}$, และ $\mathbf{I}^{\mathbf{n}} = \mathbf{I}$

<u>**Zero or Null Matrix**</u>:

0 หรือ Null คือ Matrix ที่ Elements ทุกตัวเป็นศูนย์

Matrix Inverse

Inverse ของ Matrix $\mathbf{A}$ เขียน $\mathbf{A}^{-1}$ มีคุณสมบัติโดยที่ $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ และ $\mathbf{A}^{-1}$ จะหาได้ก็ต่อเมื่อ $\mathbf{A}$ ไม่เป็น Singular หรือ $\det(\mathbf{A}) \neq 0$, คุณสมบัติที่สำคัญ $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$

Orthogonal/Unitary Matrix

ถ้า Matrix $\mathbf{A}$ มีคุณสมบัติที่ $\mathbf{A}^{T} = \mathbf{A}^{-1}$ หรือ $\mathbf{A}^{T}\mathbf{A} = \mathbf{I}$, เราเรียกว่า $\mathbf{A}$ เป็น Orthogonal Matrix, แต่ถ้าเกิด $\mathbf{A}^{*T} = \mathbf{A}^{-1}$ หรือ $\mathbf{A}^{*T}\mathbf{A} = \mathbf{I}$ เราเรียกว่า $\mathbf{A}$ เป็น Unitary Matrix

Orthogonal Vector

Orthogonal Vector:

ถ้า Vector สองตัวคือ $\mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$

และ Scalar Product ของ **A** และ **B** ซึ่งเราเขียน

$$\mathbf{A}^{\mathbf{T}}\mathbf{B} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n = \text{ Scalar Product}$$

ในกรณีที่ **A** และ **B** Orthogonal กัน, เราจะได้

$$\mathbf{A}^{\mathbf{T}}\mathbf{B} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n = 0,$$

Orthogonal Vector

ในกรณีที่ $\mathbf{A}$ และ $\mathbf{B}$ Orthogonal กัน, เราจะได้

$$\mathbf{A}^{\mathbf{T}}\mathbf{B} = a_1b_1 + a_2b_2 + \cdots + a_nb_n = 0,$$

ในกรณีของ Complex Vector(ทั้ง $\mathbf{A}$ และ $\mathbf{B}$), ถ้า $\mathbf{A}$ orthogonal กับ $\mathbf{B}$, เราได้ $\mathbf{A}^{*\mathbf{T}}\mathbf{B} = 0$,

ถ้า Vector Set $\mathbf{X_1},\mathbf{X_2},\ldots$ มีคุณสมบัติ $\mathbf{X}_j^{*T}\mathbf{X}_k = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$ เราเรียก Set ของ Vector ว่าเป็น Unitary Set, .ในกรณีที่

$\mathbf{X}_i$ มีค่าจริง(real), เราเรียก Set ของ Vector นี้ว่าเป็น Orthonormal set (หรือ Orthogonal set ของ vector)

คำว่า Orthonormal หรือ Unitary จะใช้เมื่อ Magnitude ของ Vector ดังกล่าวมีค่าเท่ากับหนึ่ง

Examples

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 \\ 3 & 2 & 6 \\ 5 & 6 & 4 \end{bmatrix} \text{is a square matrix of order } 3\times 3,\ \mathbf{A} \text{ is also a symmetric matrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & -3 & 5 \\ 3 & 0 & -6 \\ -5 & 6 & 0 \end{bmatrix} \text{is a square matrix of order } 3\times 3,\ \mathbf{B} \text{ is also a skew - symmetric matrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \text{is a diagonal matrix, Diag}(\mathbf{C}) = \{1, 2, 3\},\ \text{trace}(\mathbf{C}) = 1+2+3 = 6$$

$\mathbf{C}$ is also a symmetric matrix.

Examples

$$\mathbf{D} = \begin{bmatrix} 2 & 3 & 3 \\ 0 & -1 & 4 \\ 0 & 0 & 1 \end{bmatrix} \text{is an upper diagonal matrix}$$

$$\mathbf{E} = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 0 & 0 \\ 3 & 0 & 1 \end{bmatrix} \text{is a lower diagonal matrix}$$

$$\mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Examples

For any matrix $\mathbf{A}$ we have $\mathbf{AA}^{\mathrm{T}}$ is a symmetric matrix, $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is also a symmetric matrix.

$proof$: From $(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^T\mathbf{A}^T$ we have $(\mathbf{AA}^T)^T = (\mathbf{A}^T)^T\mathbf{A}^T = \mathbf{AA}^T$

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \text{is a zero matrix of order } 3\times 3$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \text{is a zero row vector and } \mathbf{H} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{is a zero column vector}$$

Examples

$$\mathbf{K} = 3\mathbf{i} - 2\mathbf{j} + 4\mathbf{k} \text{ can be written in matrix form as } \begin{bmatrix} 3 \\ -2 \\ 4 \end{bmatrix}$$

$$\mathbf{L} = 2\mathbf{i} + \mathbf{j} - \mathbf{k} \text{ can be written in matrix form as } \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix}$$

$$\mathbf{K} \bullet \mathbf{L} = \mathbf{K}^T\mathbf{L} = \begin{bmatrix} 3 & -2 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix} = 3 \cdot 2 + (-2) \cdot 1 + 4(-1) = 0$$

in this case vector **K** and vector **L** are orthogonal(perpendicular.)

Examples

$$\mathbf{M}=\begin{bmatrix}1 & 0 & -1\\ 2 & 1 & 1\\ 0 & 1 & -1\end{bmatrix},\quad \mathbf{N}=\begin{bmatrix}\frac{1}{2} & \frac{1}{4} & \frac{1}{4}\\ -\frac{1}{2} & \frac{1}{4} & -\frac{3}{4}\\ -\frac{1}{2} & \frac{1}{4} & \frac{1}{4}\end{bmatrix},$$

we have $\mathbf{MN}=\begin{bmatrix}1 & 0 & -1\\ 2 & 1 & 1\\ 0 & 1 & -1\end{bmatrix}\begin{bmatrix}\frac{1}{2} & \frac{1}{4} & \frac{1}{4}\\ -\frac{1}{2} & \frac{1}{4} & -\frac{3}{4}\\ -\frac{1}{2} & \frac{1}{4} & \frac{1}{4}\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$

We can say that $\mathbf{M}=\mathbf{N}^{-1}$ and $\mathbf{N}=\mathbf{M}^{-1}$ since $\mathbf{MN}=\mathbf{I}$.

If matrix is diagonal, the inverse is just the matrix with reciprocal of diagonal elements and is also diagonal.

For example, $\mathbf{O}=\begin{bmatrix}1 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 3\end{bmatrix}$ then $\mathbf{O}^{-1}=\begin{bmatrix}1 & 0 & 0\\ 0 & \frac{1}{2} & 0\\ 0 & 0 & \frac{1}{3}\end{bmatrix}$

Examples

$$P = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$ is an orthogonal matrix since

$$\mathrm{P}^{-1} = P^T = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$, we can see that

$$\mathrm{PP}^{\mathrm{T}} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$= \begin{bmatrix} \cos^2\theta + \sin^2\theta & -\cos\theta\sin\theta + \sin\theta\cos\theta \\ -\sin\theta\cos\theta + \cos\theta\sin\theta & c\cos^2\theta + \sin^2\theta \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$ for any angle θ

Determinant

ค่า ***Determinant*** ของ Matrix สามารถให้นิยามได้หลายแบบ ในกรณีนี้ เราจะใช้ Recursive Definition ดังนี้

ค่า Determinant ของ Matrix a ขนาด 1×1 ให้นิยามว่ามีค่าเท่ากับค่า Scalar a และ Determinant ของ Matrix $\mathbf{A}$ ขนาด $n \times n$ สามารถหาได้ดังนี้

$$\det(\mathbf{A}) = \sum_{j=1}^{n} (-1)^{j+1} a_{1,j} \det(\mathbf{A}_{1,j})$$

โดยที่ $\mathbf{A}_{1,j}$ เป็น Matrix ขนาด $(n-1) \times (n-1)$ ได้จากการลบแถวที่หนึ่ง และ Column ที่ j ของ Matrix $\mathbf{A}$ ออก ซึ่งค่า Determinant ที่ได้ เราเรียก *Minor* ของ Element $a_{1,j}$ ใน Matrix $\mathbf{A}$

Determinant of Matrix

- Given square matrix, determinant of a matrix **A** written **|A|** is defined by recursive equation as

$$\det(\mathbf{A}) = |\mathbf{A}| = \sum_{j=1}^{n} (-1)^{i+j} a_{i,j} \cdot Minor(a_{i,j})$$

$$= \sum_{j=1}^{n} (-1)^{j+l} a_{j,l} \cdot Minor(a_{j,l})$$

  – Starting from [$\mathbf{a}_{1x1}$] = a, and det(**a**)=a

Sign Matrix

- Given matrix **A** of order nxn
  - Sign matrix of A is the matrix in the form $\mathbf{B}=[b_{ij}]=[(-1)^{(i+j)}]$

$$\mathbf{A}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \text{sign } \mathbf{A}=\begin{bmatrix} (-1)^2 & (-1)^3 & \cdots & (-1)^{n+1} \\ (-1)^3 & (-1)^4 & \cdots & (-1)^{n+2} \\ \vdots & & & \\ \vdots & & & \\ (-1)^{n+1} & (-1)^{n+2} & \cdots & (-1)^{2n} \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 & 1 & \cdots & \cdots \\ -1 & 1 & -1 & \cdots & \cdots \\ 1 & -1 & 1 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \cdots & 1 \end{bmatrix}$$

Minor a_{ij}

- Is the determinant of matrix **A** after taken out row i^th and j^th column.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix} = \begin{bmatrix} 1 & 2 & -3 & 4 & 5 \\ -2 & 3 & 4 & -5 & 6 \\ 3 & -4 & 5 & 6 & -7 \\ 4 & 5 & -6 & 7 & 8 \\ -5 & 6 & 7 & 8 & -9 \end{bmatrix}$$

$$Minor(a_{23}) = \begin{vmatrix} a_{11} & a_{12} & a_{14} & a_{15} \\ a_{31} & a_{32} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{54} & a_{55} \end{vmatrix} = \begin{vmatrix} 1 & 2 & 4 & 5 \\ 3 & -4 & 6 & -7 \\ 4 & 5 & 7 & 8 \\ -5 & 6 & 8 & -9 \end{vmatrix}$$

Cofactor a_{ij}

- Cofactor a_{ij} is the minor a_{ij} with the sign according to sign pattern
- Matrix of cofactor of **A** is the matrix **B** which each element b_{ij} is the cofactor a_{ij}

$$Ex.\ \mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, Co(\mathbf{A}) = \begin{bmatrix} \begin{vmatrix} 5 & 6 \\ 8 & 9 \end{vmatrix} & -\begin{vmatrix} 4 & 6 \\ 7 & 9 \end{vmatrix} & \begin{vmatrix} 4 & 5 \\ 7 & 8 \end{vmatrix} \\ -\begin{vmatrix} 2 & 3 \\ 8 & 9 \end{vmatrix} & \begin{vmatrix} 1 & 3 \\ 7 & 9 \end{vmatrix} & -\begin{vmatrix} 1 & 2 \\ 7 & 8 \end{vmatrix} \\ \begin{vmatrix} 2 & 3 \\ 5 & 6 \end{vmatrix} & -\begin{vmatrix} 1 & 3 \\ 4 & 6 \end{vmatrix} & \begin{vmatrix} 1 & 2 \\ 4 & 5 \end{vmatrix} \end{bmatrix}$$

Determinant of 2x2 and 3x3

Determinants: $|\mathbf{A}|$

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

$$= a_{11} \cdot \text{Minor of } a_{11} - a_{12} \cdot \text{Minor of } a_{12} + a_{13} \cdot \text{Minor of } a_{13}$$

$$= a_{11}(a_{22}a_{33} - a_{32}a_{23}) - a_{12}(a_{21}a_{33} - a_{31}a_{23}) + a_{13}(a_{21}a_{32} - a_{31}a_{22})$$

Calculation of Determinant using Recursive Expansion

$$\det(\mathbf{A}) = |\mathbf{A}| = \sum_{j=1}^{n} (-1)^{i+j} a_{i,j} \cdot Minor(a_{i,j})$$

$$= \sum_{j=1}^{n} (-1)^{j+l} a_{j,l} \cdot Minor(a_{j,l})$$

$$|a| = a, \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

$$\begin{vmatrix} a & b & c \\ d & e & f \\ i & j & k \end{vmatrix} = a\begin{vmatrix} e & f \\ j & k \end{vmatrix} - b\begin{vmatrix} d & f \\ i & k \end{vmatrix} + c\begin{vmatrix} d & e \\ i & j \end{vmatrix}$$

$$= a(ek - jf) - b(dk - if) + c(dj - ie)$$

$$= aek\text{-}afi - bdk + bfi + cdj - cei, \quad \text{first row expand}$$

$$= -b\begin{vmatrix} d & f \\ i & k \end{vmatrix} + e\begin{vmatrix} a & c \\ i & k \end{vmatrix} - j\begin{vmatrix} a & c \\ d & f \end{vmatrix}$$

$$= -b(dk - if) + e(ak - ic) - j(af - dc)$$

$$= -bdk + bfi + aek - cei - afj + cdj, \text{ second column expand}$$

Calculation of Determinant using Recursive Expansion

$$\det(\mathbf{A}) = \left|\mathbf{A}\right| = \sum_{j=1}^{n} (-1)^{i+j} a_{i,j} \cdot Minor(a_{i,j})$$

$$= \sum_{j=1}^{n} (-1)^{j+l} a_{j,l} \cdot Minor(a_{j,l})$$

Complexity = O(n!)

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}$$

$$= a_{11}\begin{vmatrix} a_{22} & a_{23} & a_{24} \\ a_{32} & a_{33} & a_{34} \\ a_{42} & a_{43} & a_{44} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} \\ a_{41} & a_{43} & a_{44} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \\ a_{41} & a_{42} & a_{44} \end{vmatrix} - a_{14}\begin{vmatrix} a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{vmatrix}$$

ความหมายของ Determinant

ค่า Determinant ของ Matrix จะบ่งบอกว่า Matrix นั้นเป็น ***Singular*** หรือไม่ โดยที่ Matrix จะเป็น Singular ก็ต่อเมื่อค่า Determinant มีค่าเท่ากับศูนย์

Determinant ของ Matrix $\mathbf{A}$ เราใช้เครื่องหมาย $\det(\mathbf{A})$ หรือ $|\mathbf{A}|$ เครื่องหมายในกรณีหลังระวังอย่าสับสนกับค่า Magnitude ของ Vector หรือ Scalar

- เป็นค่า Scalar ที่สำคัญที่สุดที่บ่งบอกคุณสมบัติของ Square Matrix

คุณสมบัติของ Determinant

1. $\det(\mathbf{AB}) = \det(\mathbf{BA}) = \det(\mathbf{A})\det(\mathbf{B})$
2. $\det(\mathbf{A}^T) = \det(\mathbf{A})$
3. $\det(\alpha\mathbf{A}) = \alpha^n \det(\mathbf{A})$
4. $\det(\mathbf{A}^*) = \det(\mathbf{A})^*$
5. $\det(\mathbf{I}) = 1$

คุณสมบัติของ Determinant

6. ถ้ามีการสลับ Row หรือ Column ค่า Determinant ใหม่จะเท่ากับ -1 คูณค่าเดิม

7. ถ้า Matrix $\mathbf{A}$ เป็น Singular หรือ ***Linearly Dependent*** เราจะได้ $|\mathbf{A}| = 0$ อันนี้สามารถนำมาใช้ทดสอบ Non-Singular Matrix ได้

8. ถ้าแถวใดหรือ Column ใดของ Matrix เท่ากับศูนย์ หรือมีสองแถวหรือสอง Column ที่เหมือนกัน หรือเป็นจำนวนเท่าของกัน ค่า Determinant จะเป็นศูนย์

9. ถ้าเราคูณค่าคงที่ให้กับแถวใด หรือ Column ใดของ Matrix ค่า Determinant ที่ได้จะเป็นจำนวนเท่าของค่าคงที่นั้นกับค่า Determinant เดิม

10. ถ้าเราสร้าง Matrix ใหม่ ได้จากการนำค่าคงทีคูณด้วยแถว หรือ Column หนึ่งใดทำการบวกลบกับแถว หรือ Column ที่เหลือ ค่า Determinant ของ Matrix ใหม่จะยังคงเท่ากับ Matrix เดิม

Calculation of Determinant (Algorithm)

$$\det(\mathbf{A}) = |\mathbf{A}| = \sum_{j=1}^{n} (-1)^{i+j} a_{i,j} \cdot Minor(a_{i,j})$$

$$= \sum_{j=1}^{n} (-1)^{j+l} a_{j,l} \cdot Minor(a_{j,l})$$

Complexity = O(n!)

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}$$

ถ้าเราบวกลบ Column เพื่อให้ Element ในแถว(หรือคอลัมน์) ที่ต้องการขยายเป็นศูนย์หมดยกเว้น Element เดียว เราจะลงเอยด้วยการคำนวณหา Determinant ของ Matrix ที่มีขนาดลดลงหนึ่ง

$$= a_{11}\begin{vmatrix} a_{22} & a_{23} & a_{24} \\ a_{32} & a_{33} & a_{34} \\ a_{42} & a_{43} & a_{44} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} \\ a_{41} & a_{43} & a_{44} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \\ a_{41} & a_{42} & a_{44} \end{vmatrix} - a_{14}\begin{vmatrix} a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{vmatrix}$$

การบวกลบดังกล่าวต้องมีหลักการ มิฉะนั้นผลลัพธ์จะไม่ถูกต้อง
เราจะใช้คุณสมบัติข้อ 9 และ 10 ของ Determinant เพื่อกระทำดังกล่าว

Algorithm การหา Determinant ที่มีประสิทธิภาพ

- ใช้คุณสมบัติข้อ 10 ร่วมกับข้อ 9 เพื่อสร้างเป็น Algorithm
  - 1. มองหา Element ใน Matrix ที่มีค่าเท่ากับ 1 ถ้าหาไม่ได้ เลือก Element ใดก็ได้ จากนั้นหารทั้งแถว หรือหารทั้ง Column ด้วยค่าของ Element นั้นเพื่อทำให้ค่าเป็น 1 ตัวเลขที่มาหารนั้นจะต้องกลับนำมาคูณกับคำตอบที่ได้ เป็นค่า Determinant ที่ต้องการ (คุณสมบัติข้อ 9)
  - 2. พิจารณาว่าจะ Expand แบบแถวหรือ Column ผ่าน Element ที่เลือก จากนั้นกำจัด Element อื่นในแนวที่ Expand เป็นศูนย์ให้หมด(คุณสมบัติข้อ 10) สมมุติเราเลือก Element a(x,y)
    - ถ้าจะ Expand แบบแถว ให้บวกลบ Column อื่นกับ Column ที่ผ่าน Element ที่เลือก เพื่อให้ Element ในแถวที่จะ Expand เป็นศูนย์ทั้งหมด ยกเว้น Element ที่เลือก
      - Col(j) ใหม่ = Col(j) เก่า – a(x,j)*Col(y); j = 1,2,..,n ยกเว้น y
    - ถ้าจะ Expand แบบ Column ให้บวกลบแถวอื่นกับแถวที่ผ่าน Element ที่เลือก เพื่อให้ Element ใน Column ที่จะ Expand เป็นศูนย์ทั้งหมด ยกเว้น Element ที่เลือก
      - Row(i) ใหม่ = Row(i) เก่า – a(i,y)*Row(x); i = 1,2,..,n ยกเว้น x

Algorithm การหา Determinant ที่มีประสิทธิภาพ(ต่อ)

- 3.ทำการ Expand ตามสูตร เราจะลงเอยด้วยการหา Determinant ของ Matrix ที่มีขนาดลดลงหนึ่งเพียงครั้งเดียว
- 4. วิธีนี้สามารถทำเป็น Recursive เพื่อลดการหา Determinant ของ Matrix ขนาดใหญ่ เหลือแค่การหา Determinant ของ Matrix 2x2 หรือ 3x3

การหา Determinant

Example 2.1 จงหา Determinant ของ Matrix

$$\mathbf{A} = \begin{bmatrix} -3 & 2 & 2 & 5 \\ 3 & -1 & 1 & 4 \\ 6 & -2 & 3 & 7 \\ 8 & -3 & -4 & 2 \end{bmatrix}$$

คำตอบ เราจะใช้คุณสมบัติข้อ 10 ที่กล่าวข้างต้น เปลี่ยน Matrix แถวที่ 1, 3 และ 4 ให้มีค่าใน Column ที่ 3 เป็น ศูนย์ จากนั้นหาค่า Determinant ตาม Column ที่ 3 ซึ่งการคำนวณจะลดลงเหลือการหาค่า Determinant ของ Matrix ขนาด 3×3 เท่านั้น

เราสร้าง Matrix $\mathbf{B}$ จาก Matrix $\mathbf{A}$ โดย

สร้างแถวที่หนึ่งใหม่จากการนำ -2 คูณแถวที่สอง ไปบวกกับแถวที่หนึ่งเดิม

สร้างแถวที่สามใหม่จากการนำ -3 คูณแถวที่สอง ไปบวกกับแถวที่หนึ่งเดิม

สร้างแถวที่สี่ใหม่จากการนำ 4 คูณแถวที่สอง ไปบวกกับแถวที่หนึ่งเดิม

ค่า Determinant ของ Matrix ใหม่จะเท่ากับ Matrix เดิม แต่จะคำนวณน้อยกว่าเดิมมากถ้าเรา Expand ตาม Column ที่ 3 ดังนี้

$$\mathbf{B} = \begin{bmatrix} -9 & 4 & 0 & -3 \\ 3 & -1 & 1 & 4 \\ -3 & 1 & 0 & -5 \\ 20 & -7 & 0 & 18 \end{bmatrix}$$

$$|\mathbf{B}| = 0\begin{vmatrix} 3 & -1 & 4 \\ -3 & 1 & -5 \\ 20 & -7 & 18 \end{vmatrix} - 1\begin{vmatrix} -9 & 4 & -3 \\ -3 & 1 & -5 \\ 20 & -7 & 18 \end{vmatrix} + 0\begin{vmatrix} -9 & 4 & -3 \\ 3 & -1 & 4 \\ 20 & -7 & 18 \end{vmatrix} - 0\begin{vmatrix} -9 & 4 & -3 \\ 3 & -1 & 4 \\ -3 & 1 & -5 \end{vmatrix}$$

$$|\mathbf{B}| = -1\begin{vmatrix} -9 & 4 & -3 \\ -3 & 1 & -5 \\ 20 & -7 & 18 \end{vmatrix} = -1\begin{vmatrix} 3 & 4 & 17 \\ 0 & 1 & 0 \\ -1 & -7 & -17 \end{vmatrix} = -1\begin{vmatrix} 3 & 17 \\ -1 & -17 \end{vmatrix} = 34$$

สังเกตว่า ในการหา Determinant ของ Matrix 3×3 ในบรรทัดที่สอง เราใช้วิธีการเดิมในการลดรูป และทำการ Expand ในแถวที่สอง

กรรมวิธีดังกล่าวสามารถทำได้ถ้าใน Matrix มี Element ตัวใดตัวหนึ่งมีค่าเท่ากับหนึ่ง แต่ถ้าไม่มี Element ใดเท่ากับหนึ่ง เราสามารถแยก Factor แถวใดแถวหนึ่ง หรือ Column หนึ่งออกมา เพื่อให้ Element หนึ่งมีค่าเป็นหนึ่ง และใช้วิธีการดังกล่าว แต่อย่าลืมคูณค่า Factor ที่แยกออกในผลลัพธ์ที่ได้ (ดูคุณสมบัติข้อ 9)

การหา Determinant

กรรมวิธีดังกล่าวสามารถทำได้ถ้าใน Matrix มี Element ตัวใดตัวหนึ่งมีค่าเท่ากับหนึ่ง แต่ถ้าไม่มี Element ใดเท่ากับหนึ่ง เราสามารถแยก Factor แถวใดแถวหนึ่ง หรือ Column หนึ่งออกมา เพื่อให้ Element หนึ่งมีค่าเป็นหนึ่ง และใช้วิธีการดังกล่าว แต่อย่าลืมคูณค่า Factor ที่แยกออกในผลลัพธ์ที่ได้ (ดูคุณสมบัติข้อ 9)

ถ้า Matrix เป็น Diagonal Matrix หรือ ***Upper Triangular*** หรือ ***Lower Triangular*** (คือ มีค่า Element ครึ่งล่างต่ำกว่า Diagonal หรือครึ่งบนสูงกว่า Diagonal เป็นศูนย์หมด ตามลำดับ) ค่า Determinant สามารถคำนวณได้ง่าย จะเท่ากับผลคูณของ Element ใน Diagonal

Inverse of Matrix

2.5 Matrix Inverse

2.5.1 Matrix Inverse:

กำหนด Matrix **A** ที่ไม่ใช่ Singular Matrix เราต้องการหา Matrix **B** ที่ทำให้ $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$ ดังนั้น Matrix **B** ที่ได้คือ ***Inverse Matrix*** ของ **A** กล่าวคือ $\mathbf{B} = \mathbf{A}^{-1}$

ถ้า Matrix เป็น Singular จะไม่มี Inverse ดังนั้น Inverse จะมีได้สำหรับ Non-Singular Matrix เท่านั้น และ Inverse ของ Matrix จะ Unique

Inverse of Matrix

สามารถกระทำได้หลายวิธี ได้แก่

- ใช้วิธีทาง Numerical Method เช่นโดยใช้ Gauss-Jordan Elimination(จะเรียนในภายหลัง)
- Formal Evaluation ซึ่งจะกล่าวในบทนี้

ขั้นตอนการหา Matrix Inverse กระทำดังนี้

1. หา Cofactor ของ Matrix $\mathbf{A}$ โดยแทนค่าทุกๆ element ด้วยค่า Cofactor ของมัน ขั้นตอนนี้จะใช้การคำนวณมากที่สุด เพราะต้องหา Cofactor ของทุกๆ Element

2. Transpose Matrix ที่ได้ในข้อ 1 เราได้ Matrix ใหม่เรียก ***Adjoint*** ของ Matrix $\mathbf{A}$ หรือ $\mathbf{adj\ A}$

3. หาค่า Determinant ของ $\mathbf{A} = |\mathbf{A}|$

4. คำนวณ Inverse ของ Matrix โดย $\mathbf{A}^{-1} = \dfrac{\mathbf{adj\ A}}{|\mathbf{A}|}$

Inverse of Matrix

Example 2.2 จงหา Inverse ของ Matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 2 \\ 5 & -4 & 1 \\ -1 & 2 & 1 \end{bmatrix}$$

คำตอบ เราหา Matrix ของ Cofactor ดังนี้

$$\text{Cofactor} = \begin{bmatrix} \begin{vmatrix} -4 & 1 \\ 2 & 1 \end{vmatrix} & -\begin{vmatrix} 5 & 1 \\ -1 & 1 \end{vmatrix} & \begin{vmatrix} 5 & -4 \\ -1 & 2 \end{vmatrix} \\ -\begin{vmatrix} 1 & 2 \\ 2 & 1 \end{vmatrix} & \begin{vmatrix} 3 & 2 \\ -1 & 1 \end{vmatrix} & -\begin{vmatrix} 3 & 1 \\ -1 & 2 \end{vmatrix} \\ \begin{vmatrix} 1 & 2 \\ -4 & 1 \end{vmatrix} & -\begin{vmatrix} 3 & 2 \\ 5 & 1 \end{vmatrix} & \begin{vmatrix} 3 & 1 \\ 5 & -4 \end{vmatrix} \end{bmatrix} = \begin{bmatrix} -6 & -6 & 6 \\ 3 & 5 & -7 \\ 9 & 7 & -17 \end{bmatrix}$$

Inverse of Matrix

$$\text{Cofactor} = \begin{bmatrix} \begin{vmatrix} -4 & 1 \\ 2 & 1 \end{vmatrix} & -\begin{vmatrix} 5 & 1 \\ -1 & 1 \end{vmatrix} & \begin{vmatrix} 5 & -4 \\ -1 & 2 \end{vmatrix} \\ -\begin{vmatrix} 1 & 2 \\ 2 & 1 \end{vmatrix} & \begin{vmatrix} 3 & 2 \\ -1 & 1 \end{vmatrix} & -\begin{vmatrix} 3 & 1 \\ -1 & 2 \end{vmatrix} \\ \begin{vmatrix} 1 & 2 \\ -4 & 1 \end{vmatrix} & -\begin{vmatrix} 3 & 2 \\ 5 & 1 \end{vmatrix} & \begin{vmatrix} 3 & 1 \\ 5 & -4 \end{vmatrix} \end{bmatrix} = \begin{bmatrix} -6 & -6 & 6 \\ 3 & 5 & -7 \\ 9 & 7 & -17 \end{bmatrix}$$

ดังนั้น เราได้ $adj\ \mathbf{A} = \begin{bmatrix} -6 & 3 & 9 \\ -6 & 5 & 7 \\ 6 & -7 & -17 \end{bmatrix}$

ทำการหา Determinant ของ Matrix เราได้

$$|\mathbf{A}| = \begin{vmatrix} 3 & 1 & 2 \\ 5 & -4 & 1 \\ -1 & 2 & 1 \end{vmatrix} = \begin{vmatrix} 3 & 1 & 2 \\ 17 & 0 & 9 \\ -7 & 0 & -3 \end{vmatrix} = -1\begin{vmatrix} 17 & 9 \\ -7 & -3 \end{vmatrix} = -12$$

ดังนั้น เราได้ $adj\ \mathbf{A} = \begin{bmatrix} -6 & 3 & 9 \\ -6 & 5 & 7 \\ 6 & -7 & -17 \end{bmatrix}$

ทำการหา Determinant ของ Matrix เราได้

$$|\mathbf{A}| = \begin{vmatrix} 3 & 1 & 2 \\ 5 & -4 & 1 \\ -1 & 2 & 1 \end{vmatrix} = \begin{vmatrix} 3 & 1 & 2 \\ 17 & 0 & 9 \\ -7 & 0 & -3 \end{vmatrix} = -1\begin{vmatrix} 17 & 9 \\ -7 & -3 \end{vmatrix} = -12$$

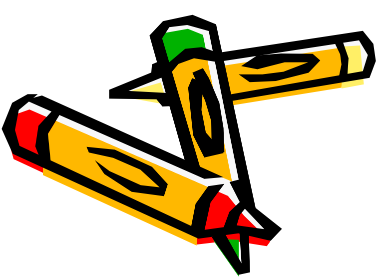
ดังนั้น

$$\mathbf{A}^{-1} = -\frac{1}{12}\begin{bmatrix} -6 & 3 & 9 \\ -6 & 5 & 7 \\ 6 & -7 & -17 \end{bmatrix}$$

ขอให้นักศึกษาลองตรวจสอบว่า $\mathbf{A}\mathbf{A}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Matrix Inverse

- การคำนวณตามสมการที่กล่าวมา จะใช้การคำนวณมากเกินไป
- หลัง Midterm เราจะมาดู Algorithm ที่มีประสิทธิภาพ เพื่อใช้ในการหา Matrix Inverse
  - Gauss-Jordan Method



Matrix Norms

2.6 Matrix Norms

สำหรับ Matrix $\mathbf{A}$ ใน $\mathbf{C}^{n\times m}$ เราให้นิยาม ***Norm*** ของ Matrix ดังนี้

$$\|\mathbf{A}\|_{pq} = \max_{x\in\mathbf{C}^m, x\neq 0} \frac{\|\mathbf{A}x\|_p}{\|x\|_q}$$

ในการนี้ Norm $\|\cdot\|_{pq}$ ได้มาจากสอง Norm คือ $\|\cdot\|_p$ และ $\|\cdot\|_q$ ซึ่งมีคุณสมบัติขิง Norm ทั้งสามประการ กล่าวคือ

1. $\|\mathbf{A}\| \geq 0 \quad \forall \mathbf{A} \in \mathbf{C}^{n\times m}$ และ $\|\mathbf{A}\| = 0 \quad \mathit{iff} \;\; \mathbf{A} = 0$

2. $\|\alpha\mathbf{A}\| = |\alpha|\|\mathbf{A}\|, \forall \mathbf{A} \in \mathbf{C}^{n\times m}, \forall \alpha \in \mathbf{C}$

3. $\|\mathbf{A}+\mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|, \forall \mathbf{A}, \mathbf{B} \in \mathbf{C}^{n\times m}$

คุณสมบัติที่สำคัญอีกอันของ Norm ก็คือ $\|\mathbf{AB}\|_p \leq \|\mathbf{A}\|_p \|\mathbf{B}\|_p$

Norm ของ Matrix ที่สำคัญมีดังนี้

Norms

$$\|\mathbf{A}\|_1 = \max_{j=1,2,\ldots,m} \sum_{i=1}^{n} |a_{ij}|$$

$$\|\mathbf{A}\|_\infty = \max_{i=1,2,\ldots,n} \sum_{j=1}^{m} |a_{ij}|$$

$$\|\mathbf{A}\|_2 = \left[\rho(\mathbf{A}^H\mathbf{A})\right]^{1/2} = \left[\rho(\mathbf{A}\mathbf{A}^H)\right]^{1/2}$$

$$\|\mathbf{A}\|_F = \left[trace(\mathbf{A}^H\mathbf{A})\right]^{1/2} = \left[trace(\mathbf{A}\mathbf{A}^H)\right]^{1/2}$$

ค่า $\rho(\mathbf{X})$ หมายถึงค่า Magnitude ที่มากที่สุดของค่า Eigenvalue ของ $\mathbf{X}$ (Eigenvalue จะกล่าวในบทหน้า)

เรียก Spectral Radius ของ X

Homework 1:Vector

- Download HW 1 Question Sheet จาก Website
  - พิมพ์ Question Sheet ลงบนกระดาษ A4
  - ทำการบ้าน โดยแสดงวิธีทำและคำตอบในช่องที่กำหนด ด้วยลายมือนักศึกษา ห้ามพิมพ์ ต้องใช้กระดาษที่พิมพ์นี้ทำการบ้าน
  - เขียนชื่อที่หัวกระดาษ ส่งอาทิตย์หน้าต้นชั่วโมง
  - ไม่รับงานที่ไม่เป็นไปตามที่กำหนด

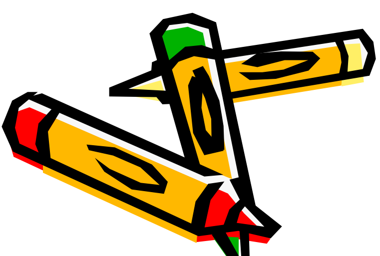
CPE 332
Computer Engineering Mathematics II

Week 3: Ch.2 Matrices Continue(Linear Equations)

Ch.3 Eigenvector

Today Topics

- Part I Chapt. 2 Matrices
- Break
- Chapter 2: Linear Equations
- Homework 1: Due
- Homework 2 ส่งสัปดาห์หน้า



Chapter 2: Cont

- Last Week
  - Matrix Concept and Notations
  - Types of Matrix
  - Concept of Determinant and Inverse
- This Week
  - More on Determinant and Matrix
  - System of Linear Equations: Homogeneous and Non-homogeneous
  - Eigenvalue/Eigenvector Concept

Determinant

Determinants: $|\mathbf{A}|$

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

$$= a_{11} \cdot \text{Minor of } a_{11} - a_{12} \cdot \text{Minor of } a_{12} + a_{13} \cdot \text{Minor of } a_{13}$$

$$= a_{11}(a_{22}a_{33} - a_{32}a_{23}) - a_{12}(a_{21}a_{33} - a_{31}a_{23}) + a_{13}(a_{21}a_{32} - a_{31}a_{22})$$

Calculation of Determinant

$$\det(\mathbf{A}) = |\mathbf{A}| = \sum_{j=1}^{n} (-1)^{i+j} a_{i,j} \cdot Minor(a_{i,j})$$

$$= \sum_{j=1}^{n} (-1)^{j+l} a_{j,l} \cdot Minor(a_{j,l})$$

$$|a| = a, \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

$$\begin{vmatrix} a & b & c \\ d & e & f \\ i & j & k \end{vmatrix} = a\begin{vmatrix} e & f \\ j & k \end{vmatrix} - b\begin{vmatrix} d & f \\ i & k \end{vmatrix} + c\begin{vmatrix} d & e \\ i & j \end{vmatrix}$$

$$= a(ek - jf) - b(dk - if) + c(dj - ie)$$

$$= aek\text{-}afi - bdk + bfi + cdj - cei, \quad \text{first row expand}$$

$$= -b\begin{vmatrix} d & f \\ i & k \end{vmatrix} + e\begin{vmatrix} a & c \\ i & k \end{vmatrix} - j\begin{vmatrix} a & c \\ d & f \end{vmatrix}$$

$$= -b(dk - if) + e(ak - ic) - j(af - dc)$$

$$= -bdk + bfi + aek - cei - afj + cdj, \text{ second column expand}$$

คุณสมบัติของ Determinant

1. $\det(\mathbf{AB}) = \det(\mathbf{BA}) = \det(\mathbf{A})\det(\mathbf{B})$
2. $\det(\mathbf{A}^T) = \det(\mathbf{A})$
3. $\det(\alpha\mathbf{A}) = \alpha^n \det(\mathbf{A})$
4. $\det(\mathbf{A}^*) = \det(\mathbf{A})^*$
5. $\det(\mathbf{I}) = 1$

คุณสมบัติของ Determinant

6. ถ้ามีการสลับ Row หรือ Column ค่า Determinant ใหม่จะเท่ากับ -1 คูณค่าเดิม

7. ถ้า Matrix $\mathbf{A}$ เป็น Singular หรือ ***Linearly Dependent*** เราจะได้ $|\mathbf{A}| = 0$ อันนี้สามารถนำมาใช้ทดสอบ Non-Singular Matrix ได้

8. ถ้าแถวใดหรือ Column ใดของ Matrix เท่ากับศูนย์ หรือมีสองแถวหรือสอง Column ที่เหมือนกัน หรือเป็นจำนวนเท่าของกัน ค่า Determinant จะเป็นศูนย์

9. ถ้าเราคูณค่าคงที่ให้กับแถวใด หรือ Column ใดของ Matrix ค่า Determinant ที่ได้จะเป็นจำนวนเท่าของค่าคงที่นั้นกับค่า Determinant เดิม

10. ถ้าเราสร้าง Matrix ใหม่ ได้จากการนำค่าคงที่คูณด้วยแถว หรือ Column หนึ่งใดทำการบวกลบกับแถว หรือ Column ที่เหลือ ค่า Determinant ของ Matrix ใหม่จะยังคงเท่ากับ Matrix เดิม

Calculation of Determinant

$$\det(\mathbf{A}) = |\mathbf{A}| = \sum_{j=1}^{n} (-1)^{i+j} a_{i,j} \cdot Minor(a_{i,j})$$

$$= \sum_{j=1}^{n} (-1)^{j+l} a_{j,l} \cdot Minor(a_{j,l})$$

Complexity = O(n!)

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}$$

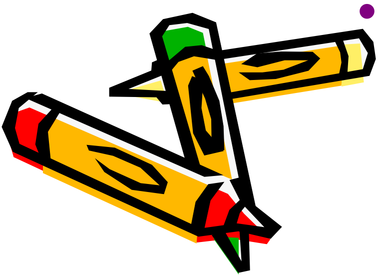
ถ้าเราบวกลบ Column เพื่อให้ Element ในแถว(หรือคอลัมน์)
ที่ต้องการขยายเป็นศูนย์หมดยกเว้น Element เดียว
เราจะลงเอยด้วยการคำนวณหา Determinant ของ Matrix
ที่มีขนาดลดลงหนึ่ง

$$= a_{11}\begin{vmatrix} a_{22} & a_{23} & a_{24} \\ a_{32} & a_{33} & a_{34} \\ a_{42} & a_{43} & a_{44} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} \\ a_{41} & a_{43} & a_{44} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \\ a_{41} & a_{42} & a_{44} \end{vmatrix} - a_{14}\begin{vmatrix} a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{vmatrix}$$

การบวกลบดังกล่าวต้องมีหลักการ มิฉะนั้นผลลัพธ์จะไม่ถูกต้อง
เราจะใช้คุณสมบัติข้อ 9 และ 10 ของ Determinant เพื่อกระทำดังกล่าว

Algorithm การหา Determinant ที่มีประสิทธิภาพ

- ใช้คุณสมบัติข้อ 10 ร่วมกับข้อ 9 เพื่อสร้างเป็น Algorithm
  - 1. มองหา Element ใน Matrix ที่มีค่าเท่ากับ 1 ถ้าหาไม่ได้ เลือก Element ใดก็ได้ จากนั้นหารทั้งแถว หรือหารทั้ง Column ด้วยค่าของ Element นั้นเพื่อทำให้ค่าเป็น 1 ตัวเลขที่มาหารนั้นจะต้องกลับนำมาคูณกับคำตอบที่ได้ เป็นค่า Determinant ที่ต้องการ (คุณสมบัติข้อ 9)
  - 2. พิจารณาว่าจะ Expand แบบแถวหรือ Column ผ่าน Element ที่เลือก จากนั้นกำจัด Element อื่นในแนวที่ Expand เป็นศูนย์ให้หมด(คุณสมบัติข้อ 10) สมมุติเราเลือก Element a(x,y)
    - ถ้าจะ Expand แบบแถว ให้บวกลบ Column อื่นกับ Column ที่ผ่าน Element ที่เลือก เพื่อให้ Element ในแถวที่จะ Expand เป็นศูนย์ทั้งหมด ยกเว้น Element ที่เลือก
      - Col(j) ใหม่ = Col(j) เก่า – a(x,j)*Col(y); j = 1,2,..,n ยกเว้น y
    - ถ้าจะ Expand แบบ Column ให้บวกลบแถวอื่นกับแถวที่ผ่าน Element ที่เลือก เพื่อให้ Element ใน Column ที่จะ Expand เป็นศูนย์ทั้งหมด ยกเว้น Element ที่เลือก
      - Row(i) ใหม่ = Row(i) เก่า – a(i,y)*Row(x); i = 1,2,..,n ยกเว้น x



Algorithm การหา Determinant ที่มีประสิทธิภาพ(ต่อ)

- 3.ทำการ Expand ตามสูตร เราจะลงเอยด้วยการหา Determinant ของ Matrix ที่มีขนาดลดลงหนึ่งเพียงครั้งเดียว
- 4. วิธีนี้สามารถทำเป็น Recursive เพื่อลดการหา Determinant ของ Matrix ขนาดใหญ่ เหลือแค่การหา Determinant ของ Matrix 2x2 หรือ 3x3

การหา Determinant

Example 2.1 จงหา Determinant ของ Matrix

$$\mathbf{A} = \begin{bmatrix} -3 & 2 & 2 & 5 \\ 3 & -1 & 1 & 4 \\ 6 & -2 & 3 & 7 \\ 8 & -3 & -4 & 2 \end{bmatrix}$$

Column 3 Expansion
Or Row 2 Expansion

คำตอบ เราจะใช้คุณสมบัติข้อ 10 ที่กล่าวข้างต้น เปลี่ยน Matrix แถวที่ 1, 3 และ 4 ให้มีค่าใน Column ที่ 3 เป็น ศูนย์ จากนั้นหาค่า Determinant ตาม Column ที่ 3 ซึ่งการคำนวณจะลดลงเหลือการหาค่า Determinant ของ Matrix ขนาด 3×3 เท่านั้น

เราสร้าง Matrix $\mathbf{B}$ จาก Matrix $\mathbf{A}$ โดย

สร้างแถวที่หนึ่งใหม่จากการนำ -2 คูณแถวที่สอง ไปบวกกับแถวที่หนึ่งเดิม

สร้างแถวที่สามใหม่จากการนำ -3 คูณแถวที่สอง ไปบวกกับแถวที่หนึ่งเดิม

สร้างแถวที่สี่ใหม่จากการนำ 4 คูณแถวที่สอง ไปบวกกับแถวที่หนึ่งเดิม

ค่า Determinant ของ Matrix ใหม่จะเท่ากับ Matrix เดิม แต่จะคำนวณน้อยกว่าเดิมมากถ้าเรา Expand ตาม Column ที่ 3 ดังนี้

$$\mathbf{B} = \begin{bmatrix} -9 & 4 & 0 & -3 \\ 3 & -1 & 1 & 4 \\ -3 & 1 & 0 & -5 \\ 20 & -7 & 0 & 18 \end{bmatrix}$$

$$|\mathbf{B}| = 0\begin{vmatrix} 3 & -1 & 4 \\ -3 & 1 & -5 \\ 20 & -7 & 18 \end{vmatrix} - 1\begin{vmatrix} -9 & 4 & -3 \\ -3 & 1 & -5 \\ 20 & -7 & 18 \end{vmatrix} + 0\begin{vmatrix} -9 & 4 & -3 \\ 3 & -1 & 4 \\ 20 & -7 & 18 \end{vmatrix} - 0\begin{vmatrix} -9 & 4 & -3 \\ 3 & -1 & 4 \\ -3 & 1 & -5 \end{vmatrix}$$

$$|\mathbf{B}| = -1\begin{vmatrix} -9 & 4 & -3 \\ -3 & 1 & -5 \\ 20 & -7 & 18 \end{vmatrix} = -1\begin{vmatrix} 3 & 4 & 17 \\ 0 & 1 & 0 \\ -1 & -7 & -17 \end{vmatrix} = -1\begin{vmatrix} 3 & 17 \\ -1 & -17 \end{vmatrix} = 34$$

สังเกตว่า ในการหา Determinant ของ Matrix 3×3 ในบรรทัดที่สอง เราใช้วิธีการเดิมในการลดรูป และทำการ Expand ในแถวที่สอง

กรรมวิธีดังกล่าวสามารถทำได้ถ้าใน Matrix มี Element ตัวใดตัวหนึ่งมีค่าเท่ากับหนึ่ง แต่ถ้าไม่มี Element ใดเท่ากับหนึ่ง เราสามารถแยก Factor แถวใดแถวหนึ่ง หรือ Column หนึ่งออกมา เพื่อให้ Element หนึ่งมีค่าเป็นหนึ่ง และใช้วิธีการดังกล่าว แต่อย่าลืมคูณค่า Factor ที่แยกออกในผลลัพธ์ที่ได้ (ดูคุณสมบัติข้อ 9)

การหา Determinant

กรรมวิธีดังกล่าวสามารถทำได้ถ้าใน Matrix มี Element ตัวใดตัวหนึ่งมีค่าเท่ากับหนึ่ง แต่ถ้าไม่มี Element ใดเท่ากับหนึ่ง เราสามารถแยก Factor แถวใดแถวหนึ่ง หรือ Column หนึ่งออกมา เพื่อให้ Element หนึ่งมีค่าเป็นหนึ่ง และใช้วิธีการดังกล่าว แต่อย่าลืมคูณค่า Factor ที่แยกออกในผลลัพธ์ที่ได้ (ดูคุณสมบัติข้อ 9)

ถ้า Matrix เป็น Diagonal Matrix หรือ ***Upper Triangular*** หรือ ***Lower Triangular*** (คือ มีค่า Element ครึ่งล่างต่ำกว่า Diagonal หรือครึ่งบนสูงกว่า Diagonal เป็นศูนย์หมด ตามลำดับ) ค่า Determinant สามารถคำนวณได้ง่าย จะเท่ากับผลคูณของ Element ใน Diagonal

Inverse of Matrix

2.5 Matrix Inverse

2.5.1 Matrix Inverse:

กำหนด Matrix $\mathbf{A}$ ที่ไม่ใช่ Singular Matrix เราต้องการหา Matrix $\mathbf{B}$ ที่ทำให้ $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$ ดังนั้น Matrix $\mathbf{B}$ ที่ได้คือ ***Inverse Matrix*** ของ $\mathbf{A}$ กล่าวคือ $\mathbf{B} = \mathbf{A}^{-1}$

ถ้า Matrix เป็น Singular จะไม่มี Inverse ดังนั้น Inverse จะมีได้สำหรับ Non-Singular Matrix เท่านั้น และ Inverse ของ Matrix จะ Unique

Inverse of Matrix

สามารถกระทำได้หลายวิธี ได้แก่

- ใช้วิธีทาง Numerical Method เช่นโดยใช้ Gauss-Jordan Elimination(จะเรียนในภายหลัง)
- Formal Evaluation ซึ่งจะกล่าวในบทนี้

ขั้นตอนการหา Matrix Inverse กระทำดังนี้

1. หา Cofactor ของ Matrix $\mathbf{A}$ โดยแทนค่าทุกๆ element ด้วยค่า Cofactor ของมัน ขั้นตอนนี้จะใช้การคำนวณมากที่สุด เพราะต้องหา Cofactor ของทุกๆ Element

2. Transpose Matrix ที่ได้ในข้อ 1 เราได้ Matrix ใหม่เรียก ***Adjoint*** ของ Matrix $\mathbf{A}$ หรือ $\mathbf{adj\ A}$

3. หาค่า Determinant ของ $\mathbf{A} = |\mathbf{A}|$

4. คำนวณ Inverse ของ Matrix โดย $\mathbf{A}^{-1} = \dfrac{\mathbf{adj\ A}}{|\mathbf{A}|}$

Inverse of Matrix

Example 2.2 จงหา Inverse ของ Matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 2 \\ 5 & -4 & 1 \\ -1 & 2 & 1 \end{bmatrix}$$

คำตอบ เราหา Matrix ของ Cofactor ดังนี้

$$\text{Cofactor} = \begin{bmatrix} \begin{vmatrix} -4 & 1 \\ 2 & 1 \end{vmatrix} & -\begin{vmatrix} 5 & 1 \\ -1 & 1 \end{vmatrix} & \begin{vmatrix} 5 & -4 \\ -1 & 2 \end{vmatrix} \\ -\begin{vmatrix} 1 & 2 \\ 2 & 1 \end{vmatrix} & \begin{vmatrix} 3 & 2 \\ -1 & 1 \end{vmatrix} & -\begin{vmatrix} 3 & 1 \\ -1 & 2 \end{vmatrix} \\ \begin{vmatrix} 1 & 2 \\ -4 & 1 \end{vmatrix} & -\begin{vmatrix} 3 & 2 \\ 5 & 1 \end{vmatrix} & \begin{vmatrix} 3 & 1 \\ 5 & -4 \end{vmatrix} \end{bmatrix} = \begin{bmatrix} -6 & -6 & 6 \\ 3 & 5 & -7 \\ 9 & 7 & -17 \end{bmatrix}$$

Inverse of Matrix

$$\text{Cofactor} = \begin{bmatrix} \begin{vmatrix} -4 & 1 \\ 2 & 1 \end{vmatrix} & -\begin{vmatrix} 5 & 1 \\ -1 & 1 \end{vmatrix} & \begin{vmatrix} 5 & -4 \\ -1 & 2 \end{vmatrix} \\ -\begin{vmatrix} 1 & 2 \\ 2 & 1 \end{vmatrix} & \begin{vmatrix} 3 & 2 \\ -1 & 1 \end{vmatrix} & -\begin{vmatrix} 3 & 1 \\ -1 & 2 \end{vmatrix} \\ \begin{vmatrix} 1 & 2 \\ -4 & 1 \end{vmatrix} & -\begin{vmatrix} 3 & 2 \\ 5 & 1 \end{vmatrix} & \begin{vmatrix} 3 & 1 \\ 5 & -4 \end{vmatrix} \end{bmatrix} = \begin{bmatrix} -6 & -6 & 6 \\ 3 & 5 & -7 \\ 9 & 7 & -17 \end{bmatrix}$$

ดังนั้น เราได้ $adj\ \mathbf{A} = \begin{bmatrix} -6 & 3 & 9 \\ -6 & 5 & 7 \\ 6 & -7 & -17 \end{bmatrix}$

ทำการหา Determinant ของ Matrix เราได้

$$|\mathbf{A}| = \begin{vmatrix} 3 & 1 & 2 \\ 5 & -4 & 1 \\ -1 & 2 & 1 \end{vmatrix} = \begin{vmatrix} 3 & 1 & 2 \\ 17 & 0 & 9 \\ -7 & 0 & -3 \end{vmatrix} = -1\begin{vmatrix} 17 & 9 \\ -7 & -3 \end{vmatrix} = -12$$

ดังนั้น เราได้ $adj\ \mathbf{A} = \begin{bmatrix} -6 & 3 & 9 \\ -6 & 5 & 7 \\ 6 & -7 & -17 \end{bmatrix}$

ทำการหา Determinant ของ Matrix เราได้

$$|\mathbf{A}| = \begin{vmatrix} 3 & 1 & 2 \\ 5 & -4 & 1 \\ -1 & 2 & 1 \end{vmatrix} = \begin{vmatrix} 3 & 1 & 2 \\ 17 & 0 & 9 \\ -7 & 0 & -3 \end{vmatrix} = -1 \begin{vmatrix} 17 & 9 \\ -7 & -3 \end{vmatrix} = -12$$

ดังนั้น

$$\mathbf{A}^{-1} = -\frac{1}{12} \begin{bmatrix} -6 & 3 & 9 \\ -6 & 5 & 7 \\ 6 & -7 & -17 \end{bmatrix}$$

ขอให้นักศึกษาลองตรวจสอบว่า $\mathbf{AA}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

สาธิตทดลอง Row Expansion

Matrix Norms

2.6 Matrix Norms

สำหรับ Matrix $\mathbf{A}$ ใน $\mathbf{C}^{n\times m}$ เราให้นิยาม ***Norm*** ของ Matrix ดังนี้

$$\|\mathbf{A}\|_{pq} = \max_{x\in\mathbf{C}^m,\, x\neq 0} \frac{\|\mathbf{A}x\|_p}{\|x\|_q}$$

ในการนี้ Norm $\|\cdot\|_{pq}$ ได้มาจากสอง Norm คือ $\|\cdot\|_p$ และ $\|\cdot\|_q$ ซึ่งมีคุณสมบัติขิง Norm ทั้งสามประการ กล่าวคือ

1. $\|\mathbf{A}\| \geq 0 \quad \forall \mathbf{A} \in \mathbf{C}^{n\times m}$ และ $\|\mathbf{A}\| = 0 \quad \mathit{iff}\ \ \mathbf{A} = 0$

2. $\|\alpha\mathbf{A}\| = |\alpha|\|\mathbf{A}\|, \forall \mathbf{A} \in \mathbf{C}^{n\times m}, \forall \alpha \in \mathbf{C}$

3. $\|\mathbf{A}+\mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|, \forall \mathbf{A}, \mathbf{B} \in \mathbf{C}^{n\times m}$

คุณสมบัติที่สำคัญอีกอันของ Norm ก็คือ $\|\mathbf{AB}\|_p \leq \|\mathbf{A}\|_p \|\mathbf{B}\|_p$

Norm ของ Matrix ที่สำคัญมีดังนี้

Norms

$$\|\mathbf{A}\|_1 = \max_{j=1,2,\ldots,m} \sum_{i=1}^{n} |a_{ij}|$$

$$\|\mathbf{A}\|_\infty = \max_{i=1,2,\ldots,n} \sum_{j=1}^{m} |a_{ij}|$$

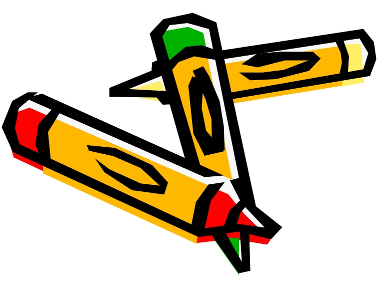
$$\|\mathbf{A}\|_2 = \left[\rho(\mathbf{A}^H\mathbf{A})\right]^{1/2} = \left[\rho(\mathbf{A}\mathbf{A}^H)\right]^{1/2}$$

$$\|\mathbf{A}\|_F = \left[trace(\mathbf{A}^H\mathbf{A})\right]^{1/2} = \left[trace(\mathbf{A}\mathbf{A}^H)\right]^{1/2}$$

ค่า $\rho(\mathbf{X})$ หมายถึงค่า Magnitude ที่มากที่สุดของค่า Eigenvalue ของ $\mathbf{X}$ (Eigenvalue จะกล่าวในบทหน้า)

เรียก Spectral Radius ของ X

Break



System of Linear Equations

พิจารณาจากระบบของ n สมการที่ประกอบด้วย m Unknown ข้างล่าง

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_m &= 0 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_m &= 0 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nm}x_m &= 0
\end{aligned}
$$

ระบบดังกล่าวเรียก ***Homogeneous*** เนื่องจากแต่ละสมการเป็น Homogeneous และเราจะเห็นได้ว่า ***Trivial Solution*** ของระบบคือเมื่อ $\mathbf{X} = \mathbf{0}$ อย่างไรก็ตาม ในกรณีนี้ เราต้องการหา ***Non-Trivial Solution***

ในระบบข้างต้นสามารถเขียนในรูป Matrix ได้เป็น $\mathbf{AX} = \mathbf{0}$ โดยที่

$$
\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ & & \vdots & \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}, \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}
$$

System of Linear Equations

สมการดังกล่าวจะมี Non-Trivial Solution เมื่อ $m > rank(\mathbf{A})$ โดยที่ ***Rank*** ของ Matrix หมายถึงจำนวนของ Independent Row ของ Matrix (จำนวนของแถวของ Matrix ที่ไม่สามารถเขียนได้ในรูปของ Linear Combination ของแถวอื่น) และสังเกตว่าในกรณีนี้ จำนวนของ Non-Trivial Solution อาจจะมีได้หลายตัว

การแก้สมการของ Homogeneous Linear Equation สามารถกระทำได้โดยใช้วิธีการ Elimination ของ Coefficient ใน Matrix เช่นใน ***Gauss-Jordan Method*** ซึ่งรายละเอียดจะกล่าวใน Part III ของวิชานี้

หลักของการ Elimination คือการเปลี่ยนรูป Matrix $\mathbf{A}$ ให้อยู่ในรูปของ ***Reduced Form*** $\mathbf{A}_R$ ซึ่งมีคุณสมบัติดังนี้

1. Element ต้น(ที่ไม่เท่ากับศูนย์) หรือ ***Leading Entry*** ของแต่ละแถวที่ไม่ใช่ศูนย์ทั้งหมดจะมีค่าเท่ากับหนึ่ง
2. ถ้ามีแถวใด มี Element ต้นอยู่ใน Column k ค่าของ Element ในแถวอื่นๆที่ Column k จะต้องเท่ากับศูนย์
3. ถ้า Row k เป็น ***Zero Row*** และ Row j ไม่ใช่ Zero Row ดังนั้น $j < k$
4. ถ้า Leading Entry ของ Row r_1 อยู่ที่ Column c_1 และ Leading Entry ของ Row r_2 อยู่ที่ Column c_2 และ $r_1 < r_2$ ดังนั้น $c_1 < c_2$

Reduced Matrix

ค่า ***Rank*** ของ Matrix $\mathbf{A}$ เขียน $rank(\mathbf{A})$ จะเท่ากับจำนวนของ ***Nonzero Row*** ใน $\mathbf{A}_R$ ซึ่งเท่ากับจำนวนของ Independent Row ของ Matrix $\mathbf{A}$ และ $rank(\mathbf{A}) = rank(\mathbf{A}_R)$

ต่อไปนี้ เป็นตัวอย่างของ ***Reduced Matrix***

$$\mathbf{A}_R = \begin{bmatrix} 1 & -4 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B}_R = \begin{bmatrix} 0 & 1 & 3 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{C}_R = \begin{bmatrix} 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{D}_R = \begin{bmatrix} 1 & 0 & 0 & 3 & 1 \\ 0 & 1 & 0 & -2 & 4 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

ค่า Rank ของ Matrix ข้างบนจะเป็นดังนี้

$$rank(\mathbf{A}_R) = 2,\ rank(\mathbf{B}_R) = 2,\ rank(\mathbf{C}_R) = 2,\ rank(\mathbf{D}_R) = 3,$$

Reduced Matrix

Example 2.3 จงหา Reduced Form และ Rank ของ Matrix

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 4 & 2 \\ 0 & 1 & 3 & 2 \\ 3 & -2 & 15 & 8 \end{bmatrix}$$

คำตอบ เราใช้วิธีการกำจัด Element ที่ไม่ต้องการทีละ Column ดังนี้

Reduced Matrix

1. กำจัด Element แรกของแถวที่ 3 ออก โดยนำแถวที่ 1 คูณด้วย -3 ไปบวกกับแถวที่ 3 เราได้

$$\begin{bmatrix} 1 & -1 & 4 & 2 \\ 0 & 1 & 3 & 2 \\ 0 & 1 & 3 & 2 \end{bmatrix}$$

2. กำจัด Element ที่ 2 ของแถวที่ 1 ออก โดยนำแถวที่ 1 บวกแถวที่ 2 ไปแทนในแถวที่ 1 เราได้

$$\begin{bmatrix} 1 & 0 & 7 & 4 \\ 0 & 1 & 3 & 2 \\ 0 & 1 & 3 & 2 \end{bmatrix}$$

3. กำจัด Element ที่ 2 ของแถวที่ 3 ออก โดยนำแถวที่ 2 ลบแถวที่ 3 ไปแทนที่ในแถวที่ 3 เราได้

$$\begin{bmatrix} 1 & 0 & 7 & 4 \\ 0 & 1 & 3 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \mathbf{A}_R$$

เราไม่สามารถ Eliminate ได้อีกต่อไป ฉะนั้น Matrix ที่ได้จะเป็น Reduced Form และในกรณีนี้ เราได้

$$rank(\mathbf{A}) = rank(\mathbf{A}_R) = 2$$

Solutions of Homogeneous

Example 2.4 จงแก้สมการ

$$x_1 - 3x_2 + 2x_3 = 0$$
$$-2x_1 + x_2 - 3x_3 = 0$$

คำตอบ เราจะใช้หลักการของการ Elimination ดังนี้ จาก $\mathbf{AX} = \mathbf{0}$

$$\begin{bmatrix} 1 & -3 & 2 \\ -2 & 1 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

ทำการกำจัด Element a_{12} และ a_{21} ออก จะได้ Reduced Matrix ของ $\mathbf{A}$ ($\mathbf{A}_R$) ตามขั้นตอนดังนี้

Solutions of Homogeneous

$$\begin{bmatrix} 1 & -3 & 2 \\ 0 & -5 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$ นำ 2 คูณแถวที่ 1 บวกกับแถวที่ 2 ไปแทนแถวที่ 2 เดิม

$$\begin{bmatrix} 1 & -3 & 2 \\ 0 & 1 & -\frac{1}{5} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$ หารแถวที่สองด้วย -5 ตลอด

$$\begin{bmatrix} 1 & 0 & \frac{7}{5} \\ 0 & 1 & -\frac{1}{5} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$ นำแถวที่สองคูณด้วย 3 บวกกับแถวที่ 1 แทนที่แถวที่ 1 เดิม

ดังนั้น Solution จะเป็น $x_1 = -\frac{7}{5}x_3$ และ $x_2 = \frac{1}{5}x_3$ โดยที่ x_3 จะเป็นค่าอะไรก็ได้ กล่าวคือเราได้คำตอบในลักษณะของอัตราส่วน ซึ่งเราเรียก ***General Solution*** ของระบบ

Solutions of Homogeneous

ปกติเรามักเขียนคำตอบในรูปความสัมพันธ์ ในลักษณะ Column Matrix ดังนี้

$$\begin{bmatrix} -\frac{7}{5}x_3 \\ \frac{1}{5}x_3 \\ x_3 \end{bmatrix} = \begin{bmatrix} -\frac{7}{5} \\ \frac{1}{5} \\ 1 \end{bmatrix} x_3 = \alpha \begin{bmatrix} -7 \\ 1 \\ 5 \end{bmatrix}$$ ($\alpha = x_3 / 5$ จะมีค่าอะไรก็ได้)

Non-homogeneous Systems

2.7.2 Nonhomogeneous Systems of Linear Equations

ในกรณีนี้ เราจะพิจารณา ***Nonhomogeneous Linear System*** ในรูปของ

$$\mathbf{AX} = \mathbf{B}$$

โดยที่ $\mathbf{A}$ เป็น Matrix ขนาด $n \times m$ ของตัวเลข, $\mathbf{X}$ เป็น Matrix ขนาด $m \times 1$ ของ Unknown ที่ต้องการหา และ $\mathbf{B}$ เป็น Matrix ขนาด $n \times 1$ ของตัวเลข ในกรณีของเราจะสมมุติว่าตัวเลขทั้งหมดเป็น Real Number (สำหรับ Complex Number จะใช้หลักการเหมือนกัน)

การแก้สมการของ Nonhomogeneous Linear Equation จะคล้ายกับ Homogeneous แต่ในการ Elimination นั้นจะต้องคำนึงถึงค่าใน $\mathbf{B}$ ด้วย ในการนี้ เวลาเราทำ Elimination เรานิยมเขียน $\mathbf{A}$ ต่อกับ $\mathbf{B}$ เพื่อความสะดวกในรูป

Non-homogeneous Systems

การแก้สมการของ Nonhomogeneous Linear Equation จะคล้ายกับ Homogeneous แต่ในการ Elimination นั้น จะต้องคำนึงถึงค่าใน $\mathbf{B}$ ด้วย ในการนี้ เวลาเราทำ Elimination เรานิยมเขียน $\mathbf{A}$ ต่อกับ $\mathbf{B}$ เพื่อความสะดวกในรูป

$$[\mathbf{A} : \mathbf{B}] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1m} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2m} & b_2 \\ & & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} & b_n \end{array}\right]$$

สมการ $\mathbf{AX} = \mathbf{B}$ จะมี Solution (เรียก ***Consistent***) เมื่อ $\mathbf{A}$ และ $[\mathbf{A} : \mathbf{B}]$ มี Rank เท่ากัน และถ้าเราพบว่า $\mathbf{U}$ เป็น Solution ของสมการ $\mathbf{AX} = \mathbf{B}$ ดังนั้นทุกๆ Solution ของ $\mathbf{AX} = \mathbf{B}$ จะอยู่ในรูป $\mathbf{U} + \mathbf{H}$ โดยที่ $\mathbf{H}$ เป็น ***General Solution*** ของ Homogeneous System $\mathbf{AX} = \mathbf{0}$ (สำหรับการพิสูจน์ในข้ออ้างข้างต้น จะไม่กล่าวรายละเอียด ผู้ที่สนใจ ขอให้หาอ่านได้จากหนังสือ Linear Algebra ทั่วไป)

Non-homogeneous Systems

Example 2.6 จงแก้สมการ หา General Solution ของ

$$-x_1 + x_2 + 3x_3 = -2$$
$$x_2 + 2x_3 = 4$$

คำตอบ นี่คือระบบของ Nonhomogeneous Linear Equation ซึ่งสามารถเขียนในรูป $\mathbf{AX} = \mathbf{B}$ ได้เป็น

$$\begin{bmatrix} -1 & 1 & 3 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 4 \end{bmatrix}$$

การหา Solution เริ่มจากเราเขียน Matrix $[\mathbf{A} : \mathbf{B}]$ ดังนี้

$$[\mathbf{A} : \mathbf{B}] = \left[\begin{array}{ccc|c} -1 & 1 & 3 & -2 \\ 0 & 1 & 2 & 4 \end{array}\right]$$

ทำการหา Reduced Form ของ Matrix เราได้

$$[\mathbf{A} : \mathbf{B}]_R = \left[\begin{array}{ccc|c} 1 & 0 & -1 & 6 \\ 0 & 1 & 2 & 4 \end{array}\right]$$

Non-homogeneous Systems

ทำการหา Reduced Form ของ Matrix เราได้

$$[\mathbf{A}:\mathbf{B}]_R = \left[\begin{array}{ccc|c} 1 & 0 & -1 & 6 \\ 0 & 1 & 2 & 4 \end{array}\right]$$

ในขณะที่

$$\mathbf{A}_R = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix}$$

สังเกตว่า $rank(\mathbf{A}) = 2 = rank([\mathbf{A}:\mathbf{B}])$ ดังนั้นระบบจะมี Solution ในรูปของความสัมพันธ์ดังนี้

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

Non-homogeneous Systems

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

ดังนั้น เราได้

$$x_1 - x_3 = 6$$
$$x_2 + 2x_3 = 4$$

เมื่อย้ายข้าง จะได้

$$x_1 = x_3 + 6$$
$$x_2 = -2x_3 + 4$$

และ Solution จะเป็น

Non-homogeneous Systems

$$x_1 = x_3 + 6$$
$$x_2 = -2x_3 + 4$$

และ Solution จะเป็น

$$\mathbf{X} = \begin{bmatrix} x_3 + 6 \\ -2x_3 + 4 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} x_3 + \begin{bmatrix} 6 \\ 4 \\ 0 \end{bmatrix}$$

โดยที่ x_3 จะมีค่าใดก็ได้ ดังนั้นเราสามารถเขียน General Solution ได้เป็น

$$\mathbf{X} = \alpha \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} + \begin{bmatrix} 6 \\ 4 \\ 0 \end{bmatrix}$$

เทอม $\alpha \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$ จะเป็น ***General Solution*** ของ Homogeneous System ของ $\mathbf{AX} = \mathbf{0}$ และเทอม $\begin{bmatrix} 6 \\ 4 \\ 0 \end{bmatrix}$ เรารียก

ว่าเป็น ***Particular Solution*** ของ $\mathbf{AX} = \mathbf{B}$ (ซึ่งเป็น Solution หนึ่งของระบบเมื่อ $\alpha = 0$)

2.7.3 Uniqueness ของ Nonhomogeneous Systems of Linear Equations

ถ้า $\mathbf{A}$ ใน $\mathbf{AX}=\mathbf{B}$ เป็น Matrix ขนาด $n \times n$ ระบบจะมี Unique Solution ก็ต่อเมื่อ $rank(\mathbf{A}) = n$ หรือกล่าวอีกนัยหนึ่งคือ $\mathbf{A}$ ไม่เป็น Singular (คือมี Determinant ไม่เท่ากับศูนย์ หรือมี Inverse)

ในกรณีดังกล่าว Solution ของสมการสามารถหาได้จากการคูณด้วย $\mathbf{A}^{-1}$ ทั้ง 2 ข้าง กล่าวคือ จาก $\mathbf{AX}=\mathbf{B}$ เราได้ $\mathbf{A}^{-1}\mathbf{AX}=\mathbf{A}^{-1}\mathbf{B}$ หรือ $\mathbf{I}_n\mathbf{X}=\mathbf{A}^{-1}\mathbf{B}$ หรือ $\mathbf{X}=\mathbf{A}^{-1}\mathbf{B}$ อย่างไรก็ตาม การแก้สมการ โดยการคูณด้วย Inverse จะใช้การคำนวณค่อนข้างมากในการหา Inverse วิธีที่นิยมมากกว่าคือขบวนการ Elimination ซึ่งรายละเอียดของ Algorithm จะกล่าวใน Part III

$$
\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = c_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = c_2 \\
\qquad\quad \cdot \qquad\quad \cdot \qquad\quad \cdot \\
\qquad\quad \cdot \qquad\quad \cdot \qquad\quad \cdot \\
\qquad\quad \cdot \qquad\quad \cdot \qquad\quad \cdot \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = c_n
\end{array}
=
\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}
=
\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}
= \mathbf{AX} = \mathbf{C}
$$

CPE 332

Computer Engineering Mathematic II

PART I: Linear Algebra

(Chapter 1-3)

Chapter 3 Intro

3.1 Introduction

กำหนด $n \times n$ Matrix $\mathbf{A}$ และ Column Vector ξ ใน n-space(หรือ Matrix ขนาด $n \times 1$) ผลคูณ $\mathbf{A}\xi$ จะเป็น Matrix ขนาด $n \times 1$ ซึ่งก็คือ Column Vector ใน n-space เช่นเดียวกัน ตามปกติแล้วทิศทางของ Vector $\mathbf{A}\xi$ จะต่างจากทิศทางของ Vector ξ อย่างไรก็ตามถ้าเราสามารถหา Vector ξ ที่ทำให้ทิศทางของ $\mathbf{A}\xi$ เหมือนกันกับทิศทางของ ξ ได้ ในกรณีนี้เรากล่าวได้ว่า Vector $\mathbf{A}\xi$ จะขนานกับ Vector ξ จากที่กล่าวมาแล้วในเรื่องของ Vector โดยที่ Vector 2 ตัวจะเท่ากันก็ต่อเมื่อมันชี้ไปในทิศทางเดียวกัน และมีขนาดเท่ากัน ดังนั้นถ้าให้ λ เป็นตัวคูณที่ทำให้ Vector $\mathbf{A}\xi$ เท่ากับ ξ เราจะได้สมการ $\mathbf{A}\xi = \lambda\xi$ ในกรณีเช่นนี้ เราเรียกค่า λ ว่าเป็น “***Eigenvalue***” และ Vector ξ ว่าเป็น “***Eigenvector***” ของ Matrix $\mathbf{A}$ ค่าทั้งสองเป็นคุณลักษณะที่สำคัญของ Matrix และสามารถนำไปใช้ประโยชน์ในการหาคำตอบทางคณิตศาสตร์หลายๆอย่างในวิศวกรรม

นิยาม 1

ให้ A เป็น Matrix ขนาด $n \times n$ ของเลขจริงหรือเลข Complex, เลขจริงหรือเลข Complex ค่า λ จะเป็นค่า Eigenvalue ก็ต่อเมื่อมี Matrix E ขนาด $n \times 1$ ที่ไม่เป็นศูนย์ ที่ทำให้ $AE = \lambda E$ และ Matrix E ดังกล่าวเราเรียก Eigenvector ที่สัมพันธ์กับค่า Eigenvalue λ นั้น

นอกจากนั้นแล้วค่า Scalar เป็นจำนวนเท่าของ Eigenvector ก็จะเป็น Eigenvector ด้วย เนื่องจาก

$$AE = \lambda E \Rightarrow A(\alpha E) = \alpha AE = \alpha(\lambda E) = \lambda(\alpha E)$$

ตัวอย่างที่ 1

จากสมการ $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0\begin{bmatrix} 0 \\ 4 \end{bmatrix}$

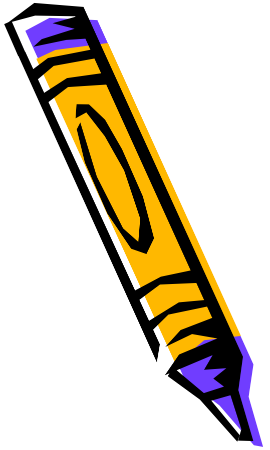
เราสรุปว่า ค่า 0 เป็น Eigenvalue ของ $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ และ $\begin{bmatrix} 0 \\ 4 \end{bmatrix}$ เป็น Eigenvector ที่สัมพันธ์กับ Eigenvalue 0 นอกจากนี้แล้วจำนวนเท่าที่ไม่ใช่ศูนย์ ของ $\begin{bmatrix} 0 \\ 4 \end{bmatrix}$ จะเป็น Eigenvector ด้วย

ตัวอย่างที่ 2

ให้ $A = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{bmatrix}$ ดังนั้นค่า 1 จะเป็น Eigenvalue และ $E = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}$ เป็น Eigenvector ที่สัมพันธ์กับค่า Eigenvalue ดังกล่าว เพราะว่า $AE = 1 \cdot E = E$ นอกจากนี้แล้ว Vector $\begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix}$ จะเป็น Eigenvector ด้วย

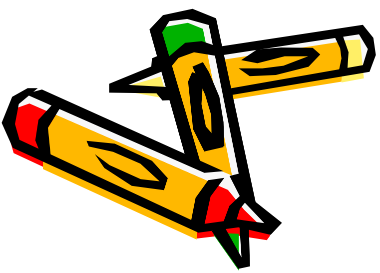
ค่า Eigenvalue ของ A อีกค่าหนึ่งก็คือ -1 และ Eigenvector ที่เกี่ยวข้องคือ $\begin{bmatrix} 1 \\ 2 \\ -4 \end{bmatrix}$ และจะรวมถึง $\begin{bmatrix} \beta \\ 2\beta \\ -4\beta \end{bmatrix}$ ซึ่งเป็น Eigenvector ของ A ที่สัมพันธ์กับค่า Eigenvalue -1 ด้วยเช่นกัน

MATLAB TUTORIAL



End of Week 3

- Download HW 2 Due Next Week
  - Chapter 1: Vector Operations
  - Chapter 2: Linear Equations
- Next week: WK 4
  - Eigenvalue/Eigenvector Continue
  - MATLAB
  - HW 3



CPE 332
Computer Engineering Mathematics II

Week 4: Ch.3 Eigenvector and Diagonalization

Ch4. Probability Review

Today Topics

- Part I Chapt. 3 Eigenvalue/Eigenvector
- Break
- Chapter 4: Review Probability
- Homework 2: Due
- Homework 3 ส่งสัปดาห์หน้า ต้นชั่วโมง

Determinant

Determinants: $|\mathbf{A}|$

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

$$= a_{11} \cdot \text{Minor of } a_{11} - a_{12} \cdot \text{Minor of } a_{12} + a_{13} \cdot \text{Minor of } a_{13}$$

$$= a_{11}(a_{22}a_{33} - a_{32}a_{23}) - a_{12}(a_{21}a_{33} - a_{31}a_{23}) + a_{13}(a_{21}a_{32} - a_{31}a_{22})$$

CPE 332

Computer Engineering Mathematic II

PART I: Linear Algebra

(Chapter 1-3)

Chapter 3 Intro

3.1 Introduction

กำหนด $n \times n$ Matrix $\mathbf{A}$ และ Column Vector ξ ใน n-space(หรือ Matrix ขนาด $n \times 1$) ผลคูณ $\mathbf{A}\xi$ จะเป็น Matrix ขนาด $n \times 1$ ซึ่งก็คือ Column Vector ใน n-space เช่นเดียวกัน ตามปกติแล้วทิศทางของ Vector $\mathbf{A}\xi$ จะต่างจากทิศทางของ Vector ξ อย่างไรก็ตามถ้าเราสามารถหา Vector ξ ที่ทำให้ทิศทางของ $\mathbf{A}\xi$ เหมือนกันกับทิศทางของ ξ ได้ ในกรณีนี้เรากล่าวได้ว่า Vector $\mathbf{A}\xi$ จะขนานกับ Vector ξ จากที่กล่าวมาแล้วในเรื่องของ Vector โดยที่ Vector 2 ตัวจะเท่ากันก็ต่อเมื่อมันชี้ไปในทิศทางเดียวกัน และมีขนาดเท่ากัน ดังนั้นถ้าให้ λ เป็นตัวคูณที่ทำให้ Vector $\mathbf{A}\xi$ เท่ากับ ξ เราจะได้สมการ $\mathbf{A}\xi = \lambda\xi$ ในกรณีเช่นนี้ เราเรียกค่า λ ว่าเป็น "***Eigenvalue***" และ Vector ξ ว่าเป็น "***Eigenvector***" ของ Matrix $\mathbf{A}$ ค่าทั้งสองเป็นคุณลักษณะที่สำคัญของ Matrix และสามารถนำไปใช้ประโยชน์ในการหาคำตอบทางคณิตศาสตร์หลายๆอย่างในวิศวกรรม

นิยาม 1

ให้ A เป็น Matrix ขนาด $n \times n$ ของเลขจริงหรือเลข Complex, เลขจริงหรือเลข Complex ค่า λ จะเป็นค่า Eigenvalue ก็ต่อเมื่อมี Matrix E ขนาด $n \times 1$ ที่ไม่เป็นศูนย์ ที่ทำให้ $AE = \lambda E$ และ Matrix E ดังกล่าวเราเรียก Eigenvector ที่สัมพันธ์กับค่า Eigenvalue λ นั้น

นอกจากนั้นแล้วค่า Scalar เป็นจำนวนเท่าของ Eigenvector ก็จะเป็น Eigenvector ด้วย เนื่องจาก

$$AE = \lambda E \Rightarrow A(\alpha E) = \alpha AE = \alpha(\lambda E) = \lambda(\alpha E)$$

ตัวอย่างที่ 1

จากสมการ $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0\begin{bmatrix} 0 \\ 4 \end{bmatrix}$

เราสรุปว่า ค่า 0 เป็น Eigenvalue ของ $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ และ $\begin{bmatrix} 0 \\ 4 \end{bmatrix}$ เป็น Eigenvector ที่สัมพันธ์กับ Eigenvalue 0 นอกจากนี้แล้วจำนวนเท่าที่ไม่ใช่ศูนย์ ของ $\begin{bmatrix} 0 \\ 4 \end{bmatrix}$ จะเป็น Eigenvector ด้วย

ตัวอย่างที่ 2

ให้ $A = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{bmatrix}$ ดังนั้นค่า 1 จะเป็น Eigenvalue และ $E = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}$ เป็น Eigenvector ที่สัมพันธ์กับค่า Eigenvalue ดังกล่าว เพราะว่า $AE = 1 \cdot E = E$ นอกจากนี้แล้ว Vector $\begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix}$ จะเป็น Eigenvector ด้วย

ค่า Eigenvalue ของ A อีกค่าหนึ่งก็คือ -1 และ Eigenvector ที่เกี่ยวข้องคือ $\begin{bmatrix} 1 \\ 2 \\ -4 \end{bmatrix}$ และจะรวมถึง $\begin{bmatrix} \beta \\ 2\beta \\ -4\beta \end{bmatrix}$ ซึ่งเป็น Eigenvector ของ A ที่สัมพันธ์กับค่า Eigenvalue -1 ด้วยเช่นกัน

3.3 การคำนวณหาค่า Eigenvalue และ Eigenvector

Theorem 1:

ให้ A เป็น $n \times n$ Matrix ของค่าจริงหรือค่า Complex ดังนั้น

1. λ จะเป็น Eigenvalue ของ A ก็ต่อเมื่อ $|\lambda I_n - A| = 0$

2. ถ้า λ เป็น Eigenvalue ของ A ดังนั้น Nontrivial Solution ใดๆของ $(\lambda I_n - A)X = 0$ จะเป็น Eigenvector

เมื่อเราขยาย Determinant ในสมการข้อที่ 1 เราจะได้ Polynomial ของ λ มี Degree เท่ากับ n Polynomial นี้เราเรียก *Characteristic Polynomial* ของ A เขียน $p_A(\lambda)$ สมการที่ได้คือสมการ $p_A(\lambda) = 0$ เราเรียก *Characteristic Equation* ของ A สังเกตว่า Characteristic Equation จะให้ค่า Eigenvalue ทั้งหมด n ค่า และบางค่าอาจจะซ้ำกัน

ตัวอย่างที่ 3

พิจารณา Matrix เดิม $A = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{bmatrix}$ เราได้ $\lambda I_3 - A = \begin{bmatrix} \lambda - 1 & 1 & 0 \\ 0 & \lambda - 1 & -1 \\ 0 & 0 & \lambda + 1 \end{bmatrix}$

ดังนั้น Characteristic Polynomial จะเป็น $\begin{vmatrix} \lambda - 1 & 1 & 0 \\ 0 & \lambda - 1 & -1 \\ 0 & 0 & \lambda + 1 \end{vmatrix} = (\lambda - 1)\begin{vmatrix} \lambda - 1 & -1 \\ 0 & \lambda + 1 \end{vmatrix} = (\lambda - 1)^2(\lambda + 1)$

Characteristic Equation $(\lambda - 1)^2(\lambda + 1) = 0$ จะมีรากของสมการเท่ากับ 1 ด้วยค่า Multiplicity เท่ากับ 2 และรากของสมการเท่ากับ -1 ด้วยค่า Multiplicity เท่ากับ 1

ในการหา Eigenvector ที่สัมพันธ์กับ 1 เราแก้สมการ $(I_3 - A)X = 0$ ดังนี้

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$ ซึ่งจะมี Solution ทั่วไปในรูปของ $$\begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix}$$

และ Nontrivial Solution($\alpha \neq 0$) จะเป็น Eigenvector ที่สัมพันธ์กับ Eigenvalue 1

คราวนี้ลองหา Eigenvector ที่สัมพันธ์กับ Eigenvalue -1

$$\begin{bmatrix} -2 & 1 & 0 \\ 0 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$ เราจะได้ Solution ในรูปของ $$\begin{bmatrix} \beta \\ 2\beta \\ -4\beta \end{bmatrix}$$

ตัวอย่างที่ 4

ให้ $A = \begin{bmatrix} 1 & -2 \\ 2 & 0 \end{bmatrix}$ ดังนั้น $\lambda I_2 - A = \begin{bmatrix} \lambda - 1 & 2 \\ -2 & \lambda \end{bmatrix}$ และ Characteristic Equation จะเป็น

$p_A(\lambda) = \begin{vmatrix} \lambda - 1 & 2 \\ -2 & \lambda \end{vmatrix} = (\lambda - 1)\lambda + 4 = \lambda^2 - \lambda + 4 = 0$ ซึ่งมีรากเป็น Eigenvector คือ $\lambda = (1 \pm \sqrt{15}i)/2$

ในการหา Eigenvector ที่สัมพันธ์กับ $(1+\sqrt{15}i)/2$ เราแก้สมการ $\left(\frac{1+\sqrt{15}i}{2} I_2 - A\right)X = 0$ กล่าวคือ

$$\begin{bmatrix} \frac{1+\sqrt{15}i}{2} - 1 & 2 \\ -2 & \frac{1+\sqrt{15}i}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$ หรือ

$$\frac{-1+\sqrt{15}i}{2} x_1 + 2x_2 = 0$$

$$-2x_1 + \frac{1+\sqrt{15}i}{2} x_2 = 0$$

ซึ่งถ้าดูจากสมการแรก เราได้ $x_2 = \frac{1-\sqrt{15}i}{4} x_1$ และถ้านำแทนค่าในสมการที่สองจะพบว่าสมการจะเป็นความจริงไม่ว่า x_1 จะมีค่าใดๆก็ตาม ดังนั้นเราให้ค่า $x_1 = \alpha$ เราจะได้ $x_2 = \frac{1-\sqrt{15}i}{4}\alpha$ นั่นก็คือคำตอบทั่วไปของระบบจะเป็น

$\alpha \begin{bmatrix} 1 \\ \frac{1-\sqrt{15}i}{4} \end{bmatrix}$ และ Matrix ใดๆก็ตามที่ค่า $\alpha \neq 0$ จะเป็น Eigenvector ที่สัมพันธ์กับค่า Eigenvalue $(1+\sqrt{15}i)/2$

ขอให้นักศึกษาหา Eigenvector อีกตัวที่สัมพันธ์กับค่า Eigenvalue $(1-\sqrt{15i})/2$ ซึ่งคำตอบจะเป็น $\beta\begin{bmatrix}1\\ \frac{1+\sqrt{15i}}{4}\end{bmatrix}$

สำหรับค่า $\beta \neq 0$

Diagonalization

นิยาม 2

Matrix ขนาด $n \times n$, $D = [d_{ij}]$ จะเรียกว่า Diagonal Matrix ถ้า $d_{ij} = 0$ เมื่อ $i \neq j$

ปกติเราเขียน Diagonal Matrix โดยเขียนเฉพาะค่า Main Diagonal ของมันคือ $d_1, d_2, \ldots, d_n$ หรือเขียน

$$\begin{bmatrix} d_1 & & & 0 \\ & d_2 & & \\ & & \ddots & \\ 0 & & & d_n \end{bmatrix}$$

สังเกตการใส่ค่า 0 ที่มุมซ้ายล่าง และขวาบน

Theorem 2:

ให้ $D = \begin{bmatrix} d_1 & & & 0 \\ & d_2 & & \\ & & \ddots & \\ 0 & & & d_n \end{bmatrix}$ และ $W = \begin{bmatrix} w_1 & & & 0 \\ & w_2 & & \\ & & \ddots & \\ 0 & & & w_n \end{bmatrix}$ ดังนั้น

1. $DW = WD = \begin{bmatrix} d_1 w_1 & & & 0 \\ & d_2 w_2 & & \\ & & \ddots & \\ 0 & & & d_n w_n \end{bmatrix}$

2. $|D| = d_1 d_2 \cdots d_n$

3. D จะเป็น Nonsingular ก็ต่อเมื่อแต่ละค่าในส่วนของ Diagonal ไม่เป็นศูนย์

4. ถ้าแต่ละค่าของ $d_i \neq 0$ ดังนั้น

$$D^{-1} = \begin{bmatrix} \frac{1}{d_1} & & & 0 \\ & \frac{1}{d_2} & & \\ & & \ddots & \\ 0 & & & \frac{1}{d_n} \end{bmatrix}$$

5. ค่า Eigenvalue ของ D คือค่าของ Main Diagonal $d_1, d_2, \ldots, d_n$

นิยาม 3

ให้ A เป็น Matrix ขนาด $n \times n$ ดังนั้น A จะถูกเรียกว่า ***Diagonalizable*** ก็ต่อเมื่อมี Matrix P ขนาด $n \times n$ ที่ทำให้ $P^{-1}AP$ เป็น Diagonal Matrix

Theorem 3:

ให้ A เป็น $n \times n$ Matrix ที่มีค่า Eigenvalue เท่ากับ $\lambda_1, \lambda_2, \ldots, \lambda_n$ และให้ $V_1, V_2, \ldots, V_n$ เป็น Eigenvector ที่สัมพันธ์กันตามลำดับ สมมุติว่า Eigenvector เหล่านี้เป็น Linearly Independent และให้ P เป็น Matrix ขนาด $n \times n$ ที่มีส่วนประกอบของ Column ที่ j คือ V_j ดังนั้น P จะเป็น Nonsingular และ

$$P^{-1}AP = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

ซึ่งเป็น Diagonal Matrix ที่มีค่า Diagonal คือ Eigenvalue ของ A ตามลำดับของ Eigenvector ที่สัมพันธ์กัน

สังเกตว่า A ไม่จำเป็นที่จะต้องมีค่า Eigenvalue ที่แตกต่างกันทั้งหมด n ตัว เพราะที่เราต้องการคือ Eigenvector n ตัวที่เป็น Linearly Independent ซึ่งกันและกัน

ตัวอย่างที่ 5

ให้ $A = \begin{bmatrix} -1 & 4 \\ 0 & 3 \end{bmatrix}$ เราพบว่าจะมี Eigenvalue คือ -1 และ 3 ด้วย Eigenvector $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ และ $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ ตามลำดับ สังเกตว่า Eigenvector ทั้งสองนั้นเป็น Linearly Independent เราสร้าง Matrix P โดยใช้ Eigenvector เป็น Column คือ

$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ และเราสามารถหา Inverse ได้เป็น $P^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$ ดังนั้น

$$P^{-1}AP = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} -1 & 4 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 3 \end{bmatrix}$$

Eigenvector อื่นๆที่สัมพันธ์กับ Eigenvalue นั้นๆสามารถนำมาใช้เพื่อทำให้ A เป็น Diagonal Matrix ได้ เช่นเราใช้ $\begin{bmatrix} 3 \\ 0 \end{bmatrix}$ และ $\begin{bmatrix} -2 \\ -2 \end{bmatrix}$ และเราสร้าง Matrix $Q = \begin{bmatrix} 3 & -2 \\ 0 & -2 \end{bmatrix}$ ซึ่งมี Inverse $Q^{-1} = \begin{bmatrix} 1/3 & -1/3 \\ 0 & -1/2 \end{bmatrix}$ และ

$$Q^{-1}AQ = \begin{bmatrix} \frac{1}{3} & -\frac{1}{3} \\ 0 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} -1 & 4 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 3 & -2 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 3 \end{bmatrix}$$

ตัวอย่างที่ 6

ให้ $A = \begin{bmatrix} -1 & 0 & 5 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix}$ เราหา Eigenvalue ได้เป็น $1,-1,-2$ และ Eigenvector ที่สัมพันธ์กันได้เป็น $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ และ $\begin{bmatrix} 5 \\ 0 \\ -1 \end{bmatrix}$ ตามลำดับ สังเกตว่าแต่ละ Eigenvector เป็น Linearly Independent ดังนั้นเราสามารถสร้าง

$$P = \begin{bmatrix} 0 & 1 & 5 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \text{ และหา } P^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 5 \\ 0 & 0 & -1 \end{bmatrix}$$

ขอให้นักศึกษาตรวจสอบว่า $P^{-1}AP = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix}$

Theorem 4:

ให้ A เป็น Diagonalizable Matrix ขนาด $n \times n$ ดังนั้น A จะมี Eigenvector จำนวน n ตัวที่เป็น Linearly Independent ยิ่งไปกว่านั้น ถ้า $Q^{-1}AQ$ เป็น Diagonal Matrix ดังนั้นค่าของ Diagonal จะเป็น Eigenvalue ของ A และ Column ของ Q จะประกอบด้วย Eigenvector ที่สัมพันธ์กัน

จากทฤษฎีที่ 4 กล่าวว่า Necessary and Sufficient Condition สำหรับ Matrix ที่สามารถ Diagonal ได้นั้นคือจะต้องมี Eigenvector ที่เป็น Linearly Independent n ตัว มิฉะนั้นแล้ว Matrix จะทำ Diagonal ไม่ได้ และการทำ Diagonal จะต้องใช้ค่า Eigenvector เท่านั้น นอกจากนี้ Diagonal Matrix ที่ได้จะประกอบด้วย Eigenvalue

Orthogonal and Symmetric

นิยาม 4

เราเรียก A ว่าเป็น Orthogonal Matrix ถ้า $AA^T = A^T A = I_n$ หรือ $A^{-1} = A^T$

Theorem 5:

ถ้า A เป็น Orthogonal Matrix ดังนั้น $|A| = \pm 1$

นิยาม 5

Matrix A ขนาด $n \times n$ จะเป็น Symmetric ก็ต่อเมื่อ $A = A^T$

Theorem 6:

ถ้า A เป็น Real Symmetric Matrix ค่า Eigenvalue จะเป็นค่า Real

Theorem 7:

ถ้า A เป็น Real Symmetric Matrix ดังนั้น Eigenvector ที่สัมพันธ์กับแต่ละ Eigenvalue จะ Orthogonal

Theorem 8:

ถ้า A เป็น Real Symmetric Matrix ดังนั้นจะมี Orthogonal Matrix ที่สามารถ Diagonalize A

- จาก Theorem ที่กล่าว จุดที่สำคัญคือ Theorem ที่ 8 สุดท้าย กล่าวคือ
  - ถ้า Matrix เป็น Real Symmetric เราจะได้ Eigenvalue เป็นค่าจริง และจะมี Orthogonal Matrix ที่จะมา Diagonalize ได้
  - เมื่อ Matrix เป็น Orthogonal มันจะหา Inverse ได้ง่าย
  - ถ้าเรา Normalized แต่ละ Eigenvector ที่คำนวณได้ให้มี Magnitude เท่ากับหนึ่ง เราจะได้ Orthogonal Matrix ที่สามารถ Diagonalize Matrix ที่ต้องการ

Break

Example 3.7 จงหา Orthogonal Matrix ที่จะมา Diagonalize Matrix $\mathbf{A}$ ดังนี้

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \sqrt{2} \\ 0 & 2 & 0 \\ \sqrt{2} & 0 & 0 \end{bmatrix}$$

คำตอบ หา Eigenvalue ของ Matrix จาก Characteristic Equation $|\lambda\mathbf{I} - \mathbf{A}| = 0$ เราได้

$$\begin{vmatrix} \lambda - 1 & 0 & -\sqrt{2} \\ 0 & \lambda - 2 & 0 \\ -\sqrt{2} & 0 & \lambda \end{vmatrix} = 0 \text{ หรือ}$$

$(\lambda - 1)(\lambda - 2)\lambda - \sqrt{2}(\sqrt{2}(\lambda - 2)) = 0$ เมื่อจัดเรียงสมการ เราได้

$$\lambda^3 - 3\lambda^2 + 4 = 0$$

เมื่อแก้สมการ เราได้ $\lambda = 2, 2, -1$

ค่า Eigenvector ที่สัมพันธ์กับ $\lambda = -1$ หาได้จากสมการ $(\lambda\mathbf{I} - \mathbf{A})\mathbf{X} = 0$ ดังนี้

$$\begin{bmatrix} -2 & 0 & -\sqrt{2} \\ 0 & -3 & 0 \\ -\sqrt{2} & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

เราได้

$$\mathbf{X}_1 = \alpha \begin{bmatrix} 1 \\ 0 \\ -\sqrt{2} \end{bmatrix}$$ เมื่อทำการ Normalized โดยหารด้วยขนาดของ Vector เราได้

$$\mathbf{V}_1 = \begin{bmatrix} \frac{1}{\sqrt{3}} \\ 0 \\ -\sqrt{\frac{2}{3}} \end{bmatrix}$$

ค่า Eigenvector ที่สัมพันธ์กับ $\lambda = 2$ หาได้จากสมการ $(\lambda\mathbf{I} - \mathbf{A})\mathbf{X} = 0$ ดังนี้

$$\begin{bmatrix} 1 & 0 & -\sqrt{2} \\ 0 & 0 & 0 \\ -\sqrt{2} & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

ค่า Eigenvector ที่สัมพันธ์กับ $\lambda = 2$ หาได้จากสมการ $(\lambda\mathbf{I} - \mathbf{A})\mathbf{X} = 0$ ดังนี้

$$\begin{bmatrix} 1 & 0 & -\sqrt{2} \\ 0 & 0 & 0 \\ -\sqrt{2} & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

ในกรณีนี้ สังเกตว่า x_2 จะเป็นอะไรก็ได้ เราสมมุติให้เท่ากับ β และเราได้ $x_1 = \sqrt{2}x_3$

เราได้ Eigenvector เป็น

$$\mathbf{X}_2 = \begin{bmatrix} \sqrt{2}\gamma \\ \beta \\ \gamma \end{bmatrix}$$ โดยที่ β และ γ จะเป็นค่าใดก็ได้ และไม่ขึ้นต่อกัน ดังนั้นเราสามารถสร้างสอง

$$\mathbf{X}_2 = \begin{bmatrix} \sqrt{2}\gamma \\ \beta \\ \gamma \end{bmatrix}$$ โดยที่ β และ γ จะเป็นค่าใดก็ได้ และไม่ขึ้นต่อกัน ดังนั้นเราสามารถสร้างสอง

Eigenvector ที่เป็น Linearly Independent จาก $\mathbf{X}_2$ เช่นในสองกรณีที่ $\beta = 0$ และ $\gamma = 0$ คือ

$$\mathbf{V}' = \gamma \begin{bmatrix} \sqrt{2} \\ 0 \\ 1 \end{bmatrix} \quad \text{และ} \quad \mathbf{V}'' = \beta \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Note: เราเลือก 2 vector ที่ Independent กัน โดยการแทนค่า β และ γ อะไรก็ได้

เมื่อทำการ Normalized สอง Eigenvector หลัง เราได้

$$\mathbf{V}_2 = \begin{bmatrix} \sqrt{\frac{2}{3}} \\ 0 \\ \frac{1}{\sqrt{3}} \end{bmatrix} \quad \text{และ} \quad \mathbf{V}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

สังเกตว่าทั้งสาม Eigenvector เป็น Linearly Independent จากนั้นเราสร้าง Matrix $\mathbf{P}$

$$\mathbf{P}=\begin{bmatrix}\frac{1}{\sqrt{3}} & \sqrt{\frac{2}{3}} & 0\\ 0 & 0 & 1\\ -\sqrt{\frac{2}{3}} & \frac{1}{\sqrt{3}} & 0\end{bmatrix}$$

ขอให้นักศึกษาตรวจสอบดูว่า

$$\mathbf{P}^{-1}=\mathbf{P}^{T}=\begin{bmatrix}\frac{1}{\sqrt{3}} & 0 & -\sqrt{\frac{2}{3}}\\ \sqrt{\frac{2}{3}} & 0 & \frac{1}{\sqrt{3}}\\ 0 & 1 & 0\end{bmatrix} \text{ และ } \mathbf{P}\mathbf{P}^{T}=\mathbf{I}_3=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

นอกจากนี้แล้วให้ตรวจสอบดูว่า

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{P}=\begin{bmatrix}-1 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 2\end{bmatrix}$$

End of Chapter 3

- HW 3 Due Next Week
- Chapter 4 Probability Review

Chapter 4 Probability

- Concept and Definition
  - Experiments, Event, Outcomes, Sample Space
  - Laplace Definition, Definition from Set
- Independent and Mutual Exclusive
  - Axioms of Probability, Vein Diagram, Independent concept, ME Concept
- Conditional Probability and Bayes
- PDF and CDF Concept and Properties
  - Continuous and Discrete

Definition

- Outcome/Sample Point
  - ผลลัพท์ที่ได้จากการทดลอง หรือสุ่มตัวอย่าง
- Sample Space
  - Set ของผลลัพธ์ทั้งหมด
- Event
  - เงื่อนไขของการทดลอง
- กำหนด Event A, ถ้าทดลอง N ครั้ง และได้ผลลัพธ์เป็นไปตามเงื่อนไข A = N_A ครั้ง

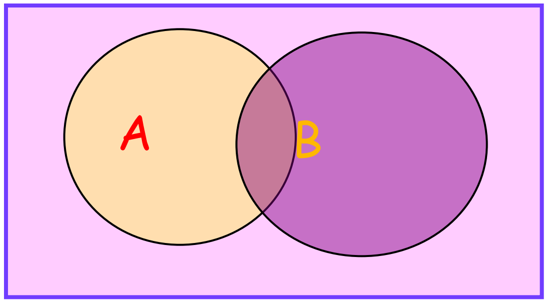
$$P[A] = \text{Probability of Event A Occur} = \lim_{N \to \infty} \frac{N_A}{N}$$

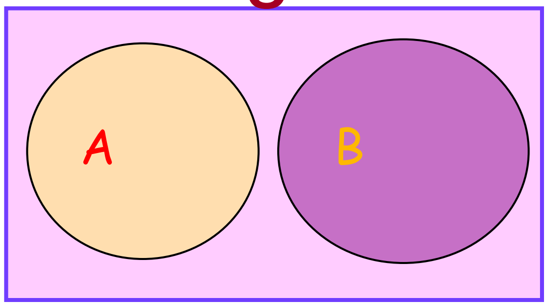
Example: Dice Roll

- Sample Space = {1,2,3,4,5,6}
- P(1)=P(2)= ... =P(6)=1/6
  - Laplace Definition of Probability
  - ถ้าแต่ละ Member ใน Sample Space มีโอกาสเกิดเท่าๆกัน
- A = Even
- A = {2,4,6}
- P(A) = |{2,4,6}|/|S| = 3/6 = ½

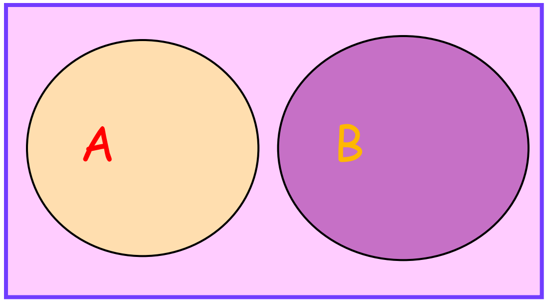
Mutually Exclusive

- A ME B
- เมื่อเกิด Event A จะเกิด Event B ไม่ได้
- เมื่อทอยลูเต๋าได้เลขคู่ จะไม่ใช่เลขคี่
  - ดังนั้นถ้า A = Even, B = Odd
  - A ME B
- P(A+B) or P(A union B) = P(A) + P(B)
- เห็นได้ชัดโดยแสดงด้วย Vein Diagram

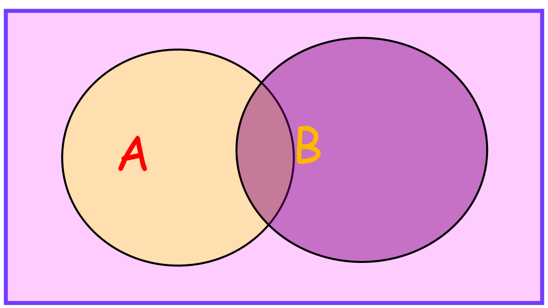




Mutually Exclusive



- ME
- $A \cap B = \varnothing$
- $A \cup B = A + B$
- $P(A \cup B) = P(A) + P(B)$



- Non ME
- $A \cap B \neq \varnothing$
- $A \cup B = A + B - A \cap B$
  - – Inclusion-Exclusion Principle

3 AXIOMS OF Probability

- 1. P(S) = 1
  - ทุกๆการทดลอง ผลลัพธ์ต้องอยู่ใน Sample Space
- 2. 0 =< P(A) =< 1
  - ค่าของ Probability ต้องอยู่ระหว่าง 0 และ 1
- 3. ME: P(A+B)=P(A)+P(B)
  - Mutually Exclusive; Probability ของ Union ของ Event เท่ากับผลบวกของ Probability ของแต่ละ Event

Conditional Probability

- Probability ของ Event หนึ่ง เมื่อกำหนดให้อีก Event หนึ่งได้เกิดขึ้น
  - Probability จะเพิ่มถ้าสอง Event เกี่ยวข้องกัน
  - Probability จะไม่เปลี่ยนถ้าสอง Event ไม่เกี่ยวกัน
    - เราเรียกว่าเป็น Statistical Independent

$$P[E_1 \setminus E_2] = \begin{cases} \dfrac{P[E_1 \cap E_2]}{P[E_2]} & ; P[E_2] \neq 0 \\ 0 & ; P[E_2] = 0 \end{cases}$$

    - นอกจากนี้

$$P[A \setminus B] = P[AB]/P[B], \quad P[B \setminus A] = P[AB\}/P[A]$$

$$P[A \setminus B]P[B] = P[B \setminus A]P[A]$$

$$P[A \setminus B] = P[B \setminus A]P[A]/P[B]$$

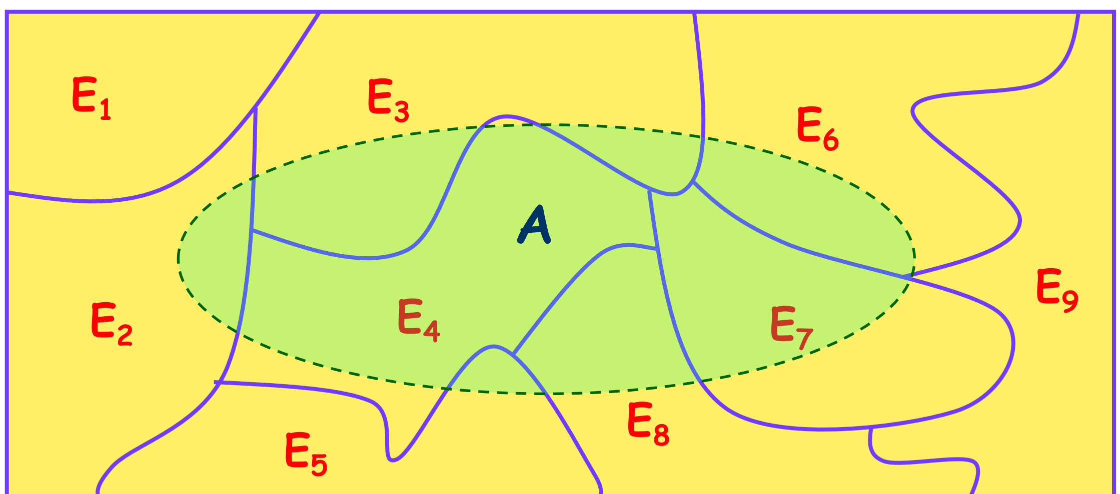
Bayes Rule

ถ้าใน Sample Space เราแบ่ง Event ออกเป็นทั้งหมด n Mutually Exclusive event $E_i; i = 1,\ldots,n$ เราได้ผลรวมของ Event $\bigcup_{i=1}^{n} E_i = \Omega$ และ $E_i \cap E_j = \phi; i \neq j$ ทำให้ $P[E_i \cap E_j] = 0$ ถ้า A เป็น Event ใดๆดังนั้น

$$P[A] = \sum_{i=1}^{n} P[E_i]P[A \backslash E_i]$$

<u>Bayes Rule:</u> $P[E_i \backslash A]$ สามารถหาได้จาก

$$P[E_i \backslash A] = \frac{P[E_i]P[A \backslash E_i]}{P[A]} = \frac{P[E_i]P[A \backslash E_i]}{\sum_{j=1}^{n} P[E_j]P[A \backslash E_j]}$$



Properties

(1) $P[\overline{E}]$ หรือ $P[E^c] = 1 - P[E]$

(2) $P[\phi] = 0$

(3) $P[E_1 \cup E_2] = P[E_1] + P[E_2] - P[E_1 \cap E_2]$

(4) ถ้า E_1 เป็น Subset ของ E_2 เขียน $E_1 \subseteq E_2$ ดังนั้น $P[E_1] \le P[E_2]$

Example 1

- ในการส่งข้อมูลแบบ Digital เป็น Frame ขนาด 50 บิต การส่งจะสมบูรณ์ได้ก็ต่อเมื่อทั้ง Frame ไปถึงอย่างถูกต้อง ถ้าการเกิด Error ในแต่ละบิตของการส่ง(BER = Bit Error Rate) มีโอกาสจะผิดพลาดได้เท่ากับ 1/1000 จงหาว่า Frame ที่ส่งจะมี Error เฉลี่ยแล้วกี่เปอร์เซ็นต์ (FER = Frame Error Rate) ถ้าการเกิด Error แต่ละ Bit เป็น Independent

$$P[\text{Bit Error}] = 10^{-3}$$

$$P[\text{Bit Not Error}] = 1 - 10^{-3} = 0.999$$

$$P[\text{Frame Not Error}] = P[\text{50 Bit Not Error}] = 0.999^{50}$$

$$P[\text{Frame Error}] = 1 - 0.999^{50} = 0.04879 = 4.879\%$$

Example 2

- จากสถิติ พบว่าโอกาสที่นักศึกษาชายจะสอบผ่านวิชา CPE332 มีค่าเท่ากับ 0.5 และโอกาสที่นักศึกษาหญิงจะสอบผ่านมีค่าเท่ากับ 0.4 ถ้าวิชา CPE332 เทอมนี้มีนักเรียนชาย 40 คนและนักเรียนหญิง 25 คน จงหาว่าเฉลี่ยแล้วจะมีนักเรียนตกกี่คน
  - สามารถคำนวนโดยใช้ค่าถ่วงน้ำหนัก
  - ให้ Sample Space ประกอบด้วยนักเรียนชาย(M) และนักเรียนหญิง(F) ดังนั้น S = {M,F}
    - สังเกตุว่า Set ทั้งสองเป็น ME คือ Sample Space ถูก Partition เป็นสอง Partition
    - P[M]=40/65 และ P[F]=25/65
  - ให้ Event A เป็นเหตุการณ์ที่นักศึกษาจะสอบผ่าน เราได้
  - P[A\M]=0.5 และ P[A\F]=0.4
  - จากสมการการ Partition

$$P[A] = \sum_{i=1}^{n} P[E_i]P[A \setminus E_i] = P[M]P[A \setminus M] + P[F]P[A \setminus F]$$

$$= \frac{40}{65} \cdot 0.5 + \frac{25}{65} \cdot 0.4 = 0.4615$$

$$P[\text{Fail CPE332}] = \text{P}[\text{A}^{\text{C}}] = 1 - P[A] = 1 - 0.4615 = 0.5385 = 53.85\%$$

Example3

- ต่อจากตัวอย่างที่ 2: ถ้าเราสุ่มตัวอย่างนักศึกษามาหนึ่งคนที่ลงวิชา CPE332 เมื่อเทอมที่แล้ว จากนั้นถามว่านักศึกษาผ่านวิชานี้หรือไม่ นักศึกษาผู้นั้นตอบว่าสอบผ่านแล้ว จงคำนวณว่า
  - 1.โอกาสที่นักศึกษาผู้นั้นจะเป็นผู้หญิงเท่ากับเท่าไร
  - 2. Probability ที่นักศึกษาสอบผ่านและเป็นผู้หญิงมีเท่าไร

- 1. ต้องการหา P[F\A]

$$P[F \backslash A] = P[FA]/P[A] = P[A \backslash F]P[F]/P[A]$$

$$= 0.4 \cdot \frac{25}{65} \cdot \frac{1}{0.4615} = 0.3334 = 33.34\%$$

- 2. ต้องการหา P[FA]=P[F∩A]

$$P[F \backslash A] = P[FA]/P[A]$$

$$P[FA] = P[F \backslash A]P[A] = P[A \backslash F]P[F]$$

$$= 0.3334 \cdot 0.4615 = 0.1539 = 15.39\%$$

Random Variables

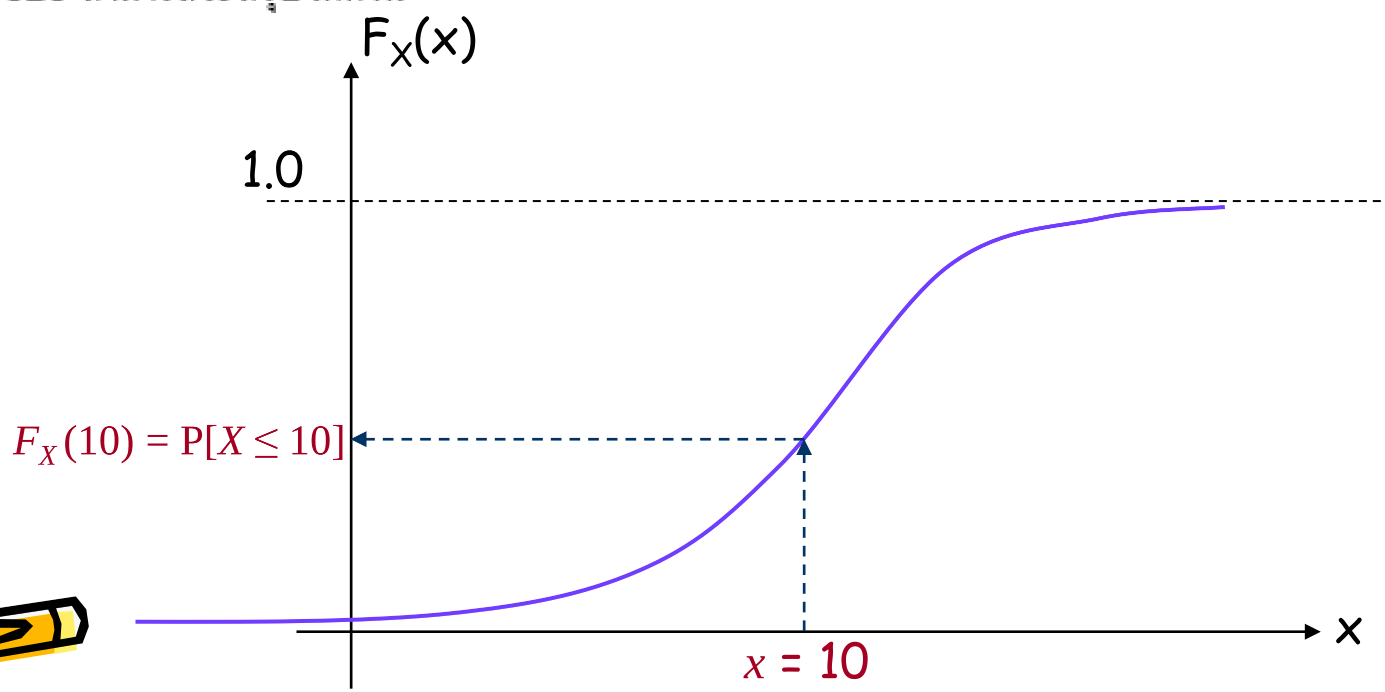
- เมื่อเรากำหนดค่าเป็นตัวเลขของทุกๆ Sample Point ใน Sample Space และถ้าให้ Variable แทนผลลัพธ์ที่ได้จากการทดลอง ดังนั้นผลของการทดลองจะมีค่าเป็นตัวเลข และเราเรียก Variable นั้นว่าเป็น **Random Variable** ซึ่งปกติแล้วเรามักจะให้ Random Variable แทนด้วยตัว Capital
- ที่สำคัญคือ การกำหนด RV ซึ่งเป็นตัวเลขทำให้เราสามารถนำไปคำนวณต่อทางคณิตศาสตร์ได้
  - Mean
  - Variance
  - Etc.

Random Variables

- เมื่อผลลัพธ์ของการทดลอง(Sample Space)ได้ Infinite Set การกำหนดตัวเลขมักจะเป็นตัวเลขที่ต่อเนื่อง
  - เราได้ Continuous Random Variable
- เมื่อผลลัพธ์เป็น Finite Set การกำหนดจะใช้ Set ของตัวเลข มักจะเป็น Integer
  - เราได้ Discrete Random Variable
- Probability ที่จะได้ผลลัพธ์การทดลองหนึ่งๆ คือ Probability ที่ Random Variable จะมีค่าตามที่เรากำหนด
- เราสามารถแสดงคุณสมบัติของ RV จากการ Plot ค่า Probability(y-axis) และค่าของ Random Variable(x-axis)
  - Cumulative Distribution Function (CDF)
  - Probability Density Function (PDF)

CDF:Cumulative Distribution Function of RV X

ถ้าให้ Function $F_X(x) = P[X \le x]$ และ Plot Graph ระหว่าง $F_X(x)$ vs. x (ซึ่งก็คือ Graph $P[X \le x]$ vs. x) Graph ที่ได้เราเรียกว่า Cumulative Distribution Function หรือ CDF(บางตำราใช้ Probability Distribution Function) ลักษณะของ CDF จะเป็น Graph ที่จะไม่มีการลดค่าเนื่องจากเป็นค่าสะสม โดยจะเริ่มจากศูนย์ และมีค่าสูงสุดเป็นหนึ่ง คุณสมบัติของ CDF สามารถจะสรุปได้ดังนี้



CDF of Normal (Gaussian) Distribution: Continuous

CDF Properties

(1) $0 \le F_X(x) \le 1$

(2) $F_X(x)$ เป็น Non-decreasing Function

(3) $\lim_{x\to-\infty} F_X(x) = 0$ และ $\lim_{x\to+\infty} F_X(x) = 1$

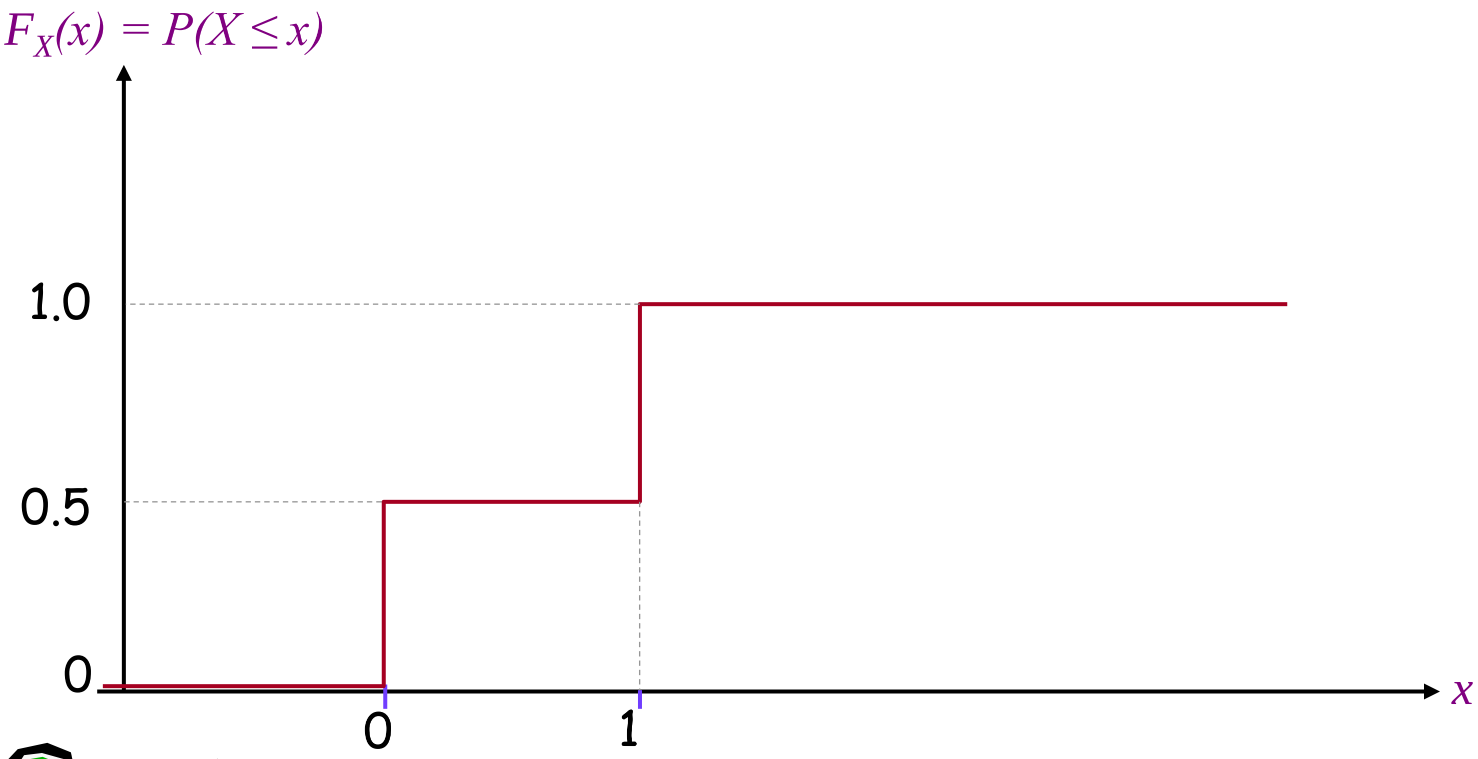
(4) $F_X(x)$ เป็น Function ที่ต่อเนื่องจากทางด้านขวา นั่นก็คือ $\lim_{\varepsilon\downarrow 0} F_X(x+\varepsilon) = F_X(x)$

(5) $P[a < X \le b] = F_X(b) - F_X(a)$

(6) $P[X = a] = F_X(a) - F_X(a^-)$

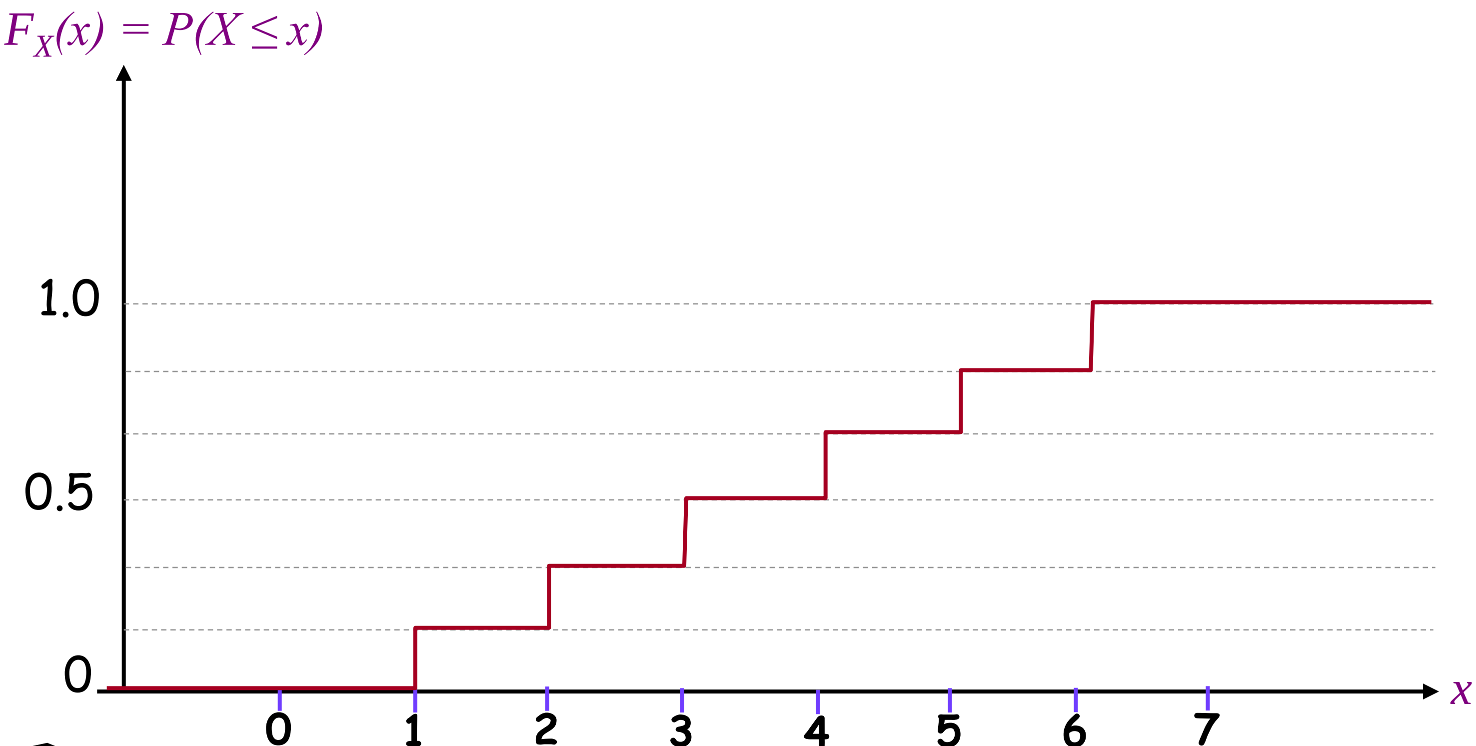
CDF การทอยเหรียญ: Discrete RV

- ให้ “หัว”= 0 และ “ก้อย” = 1



CDF การทอยลูกเต๋า: Discrete RV

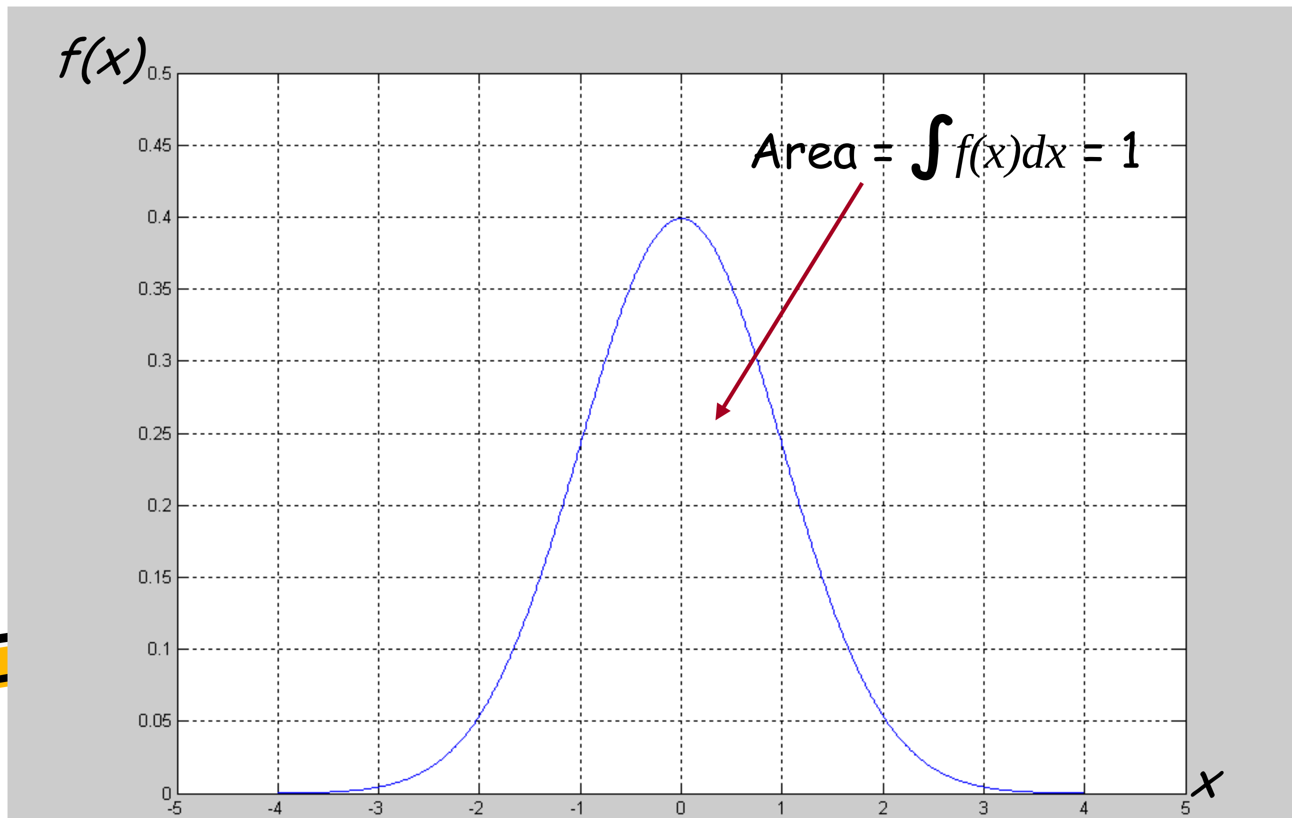
- ให้ ผลเป็นตัวเลขตามหน้าลูกเต๋า



PDF: Probability Density Function

ค่าของการเปลี่ยนแปลงใน CDF หรือ Slope ของมัน เมื่อ Plot กับแกน x เราจะได้ Probability Density Function

($f_X(x)$)หรือ PDF นั่นก็คือ $f_X(x) = \frac{dF_X(x)}{dx}$ และ $F_X(x) = \int_{-\infty}^{x^+} f_X(x)dx$ คุณสมบัติของ PDF มีดังนี้



PDF of Normal(Gaussian) Distribution : Continuous

Properties of PDF

(1) $f_X(x) \geq 0$

(2) $\int_{-\infty}^{\infty} f_X(x)dx = 1$

(3) $\int_{a^+}^{b^+} f_X(x)dx = P[a < x \leq b)$

(4) $P[X \in A] = \int_{AreaA} f_X(x)dx$

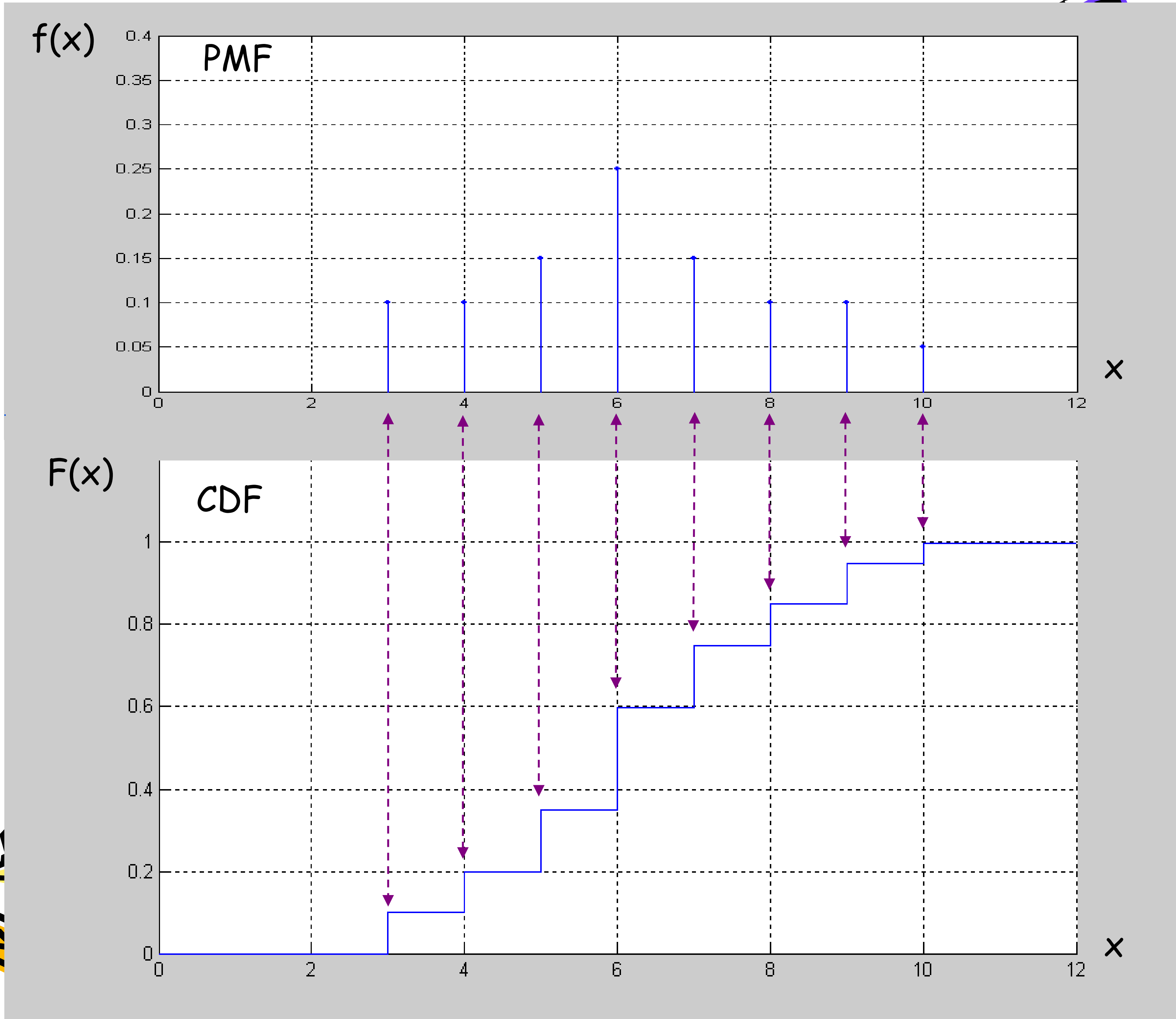
(5) $F_X(x) = \int_{-\infty}^{x^+} f_X(u)du$

ในกรณีของ Discrete Random Variable บางครั้งเราใช้คำว่า Probability Mass Function(PMF) แทน ซึ่งมีนิยามดังนี้

$$PMF = \{P_i\};\ P_i = P[X = x_i]$$

Discrete Version

– ค่าของ Variable ไม่ต่อเนื่อง
  - RV X มีค่าเฉพาะที่ $X=x_i$
– $F(x) = P(X \leq x)$
  - Function นี้มีความต่อเนื่องด้านขวามือ
  - นิยามสำหรับทุกจุดใน Domain ของ x
  - Function เป็นลักษณะขั้นบรรได
  - Monotonic Increasing Function จาก 0 ถึง 1
– $f(x_i) = P(X = x_i)$
  - นิยามเฉพาะจุด ไม่ต่อเนื่อง ค่าเป็นศูนย์ระหว่างนั้น
  - บางทีเรียก Probability Mass Function
  - $\sum f(x_i)=1$ เสมอ



f(x)
PMF
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 2 4 6 8 10 12
x
F(x)
CDF
1
0.8
0.6
0.4
0.2
0
0 2 4 6 8 10 12
x

Chapter 4 Cont. Next week

- Statistical Average
- Importance PDF
- Joint PDF
- Correlation/Covariance

- Chapter 5: Introduction to Random Process
  - Concept/Definition
  - Stationary Concept
  - Ergodic Concept
  - Autocorrelation and Cross Correlation

CPE 332
Computer Engineering Mathematics II

Part II, Chapter 4 Cont.: Statistical Average
Chapter 5: Random Process

Today Topics

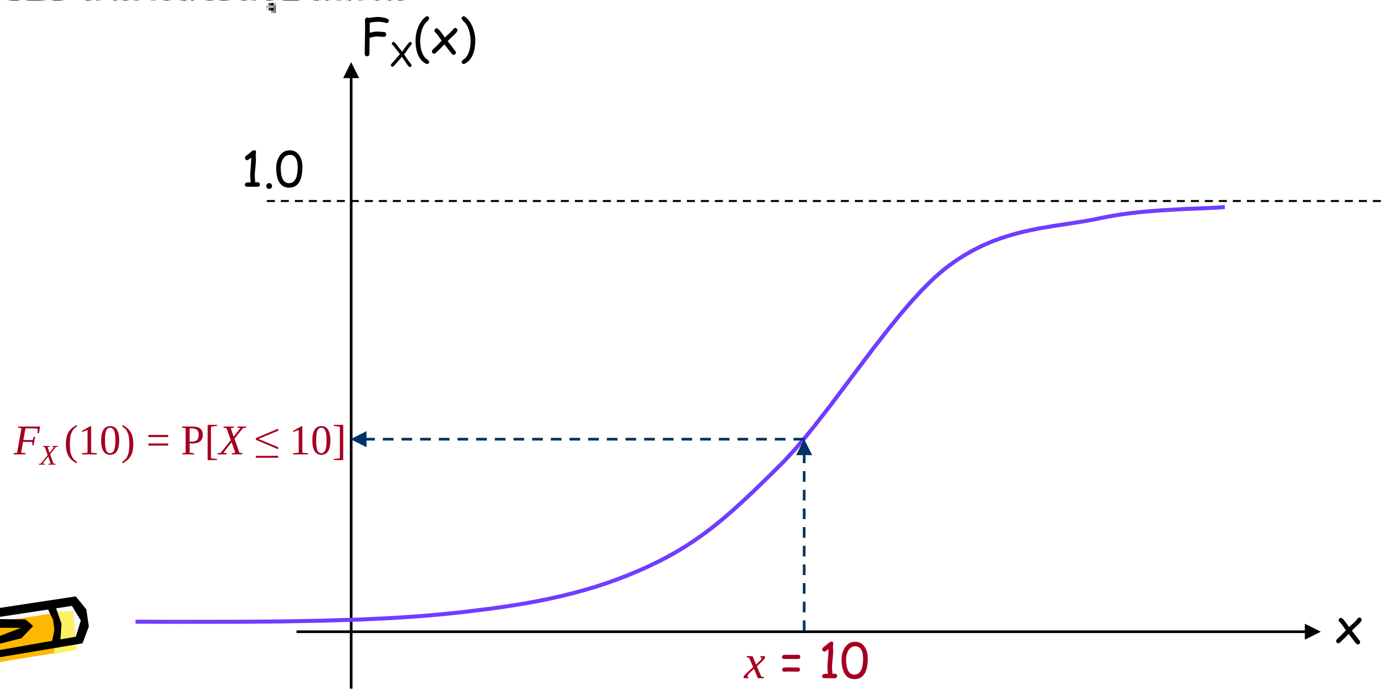
- HW 3 Due/ HW4 Due Next Week
- PDF
- Expectation
- Joint Density
- Correlation/Covariance
- Random Process
- Stationary
- Ergodic

Random Variables

- Mapping ผลลัพธ์เป็นตัวเลข (One-to-One) ดังนั้นผลการทดลองสามารถไปคำนวณต่อได้
  - ที่สำคัญคือค่าเฉลี่ยทางสถิติ
    - Mean
    - Variance
    - Etc.
  - และกราฟแสดงคุณสมบัติ Probability
    - Cumulative Distribution Function (CDF)
    - Probability Density Function (PDF)
  - เมื่อผลลัพธ์ของการทดลอง(Sample Space)ได้ Infinite Set เราได้ Continuous Random Variable
  - เมื่อผลลัพธ์เป็น Finite Set การกำหนดจะใช้ Set ของตัวเลขมักจะเป็น Integer เราได้ Discrete Random Variable

CDF:Cumulative Distribution Function of RV X

ถ้าให้ Function $F_X(x) = P[X \le x]$ และ Plot Graph ระหว่าง $F_X(x)$ vs. x (ซึ่งก็คือ Graph $P[X \le x]$ vs. x) Graph ที่ได้เราเรียกว่า Cumulative Distribution Function หรือ CDF(บางตำราใช้ Probability Distribution Function) ลักษณะของ CDF จะเป็น Graph ที่จะไม่มีการลดค่าเนื่องจากเป็นค่าสะสม โดยจะเริ่มจากศูนย์ และมีค่าสูงสุดเป็นหนึ่ง คุณสมบัติของ CDF สามารถจะสรุปได้ดังนี้



CDF of Normal (Gaussian) Distribution: Continuous

CDF Properties

(1) $0 \le F_X(x) \le 1$

(2) $F_X(x)$ เป็น Non-decreasing Function

(3) $\lim_{x \to -\infty} F_X(x) = 0$ และ $\lim_{x \to +\infty} F_X(x) = 1$

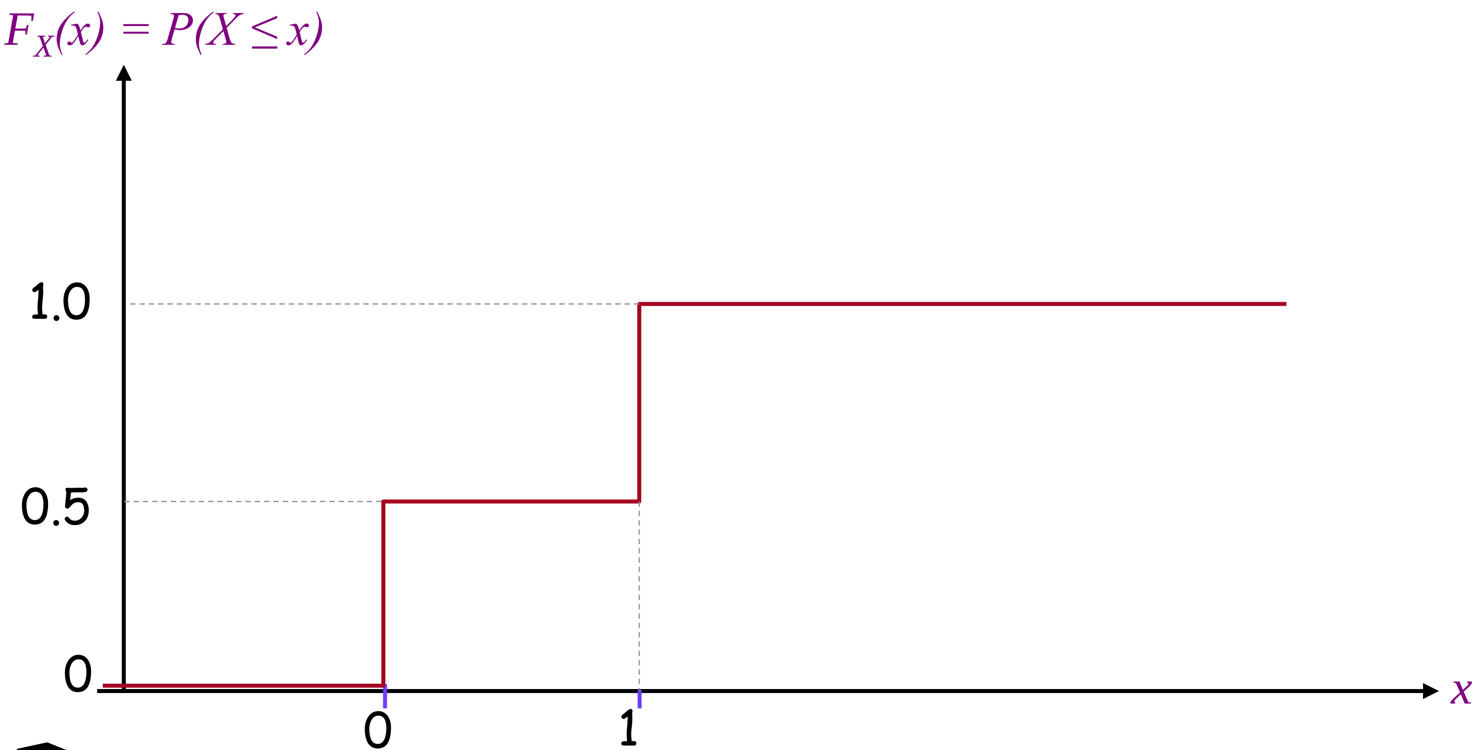
(4) $F_X(x)$ เป็น Function ที่ต่อเนื่องจากทางด้านขวา นั่นก็คือ $\lim_{\varepsilon \downarrow 0} F_X(x+\varepsilon) = F_X(x)$

(5) $P[a < X \le b] = F_X(b) - F_X(a)$

(6) $P[X = a] = F_X(a) - F_X(a^-)$

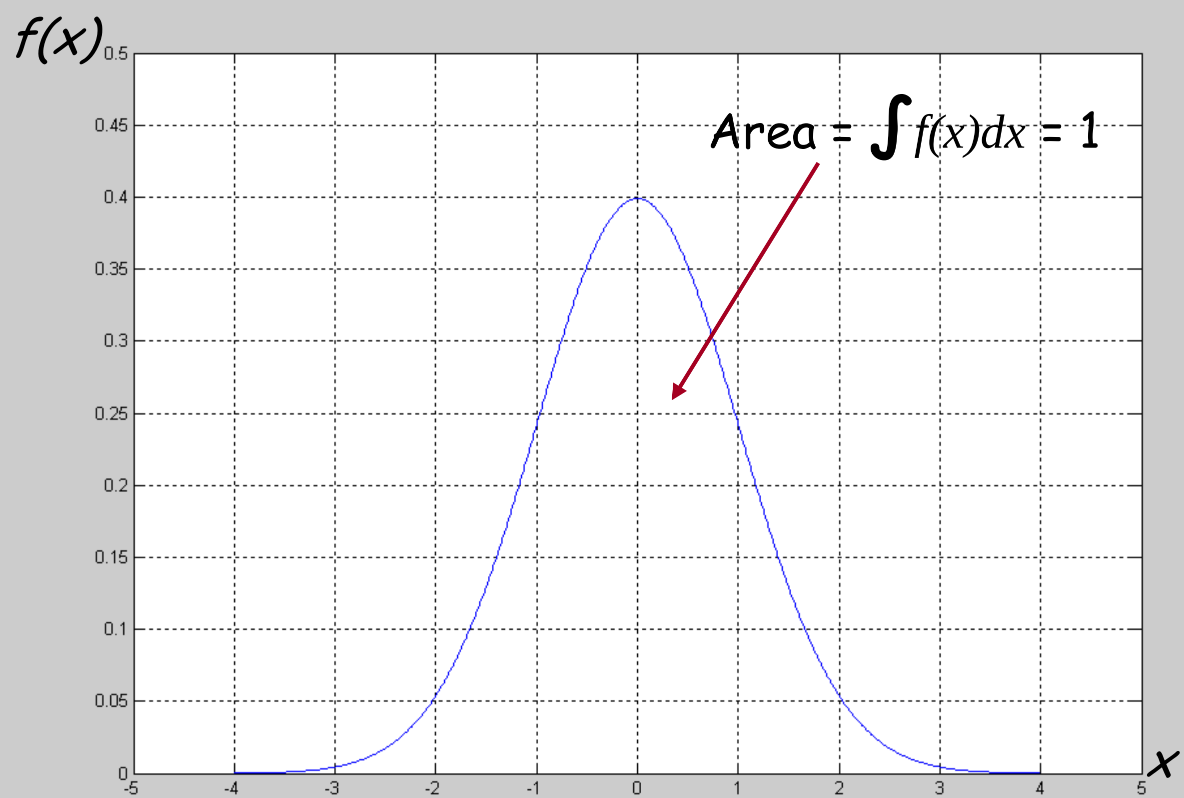
CDF การทอยเหรียญ: Discrete RV

- ให้ "หัว"= 0 และ "ก้อย" = 1



PDF: Probability Density Function

ค่าของการเปลี่ยนแปลงใน CDF หรือ Slope ของมัน เมื่อ Plot กับแกน x เราจะได้ Probability Density Function ($f_X(x)$)หรือ PDF นั่นก็คือ $f_X(x) = \dfrac{dF_X(x)}{dx}$ และ $F_X(x) = \int_{-\infty}^{x^+} f_X(x)dx$ คุณสมบัติของ PDF มีดังนี้



PDF of Normal(Gaussian) Distribution : Continuous

Properties of PDF

(1) $f_X(x) \geq 0$

(2) $\int_{-\infty}^{\infty} f_X(x)dx = 1$

(3) $\int_{a^+}^{b^+} f_X(x)dx = P[a < x \leq b)$

(4) $P[X \in A] = \int_{AreaA} f_X(x)dx$

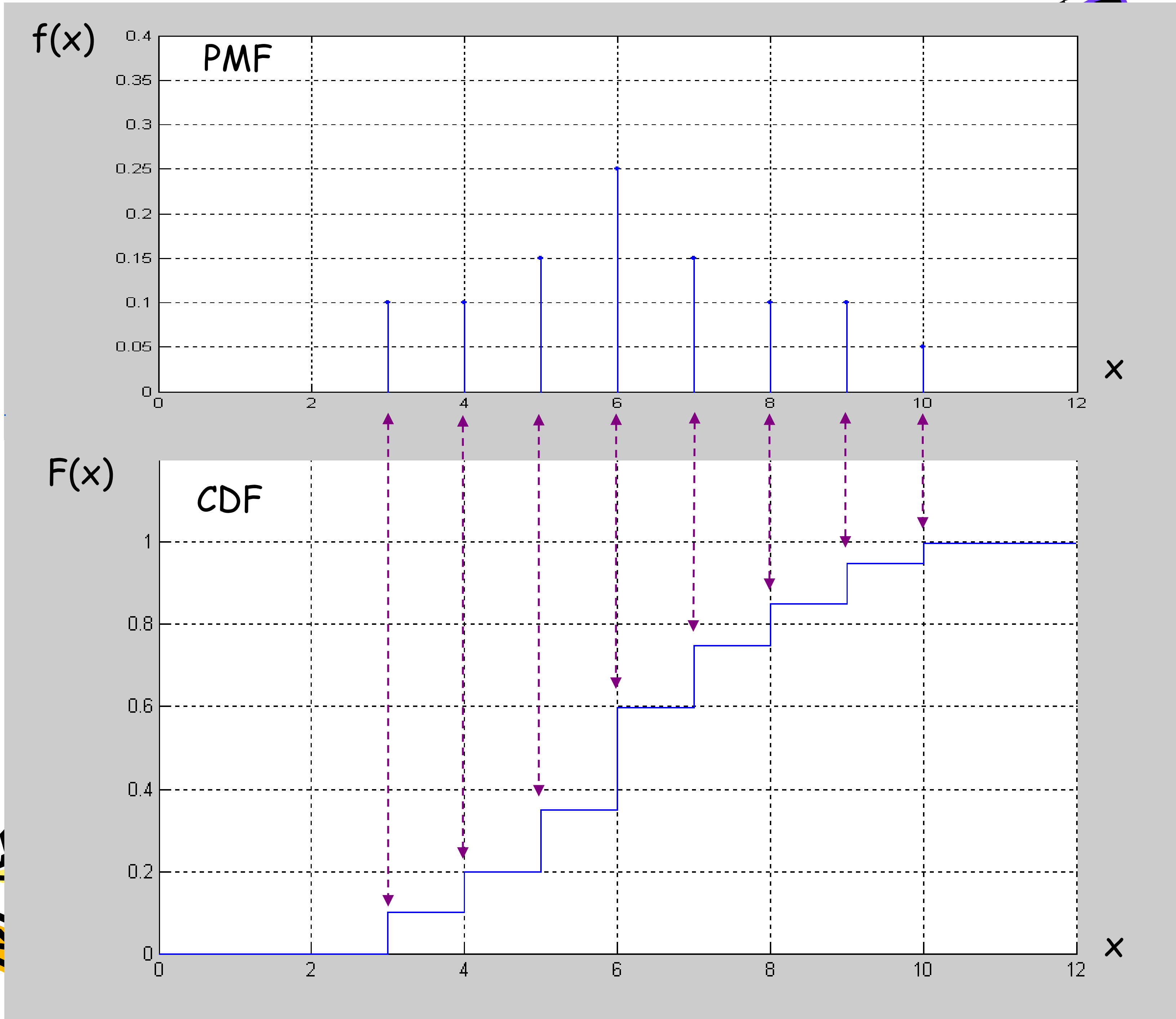
(5) $F_X(x) = \int_{-\infty}^{x^+} f_X(u)du$

ในกรณีของ Discrete Random Variable บางครั้งเราใช้คำว่า Probability Mass Function(PMF) แทน ซึ่งมีนิยามดังนี้

$$PMF = \{P_i\};\ P_i = P[X = x_i]$$

Discrete Version

- ค่าของ Variable ไม่ต่อเนื่อง
  - RV X มีค่าเฉพาะที่ $X=x_i$
- $F(x) = P(X \leq x)$
  - Function นี้มีความต่อเนื่องด้านขวามือ
  - นิยามสำหรับทุกจุดใน Domain ของ x
  - Function เป็นลักษณะขั้นบรรได
  - Monotonic Increasing Function จาก 0 ถึง 1
- $f(x_i) = P(X = x_i)$
  - นิยามเฉพาะจุด ไม่ต่อเนื่อง ค่าเป็นศูนย์ระหว่างนั้น
  - บางทีเรียก Probability Mass Function
  - $\sum f(x_i)=1$ เสมอ



f(x)
PMF
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
2
4
6
8
10
12
x
F(x)
CDF
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
x

Statistical Average

Random Variable สามารถที่จะอธิบายได้ด้วย PDF ของมัน แต่การแสดง PDF ของ RV บ่อยๆครั้งไม่จำเป็นและไม่เหมาะสม ยิ่งไปกว่านั้น ในปกติแล้ว การหา PDF ของ RV นั้นทำไม่ได้ง่ายๆ ดังนั้นเรามักจะแสดงคุณสมบัติของมันจากค่าเฉลี่ยทางสถิติซึ่งเราเรียกว่า **Expectation** หรือการคาดคะเนทางสถิติของมัน

ถ้ากำหนดให้ Function ของ RV X คือ $g(X)$ เราสามารถหา Expectation ของ $g(X)$ ได้จาก

$$E[g(X)] = \int_{-\infty}^{\infty} g(X) f_X(x) dx$$

พึงสังเกตไว้อย่างหนึ่งก็คือ Expectation หรือ $E[\cdot]$ เป็น Linear Operation นั่นก็คือ $E[X+Y]=E[X]+E[Y]$ นอกจากนี้แล้วเรายังได้ $E[C]=C$ และ $E[kX]=kE[X]$ โดยที่ C, k เป็นค่าคงที่

Discrete Version :

$$E[g(X)] = \sum_{\forall x_i \in X} g(x_i) p(x_i)$$

Importance Expectation

4.7.1 Mean (m_X หรือ μ_X หรือ $\overline{X}$) คือเมื่อ $g(X) = X$ เราได้ $E[X] = \overline{X} = \int_{-\infty}^{\infty} X f_X(x)dx$

4.7.2 Mean Square ($\overline{X^2}$) หรือ Second Moment นั่นก็คือ $g(X) = X^2$ และ $E[X^2] = \int_{-\infty}^{\infty} X^2 f_X(x)dx$

4.7.3 Variance (σ^2) หรือ Second Central Moment มาจาก $g(X) = (X - \overline{X})^2$ และ

$E[(X-\overline{X})^2] = \int_{-\infty}^{\infty} (X-\overline{X})^2 f_X(x)dx$ ค่านี้สามารถคำนวณได้จาก Linear Property ของ Expectation ดังนี้

$$\sigma^2 = E[(X-\overline{X})^2] = E[X^2] - E[2\overline{X}X] + E[\overline{X}^2] = \overline{X^2} - \overline{X}^2$$

Discrete Version :

$$E[X] = \sum_{\forall x_i \in X} x_i\, p(x_i) = \sum_{i=1}^{N} x_i\, p(x_i) = \frac{1}{N}\sum_{i=1}^{N} x_i \mid p(x_i) = \tfrac{1}{N}$$

$$E[X^2] = \sum_{\forall x_i \in X} x_i^2\, p(x_i) = \frac{1}{N}\sum_{i=1}^{N} x_i^2 \mid p(x_i) = \tfrac{1}{N}$$

$$\sigma_X^2 = \sum_{\forall x_i \in X} (x_i - \mu_X)^2\, p(x_i) = E[X^2] - E[X]$$

- Note:
  - ในกรณีของ Data ที่ได้จากการสุ่มตัวอย่าง เราได้เฉพาะค่า Estimate ของ Mean และ Mean Square เท่านั้น และ

From n Samples : $p(x_i) = \frac{1}{N}$, Each sample has the same probability

$$\hat{\mu}_X = \overline{X} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Estimate $E[X^2] = \overline{X^2} = \frac{1}{N}\sum_{i=1}^{N} x_i^2$

$\sigma_X^2 = E[X^2] - E[X]$ from estimation is biased.

We use the least biased formular :

Sample Variance(Estimated) : $V_X^2 = s_{N-1}^2 = \frac{1}{N-1}\left(\sum_{i=1}^{N}(x_i - \overline{X})^2\right)$

Multivariate: RV มากกว่า 1 ตัว

- จุดประสงค์ในการศึกษา RV มากกว่าหนึ่งตัวพร้อมๆกัน เพื่อจะหาความสัมพันธ์ระหว่าง RV
  - แสดงโดยกราฟ Joint PDF
  - หรือค่าเฉลี่ยทางสถิติ *Correlation* และ *Covariance*
- **กรณีของ Bivariate**
  - เป็นการศึกษาความสัมพันธ์ระหว่าง RV สองตัว คือ X และ Y แต่ละตัวอาจเป็น Discrete หรือ continuous
    - เช่นความสัมพันธ์ระหว่างส่วนสูงและอายุ
    - หรือความสัมพันธ์ระหว่างคะแนน Midterm และเกรดปลายเทอม
    - หรือความสัมพันธ์ระหว่าง *GPA* กับเงินเดือนที่ได้เมื่อจบการศึกษา

Multivariate: RV มากกว่า 1 ตัว

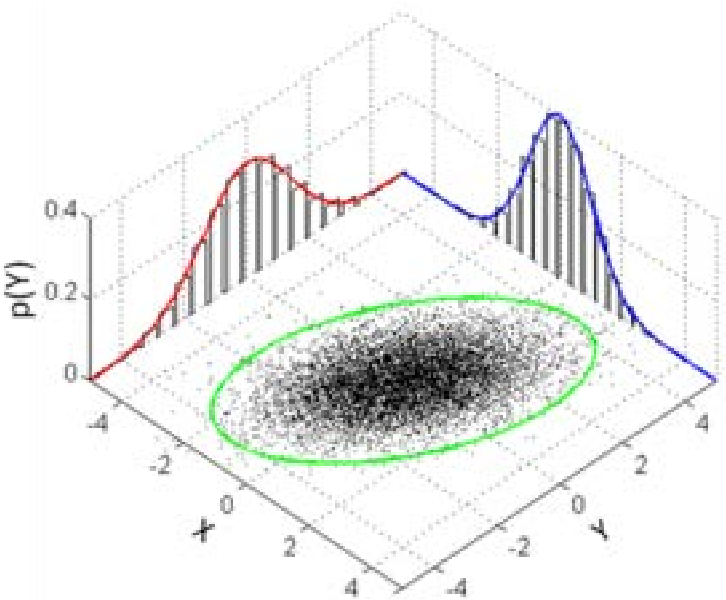
- แต่ละ X และ Y จะมี PDF ของมัน เรียก Marginal PDF บ่งบอกเฉพาะคุณสมบัติของตัวมันเอง ไม่เกี่ยวกับตัวอื่น
  - $f_X(x)$ และ $f_Y(y)$
- Marginal PDF ไม่มีข้อมูลแสดงความสัมพันธ์ของทั้งสอง RV
  - PDF ที่แสดงความสัมพันธ์ของสอง RV จะต้องเป็น Function ของทั้งสอง Variable และต่างจาก PDF แต่ละตัว

Multivariate: RV มากกว่า 1 ตัว

- PDF ที่เป็น Function ของทั้งสอง RV เรียก Joint PDF จะมีข้อมูลแสดงความสัมพันธ์เพิ่มเข้ามา (สำหรับ 2 RV ต้อง Plot แบบ 3D)
  - $f_{XY}(x, y)$ Joint PDF ไม่สามารถหาได้จาก Marginal PDF อย่างเดียว
  - ยกเว้นเฉพาะกรณีพิเศษที่ X และ Y ไม่ขึ้นต่อกันเท่านั้น (Statistical Independent) ที่ทำให้สมการล่างนี้เป็นจริง
    - $f_{XY}(x, y) = f_X(x)\, f_Y(y)$
- แต่เราสามารถแสดงได้ว่า Marginal Density
    - $f_X(x) = \int_{-\infty}^{\infty} f_{XY}(x, y)dy$ และ

Joint CDF/Marginal Density

- $F_{XY}(x,y) = P[X \leq x, Y \leq y] = \int_{-\infty}^{x}\int_{-\infty}^{y} f_{XY}(x,y)\,dx\,dy$
- Joint PDF จะมี Marginal Density ตาม PDF ของแต่ละ Random Variable
  - $f_X(x) = \int_{-\infty}^{\infty} f_{XY}(x,y)\,dy$
  - $f_Y(y) = \int_{-\infty}^{\infty} f_{XY}(x,y)\,dx$



Joint Moment

4.7.4 Joint Moment ในกรณีที่ Random Variable มากกว่าหนึ่งตัว เราได้ Joint Moment $E[g(X,Y)]$ ดังนั้นเรานิยามให้ค่า Expectation ของ Joint Moment เป็นดังนี้

$$E[g(X,Y)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(X,Y) f_{X,Y}(x,y)\,dx\,dy$$

โดยที่ $f_{X,Y}(x,y)$ คือ Joint Probability Density Function ของทั้งสอง Random Variable สังเกตว่าปกติแล้วการรู้เพียงแต่ PDF ของแต่ละ Variable นั้นจะมีข้อมูลไม่เพียงพอที่จะหา PDF ของ Joint Variable ได้ ยกเว้นในกรณีพิเศษที่ทั้งสอง Random Variable นั้นเป็น Statistical Independent ซึ่งในกรณีนี้เราได้ $f_{X,Y}(x,y) = f_X(x) f_Y(y)$

Correlation

- G(x,y) = XY

4.7.5 Correlation ในกรณี $g(X,Y)=XY$ เราได้ Correlation ของ RV 2 ตัวเรียก Cross Correlation หาได้จาก $E[XY]$ ซึ่งเราเรียกว่า Cross Correlation ของ X และ Y ให้เครื่องหมาย R_{XY} ดังนั้นเราสามารถเขียน

$$R_{XY}=E[XY]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}XYf_{XY}(x,y)\,dxdy$$

ในกรณีของ Statistical Independent เราจะได้ $R_{XY}=E[XY]=E[X]E[Y]$

$$\text{Discrete Version}: R_{XY}=\sum_{i=1}^{N}\sum_{j=1}^{M}x_i y_j p(x_i,y_j)=\sum_{i=1}^{N}\sum_{j=1}^{M}x_i y_j p(x_i)p(y_j)$$

$$=\sum_{i=1}^{N}x_i p(x_i)\sum_{i=1}^{M}y_i p(y_i)\,|\,\text{independent}=\frac{1}{N}\sum_{i=1}^{N}x_i\frac{1}{M}\sum_{i=1}^{M}y_i\,|\,p(x_i)=\tfrac{1}{N},\,p(y_i)=\tfrac{1}{M}$$

Covarience

4.7.6 Covariance ถ้า $g(X,Y)=(X-m_X)(Y-m_Y)$ เมื่อเราทำ Expectation เราได้ Covariance หรือ C_{XY} ดังนั้น

$$C_{XY}=E[(X-m_{X=})(Y-m_Y)]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(X-m_X)(Y-m_Y)f_{XY}(x,y)dxdy$$

$$=E[XY]-m_XE[Y]-m_YE[X]+m_Xm_Y=R_{XY}-m_Xm_Y$$

อีกค่าที่สำคัญก็คือ Correlation Coefficient หรือ ρ ซึ่งก็คือ Normalized Covariance กล่าวคือ

$$\rho=\frac{C_{XY}}{\sigma_X\sigma_Y}$$

Correlation and Covarience

4.7.7 Notes เกี่ยวกับ Correlation

- RV 2 ตัว จะ Uncorrelated ก็ต่อเมื่อ $C_{XY} = 0$
- RV 2 ตัว จะ Orthogonal ก็ต่อเมื่อ $R_{XY} = E[XY] = 0$
- ถ้า X หรือ Y ตัวใดตัวหนึ่งมี Zero Mean และทั้งสองตัว Orthogonal กัน ดังนั้นมันจะ Uncorrelated
- ถ้า RV 2 ตัวเป็น Statistically Independent ทั้งสองตัวจะ Uncorrelated แต่ส่วนกลับจะไม่เป็นจริงเสมอไป ยกเว้นในกรณีที่ทั้งสองตัวเป็น Jointly Gaussian
- ในกรณีที่ $Y = X$ เราได้ $C_{XY} = \sigma_X^2$

การประมาณค่า Rxy และ Cxy จาก Samples

- เราเก็บตัวอย่างเป็นคู่ (x_i,y_i) จำนวน N คู่
- Probability ของการได้แต่ละตัวอย่างเท่ากัน คือ 1/N
- ดังนั้น

$$R_{XY} = E[XY] = \sum_i \sum_j x_i y_j P(x_i, y_j) = \sum_{i=1}^{N} x_i y_i (\tfrac{1}{N}) = \frac{1}{N} \sum_{i=1}^{N} x_i y_i$$

- และ

$$C_{XY} = E[(X - m_X)(Y - m_Y)] = \sum_{i=1}^{N} (x_i - m_X)(y_i - m_Y)(\tfrac{1}{N})$$

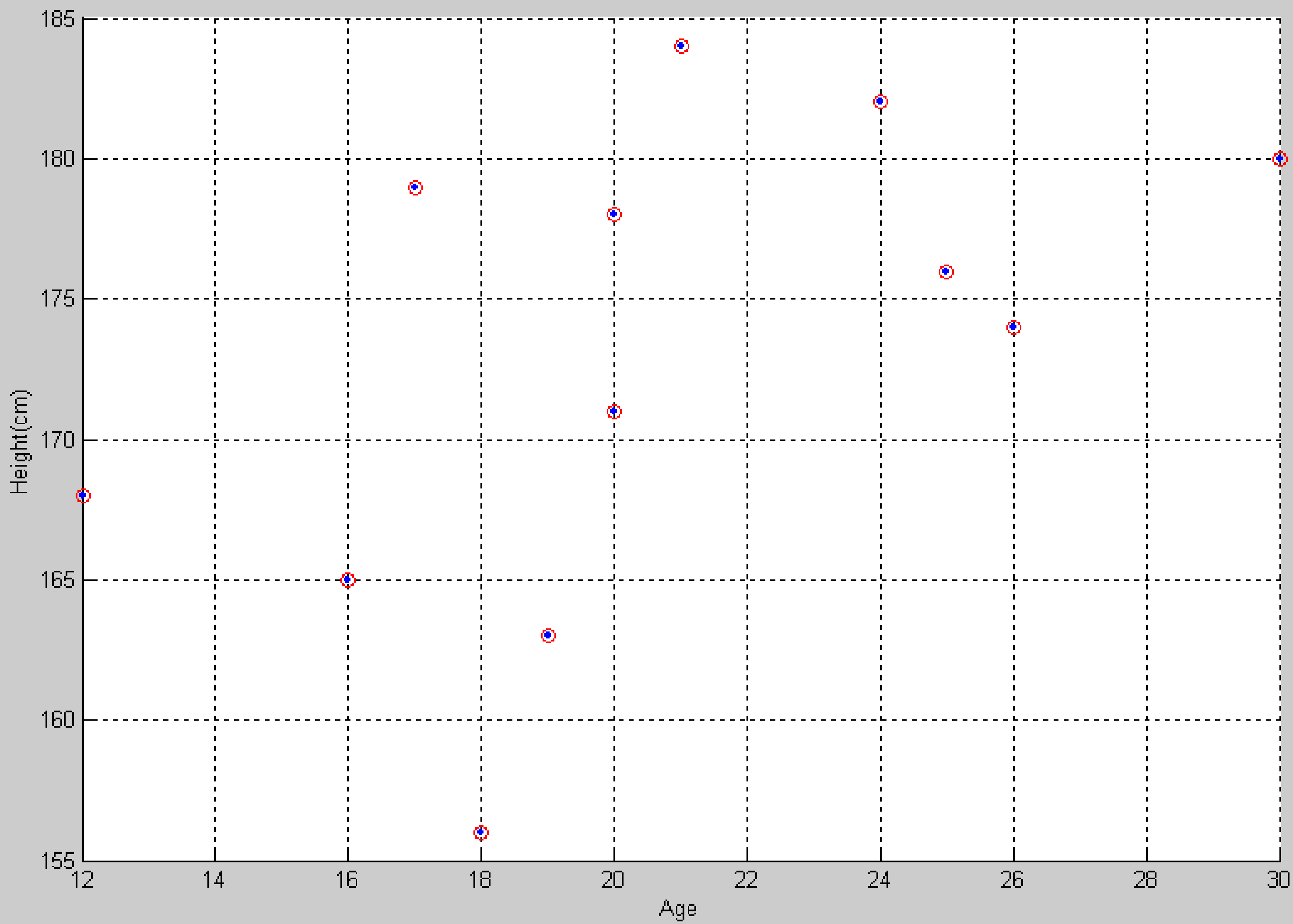
$$= \frac{1}{N} \sum_{i=1}^{N} (x_i - m_X)(y_i - m_Y) = R_{XY} - m_X m_Y$$

EX.จงหา Rxy และ Cxy จากตัวอย่างในตารางข้างล่าง

- ข้อมูลจากชายไทย อายุระหว่าง 12 - 30 ปี
- X=อายุ และ Y=ส่วนสูง

No.(i)	x_i	y_i	No.(i)	x_i	y_i
1	20	178	7	12	168
2	25	176	8	18	156
3	19	163	9	26	174
4	21	184	10	20	171
5	30	180	11	24	182
6	16	165	12	17	179

Scatter Diagram



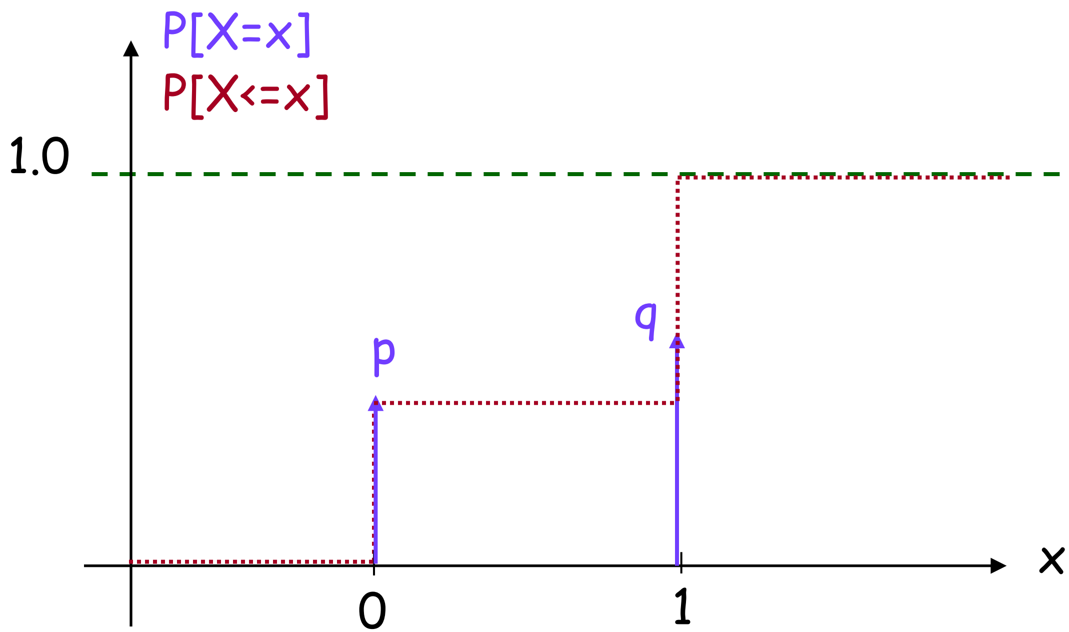
Calculation Table

i	x_i	y_i	$x_i y_i$	$x_i - m_X$	$y_i - m_y$	$(x_i - m_X)(y_i - m_Y)$
1	20	178	3560	-0.6667	5	-3.3333
2	25	176	4400	4.3333	3	13.0000
3	19	163	3097	-1.6667	-10	16.6667
4	21	184	3864	0.3333	11	3.6667
5	30	180	5400	9.3333	7	65.3333
6	16	165	2640	-4.6667	-8	37.3333
7	12	168	2016	-8.6667	-5	43.3333
8	18	156	2808	-2.6667	-17	45.3333
9	26	174	4524	5.3333	1	5.3333
10	20	171	3420	-0.6667	-2	1.3333
11	24	182	4368	3.3333	9	30.0000
12	17	179	3043	-3.6667	6	-22.0000
$\sum_{i=1}^{12}(.)$	248	2076	43140	0	0	236
Average (N=12)	20.6667	173	$R_{XY} = 3595$	0	0	$C_{XY} = 19.6667$

Notes: เราสามารถคำนวณ $C_{XY} = R_{XY} - m_X m_Y = 3595 - (20.\dot{6})(173) = 19.\dot{6}$

PDF ที่สำคัญ

4.8.1 Bernoulli Trial สมมุติว่าการทดลองมี Outcome ที่เป็นไปได้เพียง 2 แบบ เช่น 0 หรือ 1, ถูก หรือ ผิด, สำเร็จ หรือ ไม่สำเร็จ ลักษณะของการทดลองแต่ละครั้งเหล่านี้เราเรียก Bernoulli Trial โดยถ้าให้ p และ q เป็นค่า Probability ของ Outcome ทั้งสอง เราจะได้ $p+q=1$ หรือ $q=1-p$

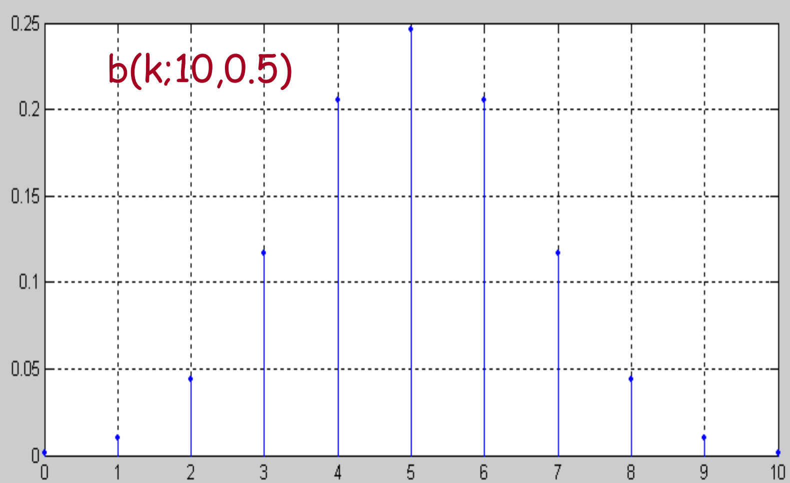


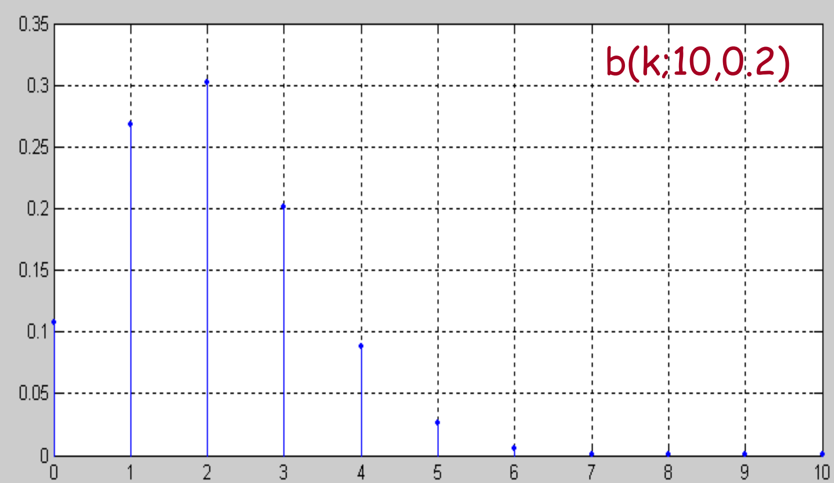
Binomial Distribution

4.8.2 Binomial Distribution มีปัญหาหลายๆแบบสามารถแก้ไขได้จากการหา Probability ที่จะสำเร็จ k ครั้งจากการทำ Bernoulli Trial ทั้งหมด n ครั้ง Probability ที่จะประสพผลสำเร็จเท่ากับ k ครั้งในการทำ Bernoulli Trial ที่ไม่ขึ้นต่อกันทั้งหมด n ครั้ง ที่มีค่า Probability ในการประสพผลสำเร็จแต่ละครั้งเท่ากับ p และไม่สำเร็จ $q = 1 - p$ จะหาได้จาก $C(n,k)p^k q^{n-k}$ ดังนั้น $p(X = k) = C(n,k)p^k q^{n-k}$ และ $E(X) = np$, $\sigma_X^2 = npq$

ค่า Probability ที่จะสำเร็จ k ครั้ง เป็น Function กับ k จัดเป็น Distribution Function ที่สำคัญอันหนึ่งที่เรียก Binomial Distribution, เขียน $b(k;n,p) = C(n,k)p^k q^{n-k}$ สังเกตว่า $\sum_{k=0}^{n} C(n,k)p^k q^{n-k} = (p+q)^n = 1$

$$C(n,k) = C_k^n = {}^nC_k = \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

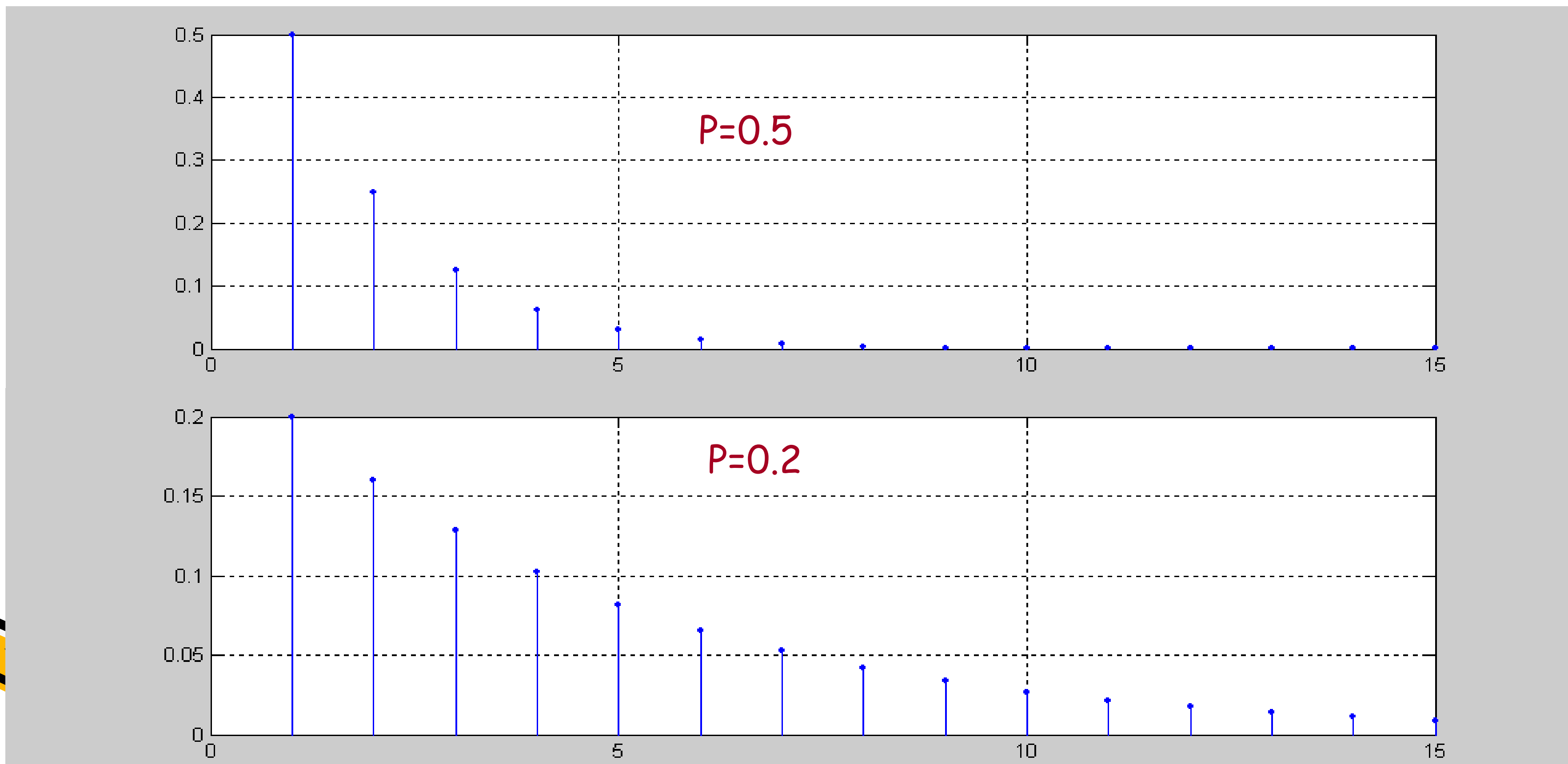




Geometric Distribution

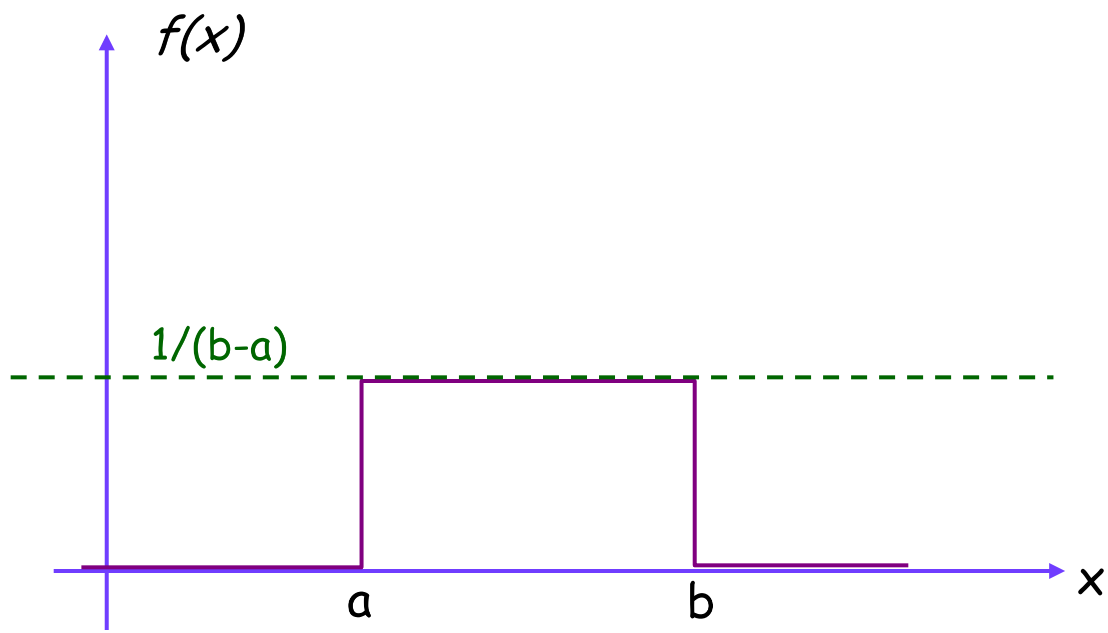
4.8.3 Geometric Distribution Random Variable X จะเป็น Geometric Distribution ด้วย Parameter p ถ้า

$$p(X=k)=(1-p)^{k-1}p \text{ สำหรับค่า } k=1,2,3,\ldots \text{ และค่า } E(X)=\frac{1}{p},\ \sigma_X^2=(1-p)/p^2$$



Uniform Distribution

4.8.4 Uniform Distribution $P(X=k)=1/(b-a)$ ถ้า X อยู่ระหว่าง $[a,b]$ และจะเป็นศูนย์ในกรณีอื่นๆ ในกรณีนี้เราได้ $E(X)=(a+b)/2,\ \sigma_X^2=(b-a)^2/12$



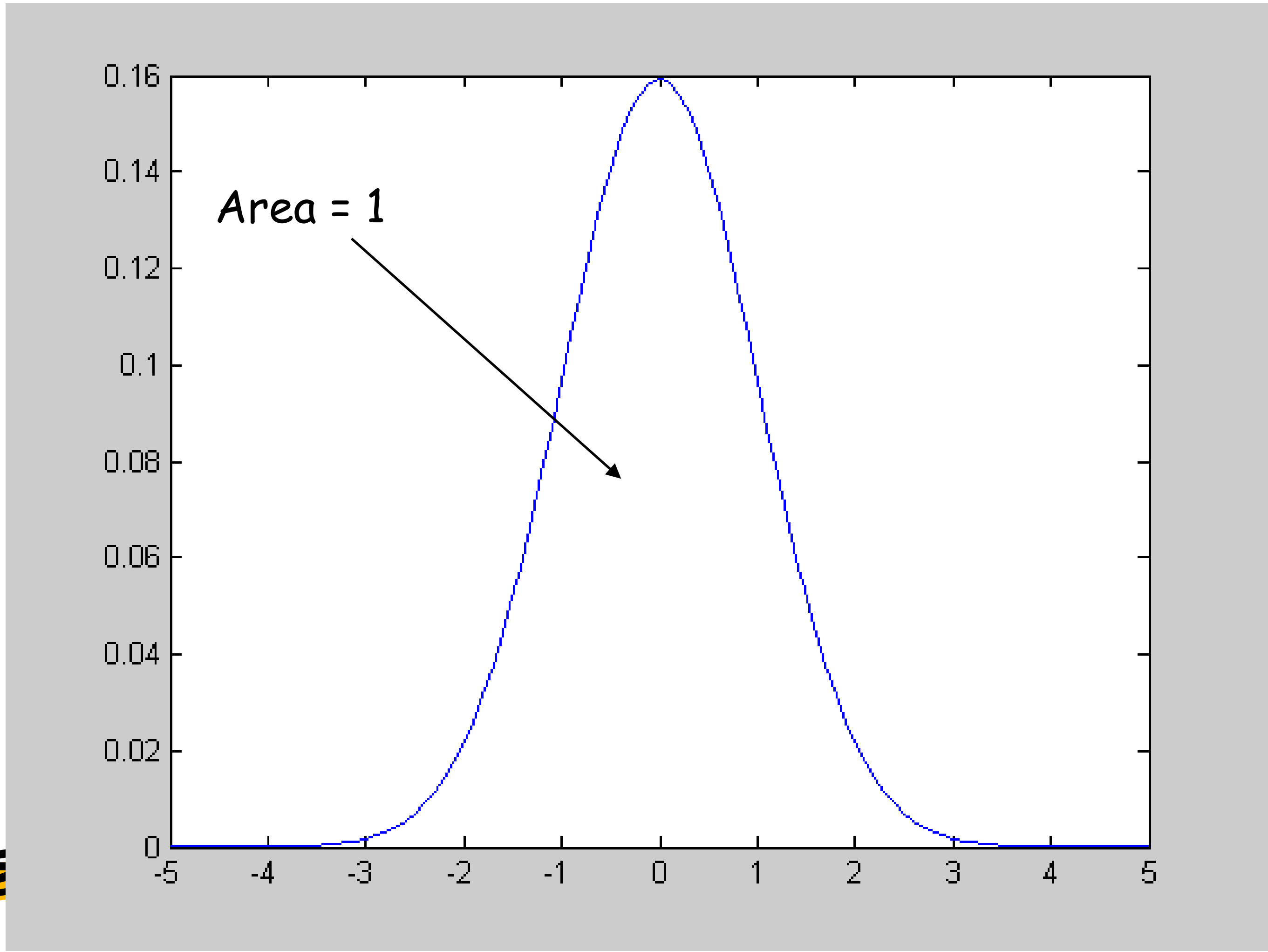
Gaussian Distribution

4.8.5 Normal Distribution(Gaussian Distribution) ด้วย Parameter μ,σ สามารถหาได้จาก

$$N(\mu,\sigma) = f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

สังเกตว่า Function ดังกล่าวไม่สามารถหา Integration ในลักษณะของ Closed Form ได้

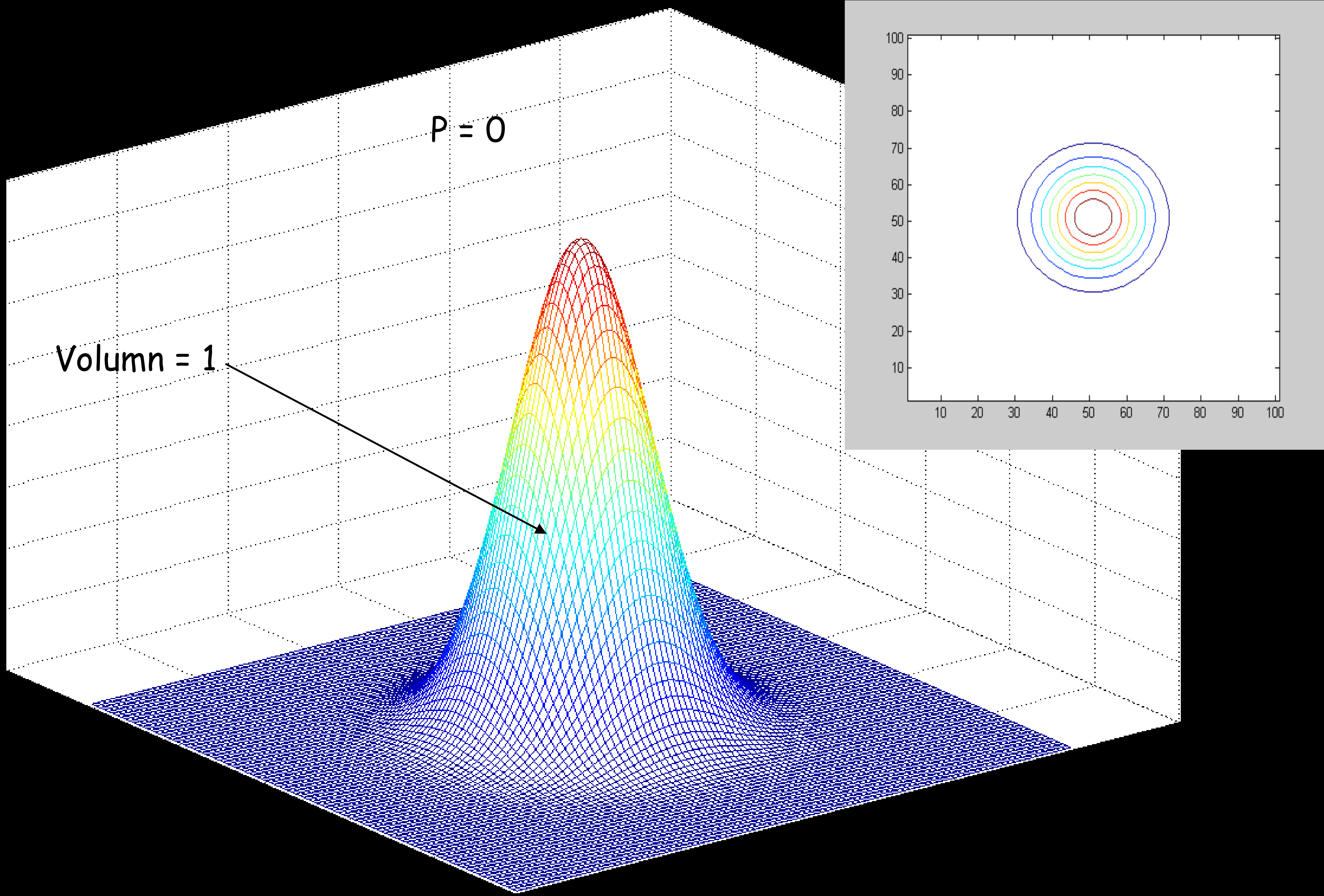
Gaussian

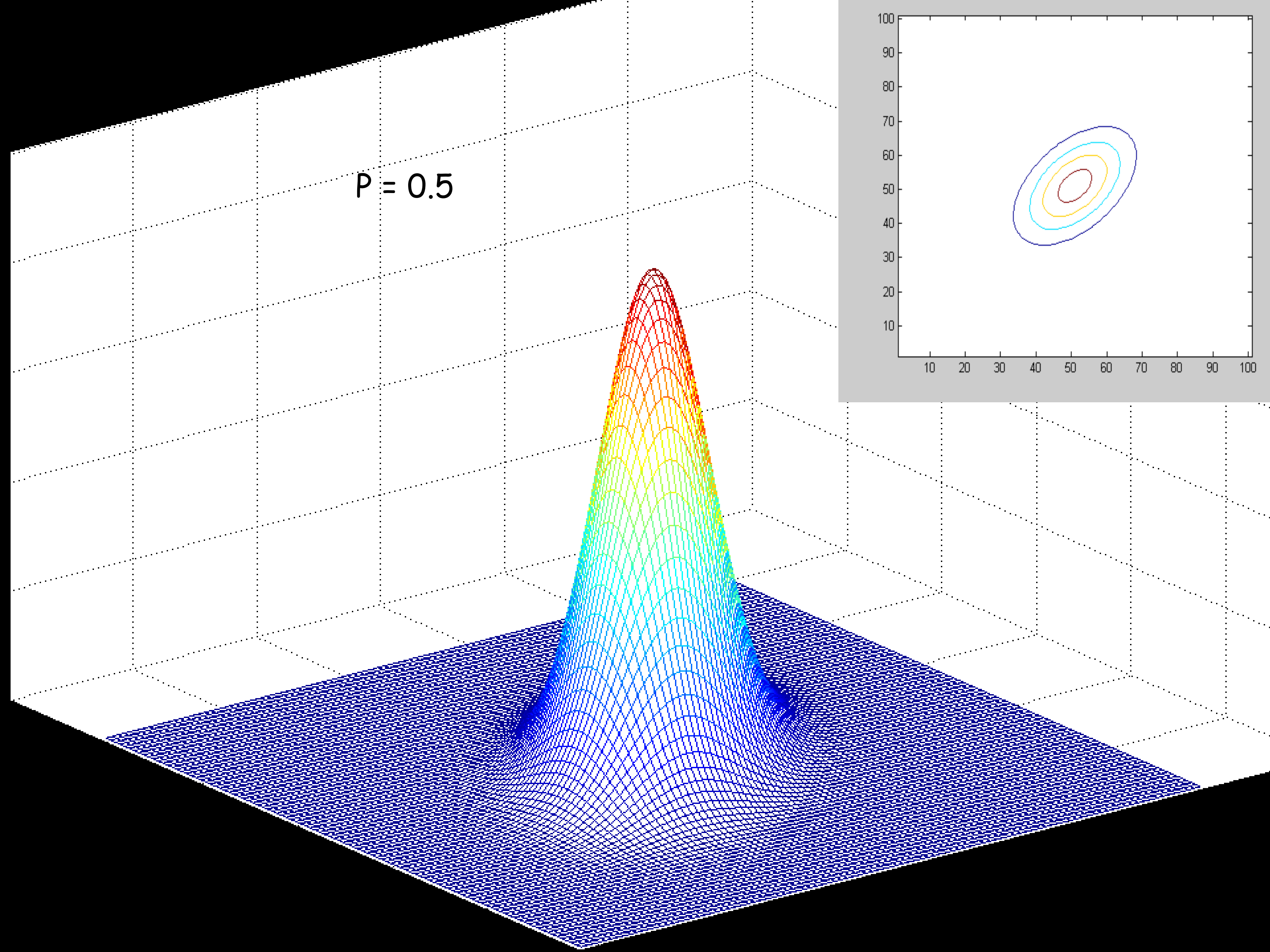


Jointly Gaussian: X, Y

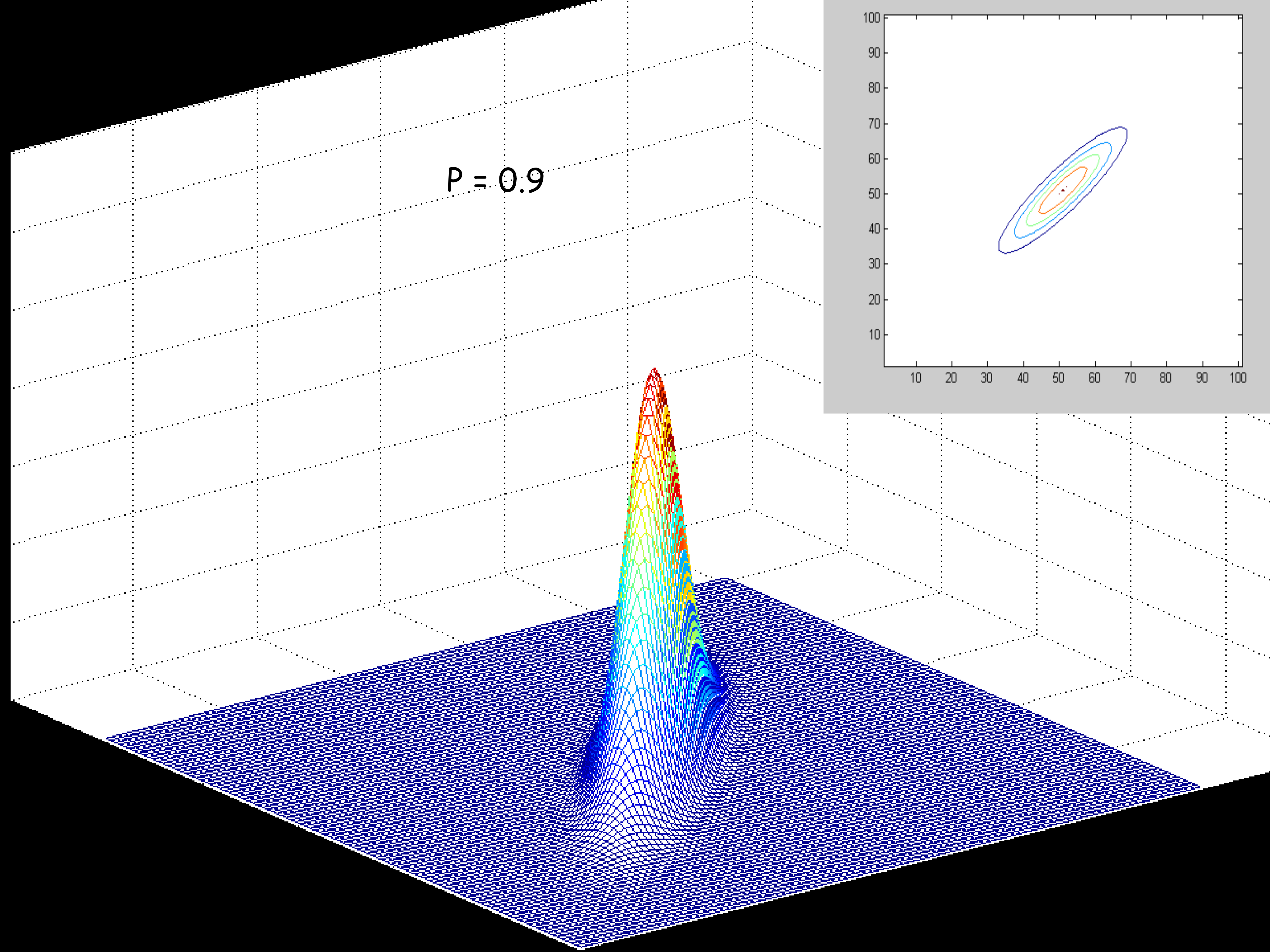
$$f_{XY}(x,y) = \frac{1}{2\cdot\pi\cdot\sigma_X\cdot\sigma_Y\cdot\sqrt{1-\rho}}\cdot e^{\left[-\frac{1}{2\cdot\left(1-\rho^2\right)}\cdot\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2-\frac{2\cdot\rho\cdot\left(x-\mu_X\right)\cdot\left(y-\mu_Y\right)}{\sigma_X\sigma_Y}+\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right]}$$

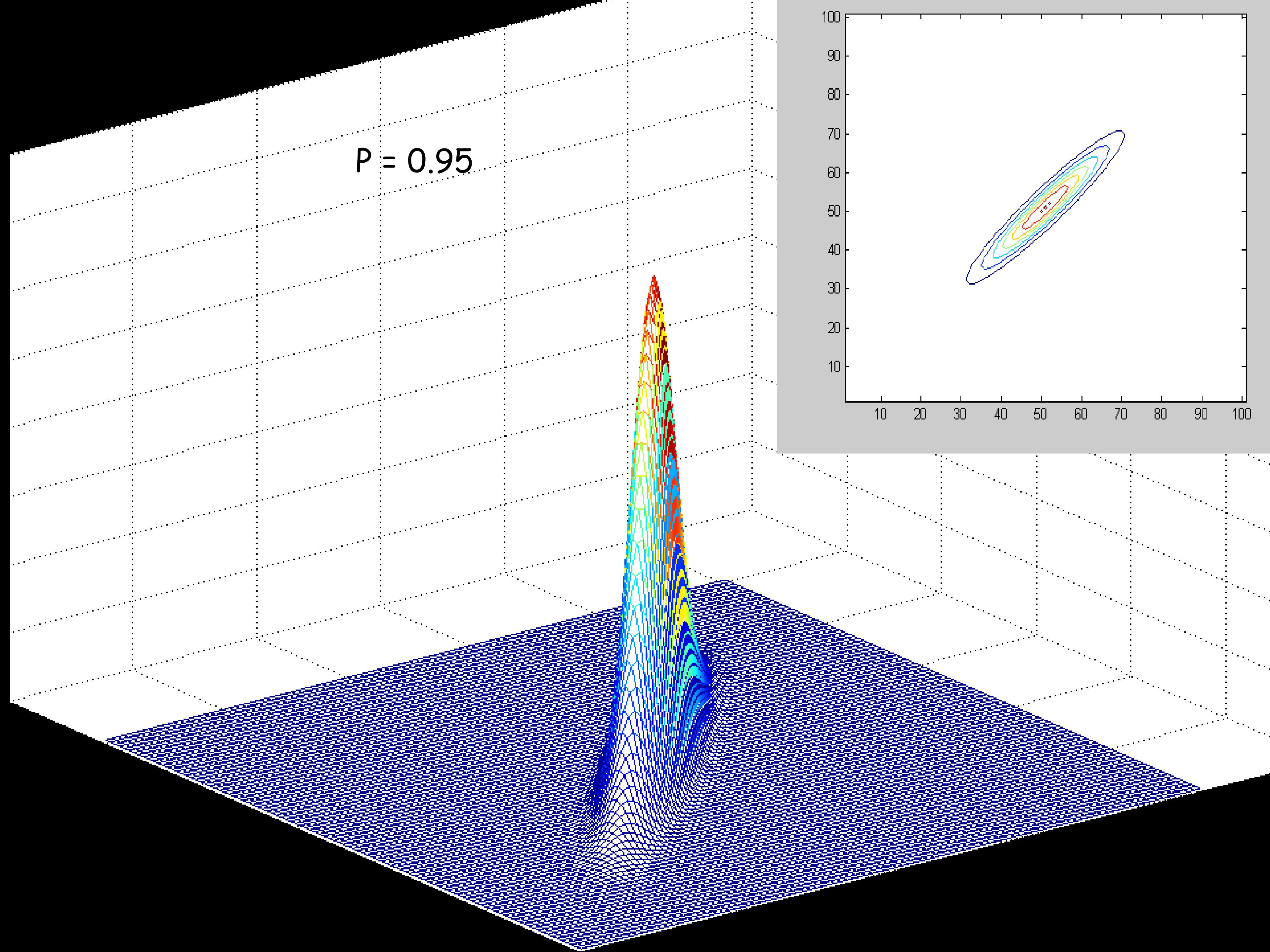
- ค่า ρ คือค่า Correlation Coefficient บ่งบอกถึงความสัมพันธ์ของสอง Variable จะมีค่าระหว่าง [-1,1]
- $\rho = \frac{cov(X,Y)}{\sigma_X\sigma_Y}$
- $f_{XY}(x,y)$ เป็น Joint PDF ที่มี Marginal

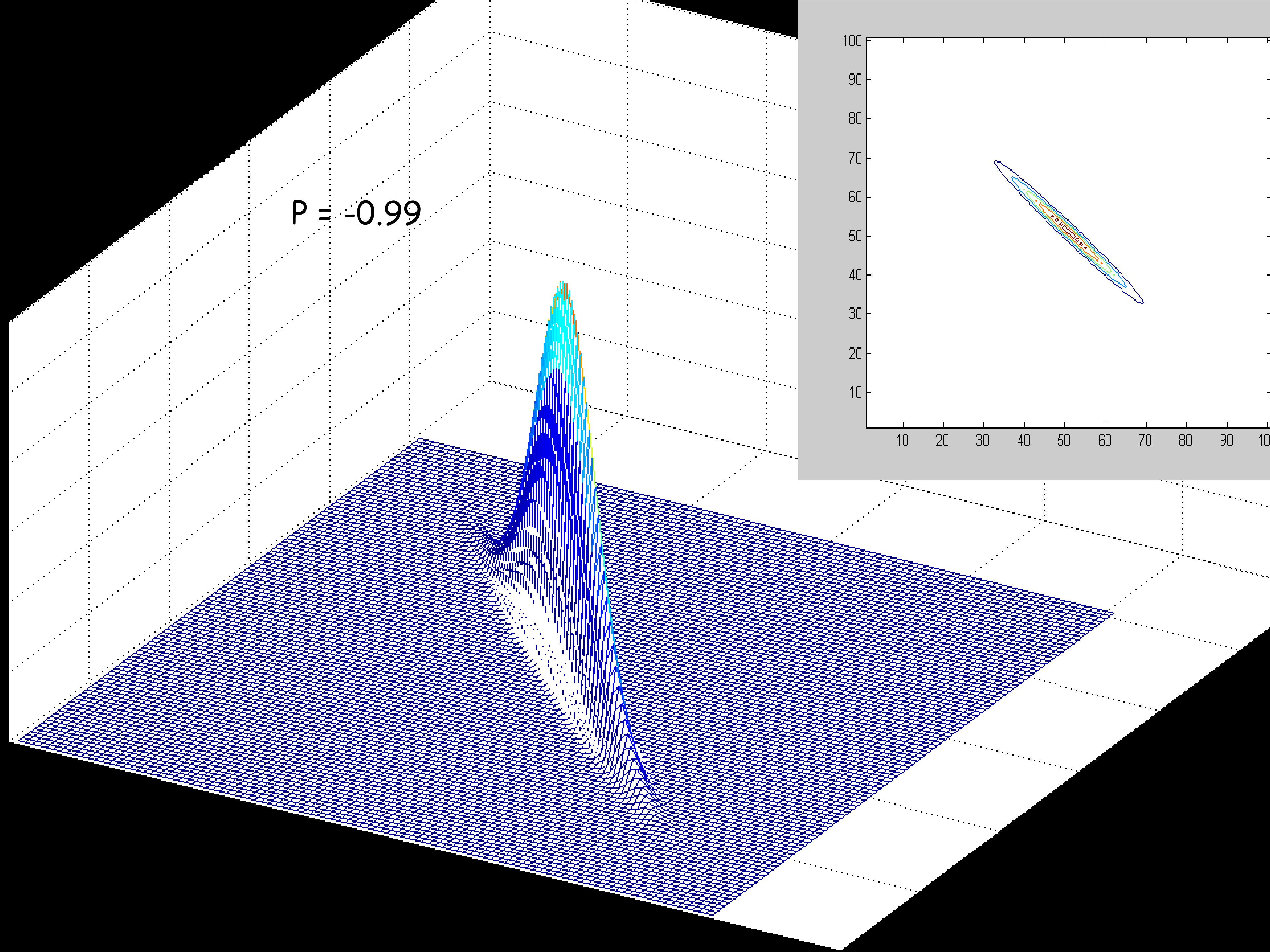

P = 0
Volumn = 1



P = 0.5
100
90
80
70
60
50
40
30
20
10
10
20
30
40
50
60
70
80
90
100


P = 0.9
100
90
80
70
60
50
40
30
20
10
10
20
30
40
50
60
70
80
90
100


P = 0.95
100
90
80
70
60
50
40
30
20
10
10
20
30
40
50
60
70
80
90
100


P = -0.99
100
90
80
70
60
50
40
30
20
10
10
20
30
40
50
60
70
80
90

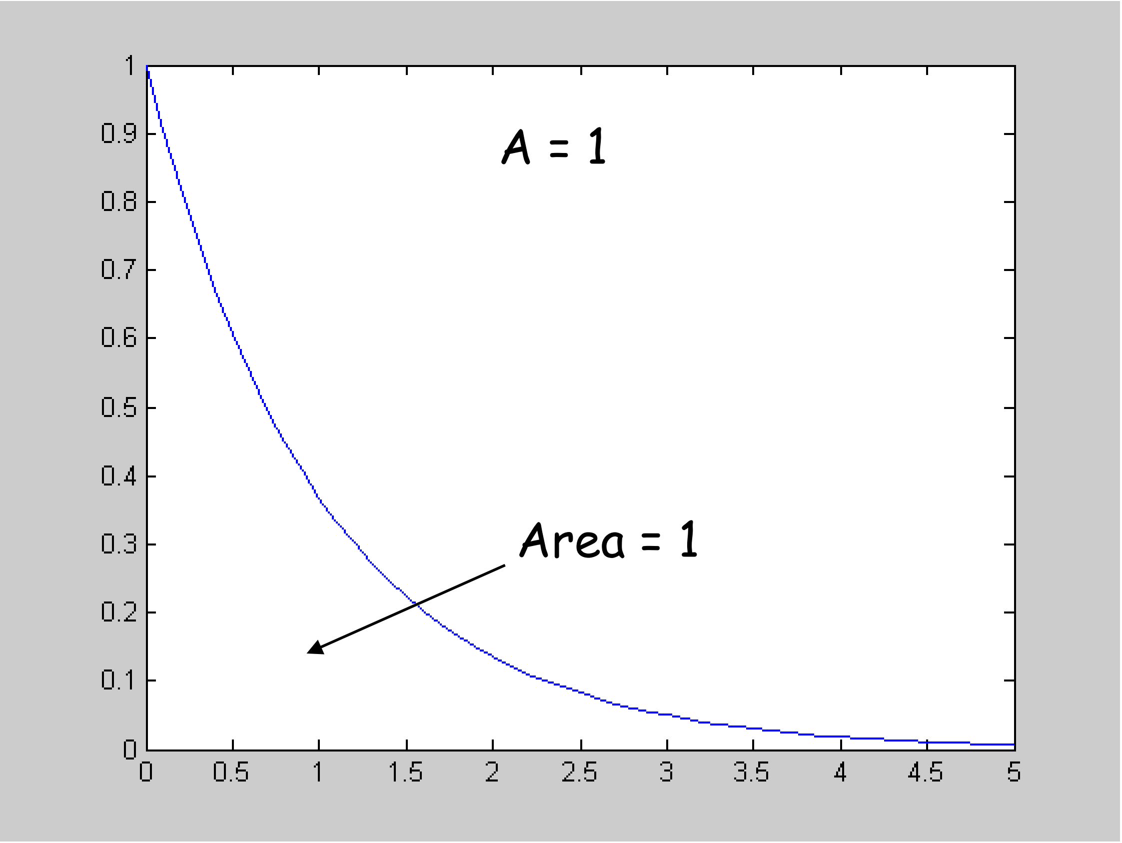
Exponential Distribution

4.8.6 Exponential Distribution Exponential Distribution ที่มี Parameter A สามารถเขียนได้ในรูป

$$f_X(x) = Ae^{-Ax} \text{ และ } F_X(x) = 1 - e^{-Ax}, \quad E(X) = \frac{1}{A} \text{ และ } \sigma_X^2 = \frac{1}{A^2}$$

Exponential Distribution เป็น Distribution ที่สำคัญอีกอันหนึ่งใน Queuing Theory ซึ่งค่า Inter-arrival Time ของสอง Random Event ที่อยู่ติดกันจะมีการกระจายแบบนี้ ซึ่งเราสามารถหา Probability ที่ Event ที่สองจะเกิดหลังจาก Event แรกเกิดแล้วภายในเวลา t ได้จาก $p(X \le t) = 1 - e^{-At}$

Exponential Distribution



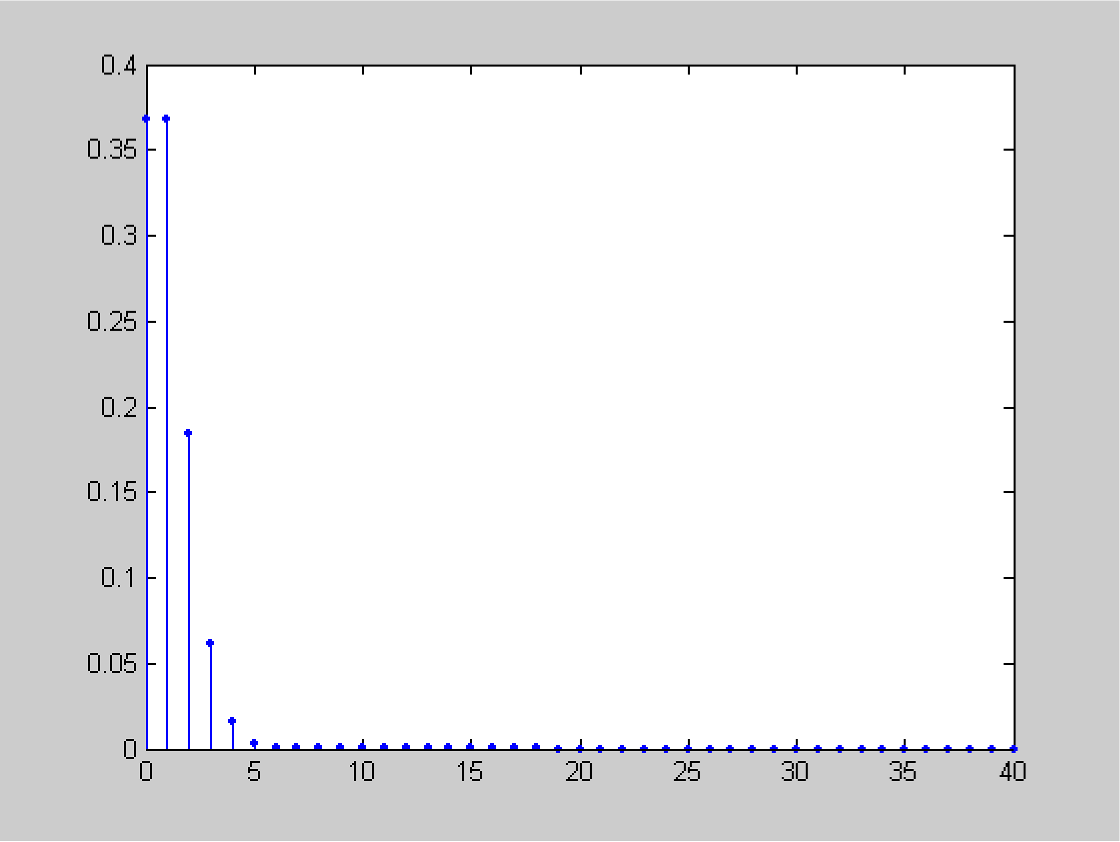
Poisson Distribution

4.8.7 Poisson Distribution เป็น Discrete Random Variable ที่สำคัญสำหรับ Queuing Theory ที่จะศึกษาต่อไป ได้จากการนับของ Independent Random Event ในช่วงเวลาใดเวลาหนึ่ง ถ้าค่าเฉลี่ยของการนับ Event ในช่วงเวลานั้นเท่ากับ λ ดังนั้น Probability ที่จะมี k Event เข้ามาในช่วงเวลาเดียวกันสามารถหาได้จาก

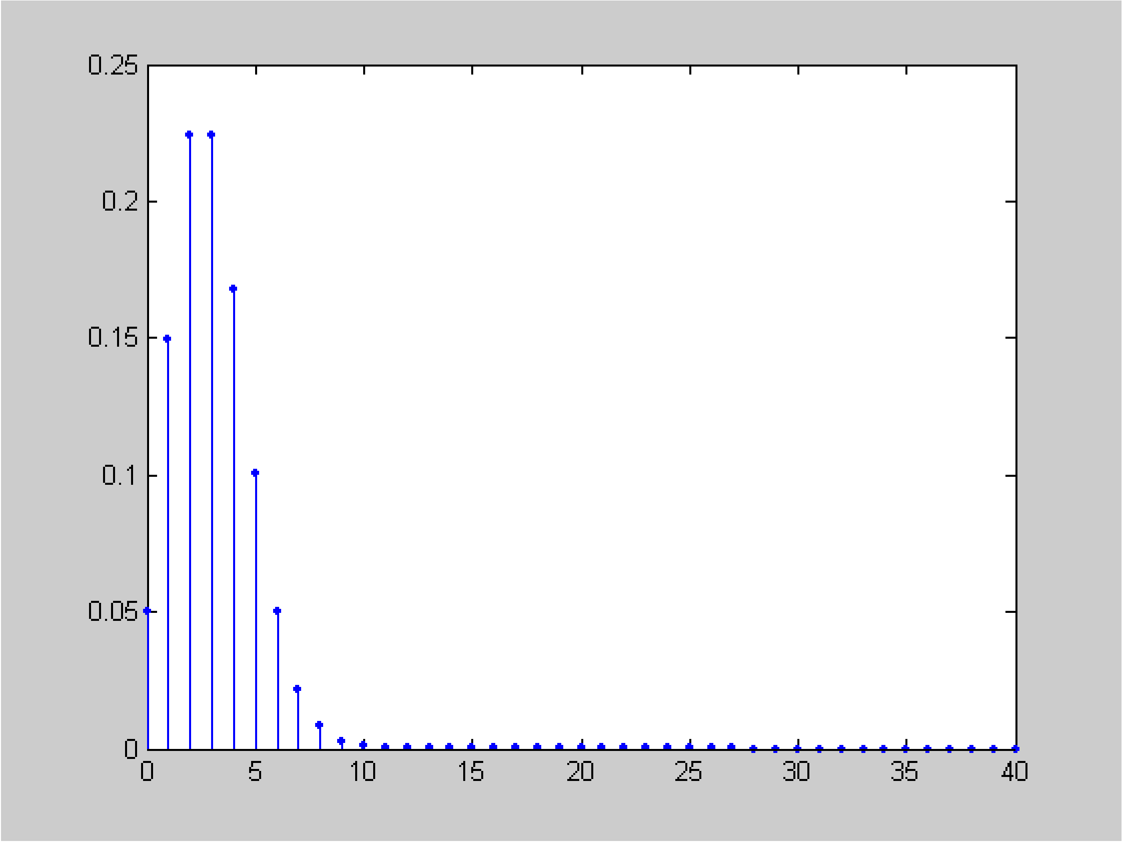
$$p(X=k)=p(k)=\frac{\lambda^k e^{-\lambda}}{k!}$$ เราสามารถพิสูจน์ได้ว่า $E(X)=\lambda,$ และ $\sigma_X^2=\lambda$

ประโยชน์อีกอย่างของ Poisson คือใช้ในการประมาณค่าของ Binomial Distribution โดยกำหนด $\lambda = np$

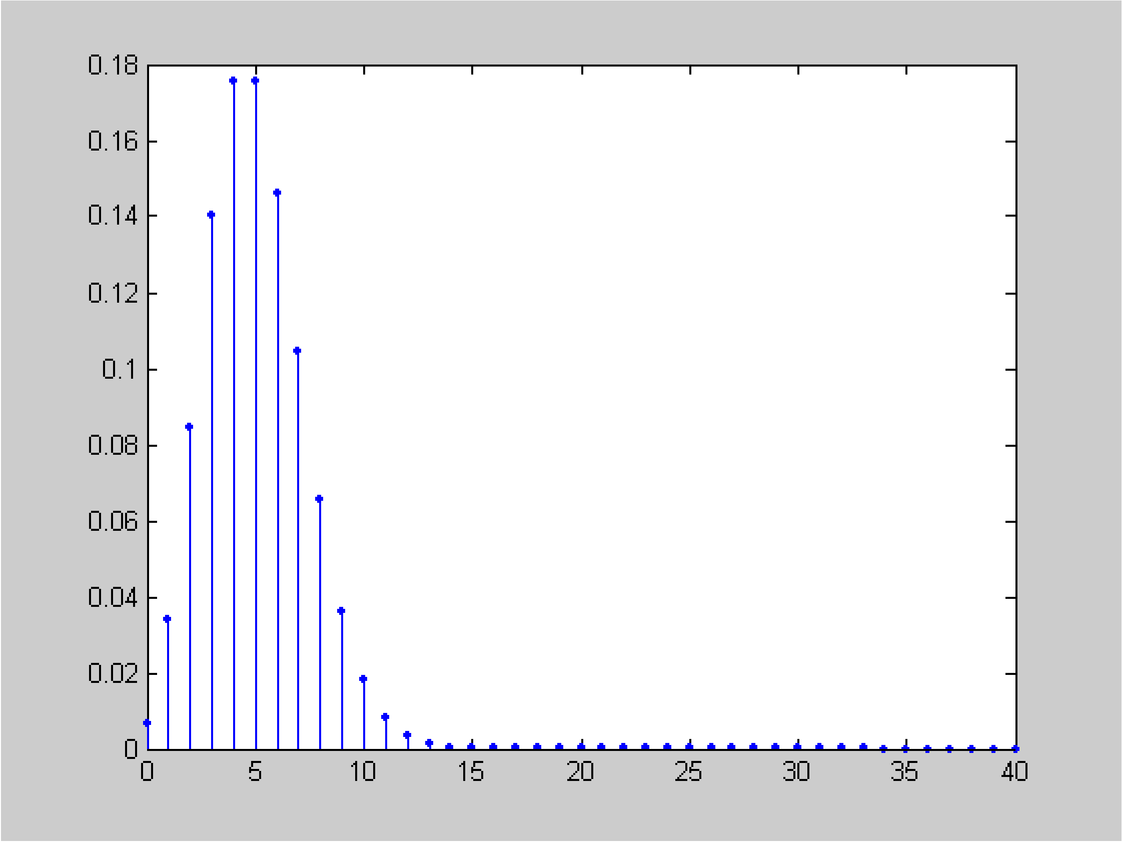
Lambda = 1



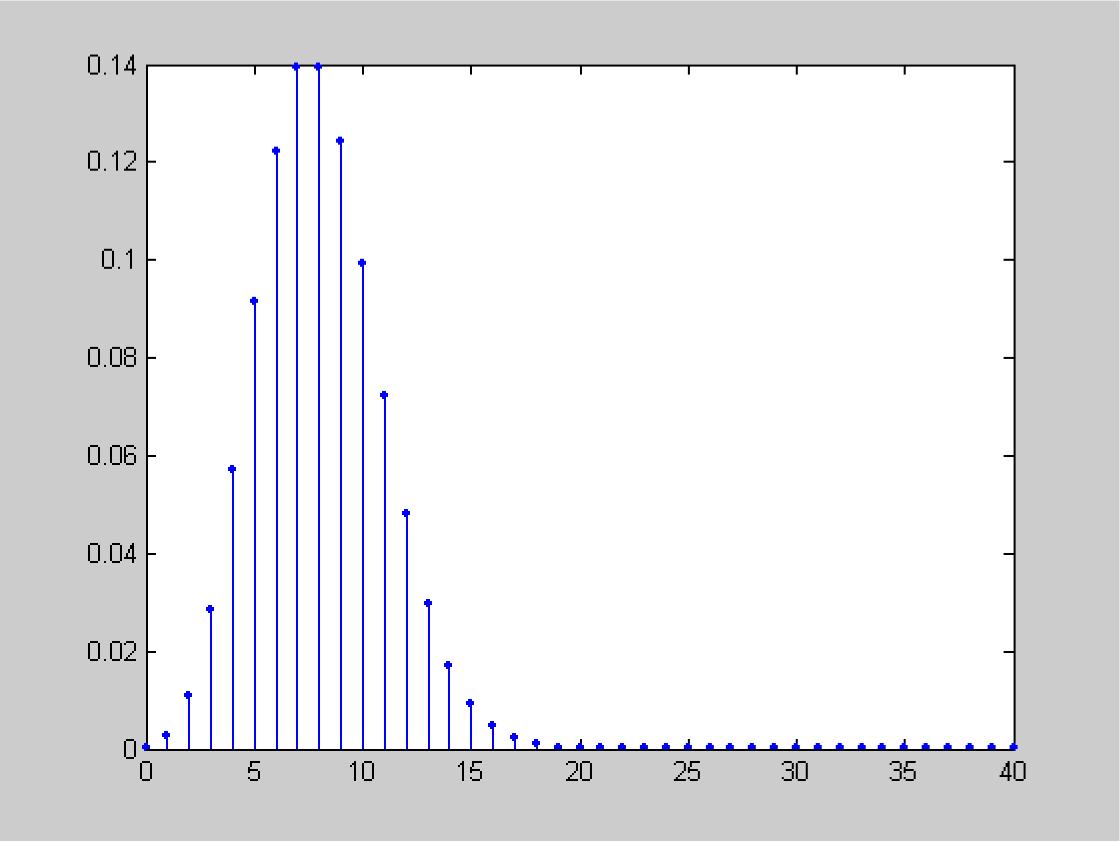
Lambda = 3



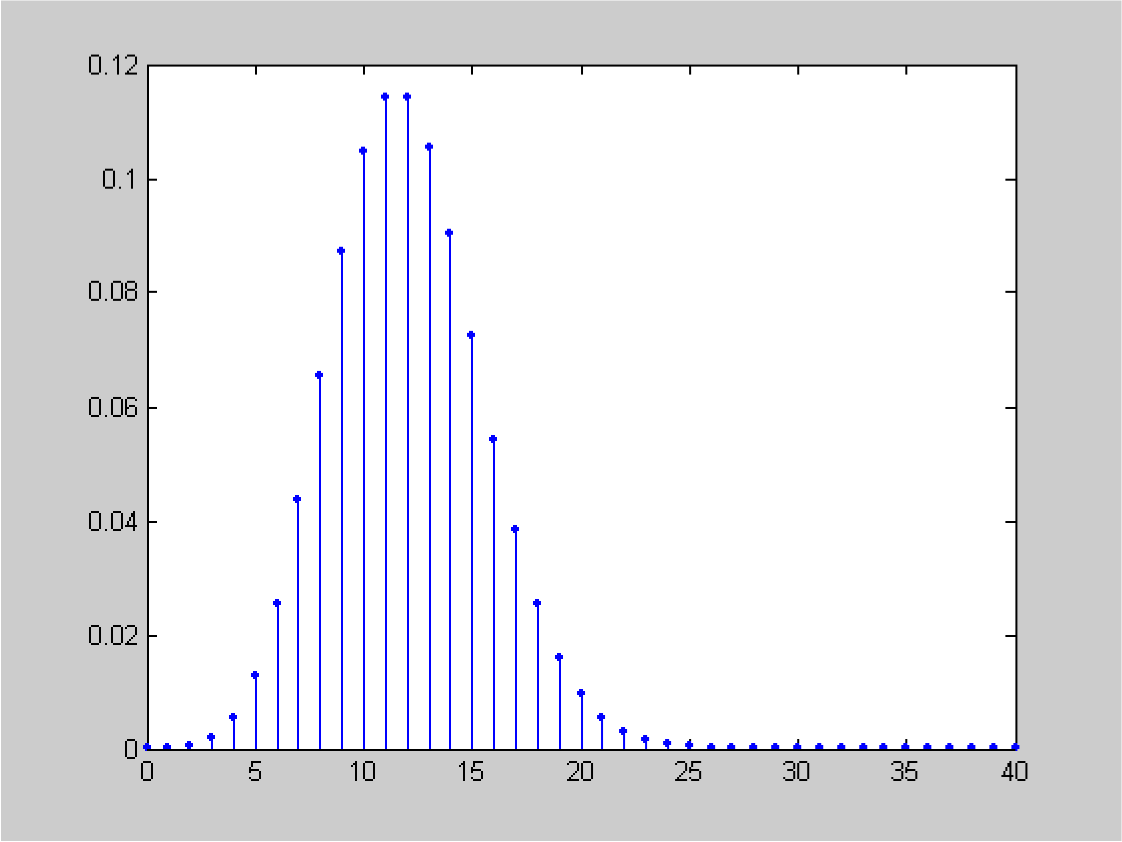
Lambda = 5



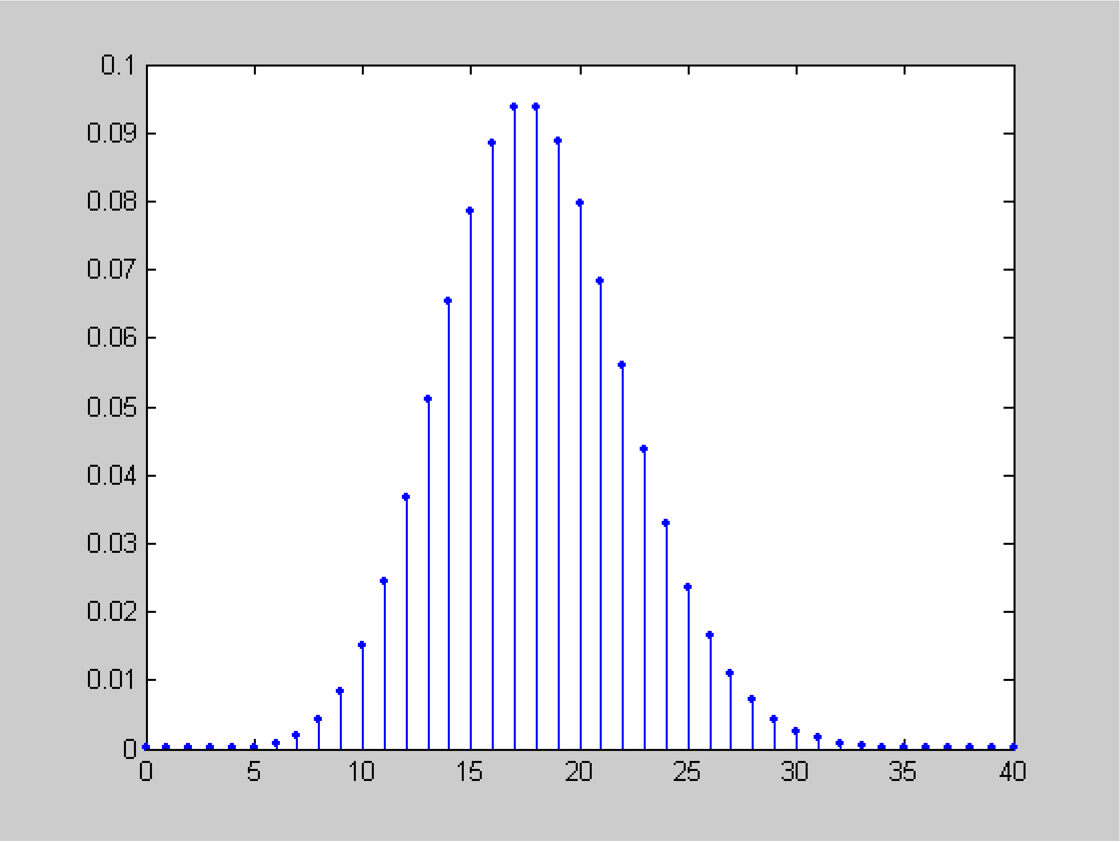
Lambda = 8



Lambda = 12



Lambda = 18



Break

- After Break Chapter 5 Random Process

Random Process

- เมื่อ Random Variable เป็น Function กับเวลา
  - สัญญาณที่มีลักษณะ Random เช่น Noise
  - การส่ง Packet ใน Network
  - จำนวนรถที่วิ่งบนถนน
  - จำนวนลูกค้าที่เข้ามาใช้บริการ
- เหล่านี้ ในแต่ละเวลาหนึ่งๆ ค่าของมันจะมีลักษณะเป็น Random คือมี PDF ตามแต่ชนิดของมัน
  - ที่เวลาต่างกัน PDF อาจจะเปลี่ยน และค่าทางสถิติจะเปลี่ยนตาม
- Random Variable ที่เป็น Function (Random Function หรือ Analytic Function) กับเวลา เราเรียก Random Process หรือ Stochastic Process
  - จาก RV **X** จะเป็น **X**(t)
  - Mean และ Variance จะเป็น Function กับเวลาด้วย
- การวิเคราะห์จะซับซ้อนกว่า RV

Random Process (Stochastic Process)

5.1 นิยามและคำจำกัดความ

Random Process เป็นการขยายคำจำกัดความของ Random Variable, เราสามารถคิดได้อย่างง่ายๆว่า Random Process คือ Random Variable ที่เป็น Function กับเวลา(หรือ Independent Variable ตัวอื่น) ซึ่งบางครั้งเราเรียกอีกชื่อหนึ่งว่า **Stochastic Process** ดังนั้นถ้าเราให้ $X(t)$ เป็น Random Process เราจะได้ว่าในแต่ละจุดของเวลา $t_1, t_2, \ldots, t_n$ Random Process ที่เวลานั้นๆ $X(t_1), X(t_2), \ldots, X(t_n)$ แต่ละตัวจะเป็น Random Variable ซึ่งอาจจะมีหรือไม่มีค่า Probability Density Function ที่เหมือนกันหรือเท่ากันก็ได้

ค่าของ Random Process ที่เวลาหนึ่งๆเราเรียกมันว่าเป็น Random Variable ที่ Ensemble ของเวลานั้น อย่างไรก็ตาม ตามปกติการวัดค่าของสัญญาณนั้นเราจะได้ข้อมูลที่เป็น Sample Point ของ Random Variable ในแต่ละจุดของเวลา ซึ่งถ้าเราวัดติดต่อกันไป เราก็จะได้ Function หนึ่งซึ่งเป็นสัญญาณที่เปลี่ยนไปตามเวลา ซึ่งเราเรียกว่า Sample Function ของ Random Process

Statistical Average

สำหรับในแต่ละเวลา ค่า $X(t_i)$ ก็คือ RV ฉะนั้นค่าทางสถิติจะหาได้จาก Expectation ซึ่งค่าที่ได้จะเป็นค่าทางสถิติที่เป็น Function กับเวลา มีค่าทางสถิติที่เพิ่มมาที่สำคัญก็คือ Autocorrelation และค่า Autocovariance

5.2.1 Mean ค่าของ Mean ในความหมายทางสถิติก็คือค่า $E[X(t)]$ เนื่องจาก $X(t)$ เป็น Function กับเวลา ดังนั้นค่า Mean ที่ได้ก็จะเปลี่ยนไปตามเวลาด้วย นั่นก็คือ $m_X(t) = E[X(t)]$

5.2.2 Variance ที่เวลา t_i เราสามารถหาค่า Variance $\sigma_X(t_i)$ ได้จาก PDF ของ $X(t_i)$ นั่นก็คือ

$$\sigma_X^2(t_i) = E[(X(t_i) - m_X(t_i))^2] = \int_{-\infty}^{\infty} (X(t_i) - m_X(t_i))^2 f_{X(t_i)}(x)dx$$

- Notations:
  - Random Variable: X
  - Random Process: $X(t)$
- PDF
  - Random Variable: $f_X(x)$
  - Random Process: $f_{X(t)}(x,t) = f_{X(t)}(x)$ที่เวลา t
- Expectation (Statistical Average)
  - $\mu_X(m_X) \rightarrow \mu_{X(t)}(t)$

    $\sigma_X^2 \rightarrow \sigma_{X(t)}^2(t)$

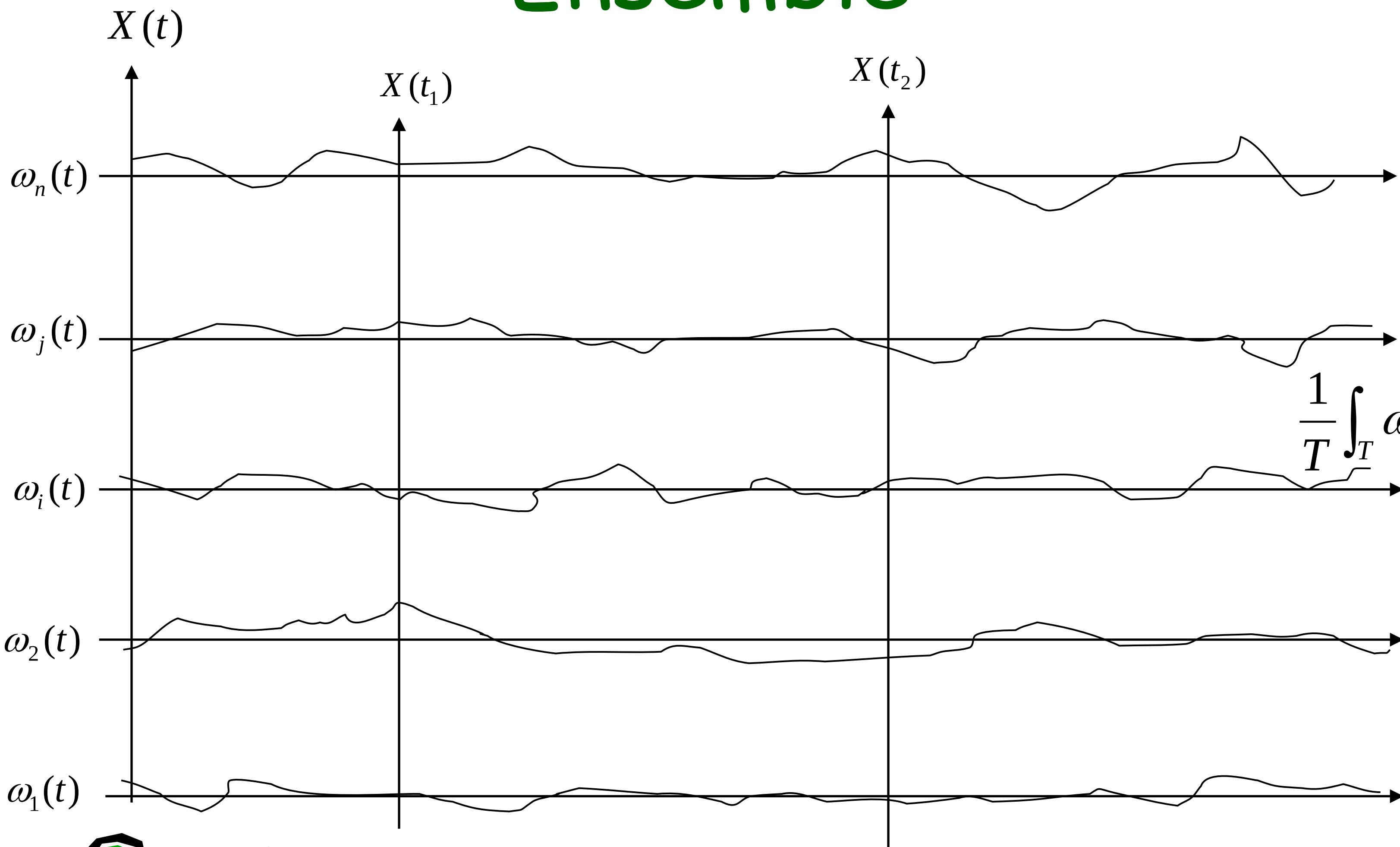
Random Process at t_i

- Random Process ที่เวลาใดเวลาหนึ่ง คือ Random Variable ที่มีค่า PDF, Mean, Variance เฉพาะ
  - เรียกค่าใน Ensemble (Ensemble Average)
- ค่าต่างๆนี้จะแปรผันไปกับเวลาได้
  - $X(t_1)$เป็น Random Variable ได้จาก Random Process ที่ t_1 จะมี PDF, Mean, Variance $f_{X(t_1)}(x),\ \mu_{X(t_1)}, \sigma_{X(t_1)}$

Sample Function: Ensemble Average and Time Average

- ในการศึกษา Random Variable เราสามารถสุ่มตัวอย่างทีละตัวอย่าง เรียก Sample Point, ω_i
- ในกรณีของ Random Process ตัวอย่างที่สุ่มจะเป็นตัวอย่างที่เป็น Function กับเวลา เรียก Sample Function, $\omega_i(t)$
- ค่าเฉลี่ยในแต่ละเวลา เราเรียก Ensemble Average
  - จะเหมือนค่าเฉลี่ยปกติของ Random Variable
- ค่าเฉลี่ยของแต่ละ Sample Function เรียก Time Average

Sample Function and Ensemble



Time Average

$$\frac{1}{T}\int_T \omega_i(t)\,dt$$

Ensemble Average

$\overline{X(t_1)}, \sigma^2_{X(t_1)}$

Ensemble Average

$\overline{X(t_2)}, \sigma^2_{X(t_2)}$

$\rightarrow \overline{X}(t), \sigma_X^2(t)$

Statistical Average: Correlation/Covariance

5.2.3 Autocorrelation ค่า Autocorrelation Function ของ Random Process $X(t)$ คือความสัมพันธ์ของตัวมันเองที่เวลาต่างกัน(เปรียบเทียบกับ Autocorrelation ของ Deterministic Signal) นั่นก็คือ

$$R_{XX}(t_1,t_2) \equiv R_X(t_1,t_2) = E[X(t_1)X(t_2)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2 f_{X(t_1)X(t_2)}(x_1,x_2)dx_1 dx_2$$

5.2.4 Autocovariance Function ในทำนองเดียวกัน เราสามารถหา Autocovariance Function ได้จาก

$$C_{XX}(t_1,t_2) \equiv C_X(t_1,t_2) = E[(X(t_1)-m_X(t_1))(X(t_2)-m_X(t_2))]$$

Stationary RP

- ค่าทางสถิติของ RP Process ปกติเปลี่ยนไปกับเวลา (Ensemble Average ที่เวลาต่างกัน)
  - Mean
  - Variance
  - PDF
- สำหรับกรณีพิเศษที่ค่าทางสถิติของ RP ไม่แปรผันกับเวลา เราเรียกว่ามันเป็น Stationary
  - $f_{X(t)}(x,t) \rightarrow f_{X(t)}(x)$
  - $\mu_{X(t)}(t) \rightarrow \mu_{X(t)}(\mu_X)$

    $\sigma^2_{X(t)}(t) \rightarrow \sigma^2_{X(t)} = \sigma^2_X$

Stationary RP

- Stationary RP
  - Strict Sense Stationary (SSS)
    - PDF ไม่เปลี่ยน ดังนั้นทุกๆ Moment (Expectation) ไม่เปลี่ยน
    - ปกติ SSS จะหาได้ยากในทางปฏิบัติ บ่อยครั้งที่เราสนใจแค่สอง Moment แรก
  - Wide Sense Stationary (WSS)
    - เฉพาะ 2 Moment แรกคงที่ แต่ PDF อาจแปรผัน
      - Mean
      - Variance (Mean Square)

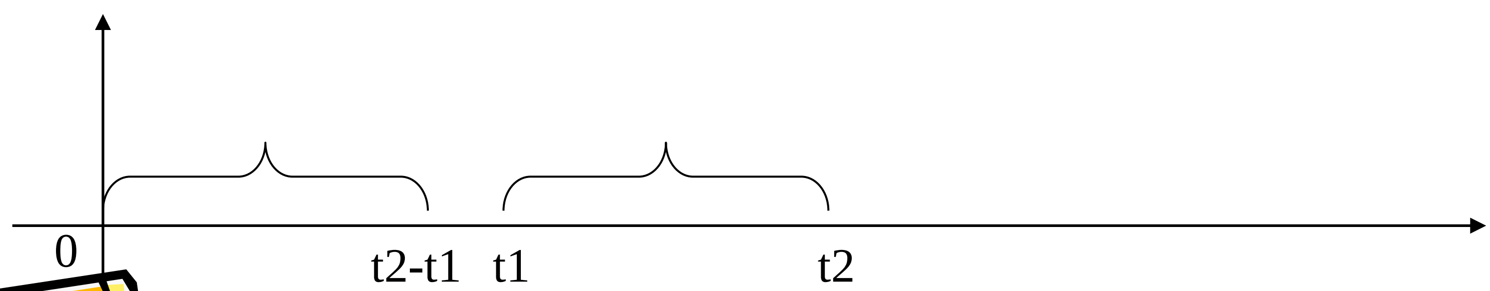
มีข้อที่น่าสังเกตอีกอันหนึ่ง คือถ้า Random Process นั้นเป็น Gaussian Process นั่นก็คือทุกจุดเวลามันจะมีลักษณะ PDF เป็นแบบ Gaussian เนื่องจาก Gaussian PDF สามารถอธิบายได้จากค่าทางสถิติเพียงแค่สองตัวคือค่า Mean และค่า Variance ดังนั้นในกรณีของ Gaussian Process คำว่า Wide-Sense Stationary จะมีความหมายเดียวกันกับ Strict-Sense Stationary

WIDE SENSE STATIONARY

- $m_X(t)$ หรือ $\mu_X(t)$ จะลดรูปลงเหลือ m_X และ $\sigma_X(t)$ จะกลายเป็น σ_X

- $R_X(t_1,t_2)$ จะไม่ขึ้นกับเวลา t_1 และ t_2 อีกต่อไป แต่จะขึ้นกับผลต่างของเวลา $t_2 - t_1 = \tau$ นั่นก็คือเราได้ $R_X(t_1,t_2) = R_X(t_2 - t_1) = R_X(\tau)$

- $C_X(t_1,t_2)$ จะลดรูปลงเหลือ $C_X(t_1,t_2) = R_X(\tau) - m_X^2$



Autocorrelation(WSS)

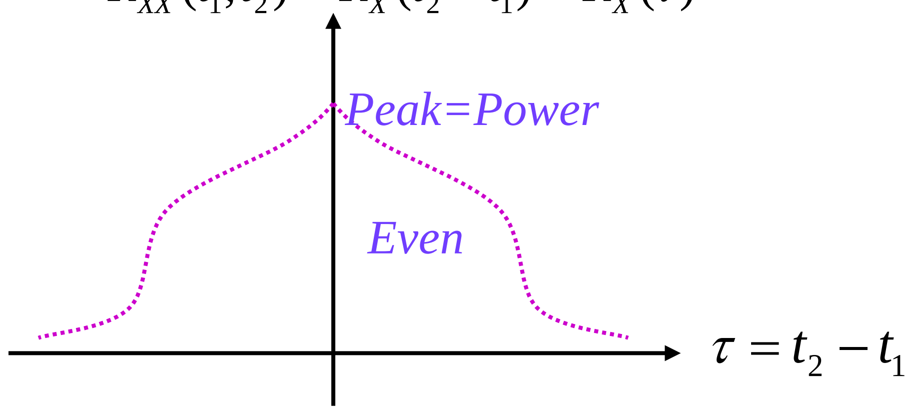
คุณสมบัติที่สำคัญก็คือการลดรูปของค่า Autocorrelation ซึ่งจะอยู่ในรูปที่เหมือนกับในกรณีของ Deterministic Signal ต่อไปนี้เป็นคุณสมบัติของ Autocorrelation

- $R_X(0) = E[X^2(t)]$ ซึ่งเกี่ยวพันกับ Power และ Energy ของสัญญาณ Random ซึ่งก็คือสัญญาณที่เป็น Random Process

- $R_X(\tau) = R_X(-\tau)$ นั่นก็คือ $R_X(\tau)$ เป็น Even Function ดังนั้นเราสามารถเขียน $R_X(\tau) = E[X(t)X(t-\tau)]$

- $|R_X(\tau)| \le R_X(0)$

$$R_{XX}(t_1,t_2) = R_X(t_2 - t_1) = R_X(\tau)$$

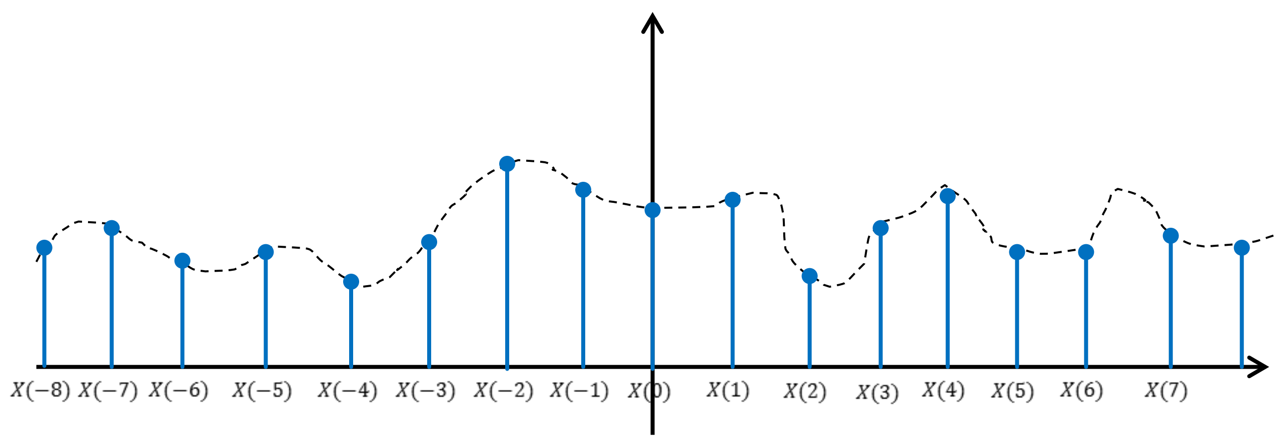


Ergodic Random Process

- ถ้า RP เป็น Ergodic
  - Time Average = Ensemble Average
  - Mean Ergodic
    - หมายถึงค่าเฉลี่ยในทางเวลา $\frac{1}{T}\int_T \omega_i(t)dt$ เท่ากับค่าเฉลี่ย(mean = E(X(t$_i$))) ของ Ensemble
  - Correlation Ergodic
    - หมายถึงค่า Variance ในทางเวลา $\frac{1}{T}\int_T (\omega_i(t) - \overline{\omega_i(t)})^2 dt$ เท่ากับค่า Variance ที่ได้จาก Ensemble $\sigma^2_{X(t_i)}$
  - เราใช้ควบคู่กับ Concept ของ Stationary Random Process

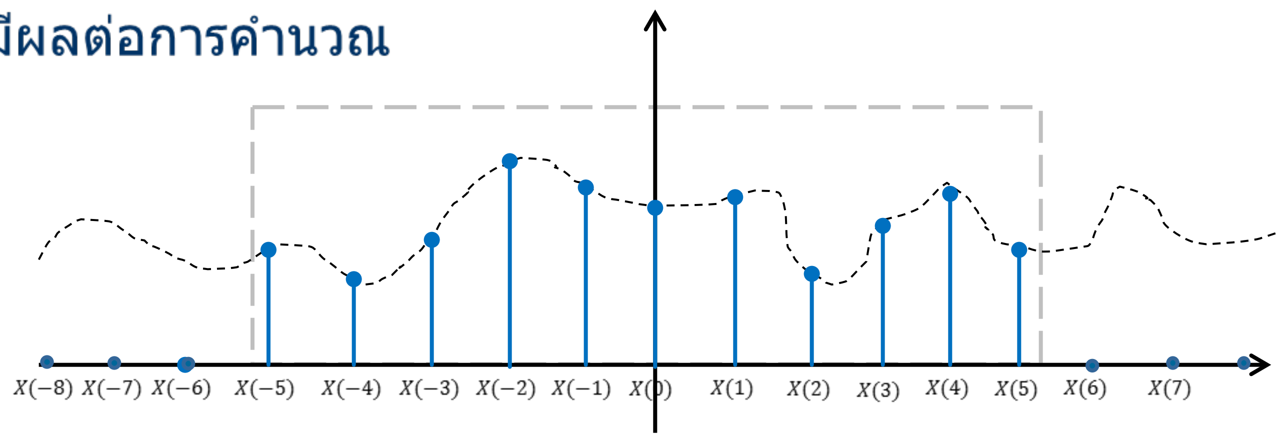
Discrete-Time Random Process

- อาจจะได้จากการสุ่มตัวอย่าง (Sampling) ของ Continuous RP
  - Uniform Sampling ด้วย Sampling Period $T = \frac{1}{fs}$
- เราได้ $\ldots, \mathrm{X}(-\mathrm{T}), X(0), X(T), X(2T), X(3T), \ldots$
- ปกติจะละ Sampling Period ไว้ฐานที่เข้าใจ เราได้ $\ldots, X(-1), X(0), X(1), X(2), X(3), \ldots$
- มักจะเขียนในลักษณะ $X(n); n = integer$



Discrete-Time Random Process (Limited Samples)

- จริงแล้ว $n = (-\infty, +\infty)$ แตในทางปฏิบัติ เราสามารถสุ่มตัวอย่างได้ในช่วงเวลาที่จำกัด
  - $n = 0,1,2, \dots, N-1$ หรือ $n = -N, -N+1, \dots, -1,0,1, \dots, N$
  - ขึ้นอยู่กับว่าเราจะใช้ Index เรียกตัวอย่างที่ได้อย่างไร
  - เสมือนเราคูณ $X(n)$ด้วย $Box\ Function$ โดยถือว่าตัวอย่างที่อยู่นอกขอบเขตของการสุ่มมีค่าเป็นศูนย์ เช่นในกรณีของ $X(n); n = 0, \pm 1, \pm 2, \dots, \pm 5$
  - จะมีผลต่อการคำนวณ



Ergodic Discrete RP

- ถ้า RP เป็น Ergodic ด้วยค่า Mean (Average จาก Ensemble = Average จาก Time)

Mean Ergodic Process

ในกรณีของ Stationary Random Process มีค่า Mean

$$m_X = E[X(n)]$$

คราวนี้ถ้าเราสมมุติค่า Time Average

$$\hat{m}_X = \frac{1}{2N+1}\sum_{n=-N}^{N} x(n)$$

และถ้า Process เป็น Ergodic, ดังนั้น

$$m_X = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N} x(n)$$

Ergodic Discrete RP

- ถ้า RP เป็น Ergodic สำหรับ Coorelation

5.4.2 Correlation Ergodic Process

เมื่อ Stationary Random Process เป็น Correlation Ergodic เราสามารถคำนวณค่า Autocorrelation และ Cross Correlation จากเพียงหนึ่ง Sample Function โดยใช้การเฉลี่ยในทางเวลาแทน ดังนั้น

$$R_{XX}(m) = R_X(m) = E[X^*(n)X(n+m)] = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N} x^*(n)x(n+m)$$

และ

$$R_{XY}(m) = E[X^*(n)Y(n+m)] = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N} x^*(n)y(n+m)$$

ค่า Estimation ของ Autocorrelation จากหนึ่ง Sample Function สามารถหาได้จาก

$$\hat{R}_x(m) = \frac{1}{2N+1}\sum_{n=-N}^{N} x^*(n)x(n+m)$$

กรณีที่จำกัด Sequence ความยาว N

ถ้าสมมุติว่าเราได้ Sample Function ของ Discrete Random Process $X(n)$ มาหนึ่งตัว คือ $\omega_i(n) = x(n)$ ดังนั้นเราจะได้

$$E[X(n)] = E(X) = m_X = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{i=-N}^{N} x(n)$$

ในทางปฏิบัติเราไม่สามารถเก็บ Sample Function ได้ตลอดทุกเวลา เราจำเป็นต้องตัดมาเพียงบางส่วน สมมุติว่าเราได้ตัวอย่างความยาว N และให้ $x(n), n = 0,\ldots,N-1$ คือจำนวนเพียง N Sample ดังนั้นค่าที่คำนวณได้จะเป็นเพียงค่า Estimation ของค่า m_X หรือ $\hat{m}_X$ ซึ่งจะถูกต้องขึ้นถ้าเพิ่มจำนวน Sample มากขึ้น และสมการจะลดรูปเป็น

$$\hat{m}_X = \frac{1}{N} \sum_{i=0}^{N-1} x(n)$$

ถ้า $x(n)$ ที่เราสนใจมีค่าอยู่ระหว่าง $n = M$ ถึง $n = N$ สมการจะลดรูปเป็น

$$\hat{m}_X = \frac{1}{N-M+1} \sum_{i=M}^{N} x(n)$$

จำกัด Sequence ความยาว N

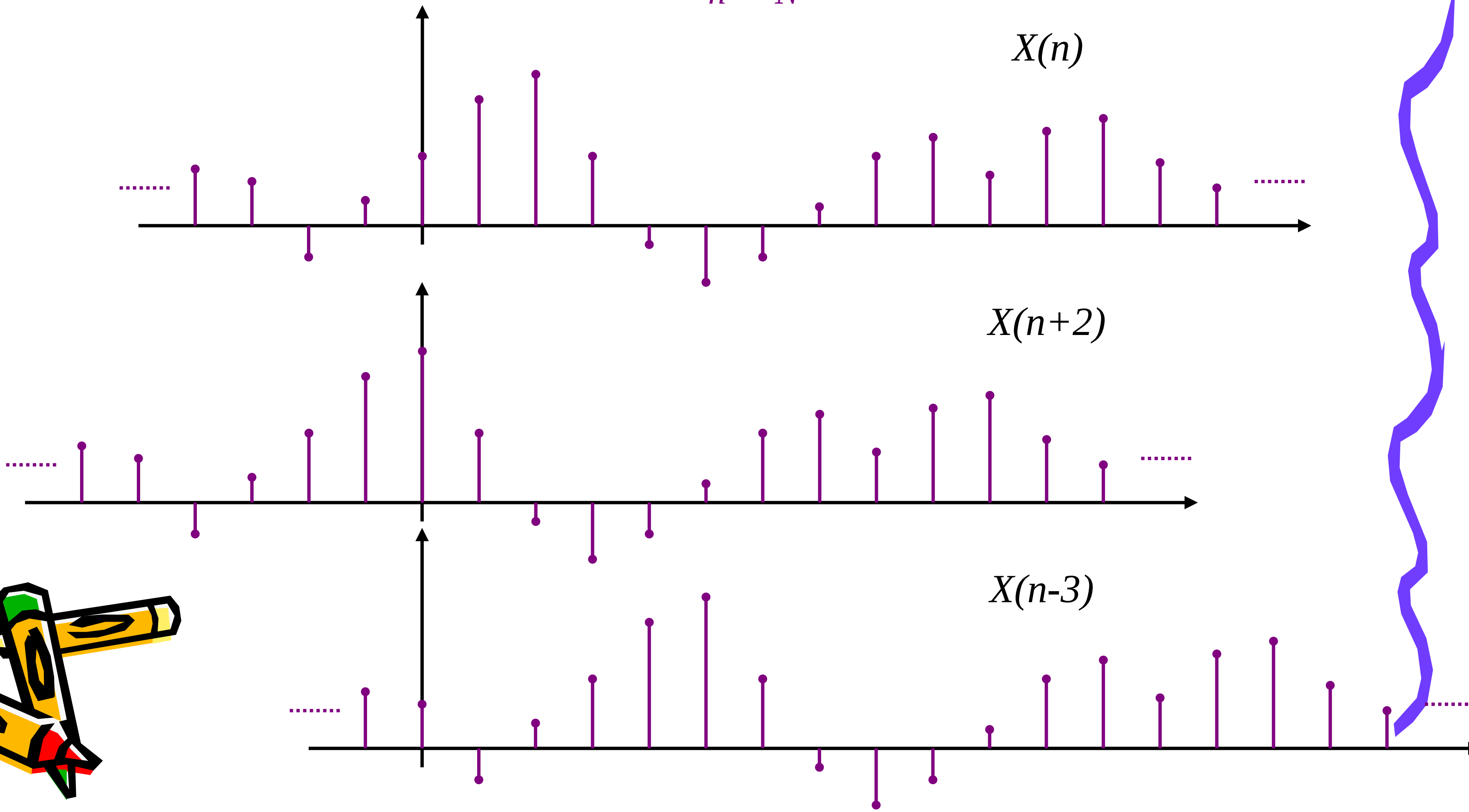
- ค่า Autocorrelation สำหรับ N Samples
  - สมการจะลดรูป เหลือแค่ Sum และเฉลี่ย N Point แต่จะเกิดการ Biased

$$R_{XX}(m) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n)x(n+m)$$

$$R_{XX}(m) = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{n=0}^{N-m-1} x(n)x(n+m); & 0 \le m \le N-1 \\ R_{XX}(-m)\,; & -N+1 \le m < 0 \end{cases} ; \quad \text{Biased}$$

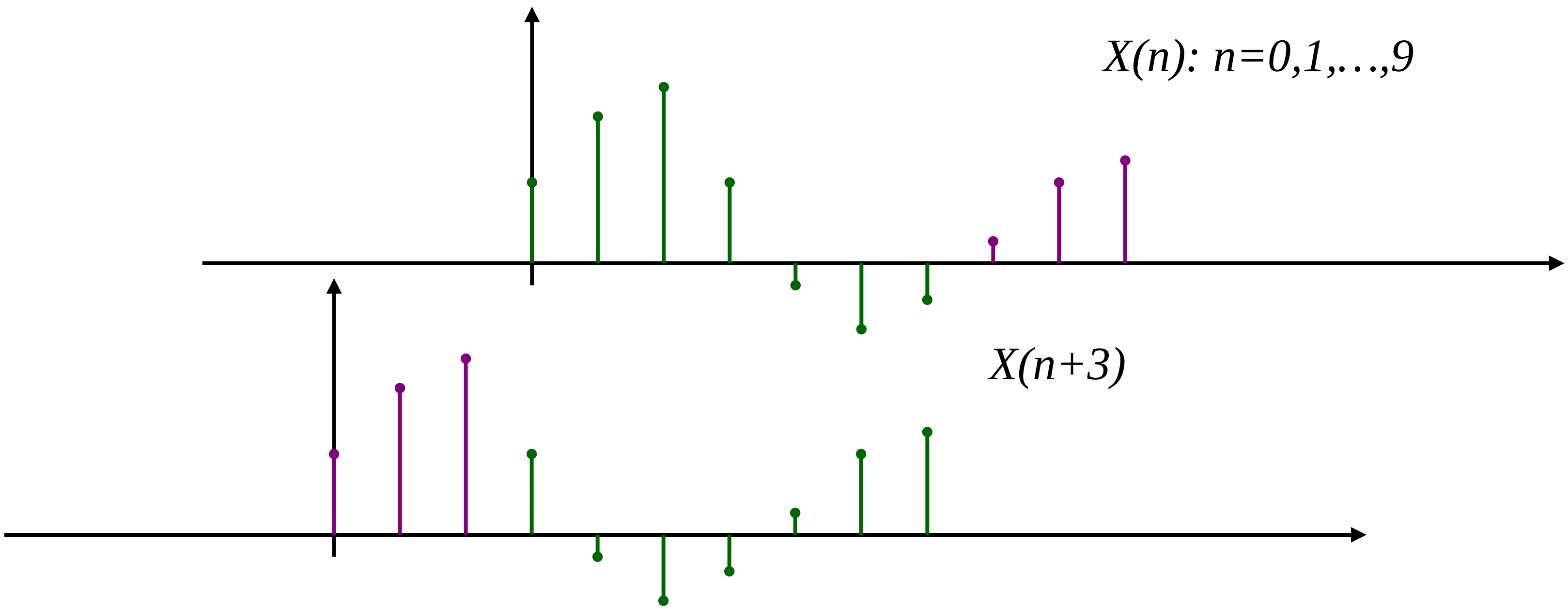
ความหมายของ x(n)x(n+m)

$$R_{XX}(m) = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n)x(n+m)$$



ความหมายของ x(n)x(n+m), N Points

$$R_{XX}(m) = \frac{1}{N}\sum_{n=0}^{N} x(n)x(n+m) = \frac{1}{N}\sum_{n=0}^{N-m+1} x(n)x(n+m)$$



เราไม่ควรเฉลี่ยทั้ง N ตัว (Biased) แต่ควรเฉลี่ย N-m ตัว (non-Biased)

จำกัด Sequence ความยาว N

- ค่า Autocorrelation สำหรับ N Samples
  - สมการจะลดรูป เหลือแค่ Sum และเฉลี่ย N Point แต่จะเกิดการ Biased เพราะเราเฉลี่ยน้อยกว่านั้น

$$R_{XX}(m) = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-m-1} x(n)x(n+m); & 0 \le m \le N-1 \\ R_{XX}(-m); & -N+1 \le m < 0 \end{cases} ; \quad \text{Biased}$$

ดังนั้นการคำนวณที่ไม่ Biased จะเป็นค่าที่ Normalized จากจำนวนจุดของการคำนวณจริงๆ และเราได้

$$R_{XX}(m) = \begin{cases} \frac{1}{N-|m|} \sum_{n=0}^{N-m-1} x(n)x(n+m); & 0 \le m \le N-1 \\ R_{XX}(-m); & -N+1 \le m < 0 \end{cases} ; \quad \text{Non - biased}$$

Sequence ความยาว N / RAW

ในการคำนวณปกติมักจะคำนวนค่าในลักษณะ “Raw Autocorrelation” และละส่วนของการ Normalization หรือส่วนของการหารไว้ สำหรับผู้ใช้จะพิจารณาเองว่าจะใช้แบบ Bised หรือ Non-Biased และเราได้

$$R_{XX}(m)=\sum_{n=0}^{N-1}x(n+m)x(n)=\begin{cases}\displaystyle\sum_{n=0}^{N-m-1}x(n)x(n+m);\ m\geq 0\\ R_{XX}(-m);\ m<0\end{cases};\quad \text{Raw Data}$$

ในทำนองเดียวกัน สำหรับ Cross Correlation เมื่อ $X(n)$ และ $Y(n)$ มีความยาว N เท่ากันโดยการเติมศูนย์เราได้

$$R_{XY}(m)=\sum_{n=0}^{N-1}x(n)y(n+m)=\begin{cases}\displaystyle\sum_{n=0}^{N-m-1}x(n)y(n+m)\ ;\ m\geq 0\\ \displaystyle\sum_{n=0}^{N-m-1}x(n-m)y(n)\ ;\ m<0\end{cases};\quad \text{Raw Data}$$

Sequence ทั่วไป

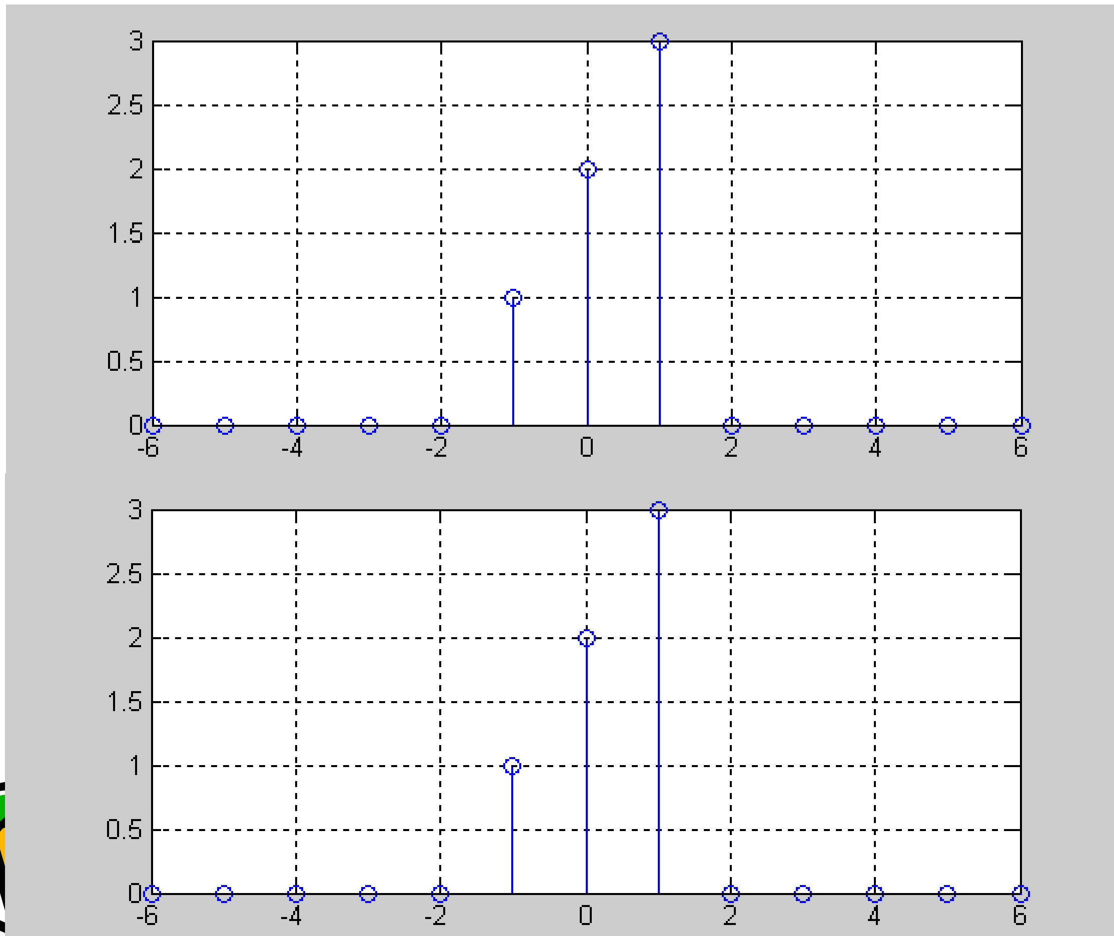
ในกรณีที่สัญญาณไม่ได้เริ่มจาก $n=0$ และ/หรือทั้งสอง Random Variable มีความยาวไม่เท่ากัน ผลลัพธ์ที่ได้จะมีความยาวของ Sequence $N+M-1$ โดยที่ N และ M เป็นความยาวของทั้งสอง Variable และในกรณีนี้ค่า Index ของ Summation จะเปลี่ยนไป อย่างไรก็ตาม สมการข้างล่างยังใช้ได้อยู่สำหรับ Raw Data ที่ไม่ได้ทำการ Normalized

$$R_{XX}(m)=\sum_{n=-\infty}^{\infty}x(n)x(n+m)\,;-\infty<m<\infty \quad \text{Raw Data}$$

$$R_{XY}(m)=\sum_{n=-\infty}^{\infty}x(n)y(n+m)\,;-\infty<m<\infty \quad \text{Raw Data}$$

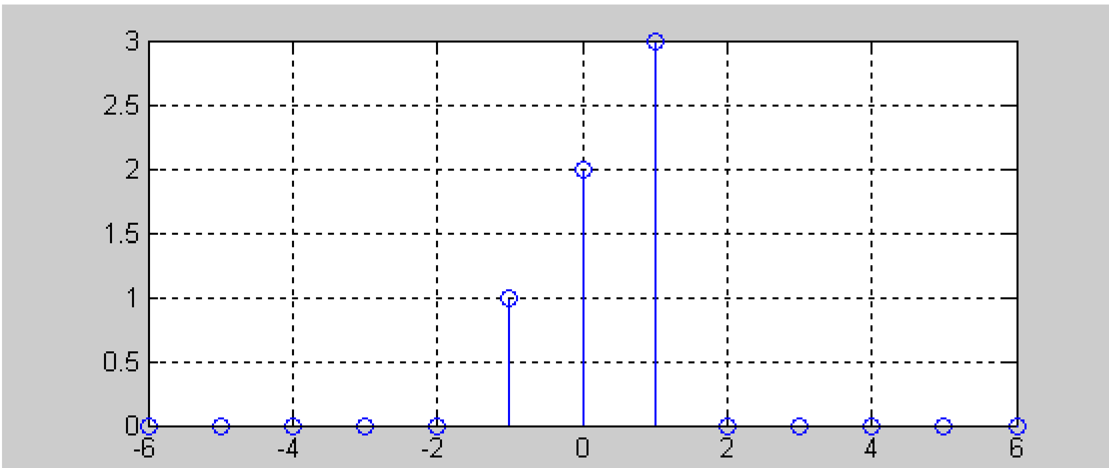
Example: Rxx

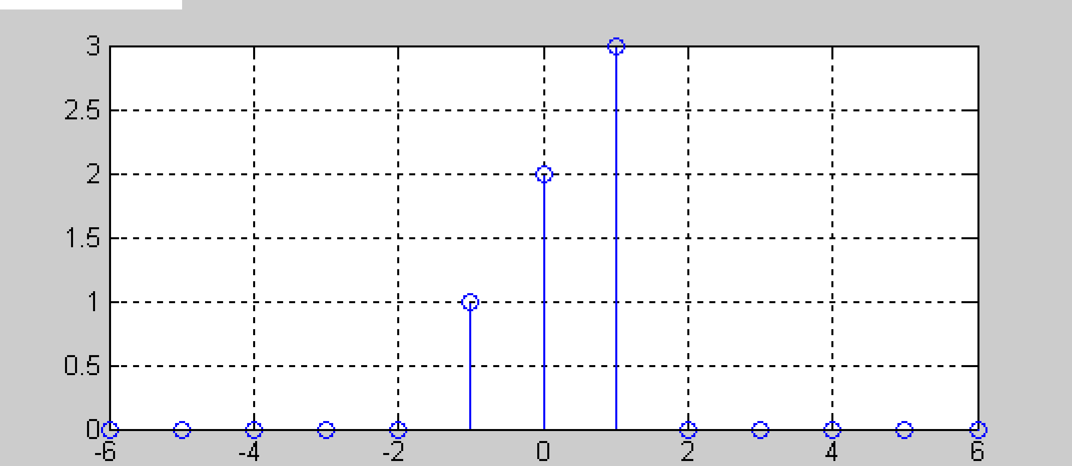
- x=[1 $\underline{2}$ 3]



Example: Rxx

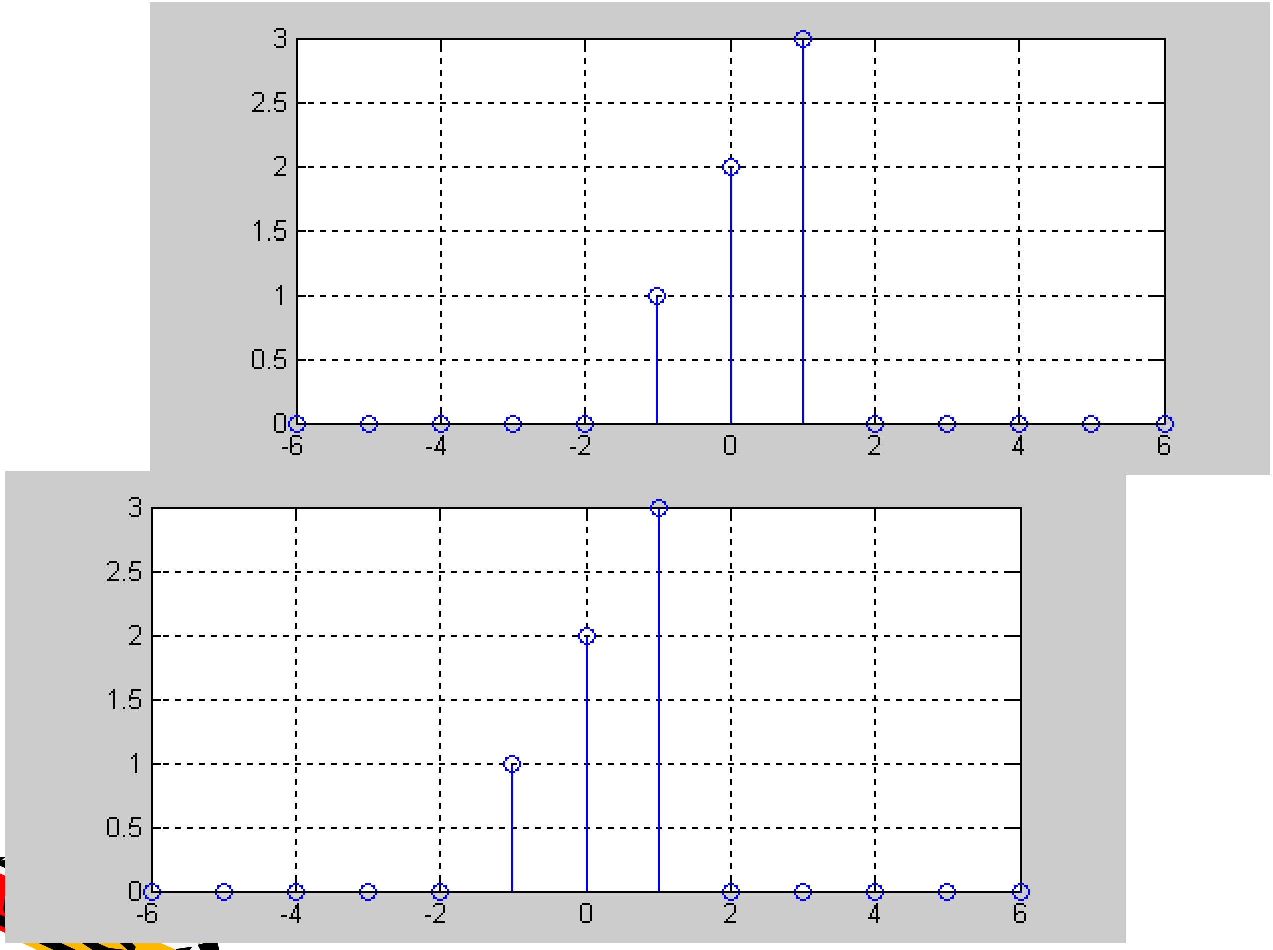
- $x=[1\ \underline{2}\ 3]$





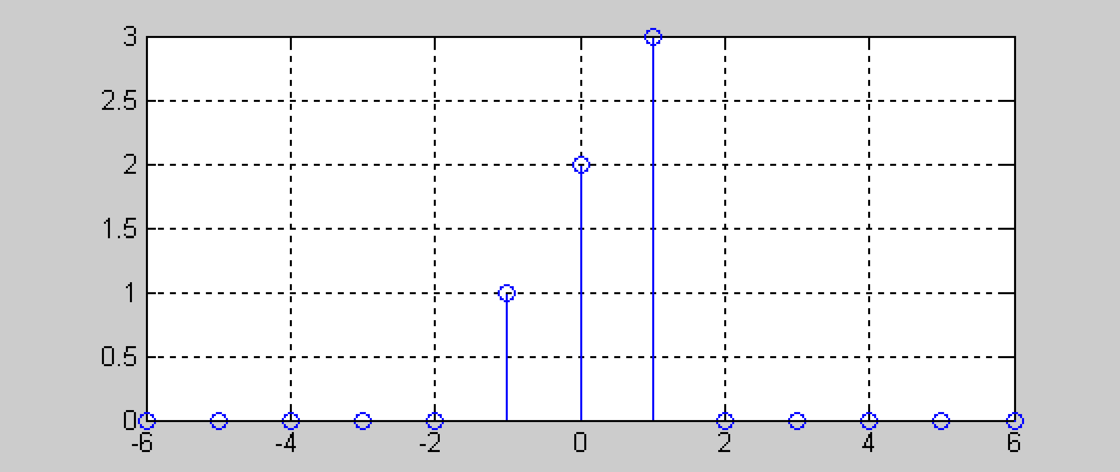
Example: Rxx

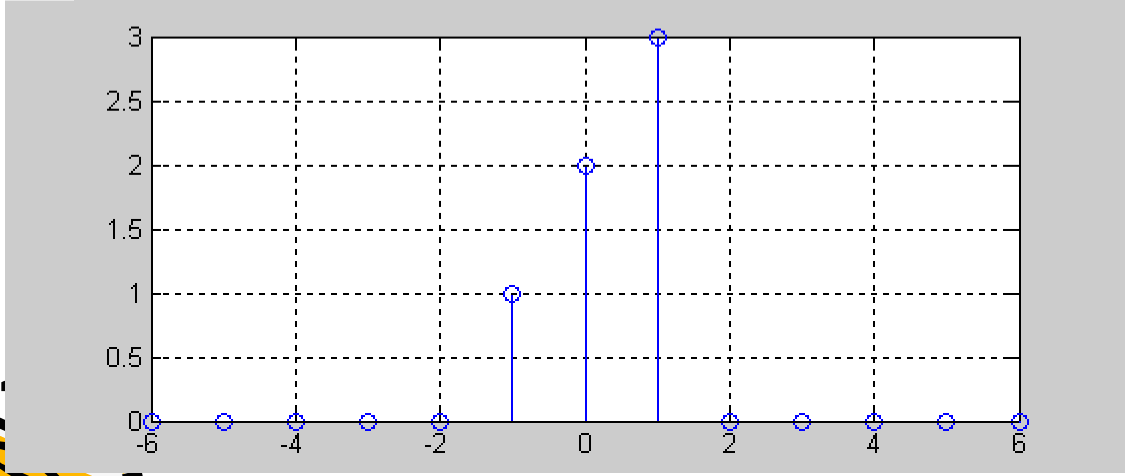
- x=[1 2 3]



Example: Rxx

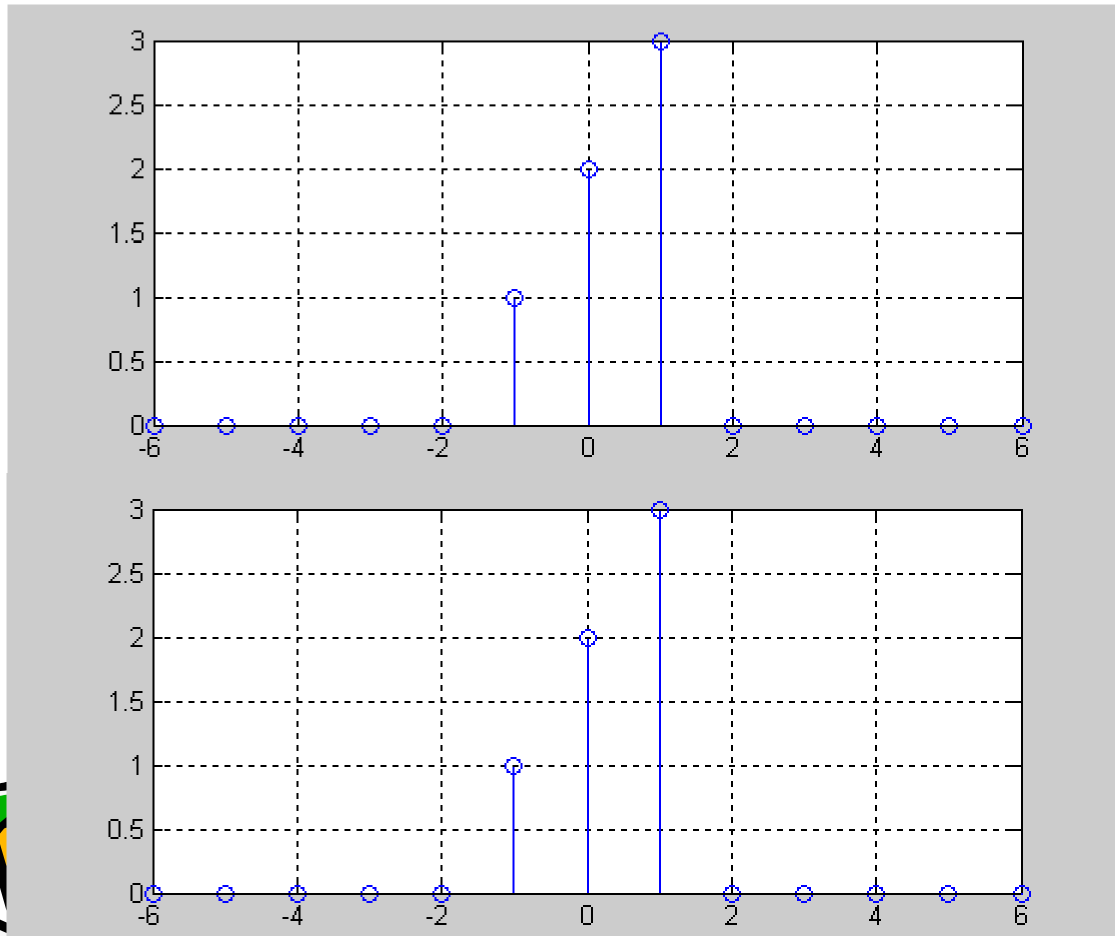
- x=[1 <u>2</u> 3]





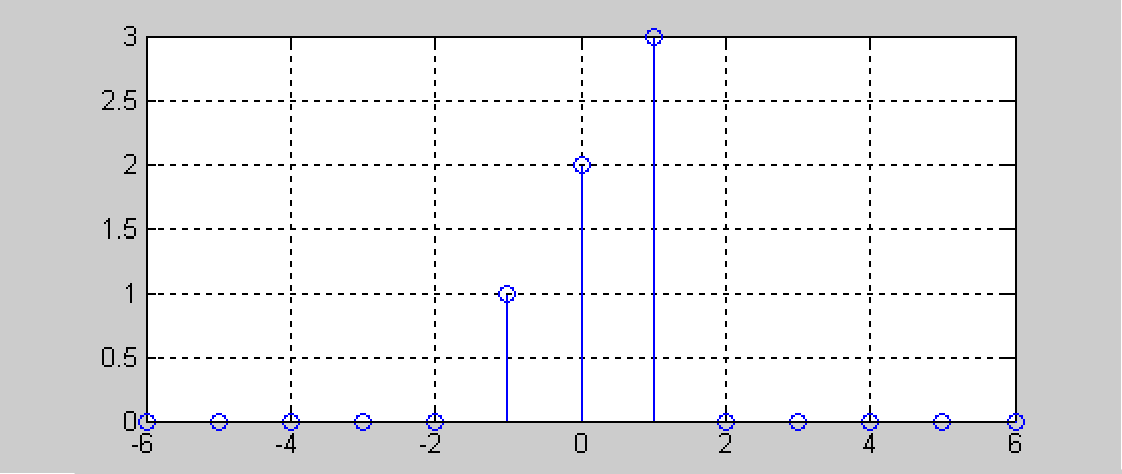
Example: Rxx

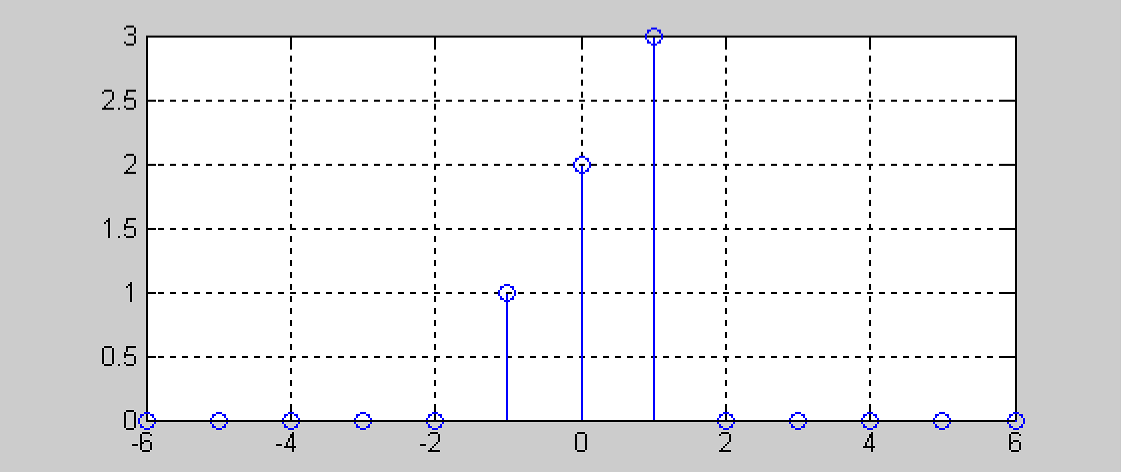
- x=[1 2 3]



Example: Rxx

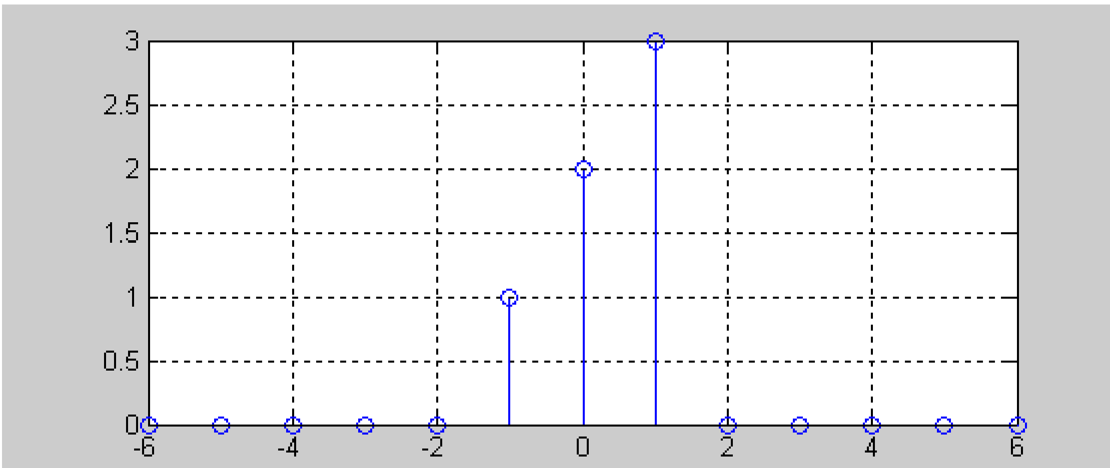
- x=[1 <u>2</u> 3]

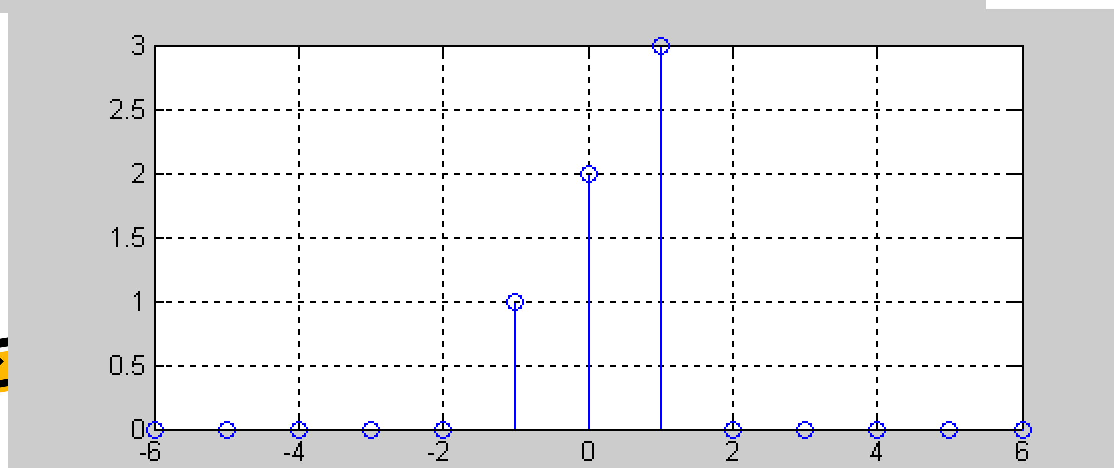




Example: Rxx

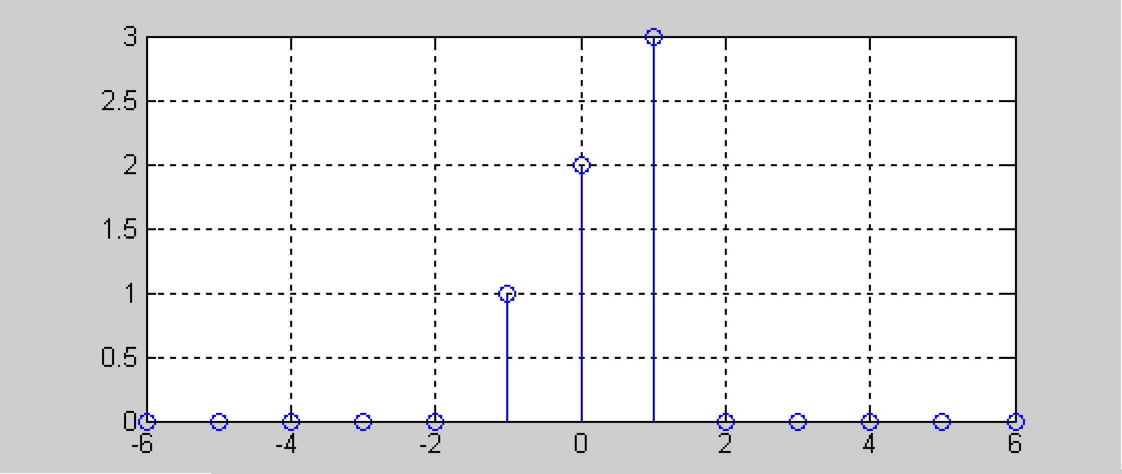
- $x=[1\ \underline{2}\ 3]$

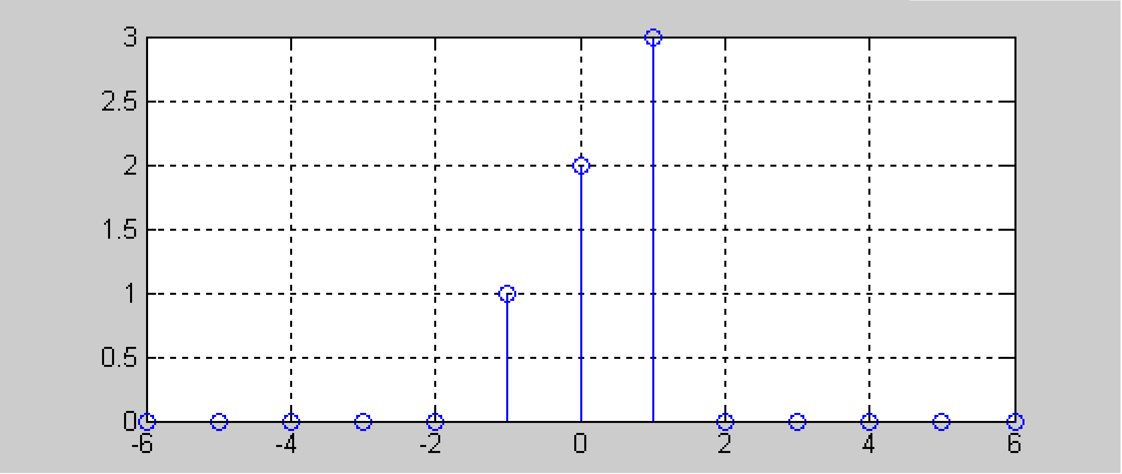




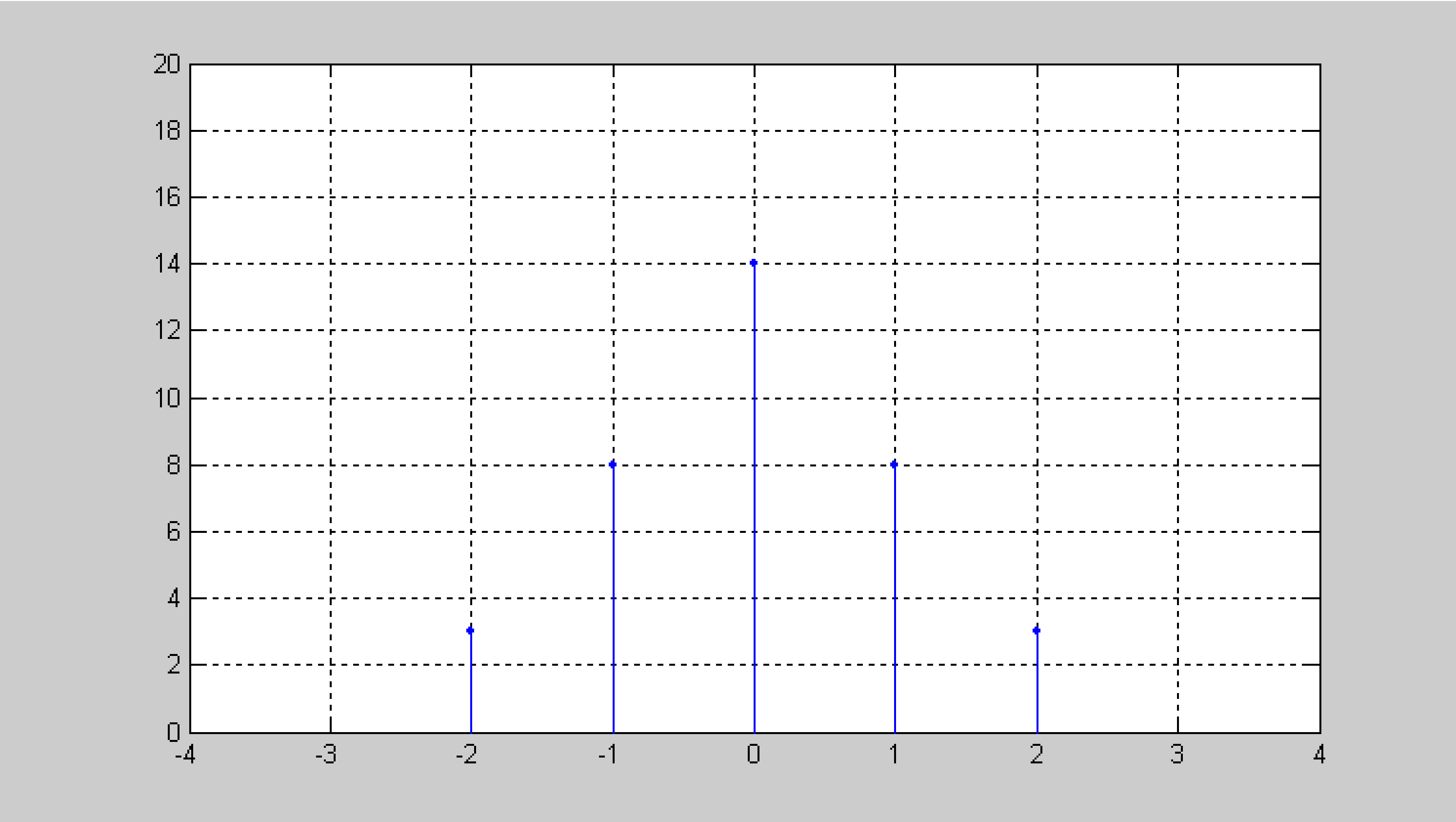
Example: Rxx

- x=[1 2 3]



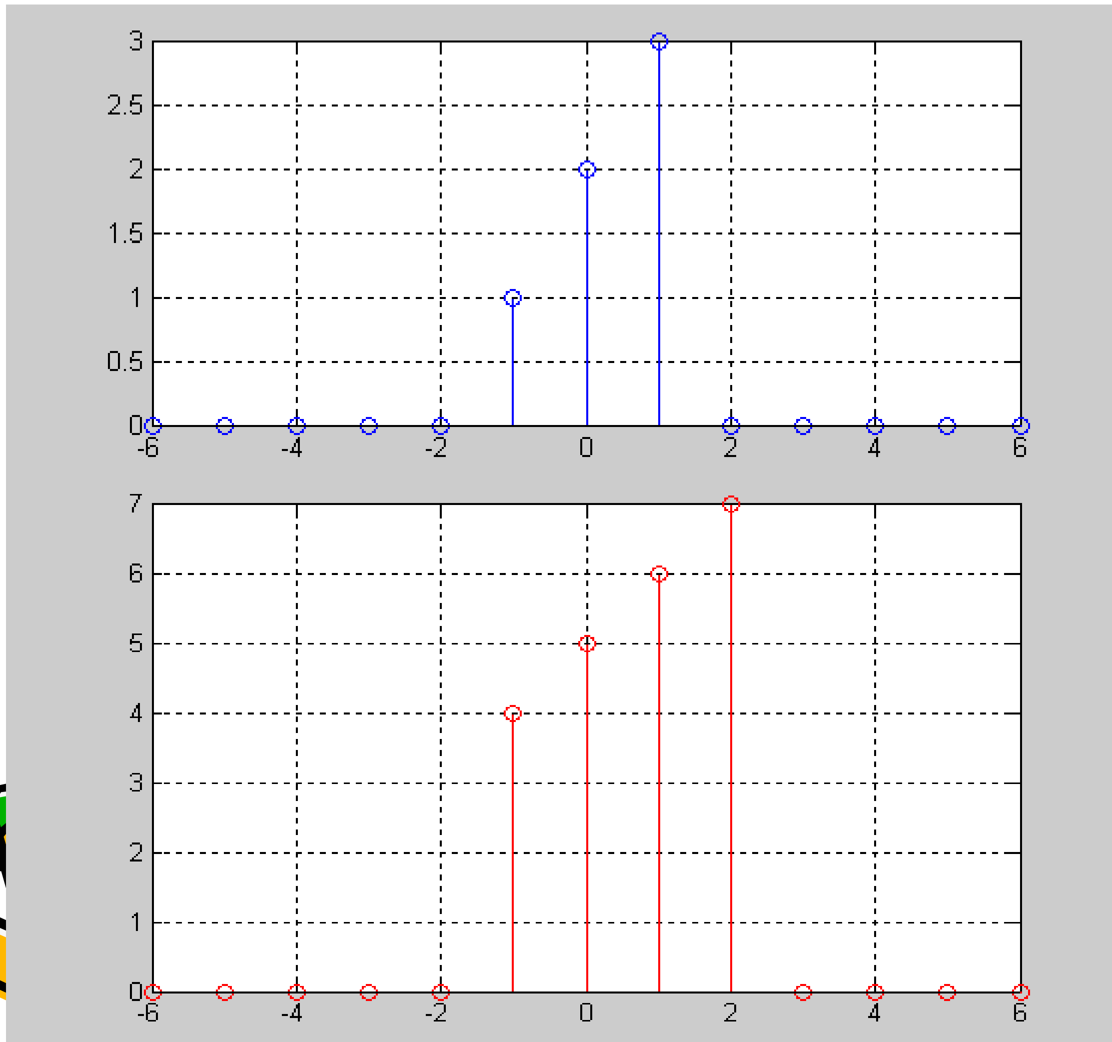


Rxx(m)



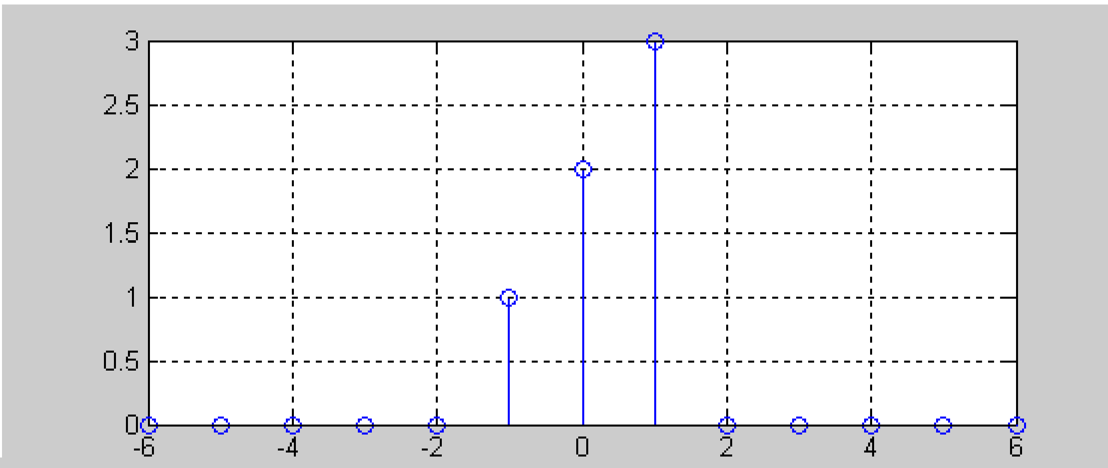
Example: Rxy

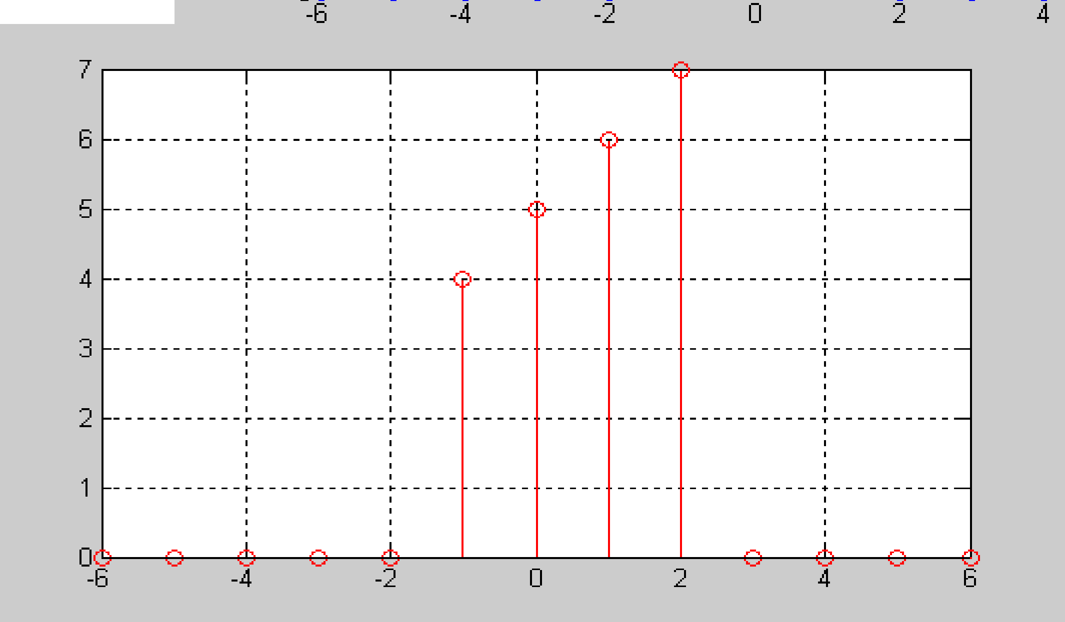
- x=[1 $\underline{2}$ 3], y=[4 $\underline{5}$ 6 7]



Example: Rxy

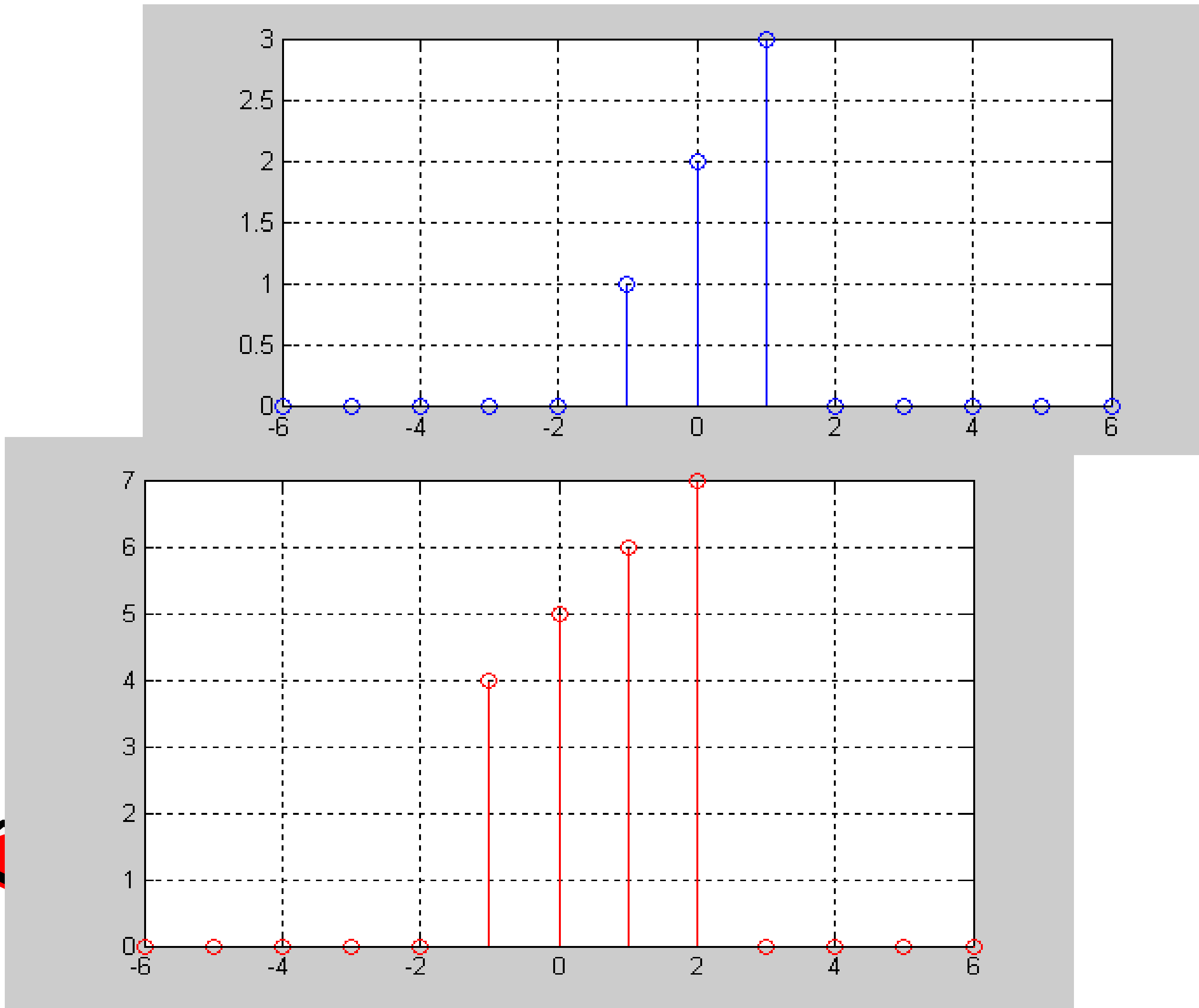
- x=[1 <u>2</u> 3], y=[4 <u>5</u> 6 7]





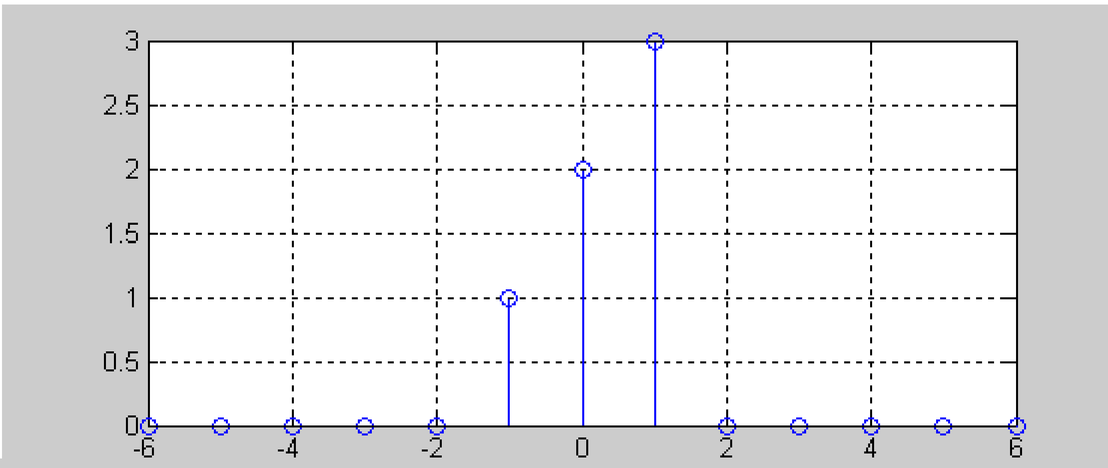
Example: Rxy

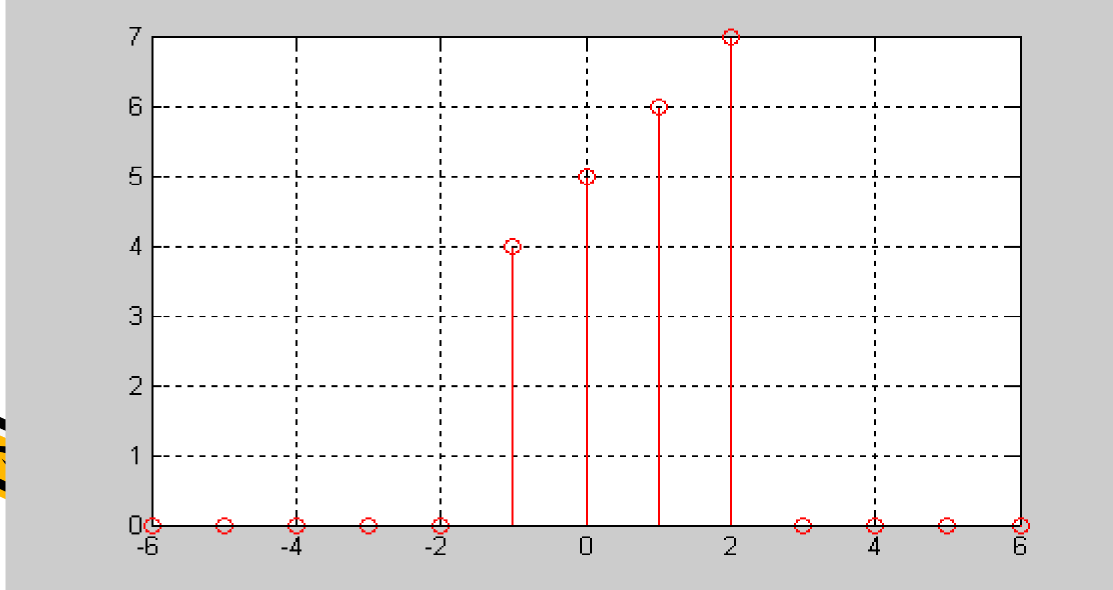
- x=[1 2 3], y=[4 5 6 7]



Example: Rxy

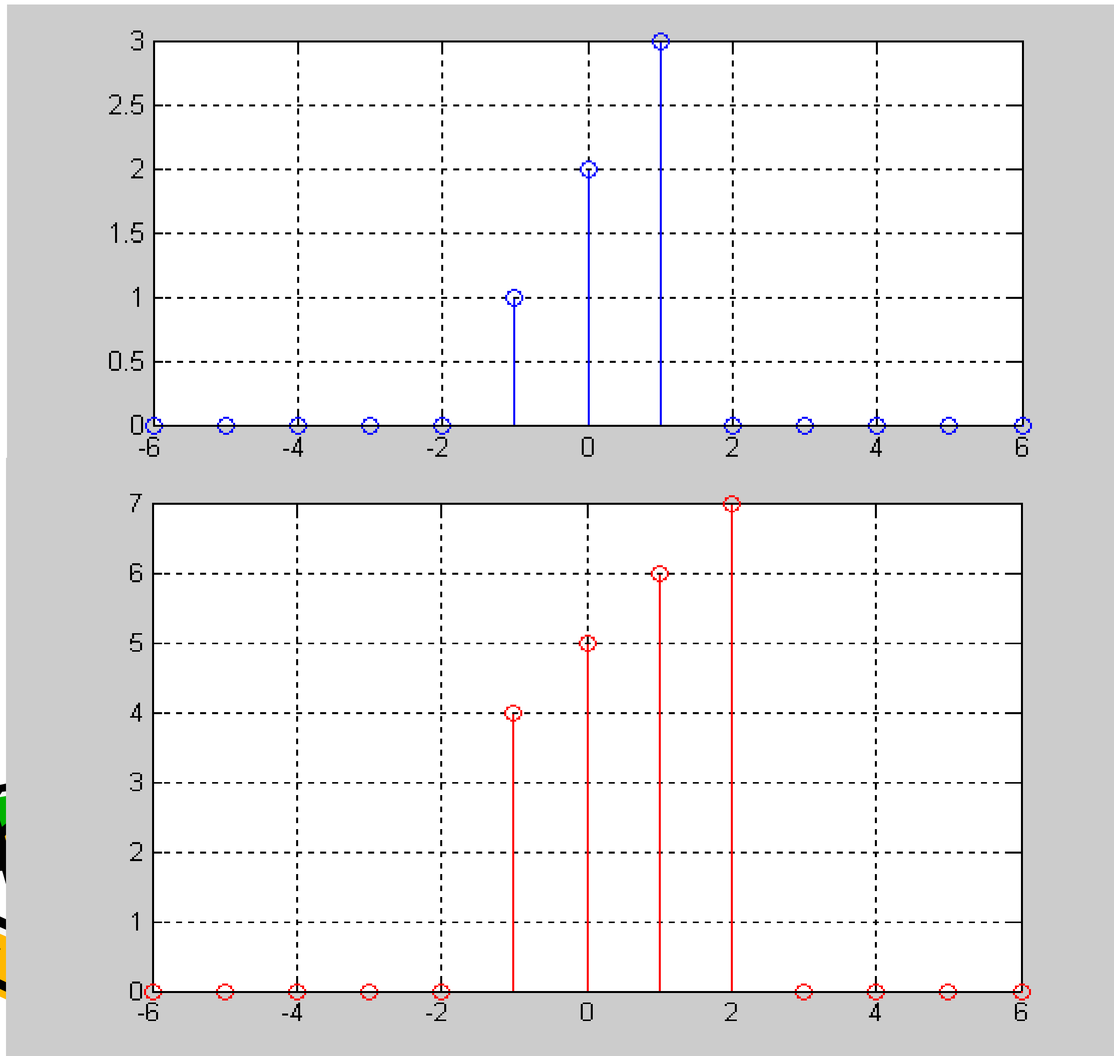
- x=[1 2 3], y=[4 5 6 7]





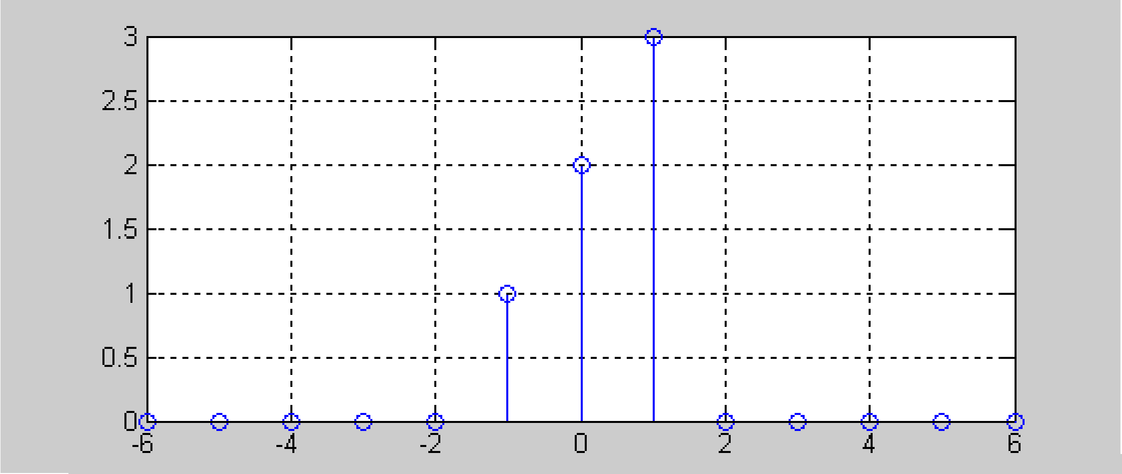
Example: Rxy

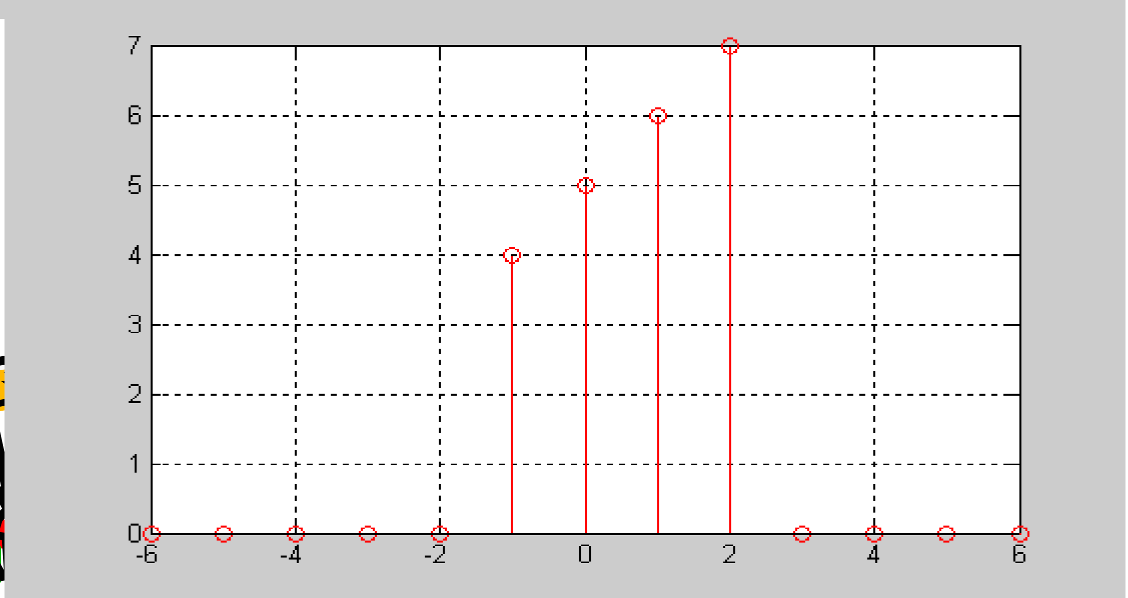
- x=[1 2 3], y=[4 5 6 7]



Example: Rxy

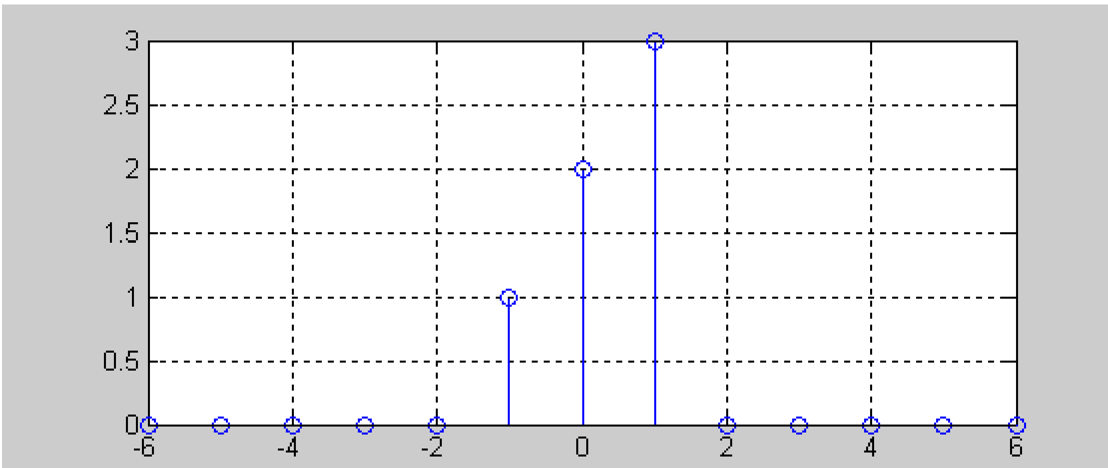
- x=[1 2 3], y=[4 5 6 7]

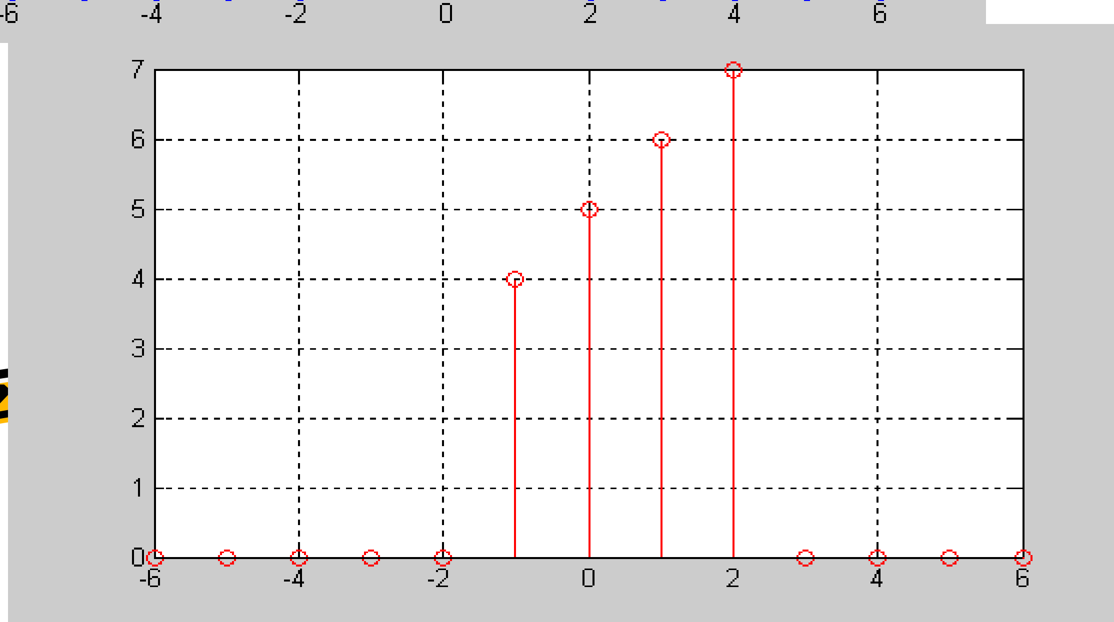




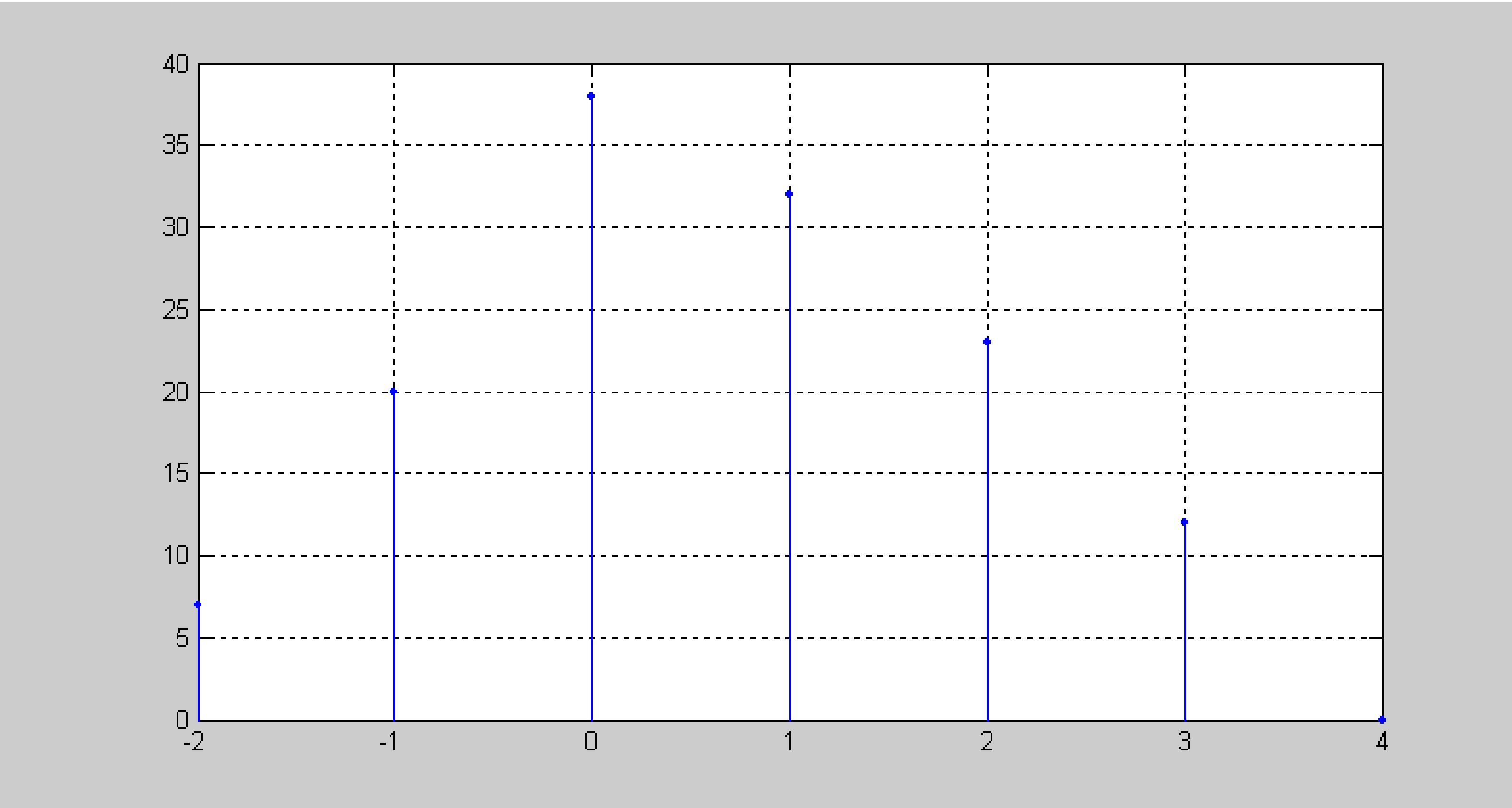
Example: Rxy

- x=[1 2 3], y=[4 5 6 7]





Rxy(m)



End of Week 5

- Download HW 4: Probability
  - ส่งต้นชั่วโมงสัปดาห์หน้า

CPE 332 Computer Engineering Mathematics II

Week 6

Part II, Chapter 5 Random Process

MarKov Process

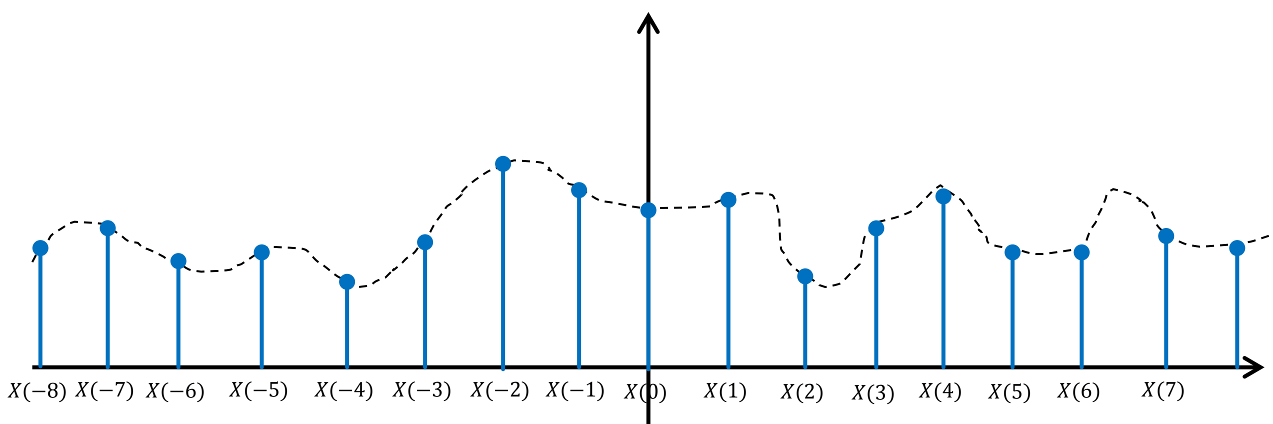
Chapter 6: Introduction to Queuing

Today Topics

- Random Process
  - Stationary: สถิติไม่เปลี่ยน
  - Ergodic: Ensemble Average=Time Average
  - Autocorrelation
  - Cross Correlation
- ต่อ
- Counting Process
- MarKov Process

Discrete-Time Random Process

- อาจจะได้จากการสุ่มตัวอย่าง (Sampling) ของ Continuous RP
  - Uniform Sampling ด้วย Sampling Period $T = \frac{1}{fs}$
- เราได้ $\dots, \mathrm{X}(-\mathrm{T}), X(0), X(T), X(2T), X(3T), \dots$
- ปกติจะละ Sampling Period ไว้ฐานที่เข้าใจ เราได้ $\dots, X(-1), X(0), X(1), X(2), X(3), \dots$
- มักจะเขียนในลักษณะ $X(n); n = integer$



จำกัด Sequence ความยาว N

- ค่า Autocorrelation สำหรับ N Samples
  - สมการจะลดรูป เหลือแค่ Sum และเฉลี่ย N Point แต่จะเกิดการ Biased เพราะเราเฉลี่ยน้อยกว่านั้น

$$R_{XX}(m) = \begin{cases} \frac{1}{N}\sum_{n=0}^{N-m-1} x(n)x(n+m); & 0 \le m \le N-1 \\ R_{XX}(-m); & -N+1 \le m < 0 \end{cases} \quad ; \quad \text{Biased}$$

ดังนั้นการคำนวณที่ไม่ Biased จะเป็นค่าที่ Normalized จากจำนวนจุดของการคำนวณจริงๆ และเราได้

$$R_{XX}(m) = \begin{cases} \frac{1}{N-|m|}\sum_{n=0}^{N-m-1} x(n)x(n+m); & 0 \le m \le N-1 \\ R_{XX}(-m); & -N+1 \le m < 0 \end{cases} \quad ; \quad \text{Non - biased}$$

Sequence ทั่วไป

ในกรณีที่สัญญาณไม่ได้เริ่มจาก $n = 0$ และ/หรือทั้งสอง Random Variable มีความยาวไม่เท่ากัน ผลลัพธ์ที่ได้จะมีความยาวของ Sequence $N + M - 1$ โดยที่ N และ M เป็นความยาวของทั้งสอง Variable และในกรณีนี้ค่า Index ของ Summation จะเปลี่ยนไป อย่างไรก็ตาม สมการข้างล่างยังใช้ได้อยู่สำหรับ Raw Data ที่ไม่ได้ทำการ Normalized

$$R_{XX}(m) = \sum_{n=-\infty}^{\infty} x(n)x(n+m)\,;\, -\infty < m < \infty \quad \text{Raw Data}$$

$$R_{XY}(m) = \sum_{n=-\infty}^{\infty} x(n)y(n+m)\,;\, -\infty < m < \infty \quad \text{Raw Data}$$

[illegible] Correlation

- 1. ด้วยวิธีกราฟ
  - X(n) คูณ X(n+m) คือ X(n) ที่เลื่อนไปซ้าย m ตำแหน่ง
    - X(n-m) จะเลื่อนไปด้านขวาแทน
  - จากนั้นทำการคูณตัวอย่างที่ตำแหน่งเดียวกันและจับผลลัพธ์มาบวกกัน(Summation)
  - ถ้าเป็น Autocorrelation ทำแค่ครึ่งเดียวเพราะ Rxx(m) เป็น Even Function
    - Cross Correlation ต้องทำทั้งสองด้าน

คุณสมบัติของ Correlation

- 2. ด้วยการแตก Summation
  - $R_{XX}(m) = \sum_{n=-\infty}^{\infty} x(n)x(n+m) = R_{XX}(-m)$
  - $R_{XY}(\pm m) = \sum_{n=-\infty}^{\infty} x(n)y(n \pm m)$, not Even
  - Index ของ Summation จะสิ้นสุดแค่ Index ของ x(n) ก็พอ เพราะที่เหลือจะเป็น ศูนย์
  - เช่น x(n) = {2, $\underline{1}$, -1 ,3}, y(n) = {1,2,1,$\underline{-1}$,-3,4,5}

    -1 0 1 2   -3 -2 -1 0 1 2 3
  - $R_{XX}(2) = \sum_{n=-1}^{2} x(n)x(n+2) = R_{XX}(-2)$
    - $= x(-1)x(1) + x(0)x(2) + x(1)x(3) + x(2)x(4)$
    - $= 2(-1) + 1(3) + (-1)\cdot 0 + 3 \cdot 0 = 1 = R_{XX}(-2)$
  - $R_{XY}(-1) = \sum_{n=-1}^{2} x(n)y(n-1) \neq R_{XY}(1)$
    - $= x(-1)y(-2) + x(0)y(-1) + x(1)y(0) + x(2)y(1)$
    - $= 2 \cdot 2 + 1 \cdot 1 + (-1)\cdot(-1) + 3 \cdot (-3) = -3$

Random Process [illegible]

- Counting Process
- Birth and Death Process
- Poisson Process
- MarKov Process
- MarKov Chain

Counting Process

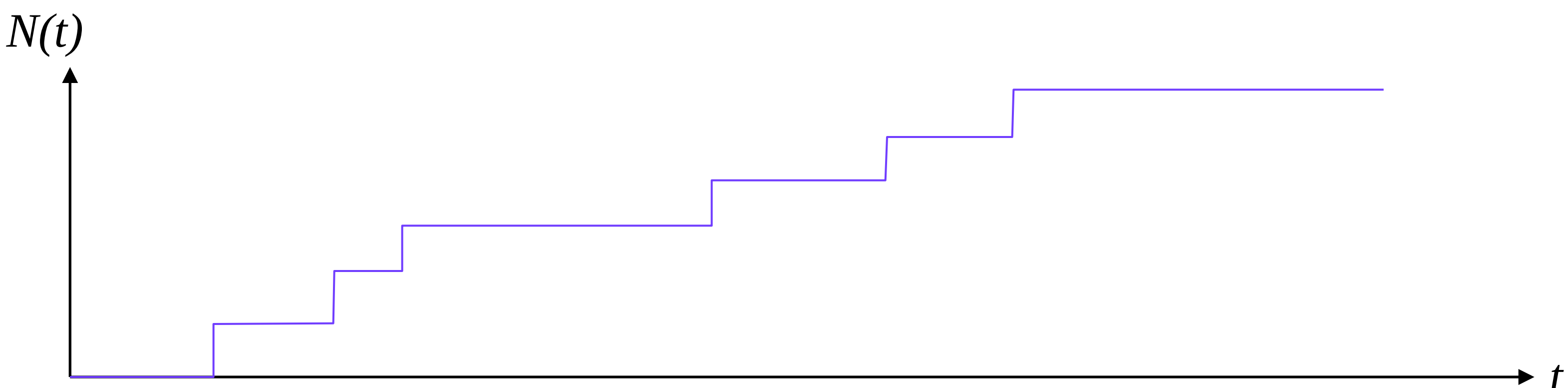
นิยาม $\{N(t), t \geq 0\}$ จัดว่าเป็น **counting process** เมื่อ

(1) $N(0) = 0$,

(2) $N(t)$ มีค่าเฉพาะเป็น integer ที่มีค่าบวก,

(3) ถ้า $s < t$ เราจะได้ $N(s) < N(t)$, และ

(4) $N(t) - N(s)$ เป็นจำนวนของ event ที่เกิดขึ้นหลัง s จนถึง t นั่นคือในช่วงเวลา $(s, t]$



Poisson Process

นิยาม Counting process $\{N(t), t \geq 0\}$ ใดๆ จะเป็น **Poisson Process** ด้วยอัตรา $\lambda > 0$ ถ้ามันเป็นไปตามข้อกำหนด 4 ข้อ ดังนี้

(1) Process เป็น independent increment (แต่ละ event ที่เกิดขึ้นในช่วงเวลาที่ไม่คาบเกี่ยวกัน จะไม่ขึ้นต่อกัน)

(2) การเพิ่มขึ้นของ process เป็น stationary (Distribution ของจำนวนของ event ในช่วงระยะเวลาหนึ่งจะขึ้นอยู่กับเฉพาะความยาวของช่วงเวลา และไม่ขึ้นกับเวลาที่มันเริ่มต้น)

(3) Probability ที่หนึ่ง event เกิดขึ้นในช่วงเวลาใดๆที่มีความยาว h จะเท่ากับ $\lambda h + o(h)$ [1] กล่าวคือ

$$P[N(h) = 1] = \lambda h + o(h)$$

(4) Probability ที่จะมีมากกว่าหนึ่ง event เกิดขึ้นในช่วงเวลาใดๆที่มีความยาว h เท่ากับ $o(h)$ กล่าวคือ

$$P[N(h) \geq 2] = o(h)$$

---

[1] $o(h)$ อ่าน “little-oh” ของ h ตามนิยามหมายถึงว่าถ้า $f(x) = o(g(x))$ แล้วแปลว่า $f(x) = O(g(x))$ แต่ $f(x) \neq \Theta(g(x))$ ซึ่งถ้า $f = o(h)$ ก็ต่อเมื่อ $\lim_{h \to 0} \frac{f(h)}{h} = 0$

Poisson Process

- ถ้าแต่ละเหตุการณ์ที่เกิดขึ้นเป็น Random และไม่ขึ้นต่อกัน มันจะเป็น Poisson
  - Probability ที่จะมี k เหตุการณ์เกิดในช่วงเวลา t สามารถคำนวณได้จากสูตร (ดูหน้าถัดไป)
  - ระยะเวลาระหว่างสองเหตุการณ์ที่เกิดขึ้น เรียก Inter-arrival time, τ, จะมีการกระจายแบบ Exponential ด้วยค่าเฉลี่ย 1/λ,
    - $F_X(\tau) = P[X \leq \tau] = 1 - \lambda e^{-\lambda\tau}, f_X(\tau) = \lambda e^{-\lambda\tau}$

Poisson Process

ทฤษฎีบท ให้ $\{N(t), t \geq 0\}$ เป็น Poisson process ด้วยอัตรา $\lambda > 0$ ดังนั้นค่า Random Variable Y ซึ่งคือจำนวนของ event ที่เกิดขึ้นในช่วงเวลาใดใดของ $t > 0$ จะมีการกระจายแบบ Poisson ที่มีค่า parameter เท่ากับ λt กล่าวคือ

$$P[Y = k] = e^{-\lambda t}\frac{(\lambda t)^k}{k!}, \quad k = 0,1,2,...$$

นั่นก็คือค่าเฉลี่ยของจำนวน event ที่เกิดขึ้นในช่วงเวลา t เท่ากับ λt

ทฤษฎีบท ให้ $\{N(t), t \geq 0\}$ เป็น Poisson process ด้วยอัตรา λ ให้ $0 < t_1 < t_2 < ...$ เป็นเวลาที่ต่อเนื่องกันที่เกิด event และให้ ช่วงเวลา $\{\tau_n\}$ เรียก **Interarrival time** โดย

$$\tau_1 = t_1, \tau_2 = t_2 - t_1, ..., \tau_k = t_k - t_{k-1}, ...$$

ดังนั้นค่า Interarrival time $\{\tau_n\}$ จะเป็น exponential random variable ที่เป็น mutually independent identically distributed(iid) ซึ่งแต่ละอันจะมีค่า mean เท่ากับ $1/\lambda$

Birth and Death Process

5.6 Birth-and-Death Process

ในระบบหนึ่งๆ นอกจากจะมี Event ที่เกิดขึ้นในระบบแล้ว ยังมี Event ที่จบลงอีกด้วย การเกิด Event เราเรียกว่า Birth ซึ่งจะถูกกำหนดโดย Birth Rate ซึ่งก็คือค่า λ ในการศึกษาระบบที่มีขนาดไม่ใหญ่นัก ค่าของ Birth Rate จะขึ้นอยู่กับจำนวนของประชากรในขณะนั้น ในกรณีนี้ค่า Birth Rate จะเป็น λ_n

สำหรับอัตราการจบลงของ Event เราเรียก Death Rate และปกติค่านี้จะขึ้นอยู่กับจำนวนของประชากรเช่นกัน ค่าของ Death Rate เราให้เป็น μ_n ในกรณีพิเศษ mujProcess ใดๆที่มีแต่การเกิด กล่าวคือ $\mu_n = 0$ เราจะเรียกระบบนั้นว่าเป็น Pure-Birth Process

ระบบที่ประกอบไปด้วยทั้งการเกิดและการตาย เราเรียก Birth-and-Death Process พึงสังเกตอย่างหนึ่งว่า Poisson Process ที่เรากล่าวในหัวข้อก่อนนั้น ความจริงแล้วก็คือกรณีพิเศษของ Birth-and-Death Process ที่เป็น Pure-Birth Process ที่มี Birth Rate คงที่

State Diagram

- พิจารณาจากระบบ มีทั้ง Birth ด้วย Birth Rate $\lambda(t)$ และ Death ด้วย Death Rate $\mu(t)$
  - เมื่อเราให้ระบบทำงาน ในระบบจะไม่มีอะไรอยู่ เราเรียกว่าอยู่ที่ State 0
  - เมื่อมีหนึ่ง Event เข้ามา หรือ Birth ระบบจะมี Event เพิ่มขึ้นและจะไปอยู่ที่ State ที่มากกว่าปัจจุบัน “หนึ่ง”
  - เมื่อมีหนึ่ง Event จบลง (Death) ระบบจะลด State ลงหนึ่ง
  - ค่า State ของระบบคือจำนวน Event ที่มีอยู่ในระบบ
  - การกระโดดไปยัง State ที่สูงกว่า หรือต่ำกว่า สามารถกำหนดด้วย Probability และเขียนได้ในลักษณะของ State Diagram
    - ถ้าระบบไม่มีการจดจำ เราเรียก Diagram นี้เป็น MarKov Model
    - ระบบคือ MarKov Process
    - การ Transition จาก State หนึ่ง ไปอีก State หนึ่ง กำหนดได้โดยค่า Probability ที่คงที่

MarKov Process and Markov Chain

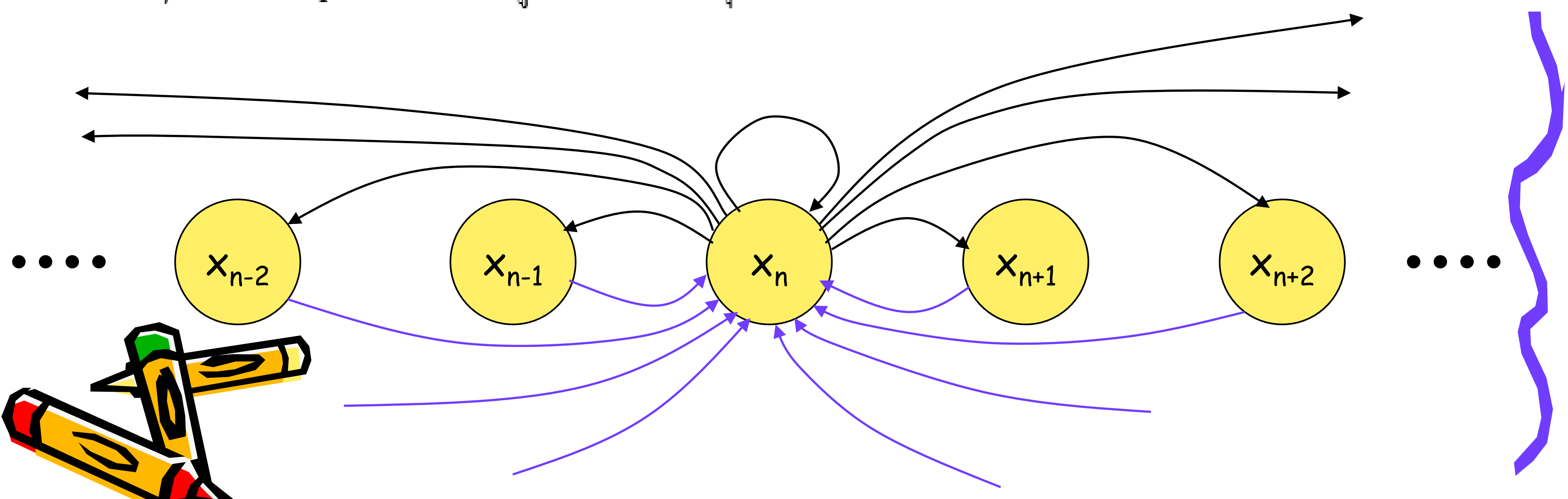
Stochastic Process ใดใดของ $\{X(t), t \in T\}$ จัดว่าเป็น **Markov Process** เมื่อมี set ที่ประกอบไปด้วย $n+1$ ค่า

ของ $t_1 < t_2 < \cdots < t_n < t_{n+1}$ ใน index set และ set ของ state $\{x_1, x_2, \ldots, x_n, x_{n+1}\}$ จำนวน $n+1$ state ที่ทำให้

$$P[X(t_{n+1}) = x_{n+1} | X(t_1) = x_1, X(t_2) = x_2, \ldots, X(t_n) = x_n]$$

$$= P[X(t_{n+1}) = x_{n+1} | X(t_n) = x_n]$$

นั่นก็คือพฤติกรรมของ process จะขึ้นอยู่กับ state ในปัจจุบัน แต่จะไม่ขึ้นกับ state ก่อนหน้านี้

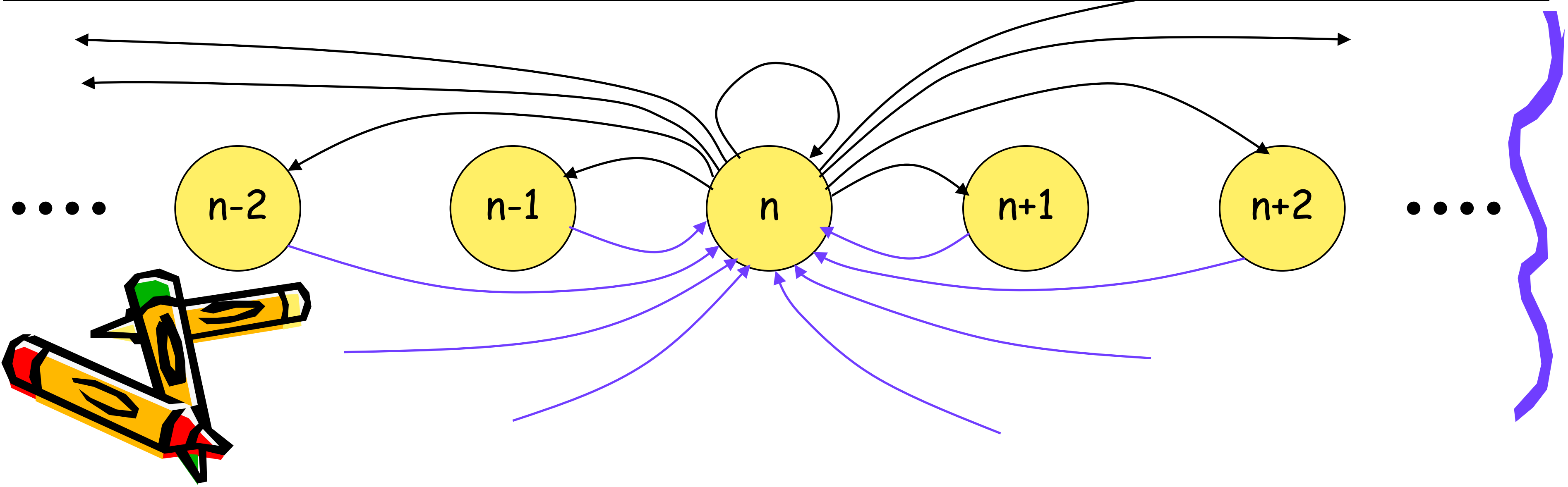


Markov Process and Markov Chain

Markov Process จะเรียกว่าเป็น **Markov Chain** ถ้า State Space นั้นเป็น Discrete ดังนั้นเราสามารถจำแนก Markov Process ได้ดังตารางที่ 1

ตารางที่ 1 Classification ของ Markov Process

ชนิดของ Parameters	**State Space**	
	Discrete	**Continuous**
Discrete Time	**Discrete Time Markov Chain**	**Discrete Time Markov Process**
Continuous Time	**Continuous Time Markov Chain**	**Continuous Time Markov Process**



MarKov Chain

- สำหรับ MarKov Chain (Discrete State แต่ Time อาจจะเป็น Continuous หรือ Discrete)
  - ในขั้นนี้ เราจะเน้นที่ Discrete Time Markov Chain
  - สมมุติแกนเวลา ถูกแบ่งเป็น Time Slot ที่เท่ากัน และเราอยู่ที่ State i ใน Time Slot ปัจจุบัน ดังนั้น จะมีเหตุการณ์เกิดได้ดังนี้
    - ระบบอยู่ที่ State เดิม ใน Time Slot หน้า ด้วยค่า Probability P_{ii}
    - ระบบมีการเปลี่ยน State ไปยัง State อื่นๆ ด้วยค่า Probability $P_{ij}; j = 0,1,2,\ldots,N, j \neq i$

Note: Continuous Time MarKov Chain สามารถ Model จาก Discrete Time MarKov Chain โดยให้ Limit ระยะห่างของ Time Slot เข้าสู่ศูนย์

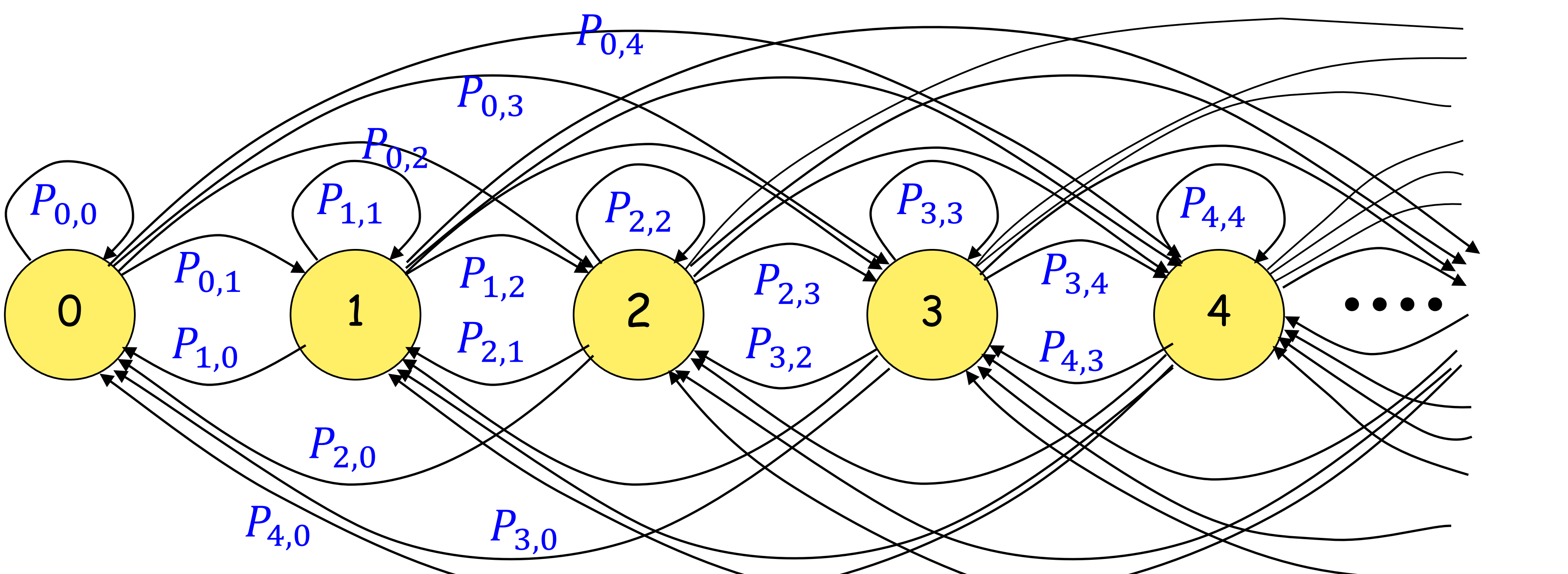
Discrete Time Markov Chain

เราจะเริ่มจาก discrete-time stochastic process ของ $\{X_n \mid n = 0,1,2,\ldots\}$ ซึ่งเป็นเซ็ทของ Nonnegative Integer โดยที่สถานะหรือ State ที่ Process นี้จะเป็นไปได้คือ $X_n = k \mid k = 0,1,2,\ldots$ โดยที่ k เป็น State ที่ X_n มีค่าอยู่ ซึ่ง Process นี้จะเรียกว่าเป็น **Markov Chain** ได้ก็ต่อเมื่อ **ถ้าที่ State** i ใดใด มีค่า **Probability** ที่แน่นอน P_{ij} ซึ่งเป็นค่า **Probability** ที่มันจะเปลี่ยนจาก **State** i ไปเป็น **State** j เมื่อ **Process** เปลี่ยนจาก X_n ไปยัง X_{n+1} โดยไม่คำนึงถึงว่าก่อนที่จะมาถึงที่ **State** i นี้ มันได้ผ่าน **State** อะไรมาก่อน นั่นก็คือในทางคณิตศาสตร์เรากล่าวได้ว่า สำหรับทุกๆค่าของ $n > 0, i_{n-1}, \ldots, i_0, i, j$ เราได้

$$P_{ij} = P\{X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0\}$$

$$= P\{X_{n+1} = j \mid X_n = i\}$$

ในรูปไม่ได้แสดง Transition หมดทุกเส้น



Discrete Time Markov Chain

เราเรียก P_{ij} ว่าเป็น **Transition Probability** ซึ่งจะต้องมีคุณสมบัติคือ

$$P_{ij} \geq 0, \quad \sum_{j=0}^{\infty} P_{ij} = 1, \quad i = 0,1,2,\ldots$$

ซึ่งเราสามารถเขียนในรูปของ Transition Probability Matrix ได้เป็น

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} & \cdots \\ P_{10} & P_{11} & P_{12} & \cdots \\ . & . & . & \cdots \\ P_{i0} & P_{i1} & P_{i2} & \cdots \\ . & . & . & \cdots \end{bmatrix}$$

ถ้าระบบมีได้ถึง State N, (State 0,1,2,....,N), Transition Matrix จะมีขนาด (N+1)คูณ(N+1)

Discrete Time Markov Chain

$$
P = \begin{bmatrix}
P_{00} & P_{01} & P_{02} & \cdots \\
P_{10} & P_{11} & P_{12} & \cdots \\
. & . & . & \cdots \\
P_{i0} & P_{i1} & P_{i2} & \cdots \\
. & . & . & \cdots
\end{bmatrix}
$$

คราวนี้ลองมาพิจารณา n-step Transition Probability

$$P_{ij}^{n} = P\{X_{m+n} = j \mid X_m = i\}; \quad n \geq 0; i, j \geq 0$$

ค่าP_{ij}^{n} สามารถคำนวณได้จากสมการของ **Chapman-Kolmogorov** ดังนี้

$$P_{ij}^{n+m} = \sum_{k=0}^{\infty} P_{ik}^{n} P_{kj}^{m}; \quad n, m \geq 0; i, j \geq 0$$

Notations: สำหรับ Discrete Time Markov Chain

- State Probability
  - คือ Probability ที่จะพบว่าระบบอยู่ที่ State ใดๆ $p_i = Probability\ State = i, i = 0,1,\ldots,N$
  - $\sum_{x=0}^{N} p_x = 1$
- Transition Probability
  - คือ Probability ที่ระบบจะกระโดดจาก State หนึ่งใน Time Slot ปัจจุบัน ไปยังอีก State หนึ่งใน Time Slot หน้า อย่าลืมว่าระบบไม่มีการจำ
  - $P_{ij} = P[X_n = i, X_{n+1} = j]; i,j = 0,1,\ldots,N$
  - $\sum_{j=0}^{N} P_{ij} = 1$
- ระบบอยู่ที่ Equilibrium ค่าเหล่านี้จะไม่เปลี่ยน

Discrete Time Markov Chain

สอง State i และ j จะเรียกว่ามีการ **Communicate** ถ้าสำหรับบางค่าของ n และ n' เราได้ $P_{ij}^{n} > 0$ และ $P_{ji}^{n'} > 0$ และถ้าทุกๆ state มีการ Communicate แล้ว เรากล่าวได้ว่า Markov Chain นั้น **Irreducible**

ถ้าแต่ละ state i ของ Markov Chain เราไม่สามารถหาค่า integer $d \geq 2$ ที่ทำให้ $P_{ii}^{n} = 0$ ยกเว้นเมื่อ n เป็นจำนวนเท่าของ d เรากล่าวได้ว่า Markov Chain นั้น เป็น **Aperiodic**

สุดท้าย ค่า probability distribution ของแต่ละ State $\{p_j \,|\, j \geq 0\}$ จะเรียกว่าเป็น **Stationary Distribution** สำหรับ Markov Chain ถ้า

$$p_j = \sum_{i=0}^{\infty} p_i P_{ij}, \quad j \geq 0 \qquad (1)$$

อย่างไรก็ตาม เราจะสนใจเฉพาะในกรณีของ irreducible และ aperiodic Markov Chain เท่านั้น เนื่องจากเป็นชนิดเดียวที่เราจะพบในความเป็นจริง(อย่างน้อยก็สำหรับวิชานี้) ซึ่งในกรณีนี้ ค่า state probability distribution จะได้

$$p_j = \lim_{n \to \infty} P_{jj}^{n}, \quad j \geq 0$$

Discrete Time Markov Chain

อย่างไรก็ตาม เราจะสนใจเฉพาะในกรณีของ irreducible และ aperiodic Markov Chain เท่านั้น เนื่องจากเป็นชนิดเดียวที่เราจะพบในความเป็นจริง(อย่างน้อยก็สำหรับวิชานี้) ซึ่งในกรณีนี้ ค่า state probability distribution จะได้

$$p_j = \lim_{n \to \infty} P_{jj}^n, \quad j \geq 0$$

เราสามารถพิสูจน์ได้ว่าค่า limit ข้างบนสามารถหาค่าได้ และเมื่อ $p_j > 0$ เราจะได้ $1/p_j$ เท่ากับค่า **mean recurrence time** ของ j (ค่า Expectation ของจำนวนของการ Transition ที่เกิดขึ้นในระหว่างที่มีการเข้าไปอยู่ที่ State j สองครั้งติดกัน) แต่ถ้า $p_j = 0$ ค่า mean recurrence time จะเป็นค่า infinity มองในอีกแง่หนึ่งก็คือค่า p_j สะท้อนถึงค่าเฉลี่ยของ propagation time ของ process ที่จะไปอยู่ที่ state j ดังนั้นใจความสำคัญสามารถสรุปเป็นทฤษฎีบทได้ดังนี้

Discrete Time Markov Chain

Theorem. ใน Markov Chain ที่เป็น irreducible และ aperiodic จะมีสิ่งที่เป็นไปได้สองแบบ

1. $p_j = 0$ สำหรับทุกๆค่าของ $j \geq 0$ ซึ่ง chain จะไม่มี Stationary Distribution
2. $p_j > 0$ สำหรับทุกๆค่าของ $j \geq 0$ ซึ่งค่า $\{p_j \mid j \geq 0\}$ เป็นค่า Stationary Distribution เฉพาะของ chain นั้น (unique stationary distribution)

ตัวอย่างที่เห็นได้ชัดสำหรับกรณีแรกข้างบนก็คือ $M/M/1$ queuing system เมื่อค่า Arrival rate λ มีค่าสูงกว่า service rate μ

ในกรณีที่สอง ซึ่งจะมีเรื่องของการแสดงคุณลักษณะ(Characteristic) ของ Stationary Distribution $\{p_j \mid j \geq 0\}$ เข้ามาเกี่ยวข้อง สำหรับในกรณีของ Queuing System เรามักจะใช้เทคนิคดังนี้

Discrete Time Markov Chain

จากสมการ $P_{jj} + \sum_{i=0,i\neq j}^{\infty} P_{ji} = 1$ เราคูณด้วย p_j ทั้งสองข้าง และใช้สมการข้างบน(สมการ 1) แทนค่า เราได้

$$p_j \sum_{i=0,i\neq j}^{\infty} P_{ji} = \sum_{i=0,i\neq j}^{\infty} p_i P_{ij} \qquad (2)$$

สมการเหล่านี้รู้จักกันในนามของ **Global Balance Equation** ซึ่งกล่าวว่า ที่ Equilibrium ค่า Probability ของการ Transition ออกจาก j (ด้านซ้ายของสมการที่ 2) เท่ากับ Probability ของการ Transition เข้าสู่ j (ด้านขวาของสมการที่ 2)

สมการ Global Balance Equation สามารถจะถูก Generalized โดยนำมาใช้กับทุกๆ State ใน Set ลองพิจารณาจาก Subset ของ State S โดยการรวมสมการที่(2) ตลอดทุกๆ $j \in S$ เราจะได้

$$\sum_{j\in S} p_j \sum_{i\notin S} P_{ji} = \sum_{i\notin S} p_i \sum_{j\in S} P_{ij} \qquad (3)$$

สมการข้างบนแสดงให้เห็นว่า probability ของการ transition ออกจาก set ของ state S เท่ากับ probability ของการ transition เข้าสู่ state S

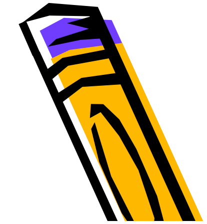
Discrete Time Markov Chain

เหตุที่เป็นเช่นนั้นก็เนื่องมาจากว่า เมื่อ Markov Chain เป็น irreducible สถานะของมันจะกลับเข้ามายัง state S (ด้วย probability เท่ากับหนึ่ง) นับครั้งไม่ถ้วน ดังนั้นแต่ละครั้งที่มีการ transition ออกมาจาก state S มันจะต้องมี(ด้วย probability เท่ากับหนึ่ง) การ transition กลับมาใน state S ในเวลาต่อมา ผลลัพธ์ก็คือ อัตราส่วนของการ transition ออกจาก S เทียบกับการ transition ทั้งหมด จะเท่ากับอัตราส่วนของการ transition เข้าสู่ S อันนี้คือความหมายที่แท้จริงของ global balance equation ในสมการที่ 3

สรุป MarKov Chain

- สถานะของระบบ ดูได้จากจำนวณ Event ที่อยู่ในระบบ เรียก State ของระบบ
  - ถ้า State เป็น Discrete เราได้ MarKov Chain
- มีค่า Probability สองชุดที่อธิบายการทำงานของระบบ
  - State Probability: Probability ที่ระบบจะอยู่ที่ State ใด State หนึ่ง
    - ผลรวมของ State Probability จะต้องเท่ากับ 1
  - Transition Probability: Probability ที่ระบบจะมีการเปลี่ยน State
    - อธิบายจาก Transition Matrix
    - ผลรวมของ Transition Probability แต่ละแถว จะต้องเท่ากับ 1
- ถ้าระบบอยู่ที่ Equilibrium ค่า Probability ของ State จะไม่เปลี่ยน และสามารถอธิบายได้ด้วย Global Balance Equation
- MarKov Chain ที่เราสนใจคือ Irreducible และ Aperiodic

Detailed Balance Equation: Simple MarKov Chain

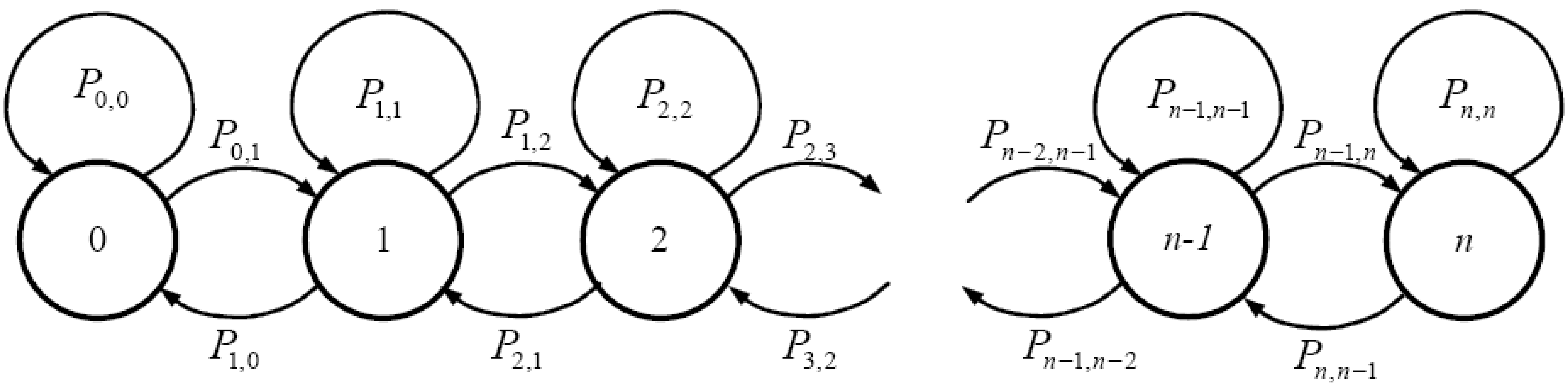


เพื่อให้เข้าใจถึงการนำ global balance equation ไปประยุกต์ใช้ ขอให้เราพิจารณาจาก Markov Chain ที่พบทั่วไปใน Queuing System กล่าวคือ **Birth-Death System** โดยที่สอง state ที่อยู่ติดกันจะเป็น state ที่ต่างกันเพียงหนึ่งหมายเลข ดังแสดงในรูปที่ 1 เราจะสมมุติว่า $P_{i,i+1} > 0$ และ $P_{i+1,i} > 0$ สำหรับทุกๆค่าของ i ซึ่งอันนี้จะเป็น Condition ที่จำเป็นและเพียงพอ (necessary and sufficient condition) สำหรับ Chain ที่จะเป็น irreducible   คราวนี้มาพิจารณาจาก set ของ state

$$S = \{0,1,2,\dots, n\}$$

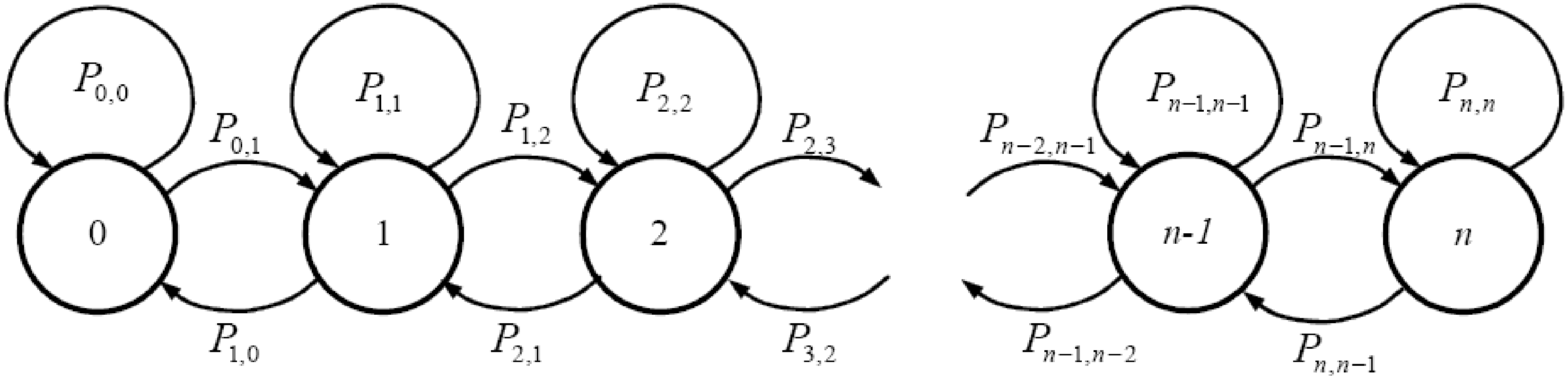
โดยใช้สมการที่ 3, เราได้

ระบบจะอยู่ที่ State เดิม หรือกระโดดไปยัง State ข้างเคียงเท่านั้น



รูปที่ 1   Transition Probability Diagram ของ Birth-Death Process

Markov Chain(Detailed Bal Eq)



รูปที่ 1 Transition Probability Diagram ของ Birth-Death Process

$$p_n P_{n,n+1} = p_{n+1} P_{n+1,n}, \quad n = 0,1,2,\ldots \tag{4}$$

กล่าวคือ ในสถานะ steady-state ค่า probability ของการ transition จาก n ไปเป็น $n+1$ จะเท่ากับ ค่า probability ของการ transition จาก $n+1$ ไปเป็น n สมการนี้มีประโยชน์ในการคำนวณค่า stationary distribution $\{p_j \mid j \geq 0\}$

สมการที่ 4 นี้ เป็นกรณีพิเศษของสมการ

$$p_j P_{ji} = p_i P_{ij}, \quad i, j \geq 0 \tag{5}$$

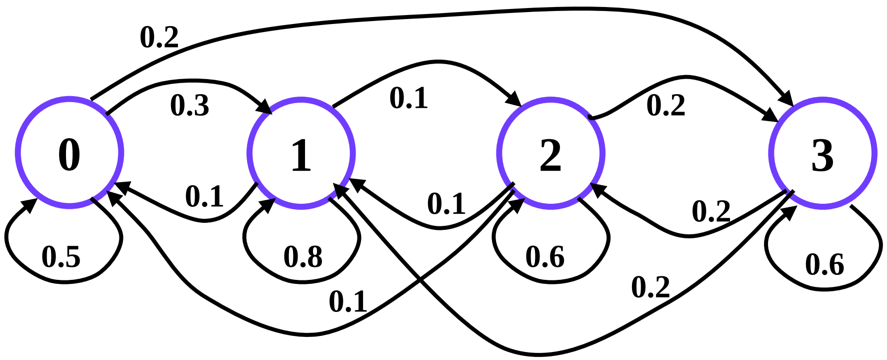
ซึ่งเรียกว่า **Detailed Balance Equation** สมการนี้ไม่จำเป็นต้องเป็นจริงเสมอไปสำหรับทุกๆ Markov Chain อย่างไรก็ตาม ในกรณีพิเศษหลายๆกรณีที่สำคัญมันจะเป็นจริง และถ้าเป็นเช่นนั้นแล้ว ก็จะช่วยให้การคำนวณค่า state stationary distribution ง่ายขึ้นมาก

- Transition Matrix ของ Simple Markov Chain จะมีลักษณะเป็น Tridiagonal

$$\mathbf{P} = \begin{bmatrix} P_{00} & P_{01} & & & 0 \\ P_{10} & P_{11} & P_{12} & & \\ & P_{21} & P_{22} & P_{23} & \\ & & & \ddots & P_{n-1,n} \\ 0 & & & P_{n,n-1} & P_{nn} \end{bmatrix}$$

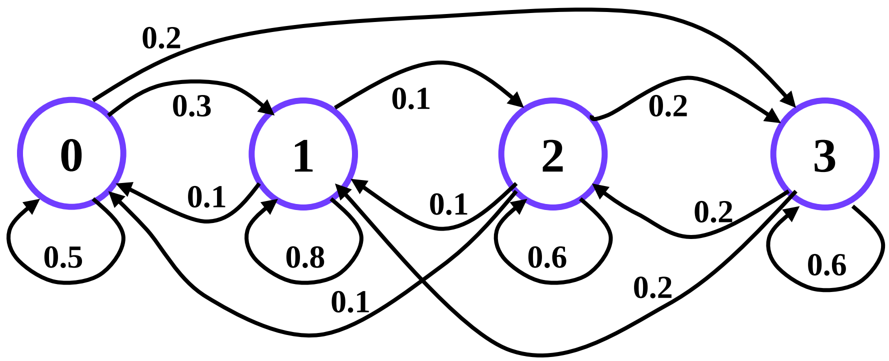
Example

- MarKov Chain แสดงด้วย Markov Model ดังรูปข้างล่าง
  - 1. จงหา Transition Matrix
  - 2. จาก Transition Matrix จงคำนวณหา State Probability



Example

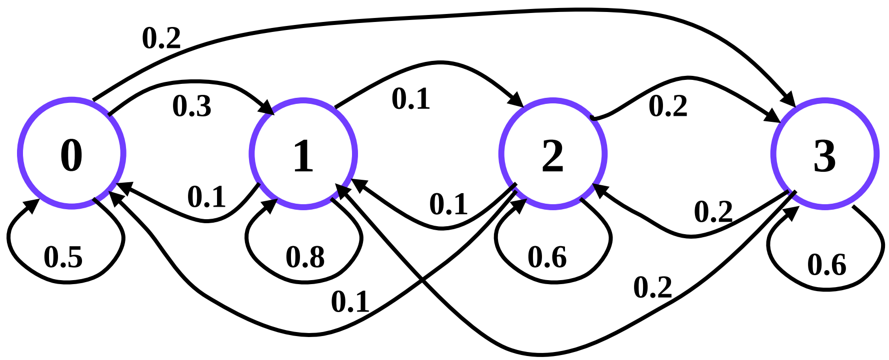
– 1. จงหา Transition Matrix



$$\mathbf{P} = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix}$$

Example

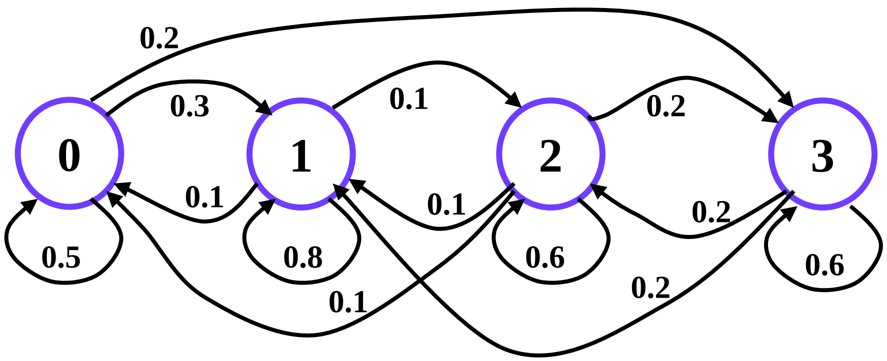
– 1. จงหา Transition Matrix



$$\mathbf{P} = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0 & 0.2 & 0.2 & 0.6 \end{bmatrix}$$

Example

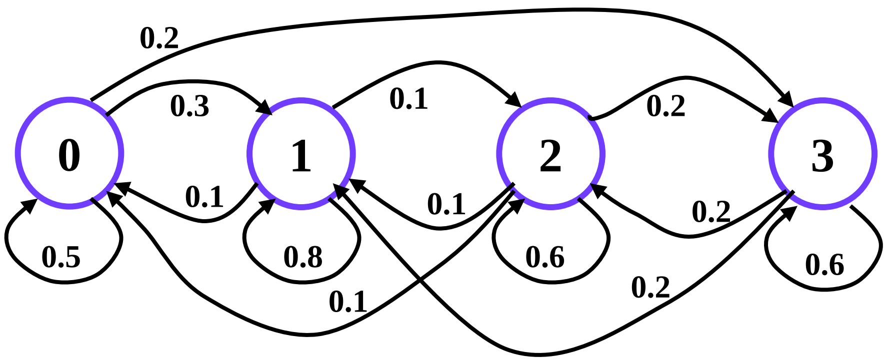
– 2. จงคำนวณหา State Probability



$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0 & 0.2 & 0.2 & 0.6 \end{bmatrix}$$

Example

– 2. จงคำนวณหา State Probability



$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0 & 0.2 & 0.2 & 0.6 \end{bmatrix}$$

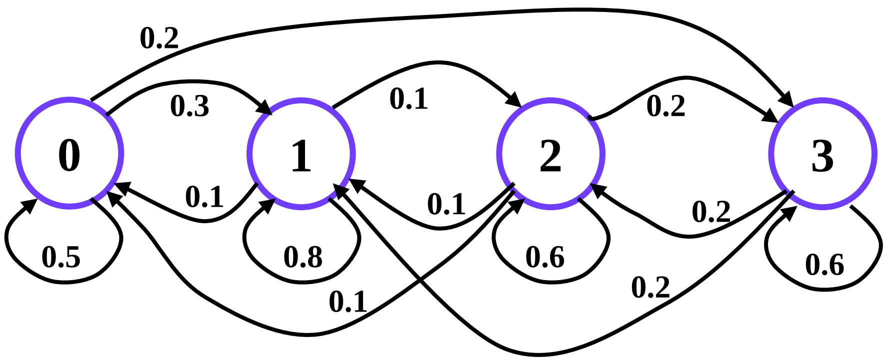
$$p_0 + p_1 + p_2 + p_3 = 1.0$$

Global Balanced Equation :

$$p_j \sum_{i=0, i\neq j}^{\infty} P_{ji} = \sum_{i=0, i\neq j}^{\infty} p_i P_{ij}$$

Example

– 2. จงคำนวณหา State Probability



$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0 & 0.2 & 0.2 & 0.6 \end{bmatrix}$$

$$p_0 + p_1 + p_2 + p_3 = 1.0$$

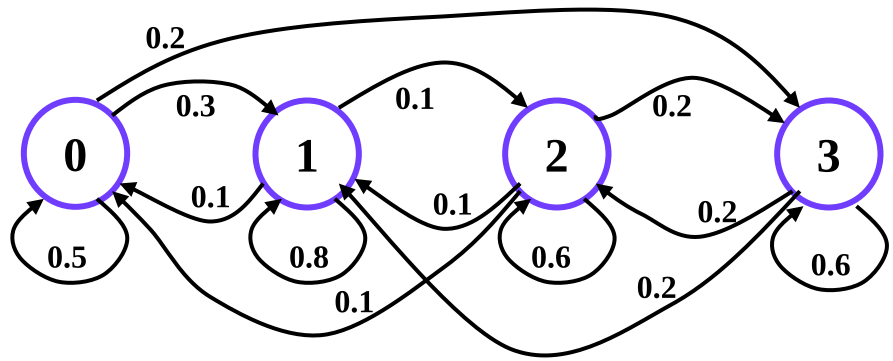
Global Balanced Equation :

$$p_j \sum_{i=0, i \neq j}^{\infty} P_{ji} = \sum_{i=0, i \neq j}^{\infty} p_i P_{ij}$$

$$p_0[P_{01} + P_{02} + P_{03}] = p_1 P_{10} + p_2 P_{20} + p_3 P_{30}$$

Example

– 2. จงคำนวณหา State Probability



$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0 & 0.2 & 0.2 & 0.6 \end{bmatrix}$$

$$p_0 + p_1 + p_2 + p_3 = 1.0$$

Global Balanced Equation :

$$p_j \sum_{i=0, i\neq j}^{\infty} P_{ji} = \sum_{i=0, i\neq j}^{\infty} p_i P_{ij}$$

$$p_0[P_{01} + P_{02} + P_{03}] = p_1 P_{10} + p_2 P_{20} + p_3 P_{30}$$

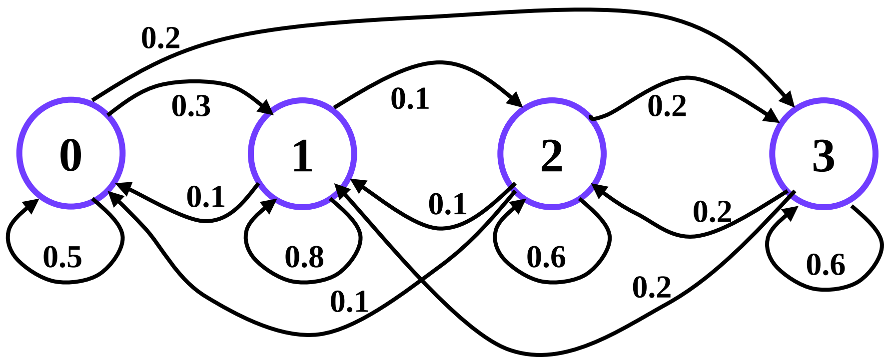
$$p_1[P_{10} + P_{12} + P_{13}] = p_0 P_{01} + p_2 P_{21} + p_3 P_{31}$$

$$p_2[P_{20} + P_{21} + P_{23}] = p_0 P_{02} + p_1 P_{12} + p_3 P_{32}$$

$$p_3[P_{30} + P_{31} + P_{32}] = p_0 P_{03} + p_1 P_{13} + p_2 P_{23}$$

Example

– 2. จงคำนวณหา State Probability



$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0 & 0.2 & 0.2 & 0.6 \end{bmatrix}$$

$$p_0 + p_1 + p_2 + p_3 = 1.0$$

Global Balanced Equation :

$$p_j \sum_{i=0, i \neq j}^{\infty} P_{ji} = \sum_{i=0, i \neq j}^{\infty} p_i P_{ij}$$

$$p_0[0.3 + 0 + 0.2] = 0.1p_1 + 0.1p_2 + 0.p_3$$

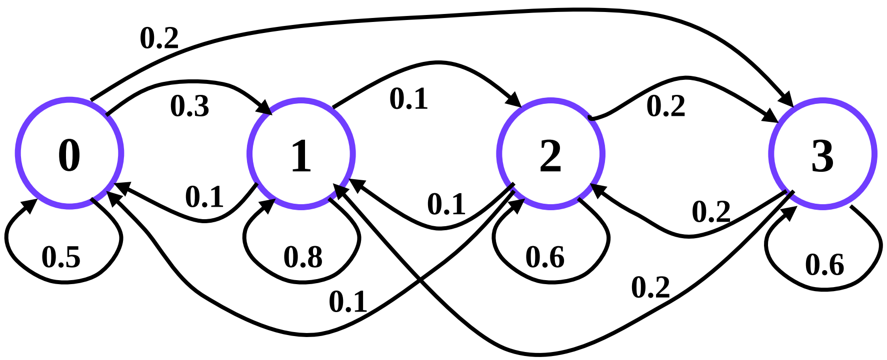
$$p_1[0.1 + 0.1 + 0] = 0.3p_0 + 0.1p_2 + 0.2p_3$$

$$p_2[0.1 + 0.1 + 0.2] = 0.p_0 + 0.1p_1 + 0.2p_3$$

$$p_3[0 + 0.2 + 0.2] = 0.2p_0 + 0.p_1 + 0.2p_2$$

Example

– 2. จงคำนวณหา State Probability



$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0 & 0.2 & 0.2 & 0.6 \end{bmatrix}$$

$$p_0 + p_1 + p_2 + p_3 = 1.0$$

Global Balanced Equation :

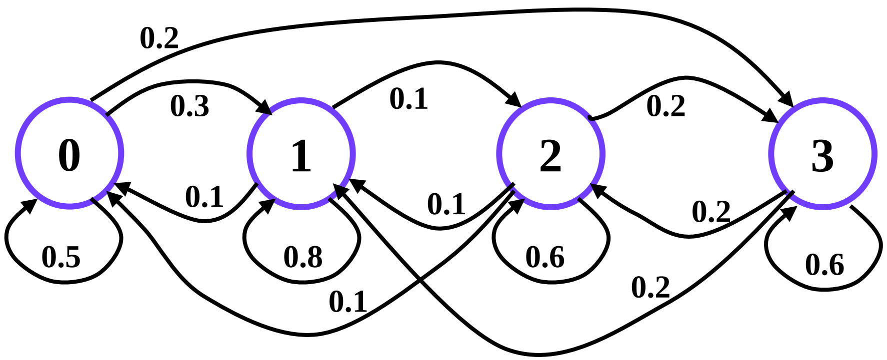
$$p_j \sum_{i=0, i\neq j}^{\infty} P_{ji} = \sum_{i=0, i\neq j}^{\infty} p_i P_{ij}$$

$$\begin{aligned} -0.5p_0 + 0.1p_1 + 0.1p_2 \quad\quad\quad &= 0 \\ 0.3p_0 - 0.2p_1 + 0.1p_2 + 0.2p_3 &= 0 \\ 0.1p_1 - 0.4p_2 + 0.2p_3 &= 0 \\ 0.2p_0 \quad\quad + 0.2p_2 - 0.4p_3 &= 0 \end{aligned}$$

Homogeneous Linear Eq. *ยังแก้สมการไม่ได้*

Example

– 2. จงคำนวณหา State Probability



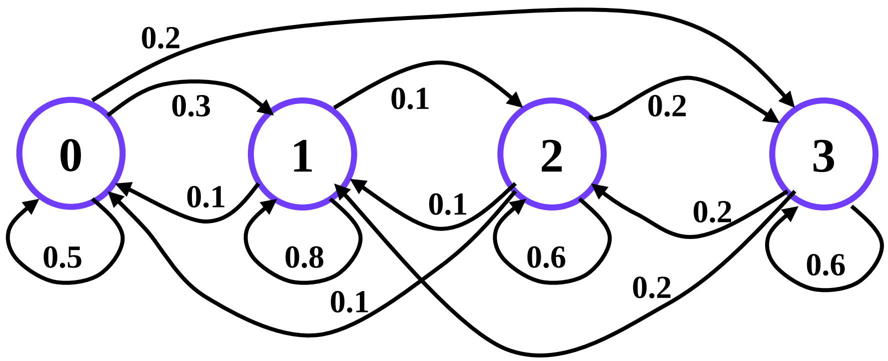
$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0 & 0.2 & 0.2 & 0.6 \end{bmatrix}$$

$$\begin{aligned} -0.5p_0 + 0.1p_1 + 0.1p_2 \qquad\qquad &= 0 \\ 0.3p_0 - 0.2p_1 + 0.1p_2 + 0.2p_3 &= 0 \\ 0.1p_1 - 0.4p_2 + 0.2p_3 &= 0 \\ p_0 \quad + p_1 \quad + p_2 \quad + p_3 &= 1.0 \end{aligned}$$

ต้องนำสมการที่ 4 มาใส่เสมอ + อีก 3 สมการอะไรก็ได้
สมการ Non Homogeneous System of Linear Equation
แก้ได้โดยวิธี Elimination จะได้ Unique Solution

Example

– 2. จงคำนวณหา State Probability



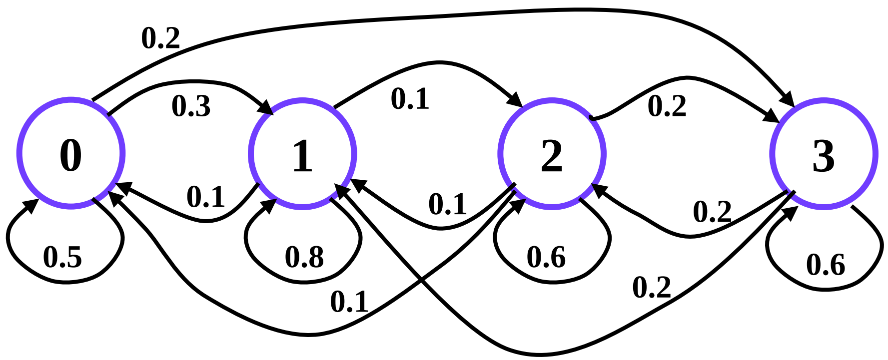
$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0 & 0.2 & 0.2 & 0.6 \end{bmatrix}$$

$$\begin{aligned} -0.5p_0 + 0.1p_1 + 0.1p_2 \qquad &= 0 \\ 0.3p_0 - 0.2p_1 + 0.1p_2 + 0.2p_3 &= 0 \\ 0.1p_1 - 0.4p_2 + 0.2p_3 &= 0 \\ p_0 \quad + p_1 \quad + p_2 \quad + p_3 &= 1.0 \end{aligned} \Rightarrow \begin{bmatrix} -0.5 & 0.1 & 0.1 & 0 \\ 0.3 & -0.2 & 0.1 & 0.2 \\ 0 & 0.1 & -0.4 & 0.2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Non Homogeneous System of Linear Equation
แก้ได้โดยวิธี Elimination จะได้ Unique Solution
Algorithm ในการแก้สมการ จะกล่าวภายหลัง

Example

– 2. จงคำนวณหา State Probability



$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0 & 0.2 & 0.2 & 0.6 \end{bmatrix}$$

$$-0.5p_0 + 0.1p_1 + 0.1p_2 = 0 \quad (1)$$
$$0.3p_0 - 0.2p_1 + 0.1p_2 + 0.2p_3 = 0 \quad (2)$$
$$0.1p_1 - 0.4p_2 + 0.2p_3 = 0 \quad (3)$$
$$p_0 + p_1 + p_2 + p_3 = 1.0 \quad (4)$$

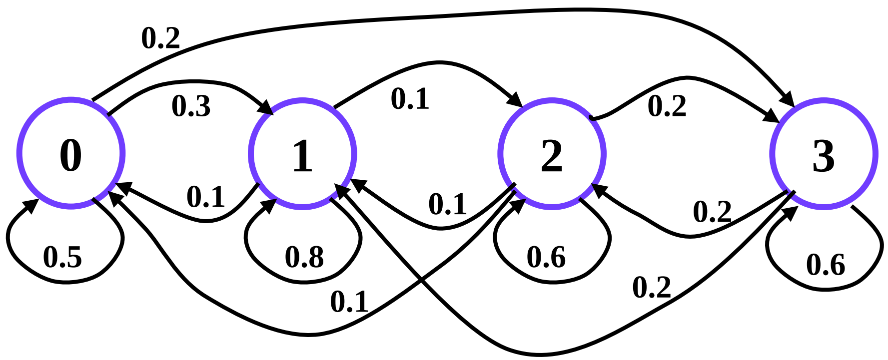
$$(4) - 5\times(2): \quad -0.5p_0 + 2p_1 + 0.5p_2 = 1 \quad (5)$$
$$(4) - 5\times(3): \quad p_0 + 0.5p_1 + 3p_2 = 1 \quad (6)$$

จากสมการที่ (1),(5),(6) เราได้

$$-0.5p_0 + 0.1p_1 + 0.1p_2 = 0 \quad (1)$$
$$-0.5p_0 + 2p_1 + 0.5p_2 = 1 \quad (5)$$
$$p_0 + 0.5p_1 + 3p_2 = 1 \quad (6)$$

Example

– 2. จงคำนวณหา State Probability



$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0 & 0.2 & 0.2 & 0.6 \end{bmatrix}$$

$$-0.5p_0 + 0.1p_1 + 0.1p_2 = 0 \quad (1)$$

$$-0.5p_0 + 2p_1 \quad + 0.5p_2 = 1 \quad (5)$$

$$p_0 + 0.5p_1 \ + \ 3p_2 = 1 \quad (6)$$

$$(6) + 2\times(1): \quad 0.7p_1 + 3.2p_2 = 1 \quad (7)$$

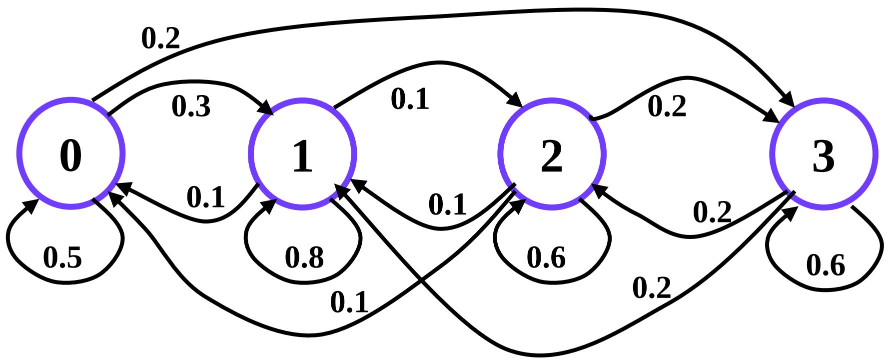
$$(6) + 2\times(5): \quad 4.5p_1 + \ 4p_2 = 3 \quad (8)$$

$$(8) - \frac{4.5}{0.7}(7): \quad (4 - \frac{4.5}{0.7}3.2)p_2 = 3 - \frac{4.5}{0.7}$$

$$p_2 = \frac{-3.4285714}{-16.57142857} = 0.2069$$

Example

– 2. จงคำนวณหา State Probability



$$\mathbf{P}=\begin{bmatrix}0.5 & 0.3 & 0 & 0.2\\ 0.1 & 0.8 & 0.1 & 0\\ 0.1 & 0.1 & 0.6 & 0.2\\ 0 & 0.2 & 0.2 & 0.6\end{bmatrix}$$

$$p_2 = 0.2069$$

$$\text{From}\,(7):\, p_1 = \frac{1-3.2p_2}{0.7} = 0.4828$$

$$From\,(1):\, p_0 = 0.1379$$

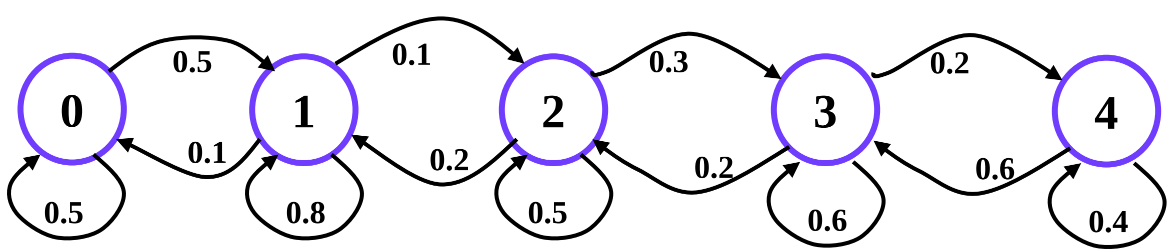
$$From\,(4):\, p_3 = 0.1724$$

สมการ System of Linear Equation แก้ได้โดยวิธี Elimination

$$p_0 = 0.1379,\ p_1 = 0.4828,\ p_2 = 0.2069,\ p_3 = 0.1724$$

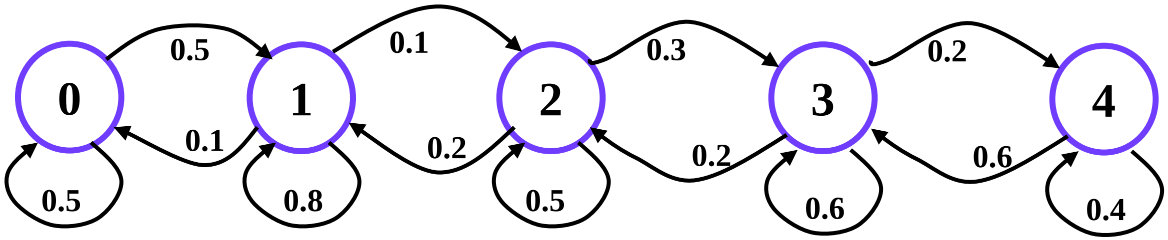
Example: Simple MarKov

- MarKov Chain แสดงด้วย Markov Model ดังรูปข้างล่าง
  - 1. จงหา Transition Matrix
  - 2. จาก Transition Matrix จงคำนวณหา State Probability



Example: Simple MarKov

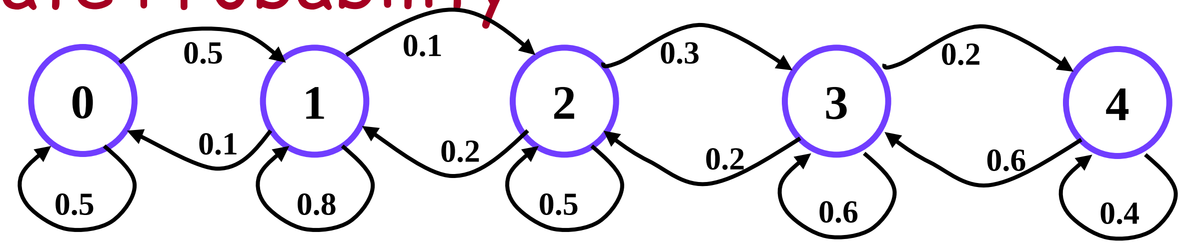
- Transition Matrix



$$\mathbf{P} == \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} & P_{2,4} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} & P_{3,4} \\ P_{4,0} & P_{4,1} & P_{4,2} & P_{4,3} & P_{4,4} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0.1 & 0.8 & 0.1 & 0 & 0 \\ 0 & 0.2 & 0.5 & 0.3 & 0 \\ 0 & 0 & 0.2 & 0.6 & 0.2 \\ 0 & 0 & 0 & 0.6 & 0.4 \end{bmatrix}$$

Example: Simple MarKov

- State Probability



$$\mathbf{P} == \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} & P_{2,4} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} & P_{3,4} \\ P_{4,0} & P_{4,1} & P_{4,2} & P_{4,3} & P_{4,4} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0.1 & 0.8 & 0.1 & 0 & 0 \\ 0 & 0.2 & 0.5 & 0.3 & 0 \\ 0 & 0 & 0.2 & 0.6 & 0.2 \\ 0 & 0 & 0 & 0.6 & 0.4 \end{bmatrix}$$

$$\text{Detailed Balance Equation} : p_i P_{i,j} = p_j P_{j,i}$$

$$\begin{array}{l} p_0 P_{0,1} = p_1 P_{1,0} \\ p_1 P_{1,2} = p_2 P_{2,1} \\ p_2 P_{2,3} = p_3 P_{3,2} \\ p_3 P_{3,4} = p_4 P_{4,3} \end{array} \Rightarrow \begin{array}{l} 0.5 p_0 = 0.1 p_1 \text{ or } p_1 = 5 p_0 \\ 0.1 p_1 = 0.2 p_2 \text{ or } p_2 = 0.5 p_1 = 2.5 p_0 \\ 0.3 p_2 = 0.2 p_3 \text{ or } p_3 = 1.5 p_2 = 3.75 p_0 \\ 0.2 p_3 = 0.6 p_4 \text{ or } p_4 = \tfrac{1}{3} p_3 = 1.25 p_0 \end{array}$$

Example: Simple MarKov

- State Probability

Detailed Balance Equation : $p_i P_{i,j} = p_j P_{j,i}$

$$\begin{aligned}
p_0 P_{0,1} &= p_1 P_{1,0} & & 0.5 p_0 = 0.1 p_1 \text{ or } p_1 = 5 p_0 \\
p_1 P_{1,2} &= p_2 P_{2,1} & & 0.1 p_1 = 0.2 p_2 \text{ or } p_2 = 0.5 p_1 = 2.5 p_0 \\
p_2 P_{2,3} &= p_3 P_{3,2} & \Rightarrow & 0.3 p_2 = 0.2 p_3 \text{ or } p_3 = 1.5 p_2 = 3.75 p_0 \\
p_3 P_{3,4} &= p_4 P_{4,3} & & 0.2 p_3 = 0.6 p_4 \text{ or } p_4 = \tfrac{1}{3} p_3 = 1.25 p_0
\end{aligned}$$

$$\textit{From} \quad p_0 + p_1 + p_2 + p_3 + p_4 = 1$$

$$p_0 + 5 p_0 + 2.5 p_0 + 3.75 p_0 + 1.25 p_0 = 1$$

$$p_0 [1 + 5 + 2.5 + 3.75 + 1.25] = 1$$

$$p_0 = 0.074074 \quad \textit{then}$$

$$p_1 = 0.37037,\ p_2 = 0.185185,\ p_3 = 0.277778,\ p_4 = 0.092593$$

Example: Simple MarKov

- State Probability

Detailed Balance Equation : $p_i P_{i,j} = p_j P_{j,i}$

$$\begin{array}{ll} p_0 P_{0,1} = p_1 P_{1,0} & 0.5p_0 = 0.1p_1 \text{ or } p_1 = 5p_0 \\ p_1 P_{1,2} = p_2 P_{2,1} & 0.1p_1 = 0.2p_2 \text{ or } p_2 = 0.5p_1 = 2.5p_0 \\ p_2 P_{2,3} = p_3 P_{3,2} \;\Rightarrow & 0.3p_2 = 0.2p_3 \text{ or } p_3 = 1.5p_2 = 3.75p_0 \\ p_3 P_{3,4} = p_4 P_{4,3} & 0.2p_3 = 0.6p_4 \text{ or } p_4 = \tfrac{1}{3}p_3 = 1.25p_0 \end{array}$$

From $p_0 + p_1 + p_2 + p_3 + p_4 = 1$ **ใช้วิธีของ Matrix**

$$\begin{array}{l} 0.5p_0 - 0.1p_1 = 0 \\ 0.1p_1 - 0.2p_2 = 0 \\ 0.3p_2 - 0.2p_3 = 0 \\ 0.2p_3 - 0.6p_4 = 0 \\ p_0 + p_1 + p_2 + p_3 + p_4 = 1 \end{array} \Rightarrow \begin{bmatrix} 0.5 & -0.1 & 0 & 0 & 0 \\ 0 & 0.1 & -0.2 & 0 & 0 \\ 0 & 0 & 0.3 & -0.2 & 0 \\ 0 & 0 & 0 & 0.2 & -0.6 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

End of Chapter 5

- Chapter 6
  - Introduction to Queuing Theory
- Homework Chapter 5 Download
  - เน้นที่ Correlation ของ Stationary RP และการใช้ Global/Detailed Balance Equation ใน MarKov Chain

CPE 332 Computer Engineering Mathematics II

Week 6

Part II, Chapter 6

Queuing System

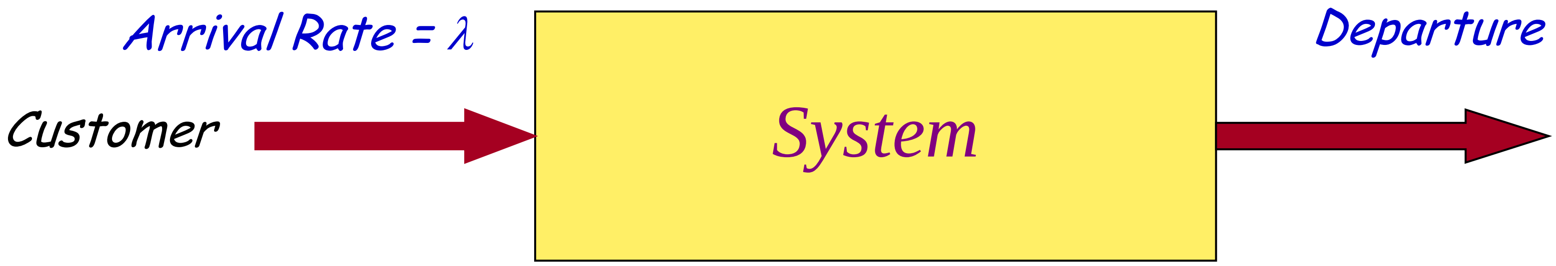
Topics

- Birth and Death Process
- Unlimited Server
- N Servers
- Single Server, M/M/1
- Kendal Notation
- Applications

System

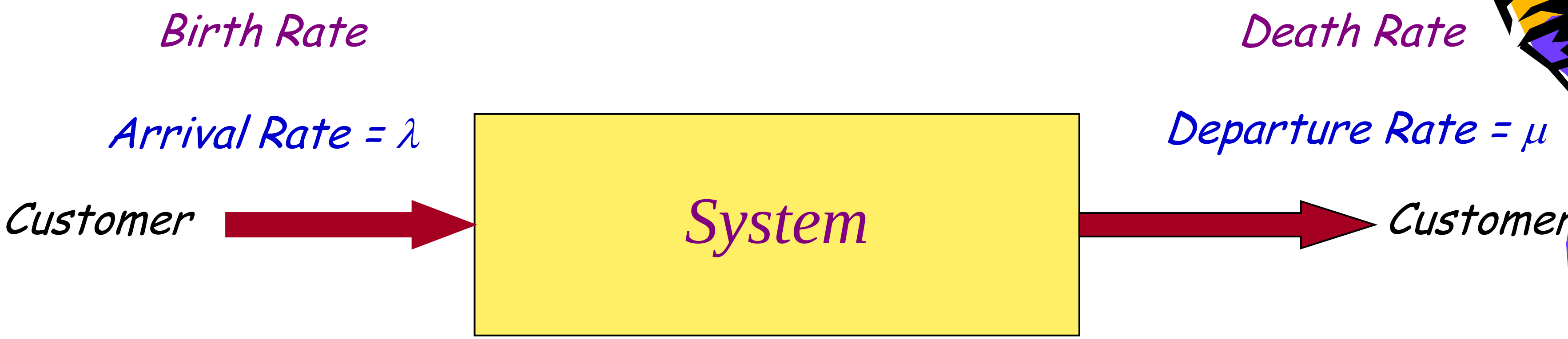
- ระบบประกอบด้วย Input และ Output
- พิจารณาระบบที่มีการให้บริการ(Service) แก่ลูกค้า (Customer)
  - ลูกค้าเข้ามาในระบบเพื่อขอรับบริการ (Input)
  - ระบบมี Resource ที่จำกัดในการให้บริการ
  - ลูกค้า เมื่อได้รับบริการแล้ว ออกไปจากระบบ (Output)
- ระบบขายของหน้าร้าน, ระบบหน้าธนาคาร, ระบบการจราจร, ระบบ Operating System ในคอมพิวเตอร์, สถานีน้ำมัน/แก๊ส, คิวจ่ายของ/อาหาร, ระบบสื่อสารข้อมูล, ระบบโทรศัพท์ และอื่นๆอีกมาก

Queuing System



1.อัตราการเข้ามาของลูกค้า คือ Arrival Rate
2.อัตราการออกไปของลูกค้าเมื่อได้รับบริการเสร็จคือ Departure Rate
3. State ของระบบคือจำนวนลูกค้าที่อยู่ในระบบ ที่รอบริการ หรือกำลังถูกบริการ
4. ถ้าระบบไม่มีการจดจำ การบริการลูกค้าแต่ละรายเหมือนกัน และไม่ขึ้นกับอดีต เราสามารถใช้รูปแบบ MarKov Chain อธิบายระบบ

Queuing System



5. ถ้าเราแบ่งช่วงเวลาการเข้ามาของลูกค้าเป็นช่วง Time Slot
และบันทึกเหตุการณ์ที่เกิดขึ้นในแต่ละ Time Slot
a) มีลูกค้าใหม่เข้ามา เพื่อขอใช้บริการ (State ของระบบจะเพิ่ม)
หรือ b) มีลูกค้าที่ได้รับบริการแล้วออกจากระบบไป (State จะลด)
หรือ c) ไม่มีลูกค้าใหม่ และไม่มีลูกค้าที่ให้บริการเสร็จ (State คงเดิม)

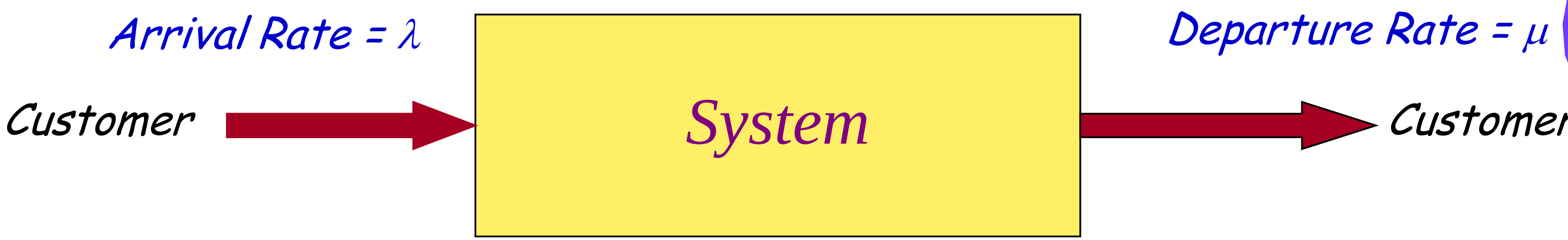
6. จาก Model ข้อ 5 เราจะได้ Discrete Time Markov Chain

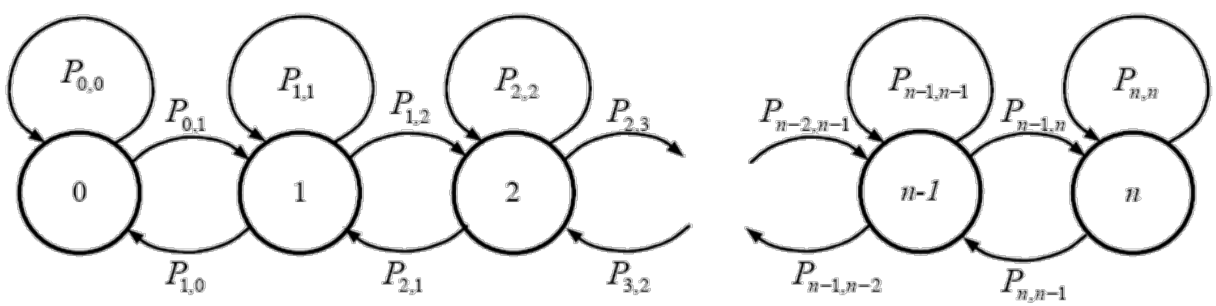
7. ถ้าช่วงเวลาของ Time Slot สั้นมากจนกระทั่งลูกค้าที่เข้ามา หรือออกไป
ในช่วงหนึ่ง Time Slot สามารถมีได้แค่คนเดียว
เราจะสามารถ Model ระบบได้เป็น Simple MarKov Model

Queuing System

ถ้า λ < μ ระบบจะสามารถ
เข้าสู่ Equilibrium ได้

Birth Rate $\rho = \frac{\lambda}{\mu}$; Server Utilization Death Rate

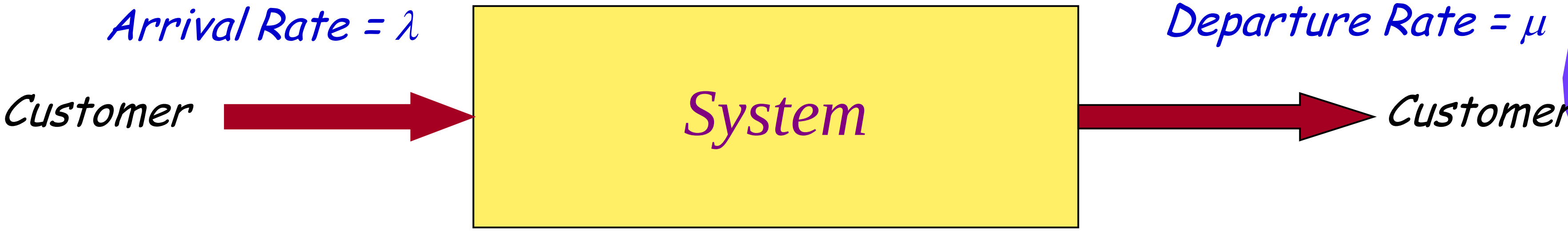


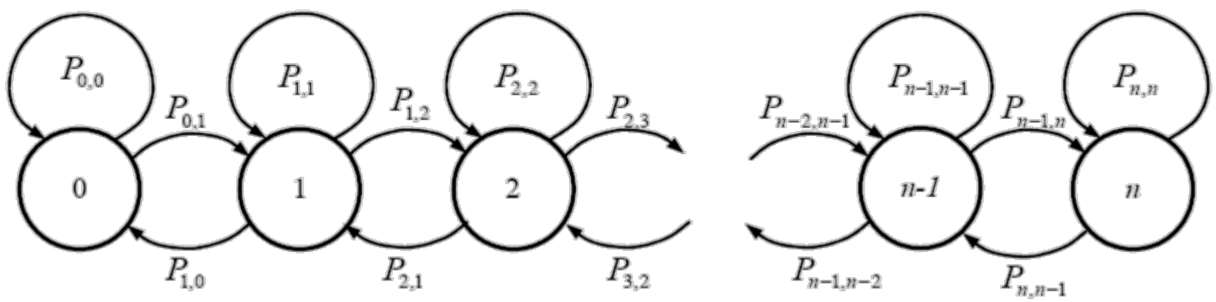


Queuing System
Simple MarKov Model

Birth Rate $\lambda < \mu$ Death Rate

Arrival Rate = λ Departure Rate = μ





Detailed Balance Equation: $p_i P_{ij} = p_j P_{ji}$

สำหรับสอง State i, j ที่อยู่ติดกันใดๆ

Queuing System
Simple MarKov Model

$$p_i P_{ij} = p_j P_{ji}$$

$$\lambda < \mu$$

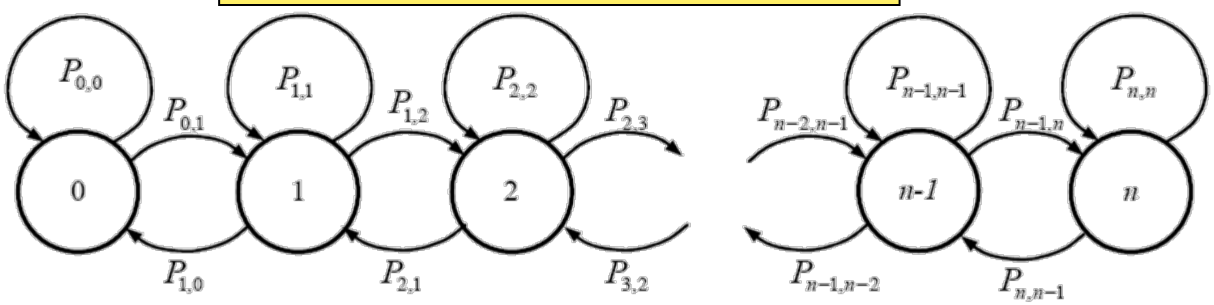
Birth Rate

Death Rate

Arrival Rate = λ

Departure Rate = μ

Customer → **System** → Customer



ในที่นี้เราจะพิจารณากรณี

1. ที่ลูกค้าแต่ละรายเข้ามาในระบบแบบ Random และเป็น Poisson Process
2. เวลาในการให้บริการของลูกค้าแต่ละราย เป็น Random มีการกระจายแบบ Exponential

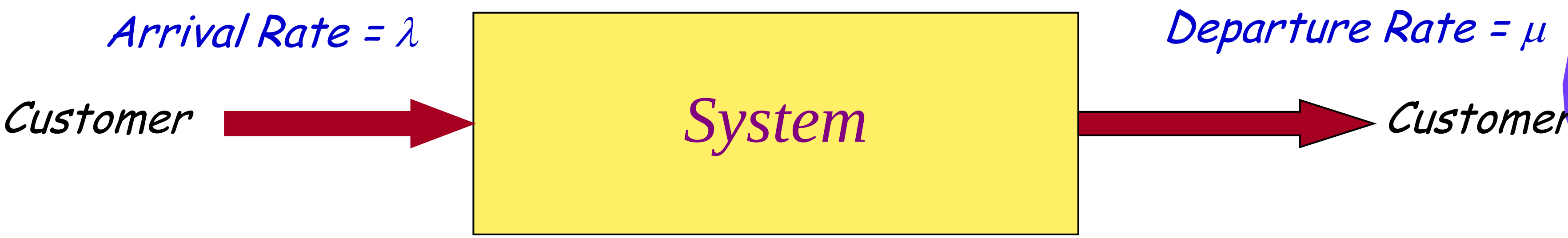
Queuing System
Simple MarKov Model

$$p_i P_{ij} = p_j P_{ji}$$

$$\lambda < \mu$$

Birth Rate

Death Rate



Arrival: 1. Probability ที่จะมีลูกค้า k คนเข้ามาในช่วง T วินาที

$$P[k] = P[X = k] = \frac{(\lambda T)^k e^{\lambda T}}{k!}; k = 0, 1, 2, \ldots$$

เมื่อ λ เป็นค่าเฉลี่ยจำนวนลูกค้าที่เข้ามา ต่อวินาที

2. ค่า Inter-arrival Time, τ จะมีการกระจายแบบ Exponential ด้วยค่าเฉลี่ย ${}^{1}/_{\lambda}$

$$P[T \leq \tau] = 1 - \lambda e^{-\lambda \tau}$$

Queuing System
Simple MarKov Model

$p_i P_{ij} = p_j P_{ji}$

$\lambda < \mu$

Birth Rate

Death Rate

Arrival Rate = λ

Departure Rate = μ

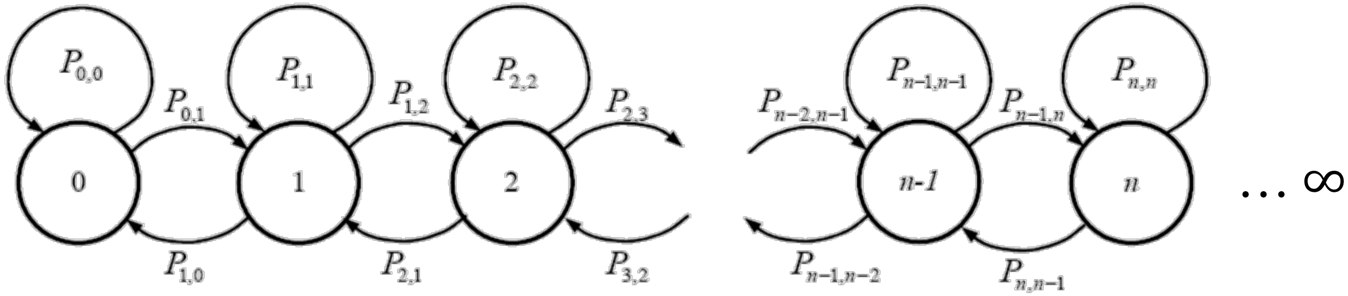
Customer → System → Customer

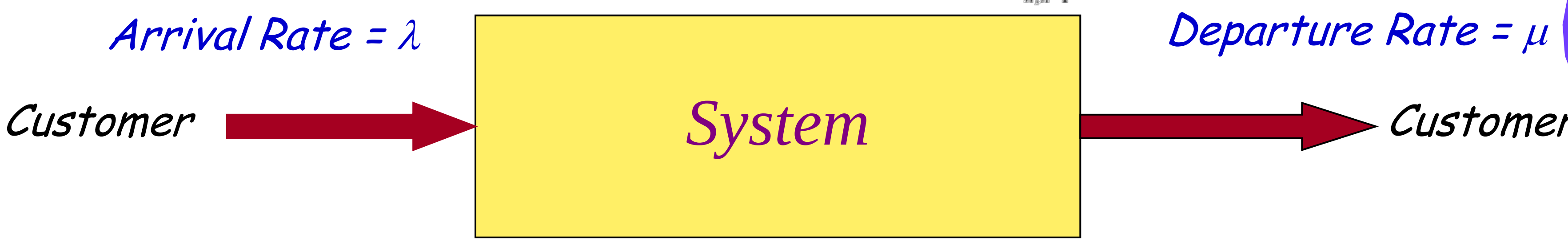
Departure: 1. เวลาเฉลี่ยที่ลูกค้าใช้บริการ $= T_s$(Service Time)
จะมีการกระจายแบบ Exponential
Probability ที่ลูกค้าจะใช้เวลาบริการ น้อยกว่าหรือเท่ากับ t $P[T \le t] = 1 - \frac{1}{T_s} e^{-t/T_s}$

2. Departure Rate คืออัตราที่ลูกค้าออกจากระบบ
เมื่อได้รับบริการเสร็จ(อัตราการให้บริการแก่ลูกค้า)

$$\mu = \frac{1}{T_s}$$

Queuing System Case 1: Unlimited Server; No Queue





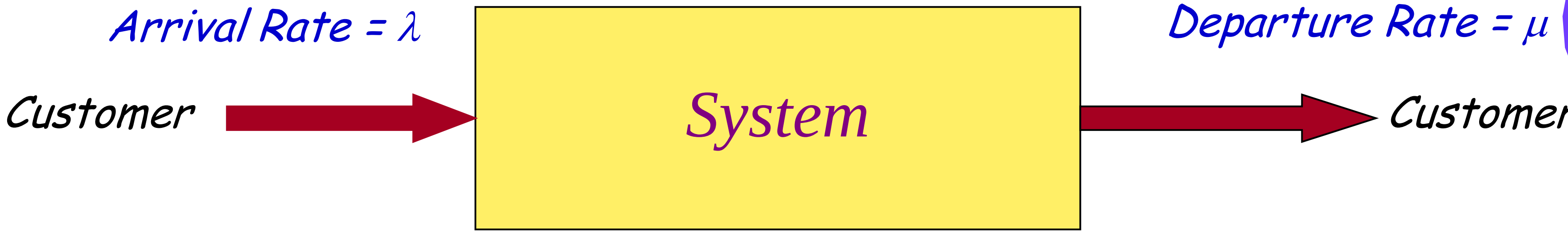
1. ลูกค้าที่เข้ามา เป็น Random ด้วยอัตราเฉลี่ย λ และมีการกระจายแบบ Poisson
2. สมมุติว่า Customer แต่ละคนที่เข้ามาได้รับการ Service จากระบบทันที (ระบบมี Server จำนวนไม่จำกัด และรับ Customer ได้ไม่จำกัด)
3. เวลาที่ใช้ในการ Service เป็น เป็น Exponential ด้วยเวลาเฉลี่ย T_S
4. ระบบสามารถรับ Customer ได้ไม่จำกัด
5. ลูกค้าเข้ามาได้ทีละคน และออกทีละคน
6. ระบบนี้เรียก M/M/∞

Queuing System Case 1:
Unlimited Server; No Queue

$$\lambda < \mu$$

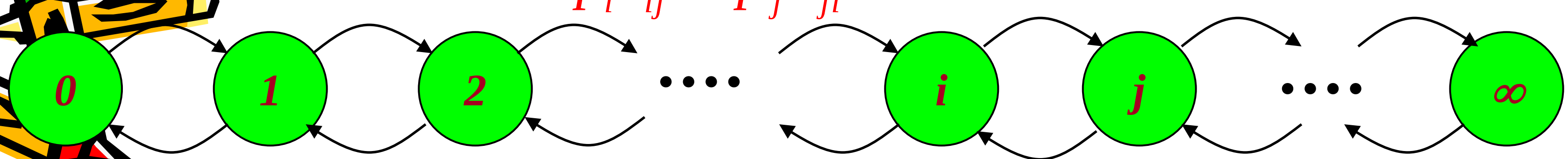
$$P[k] = \frac{\lambda^k e^{-\lambda}}{k!} \qquad p_x = \frac{\rho^x e^{-\rho}}{x!};\ \rho = \lambda / \mu \qquad P[T \geq t] = e^{-t/T_s};\ T_s = 1/\mu$$

Arrival Rate = λ *Departure Rate = μ*



1. *สมมุติว่า Customer แต่ละคนที่เข้ามาเป็น Poisson และได้รับการ Service จากระบบทันที*
2. *เวลาที่ใช้ในการ Service เป็น Random สมมุติว่าเป็น Exponential ด้วยเวลาเฉลี่ย T*
3. *ระบบสามารถรับ Customer ได้ไม่จำกัด แต่เข้ามาได้ทีละคน และออกทีละคน*
4. *ระบบนี้เรียก M/M/∞ แสดงได้ด้วย Simple Markov Model*
5. *เราสามารถพิสูจน์ได้ว่าค่า State Probability จะมีการกระจายแบบ Poisson*

$$p_i P_{ij} = p_j P_{ji}$$



Queuing System Case 2: Lost System
Limited Server=N; No Queue

$$\lambda < \mu$$

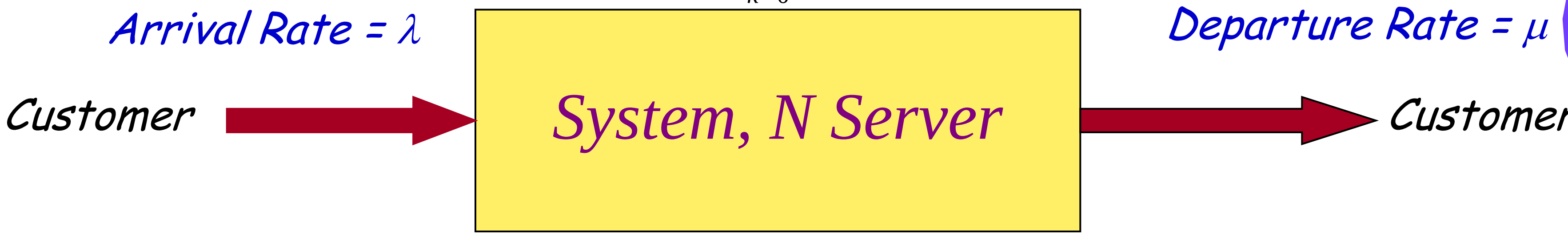
$$P[k] = \frac{\lambda^k e^{-\lambda}}{k!}$$

$$p_x = \frac{\rho^x / x!}{\sum_{k=0}^{N} \frac{\rho^k}{k!}}$$

$$P[T \geq t] = e^{-t/T_s};\ T_s = {}^{1}\!/_{\mu}$$

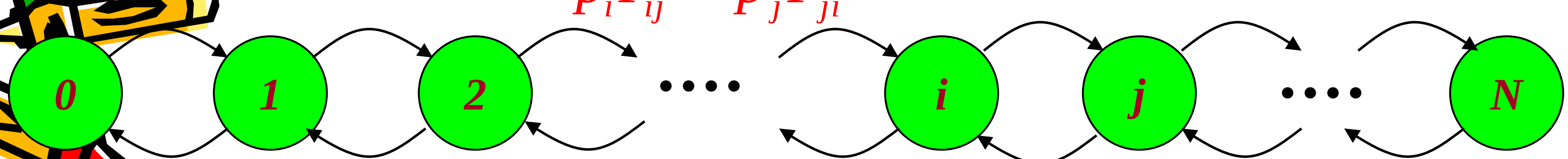
Arrival Rate = λ

Departure Rate = μ



1. *สมมุติว่า Customer แต่ละคนที่เข้ามาเป็น Poisson และได้รับการ Service จากระบบทันที*
2. *เวลาที่ใช้ในการ Service เป็น Random สมมุติว่าเป็น Exponential ด้วยเวลาเฉลี่ย T*
3. *ระบบสามารถรับ Customer ได้ N ถ้าทุก Server เต็ม จะรับ Customer ใหม่ไม่ได้ (Lost)*
4. *ระบบนี้เรียก M/M/N/N แสดงได้ด้วย N-state Simple Markov Model*
5. *State Probability จะมีการกระจายแบบ First Erlang (Erlang B) Distribution*

$$p_i P_{ij} = p_j P_{ji}$$



Queuing System Case 3: Delay System

Limited Server=N; With Unlimited Queue

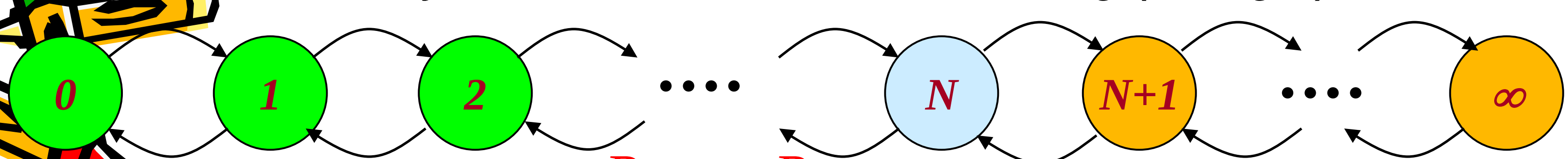
$$\lambda < \mu$$

$$P[k] = \frac{\lambda^k e^{-\lambda}}{k!} \qquad p_x = \begin{cases} \dfrac{\rho^x}{x!} p_0 ; 0 \le x \le N \\ \dfrac{N^N}{N!}\left(\dfrac{\rho}{N}\right)^x p_0 ; x \ge N \end{cases} ; p_0 = \left[\frac{N\rho^N}{N!(N-\rho)} + \sum_{k=0}^{N-1} \frac{\rho^k}{k!} \right]^{-1} \qquad P[T \ge t] = e^{-t/T_s} ; T_s = 1/\mu$$

Arrival Rate = λ *Departure Rate = μ*

Customer → **System, N Server** → *Customer*

1. สมมุติว่า Customer แต่ละคนที่เข้ามาเป็น Poisson และได้รับการ Service จากระบบทันที
2. เวลาที่ใช้ในการ Service เป็น Random สมมุติว่าเป็น Exponential ด้วยเวลาเฉลี่ย T
3. ระบบสามารถรับ Customer ได้ไม่จำกัด แต่จะ Service ได้สูงสุด N พร้อมๆกัน
4. ถ้าทุก Server เต็ม Customer ใหม่จะต้องรอใน Queue ในกรณีนี้จะเกิด Queuing Delay
5. ระบบนี้เรียก M/M/N หรือ M/M/N/∞ แสดงได้ด้วย Simple Markov Model
6. State Probability จะมีการกระจายแบบ Second Erlang (Erlang C) Distribution



$$p_i P_{ij} = p_j P_{ji}$$

Queuing System Case 3: Delay System
Server=1; With Unlimited Queue; M/M/1

$$\lambda < \mu$$

$$P[k] = \frac{\lambda^k e^{-\lambda}}{k!}$$

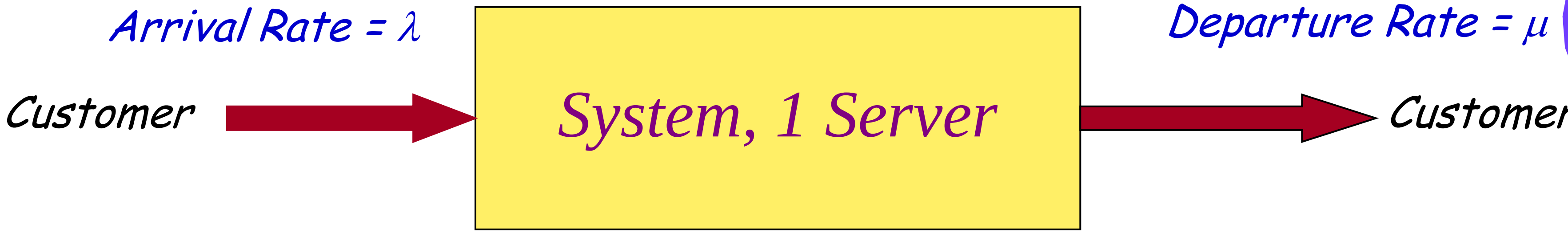
$$p_0 = \left[\frac{\rho}{(1-\rho)} + 1\right]^{-1} = 1-\rho$$

$$p_x = \rho^x (1-\rho)$$

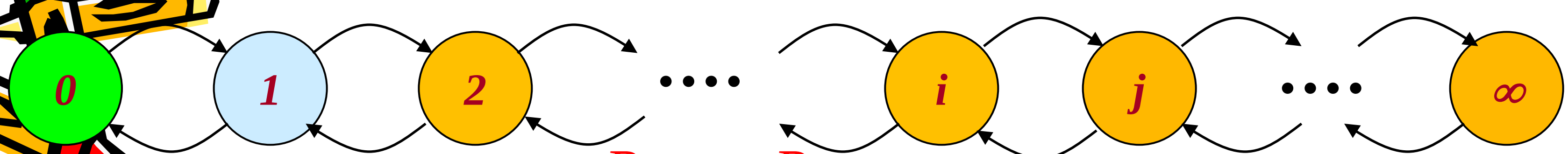
$$P[T \geq t] = e^{-t/T_s};\ T_s = 1/\mu$$

Arrival Rate = λ

Departure Rate = μ

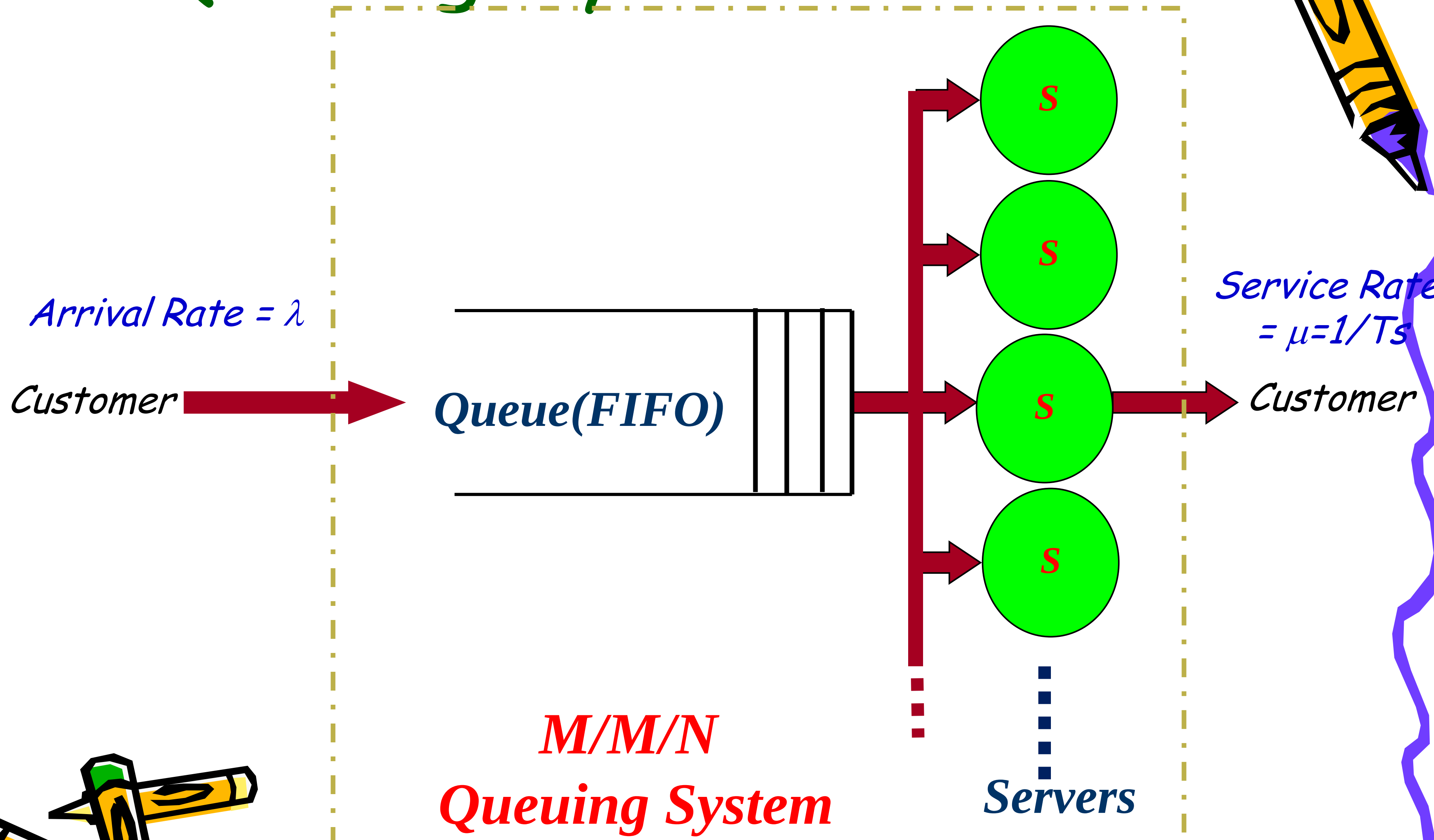


1. สมมุติว่า Customer แต่ละคนที่เข้ามาเป็น Poisson และได้รับการ Service จากระบบทันที
2. เวลาที่ใช้ในการ Service เป็น Random สมมุติว่าเป็น Exponential ด้วยเวลาเฉลี่ย T
3. ระบบสามารถรับ Customer ได้ไม่จำกัด แต่จะ Service ได้ครั้งละคน
4. ถ้าทุก Server เต็ม Customer ใหม่จะต้องรอใน Queue ในกรณีนี้จะเกิด Queuing Delay
5. ระบบนี้เรียก M/M/1 หรือ M/M/1/∞ แสดงได้ด้วย Simple Markov Model
6. State Probability จะมีการกระจายแบบ Second Erlang (Erlang C) Distribution

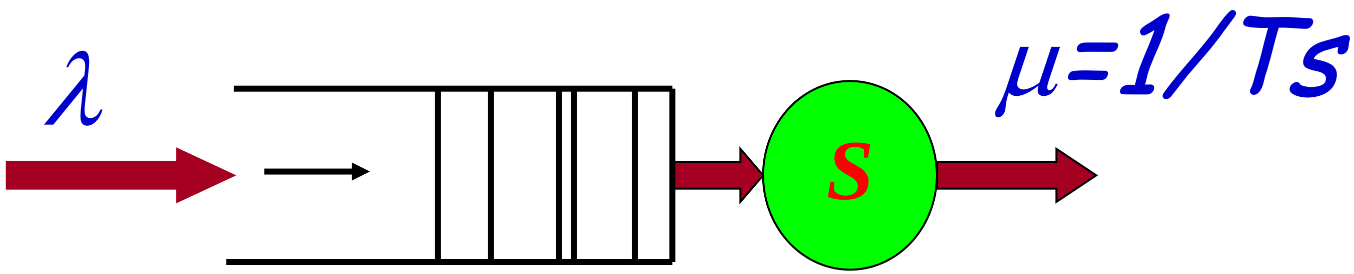


$$p_i P_{ij} = p_j P_{ji}$$

Queuing System: Model

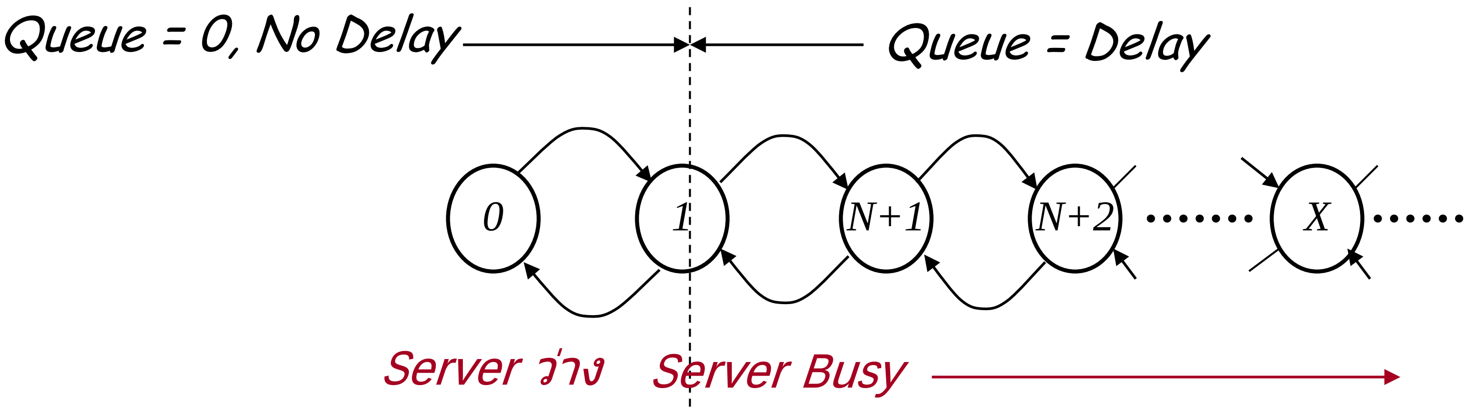


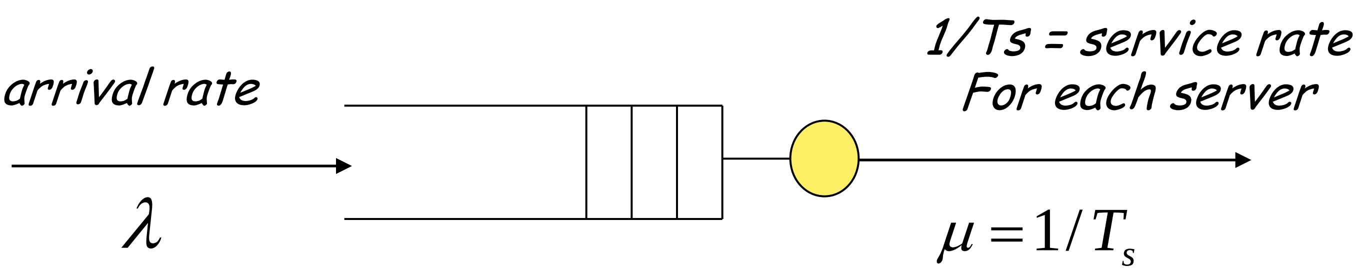
M/M/1: Summary



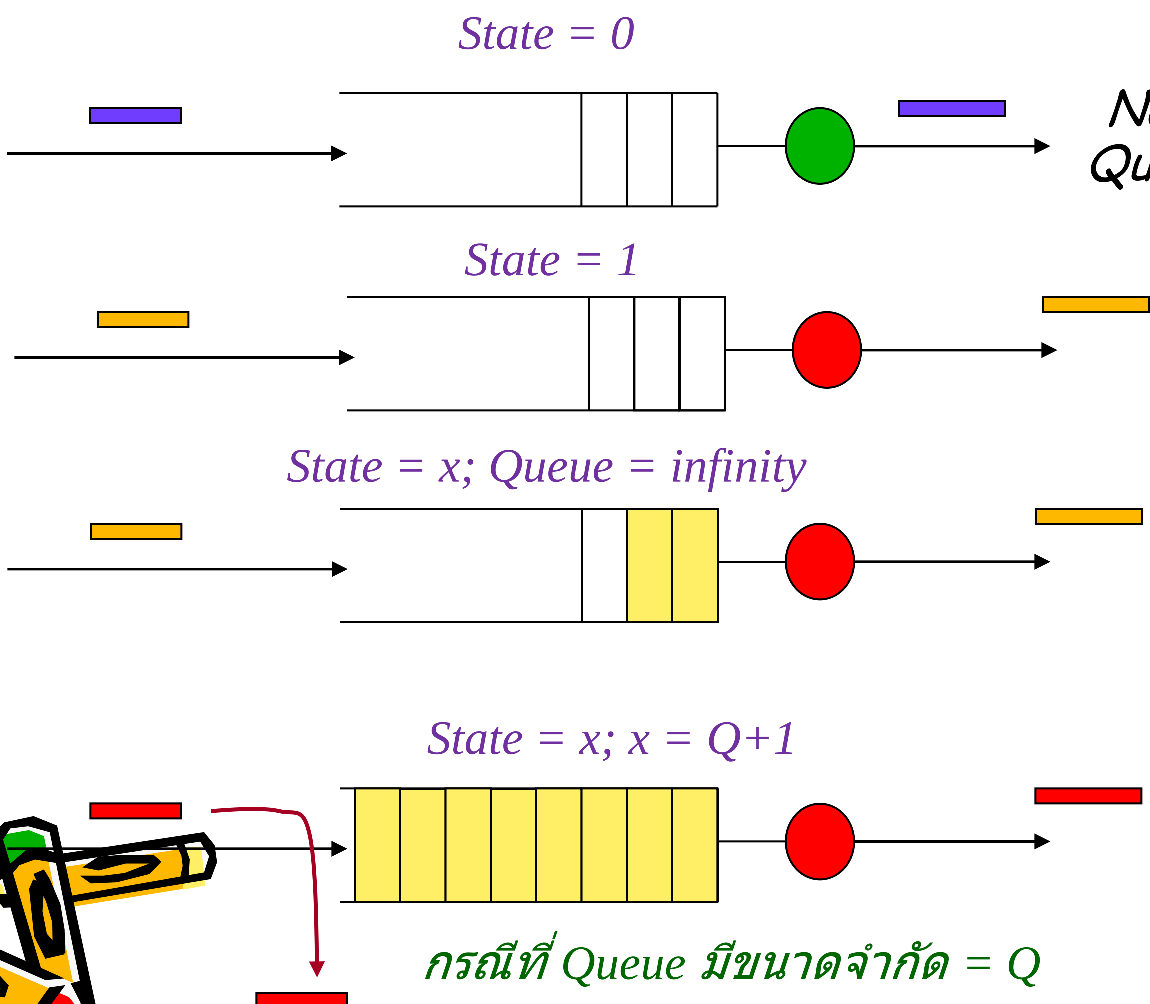
Arrival = Poisson, λ
Inter Arrival Time = Exponential, $1/\lambda$
Service Rate, μ
Service Time, Ts ($1/\mu$) = Exponential
Queue = FIFO
1 Server

Queuing Model(1 Server); M/M/1

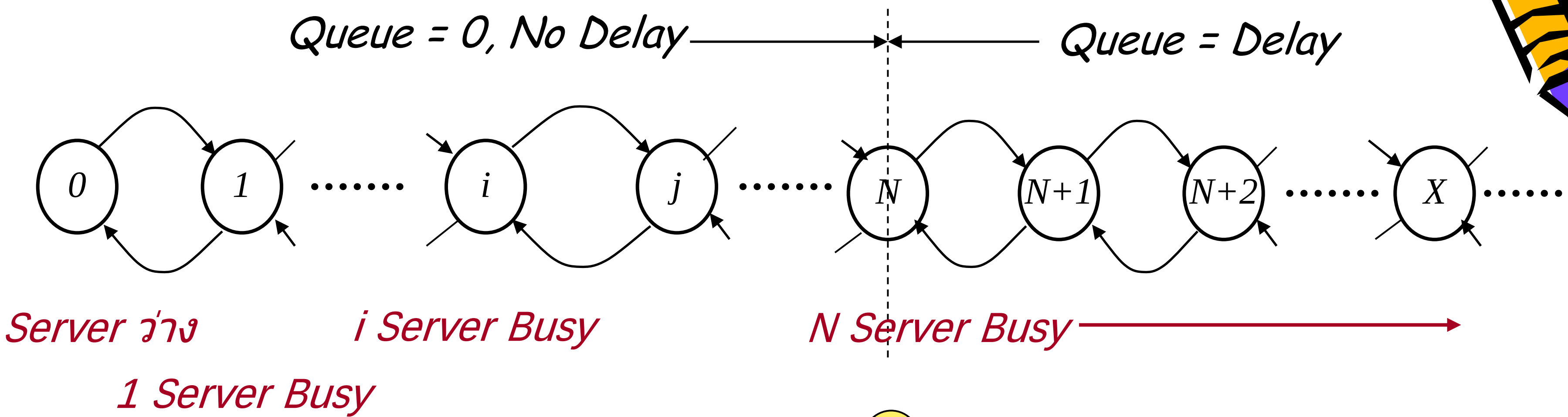




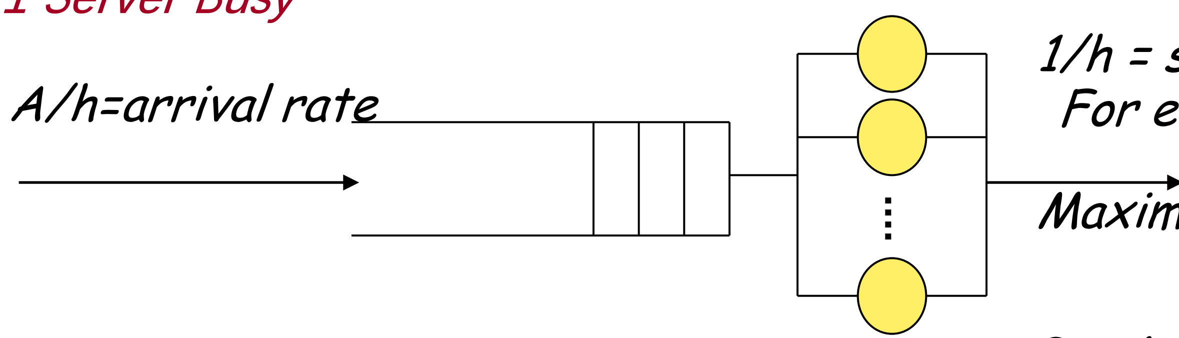
การทำงานของ M/M/1



เปรียบเทียบ Queuing Model (N Server); M/M/N



A/h=arrival rate

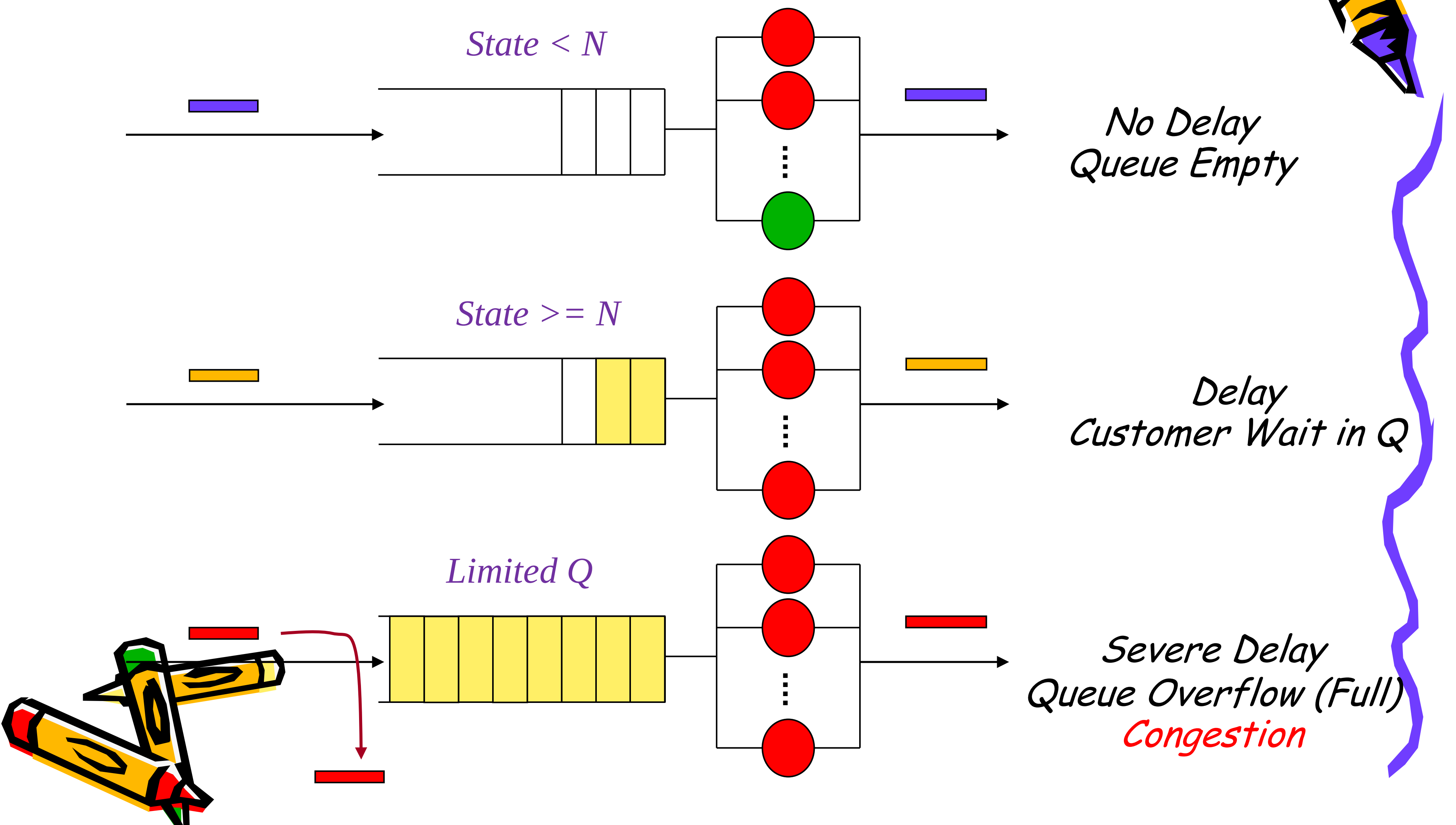


1/h = service rate
For each server

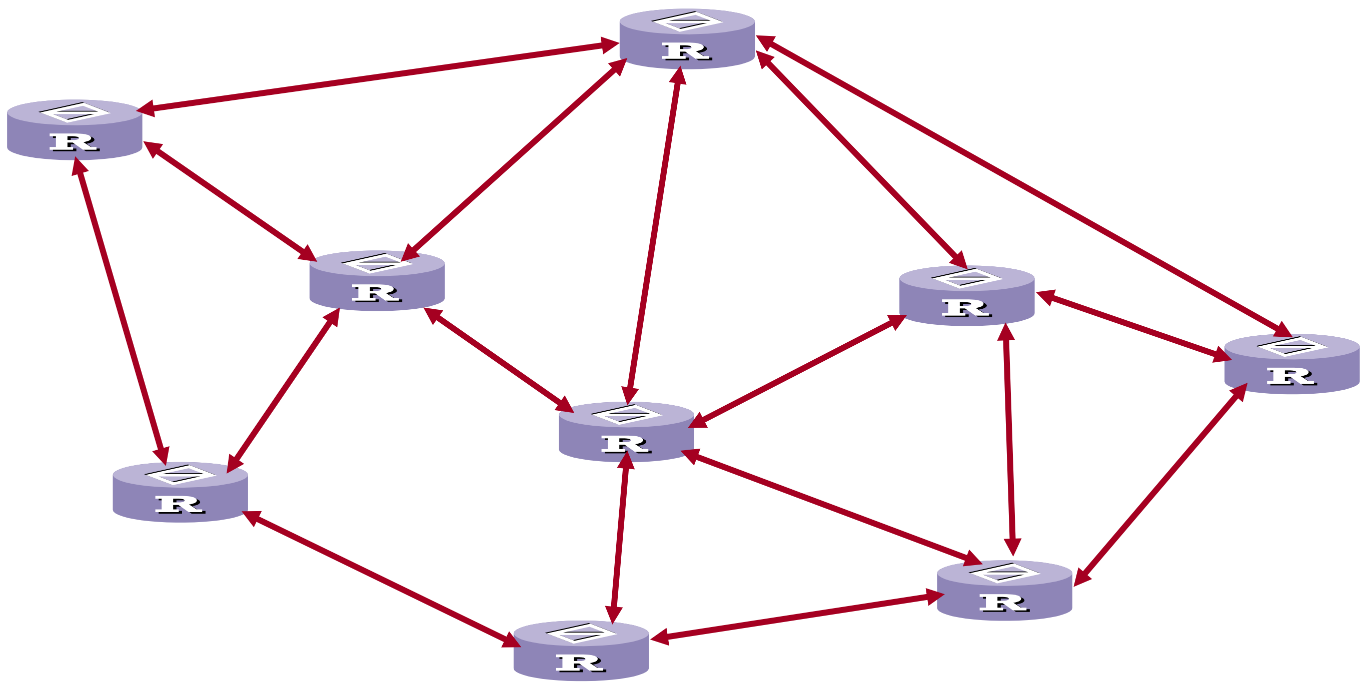
Maximum Service Rate
= N/h

Service Rate at State
k = k/h

M/M/N

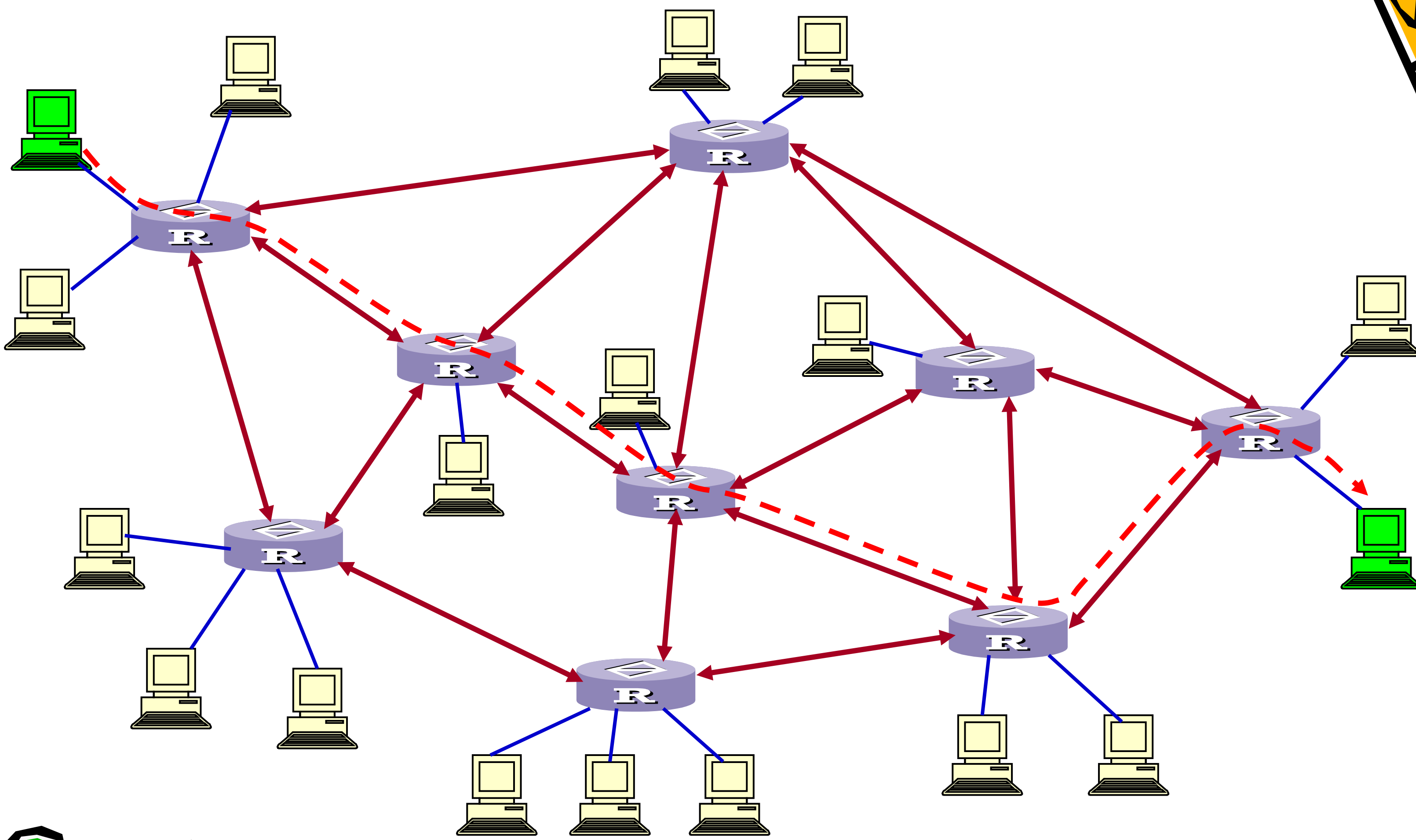

State < N
No Delay
Queue Empty
State >= N
Delay
Customer Wait in Q
Limited Q
Severe Delay
Queue Overflow (Full)
Congestion

Network Model using M/M/1



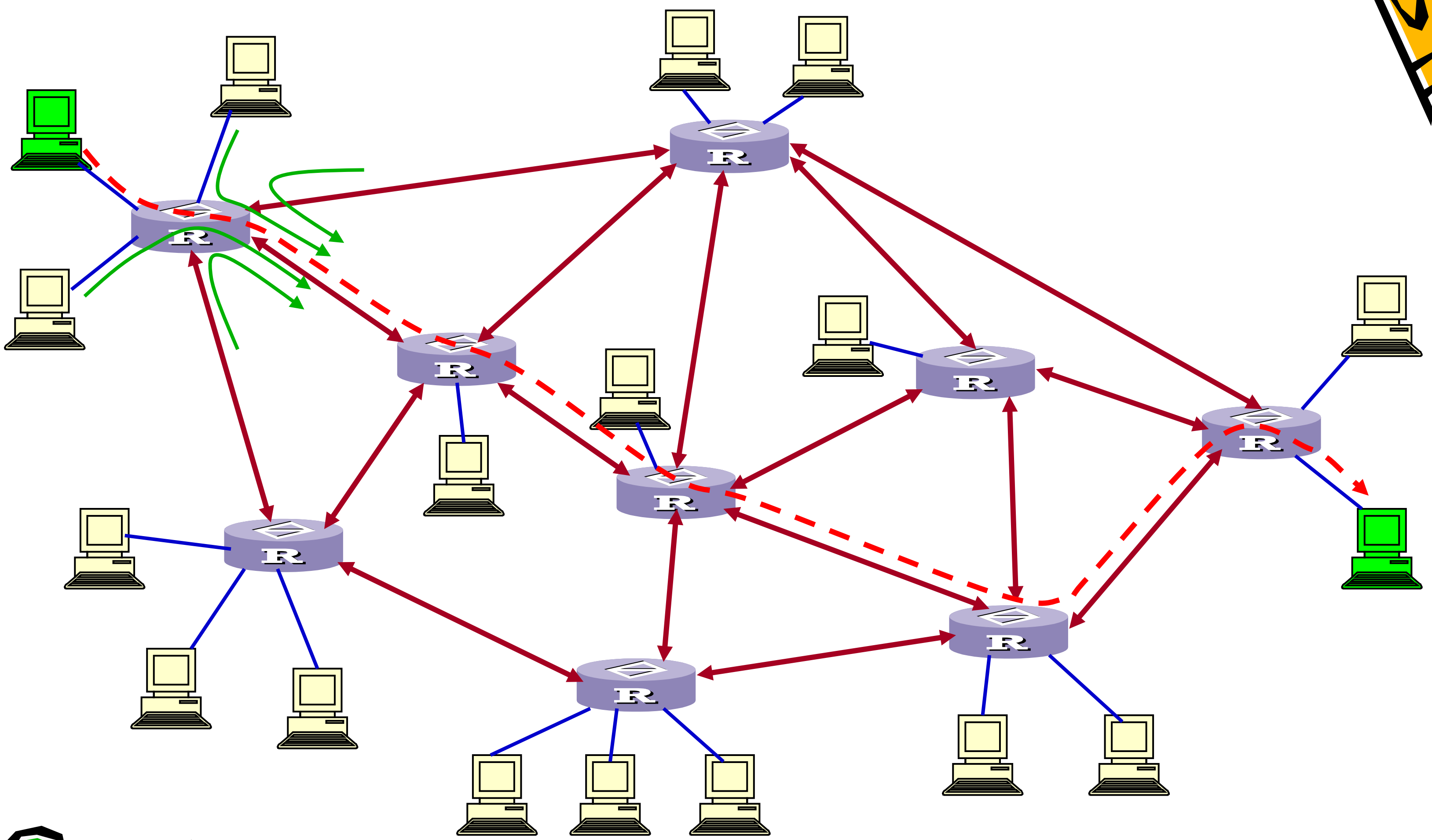
แต่ละ Router เชื่อมต่อกันด้วย Logical Link เดียว
เสมือนว่ามี Transmitter ตัวเดียวในการส่งข้อมูลผ่าน Link

Network Model (M/M/1)



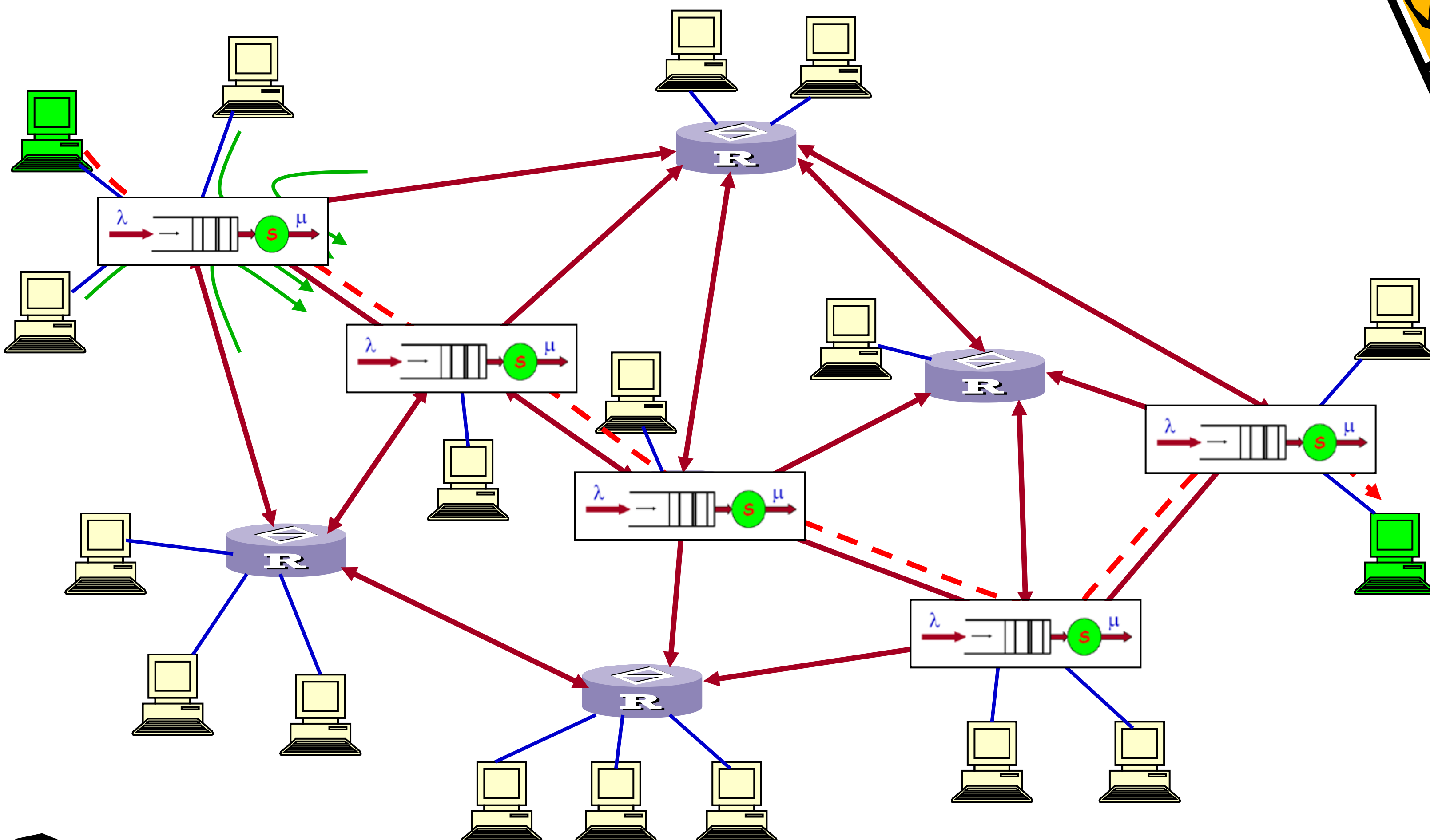
สมมุติว่าคอมพิวเตอร์ที่ต่อกับ Router
ต้องการส่งข้อมูลถึงกัน ตามเส้นทางที่กำหนด

Network Model (M/M/1)



ที่ Output ของ Router สามารถ Model โดยใช้ M/M/1

Network Model (M/M/1)



ถ้าเราให้ทุก Model เป็น M/M/1
ดังนั้น Delay จะเป็นผลรวมของ Delay แต่ละอัน

Kendal Notation

David Kendall[1] ได้คิดการให้ชื่อเพื่อจะอธิบายระบบ Queuing ซึ่งรู้จักกันในนามของ **Kendal Notation** ในรูปแบบดังนี้

$$A / B / c / K / m / Z$$

โดยที่

A หมายถึง Interarrival Time Distribution

B หมายถึง Service Time Distribution

c หมายถึงจำนวนของ Server

K หมายถึง System Capacity หรือ จำนวนของ Customer สูงสุดที่ยอมให้มีในระบบ

m หมายถึง จำนวนของ Population หรือ Source

Z หมายถึง Queue Discipline

Kendal Notation

ปกติที่เราเจอมักอยู่ในรูป Short Notation กล่าวคือ $A/B/c$ โดยสมมุติให้ความยาวของ Queue มีไม่จำกัด ($K=\infty$), จำนวนของ Source มีขนาดไม่จำกัด($m=\infty$) และ Queue Discipline เป็น FCFS(First-come first-serve) หรือ FIFO(First-in first-out)

Distribution ที่ใช้ใน A และ B คือ

GI	General Independent Interarrival Time
G	General Service Time
H_k	k-stage Hyperexponential Interarrival หรือ Service Time Distribution
E_k	Erlang-k Interarrival หรือ Service Time Distribution
M	Exponential Interarrival หรือ Service Time Distribution
D	Deterministic(Constant) Interarrival หรือ Service Time Distribution
U	Uniform Interarrival หรือ Service Time Distribution

Kendal Notation

ส่วนของ Queue Discipline ที่นิยมได้แก่

FCFS หรือ FIFO	First-come first-serve หรือ First-in first-out
LCFS หรือ LIFO	Last-Come First-Serve หรือ Last-in first-out
RSS หรือ SIRO	Random selection for service หรือ Service-in-random order
PRI	Priority Service

Next Week

- M/M/1 Analysis and Examples
- HW 5 Due

CPE 332 Computer Engineering Mathematics II

Week 7

Part II, Chapter 6

Queuing System Continue

Extra: PRNG

Topics

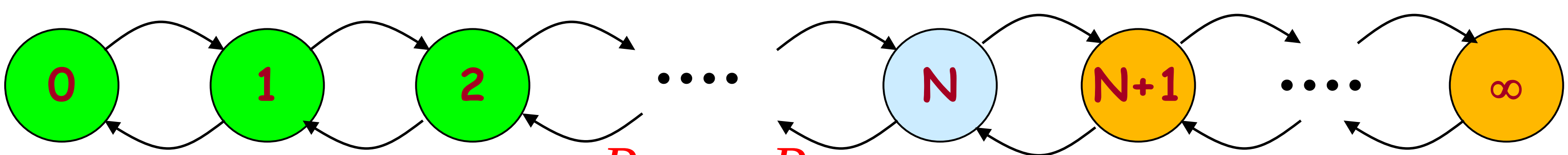
- Single Server, M/M/1
- Kendal Notation
- Applications

Queuing System Case 3: Delay System
Limited Server=N; With Unlimited Queue

$\lambda < \mu$

$$P[k] = \frac{\lambda^k e^{-\lambda}}{k!} \qquad p_x = \begin{cases} \dfrac{\rho^x}{x!} p_0 ; 0 \le x \le N \\ \dfrac{N^N}{N!}\left(\dfrac{\rho}{N}\right)^x p_0 ; x \ge N \end{cases} ; p_0 = \left[\frac{N\rho^N}{N!(N-\rho)} + \sum_{k=0}^{N-1} \frac{\rho^k}{k!} \right]^{-1} \qquad P[T \ge t] = e^{-t/T_s} ; T_s = 1/\mu$$

1. สมมุติว่า Customer แต่ละคนที่เข้ามาเป็น Poisson และได้รับการ Service จากระบบทันที
2. เวลาที่ใช้ในการ Service เป็น Random สมมุติว่าเป็น Exponential ด้วยเวลาเฉลี่ย T
3. ระบบสามารถรับ Customer ได้ไม่จำกัด แต่จะ Service ได้สูงสุด N พร้อมๆกัน
4. ถ้าทุก Server เต็ม Customer ใหม่จะต้องรอใน Queue ในกรณีนี้จะเกิด Queuing Delay
5. ระบบนี้เรียก M/M/N หรือ M/M/N/∞ แสดงได้ด้วย Simple Markov Model
6. State Probability จะมีการกระจายแบบ Second Erlang (Erlang C) Distribution



$$p_i P_{ij} = p_j P_{ji}$$

Queuing System Case 3: Delay System
Server=1; With Unlimited Queue; M/M/1

$\lambda < \mu$

$$p_0 = \left[\frac{\rho}{(1-\rho)} + 1\right]^{-1} = 1-\rho$$

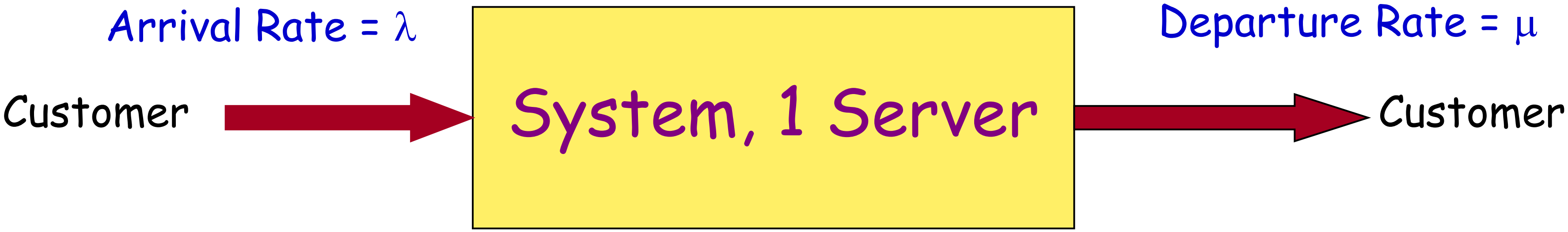
$$P[k] = \frac{\lambda^k e^{-\lambda}}{k!}$$

$$p_x = \rho^x(1-\rho)$$

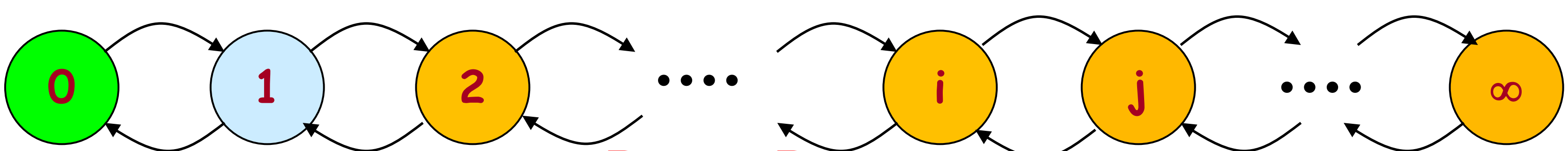
$$P[T \ge t] = e^{-t/T_s};\ T_s = 1/\mu$$

Arrival Rate = λ

Departure Rate = μ

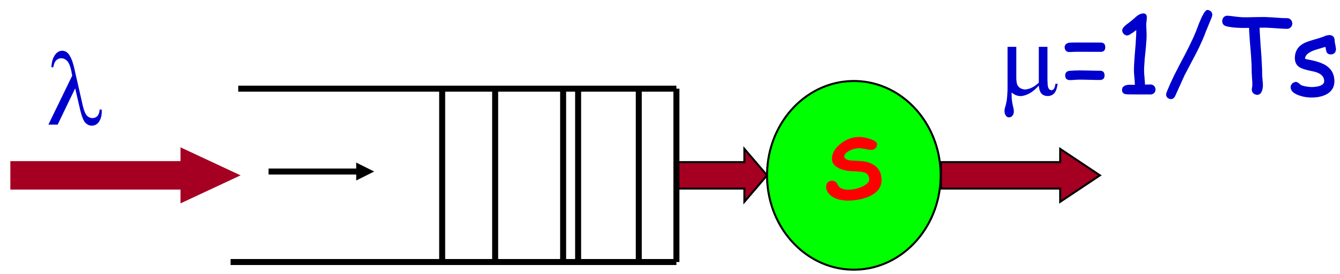


1. สมมุติว่า Customer แต่ละคนที่เข้ามาเป็น Poisson และได้รับการ Service จากระบบทันที
2. เวลาที่ใช้ในการ Service เป็น Random สมมุติว่าเป็น Exponential ด้วยเวลาเฉลี่ย T
3. ระบบสามารถรับ Customer ได้ไม่จำกัด แต่จะ Service ได้ครั้งละคน
4. ถ้าทุก Server เต็ม Customer ใหม่จะต้องรอใน Queue ในกรณีนี้จะเกิด Queuing Delay
5. ระบบนี้เรียก M/M/1 หรือ M/M/1/∞ แสดงได้ด้วย Simple Markov Model
6. State Probability จะมีการกระจายแบบ Second Erlang (Erlang C) Distribution



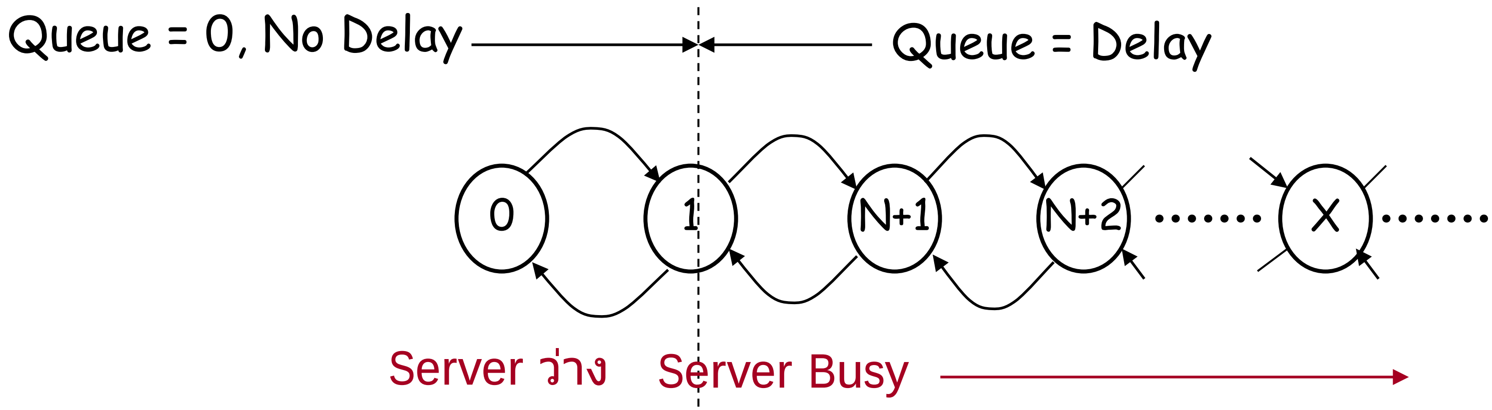
$$p_i P_{ij} = p_j P_{ji}$$

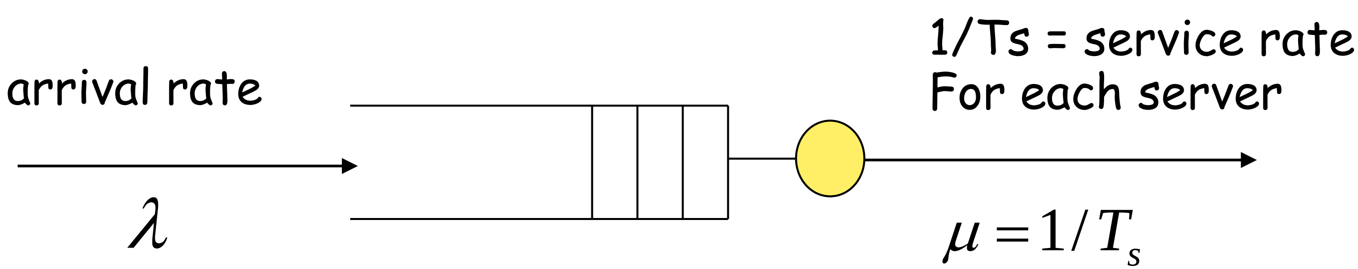
M/M/1: Summary



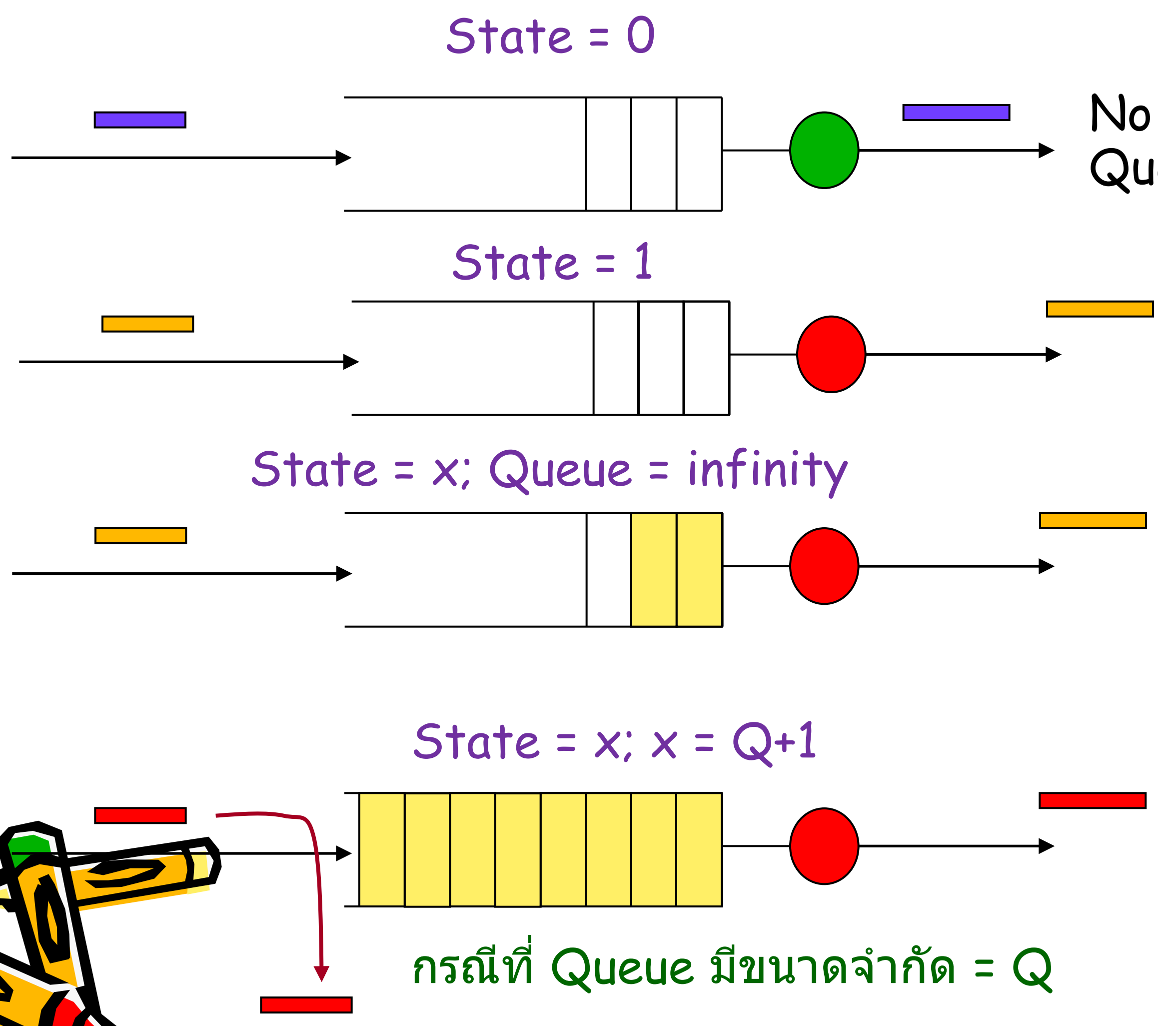
Arrival = Poisson, λ
Inter Arrival Time = Exponential, $1/\lambda$
Service Rate, μ
Service Time, Ts $(1/\mu)$ = Exponential
Queue = FIFO
1 Server

Queuing Model(1 Server); M/M/1





การทำงานของ M/M/1



No Q Delay
Queue Empty

Delay
Customer Wait in Q

Severe Delay
Queue Overflow (Full)
Congestion
Packet Lost

Kendal Notation

David Kendall[1] ได้คิดการให้ชื่อเพื่อจะอธิบายระบบ Queuing ซึ่งรู้จักกันในนามของ **Kendal Notation** ในรูปแบบดังนี้

$$A \,/\, B \,/\, c \,/\, K \,/\, m \,/\, Z$$

โดยที่

A หมายถึง Interarrival Time Distribution

B หมายถึง Service Time Distribution

c หมายถึงจำนวนของ Server

K หมายถึง System Capacity หรือ จำนวนของ Customer สูงสุดที่ยอมให้มีในระบบ

m หมายถึง จำนวนของ Population หรือ Source

Z หมายถึง Queue Discipline

Kendal Notation

ปกติที่เราเจอมักอยู่ในรูป Short Notation กล่าวคือ $A/B/c$ โดยสมมุติให้ความยาวของ Queue มีไม่จำกัด ($K = \infty$), จำนวนของ Source มีขนาดไม่จำกัด($m = \infty$) และ Queue Discipline เป็น FCFS(First-come first-serve) หรือ FIFO(First-in first-out)

Distribution ที่ใช้ใน A และ B คือ

GI	General Independent Interarrival Time
G	General Service Time
H_k	k-stage Hyperexponential Interarrival หรือ Service Time Distribution
E_k	Erlang-k Interarrival หรือ Service Time Distribution
M	Exponential Interarrival หรือ Service Time Distribution
D	Deterministic(Constant) Interarrival หรือ Service Time Distribution
U	Uniform Interarrival หรือ Service Time Distribution

Kendal Notation

ส่วนของ Queue Discipline ที่นิยมได้แก่

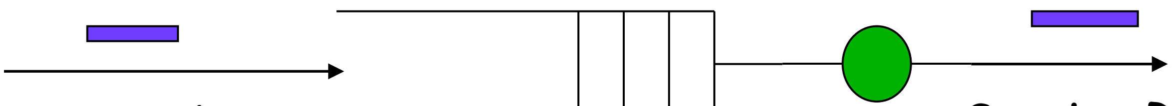
FCFS หรือ FIFO	First-come first-serve หรือ First-in first-out
LCFS หรือ LIFO	Last-Come First-Serve หรือ Last-in first-out
RSS หรือ SIRO	Random selection for service หรือ Service-in-random order
PRI	Priority Service

Analysis ของ M/M/1

- สมมุติตอนแรกว่า Queue มีขนาดไม่จำกัด
- ใช้ M/M/1 ในการ Model แต่ละ Port ของ Router (หรือ Switch L3)
- Arrival คือจำนวน Packet ที่เข้ามาในช่วงเวลาหนึ่ง ปกติวัดเป็น pps
- ขนาดของ Packet สมมุติว่าไม่แน่นอน แต่มีการกระจายแบบ Exponential
  - Service Time ของแต่ละ Packet จะเป็น Exponential ด้วย ทั้งนี้ขึ้นอยู่กับค่า Link Speed ของ Output Port
- ค่า Server Utilization เท่ากับอัตราส่วนของ Arrival Rate หารด้วย Service Rate จะบ่งบอกอัตราส่วนที่ Server จะ Busy และคือ Link Utilization ของ Output Port ด้วย

$$\rho = \lambda / \mu$$

Queuing in Communication NW and M/M/1



Arrival Rate

$$\lambda$$

Service Rate = 1/Service Time

$$\mu = 1/T_s$$

$$\rho = \lambda/\mu = \lambda T_s$$

Example

- Router ได้รับ Packet เฉลี่ย 8 pps
  - ความยาวของ Packet มีการกระจายแบบ Exponential ด้วยความยาวเฉลี่ย 500 Octet
  - Link ที่จะส่งออกไป มีความเร็ว 64 kbps
- 1. Arrival Rate, λ = 8 pps
- 2. ความยาวเฉลี่ยของ Packet = 4000 bit
- 3. ความเร็ว Link = 64 k ดังนั้น Service Time, Ts ของแต่ละ Packet = 4000/64k = 1/16
- 4. Service Rate(μ) = 16 pps
- 5. Server Utilization = 8/16 =0.5 = 50%

Assumption

- 1. อย่าลืมว่า Packet ที่เข้ามา ต้องเป็น Independent และ Random มันจึงเป็น Poisson
- 2. ความยาวของ Packet จะสมมุติว่าเป็น Exponential ดังนั้น Service Time จะเป็น Exponential ด้วย แม้ว่าสมมุติฐานนี้จะไม่ถูกต้องนัก
- 3. มี Output Link เดียว คือเป็น Single Server
- 4. ดังนี้แล้ว จึงจะเป็น M/M/1

Utilization

- Utilization บอกอัตราส่วนที่ Server จะ Busy และสัมพันธ์กับ Probability ที่ Queue จะว่าง
  - Probability ที่ Server ว่าง $1-\rho$
- ใน Network คือคือ Probability ที่ Output Link จะ Busy ด้วย

$$\boxed{\rho = \lambda / \mu = \lambda T_s}$$

Arrival Rate

- เนื่องจาก Arrival Rate มีการกระจายแบบ Poisson ดังนั้นถ้าให้ λ เป็นอัตราเฉลี่ยของ Customer (Packet) ที่เข้ามาในช่วงเวลา 1 วินาที
  - Probability ที่จะมี k customer (Packet) เข้ามาในช่วงเวลา T วินาทีจะหาได้จาก

$$p(X = k) = p(k) = \frac{(\lambda T)^k e^{-\lambda T}}{k!}$$

$$p(0) = e^{-\lambda T}$$

Service Time

- T_s เป็น Service Time เฉลี่ย และ Service Rate หาได้จาก $\mu = 1/T_s$
- เนื่องจาก Service Time เป็น Random Variable ที่มีการกระจายแบบ Exponential ดังนั้น Probability ที่ Service Time จะมีค่าน้อยกว่า T จะเป็น

$$p(t < T) = 1 - e^{-T/T_s}$$

Queue Distribution

- การกระจายของ Customer (State Probability) สามารถคำนวณได้จาก Probability ที่ ระบบ จะมี k Packet อยู่ดังนี้

$$p_k = p_0(\lambda T_s)^k$$

- โดยที่ p_0 คือ Probability ที่ ระบบ จะว่าง

$$p_0 = 1 - \rho = 1 - \lambda T_s$$

- ดังนั้น เราได้ $p_k = (1-\lambda T_s)(\lambda T_s)^k = (1-\rho)\rho^k$
- กล่าวคือ การกระจายของ Customer ในระบบ หรือค่า State Probability จะเป็น Geometric Distribution

[illegible] Customer [illegible], N [illegible] State, $E[X]$

จาก $P[X=x]=p_x=(1-\rho)\rho^x$ เราได้

$$E[X]=\sum_{x=0}^{\infty} x\,p_x=(1-\rho)\sum_{x=0}^{\infty} x\rho^x$$

แต่จาก (CPE231) $\sum_{x=0}^{\infty} x\rho^{x-1}=\frac{1}{(1-\rho)^2}$

ดังนั้น

$$[X]=(1-\rho)\rho\sum_{x=0}^{\infty} x\rho^{x-1}=\frac{\rho}{1-\rho}$$

Queuing Delay

- จาก Geometric Distribution ค่าเฉลี่ย คือจำนวน Customer เฉลี่ย คือจำนวน Packet เฉลี่ยในระบบ จะหาได้จาก

$$N = \frac{\rho}{1-\rho}$$

- ถ้าคิดเฉพาะจำนวน Customer เฉลี่ยใน Queue เราจะได้

$$N_Q = N - \rho = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho}$$

Queuing Delay

- สำหรับ Network ค่าเฉลี่ย จำนวน Packet เฉลี่ยใน ระบบ และใน Queue จะหาได้จาก

$$N = \frac{\rho}{1-\rho} \qquad N_Q = \frac{\rho^2}{1-\rho}$$

- ถ้าแต่ละ Packet ต้องใช้เวลาเฉลี่ยในการ Service T_s ดังนั้น ค่า Queuing Delay จะเป็น

$$W = \frac{\rho T_s}{1-\rho} = \frac{\rho}{\mu - \lambda}$$

System Delay

- แต่ละ Packet ต้องใช้เวลาเฉลี่ยในการ Service T_s ดังนั้น ค่า Queuing Delay จะเป็น

$$W = \frac{\rho T_s}{1-\rho} = \frac{\rho}{\mu - \lambda}$$

- และเวลาเฉลี่ยทั้งหมดที่ลูกค้าจะต้องรอในระบบทั้งหมดจะเป็น

$$T = W + T_s = \frac{\rho}{\mu - \lambda} + \frac{1}{\mu} = \frac{1}{\mu - \lambda}$$

Little's Theorem

- ถ้า T เป็นเวลาเฉลี่ยที่ลูกค้าอยู่ในระบบ และ λ เป็น Arrival Rate ดังนั้นจำนวนลูกค้าเฉลี่ยในระบบจะเท่ากับ

$$N = \lambda T$$

$$N_Q = \lambda W$$

สรุป M/M/1

I. Notations

p_n : เป็นค่า Steady-state probability ที่จะมี n customer ในระบบ

λ : เป็นค่า Arrival rate(เป็นค่าส่วนกลับของค่าเฉลี่ยของ inter arrival time)

μ : ค่า Service rate(ส่วนกลับของค่าเฉลี่ยของ service time) $= 1/T_s$

N : ค่าเฉลี่ยของจำนวนของ Customer ในระบบ

N_Q : ค่าเฉลี่ยของจำนวนของ customer ที่รอใน Queue

T : ค่าเฉลี่ยของเวลาของ Customer ที่อยู่ในระบบ

W : ค่าเฉลี่ยของเวลาที่ Customer ต้องรอใน Queue ไม่รวม Service Time

$\bar{X}$: ค่าเฉลี่ยของ Service Time

$\overline{X^2}$: Second Moment ของ Service Time

สรุป M/M/1

II. Little's Theorem

$$N = \lambda T$$

$$N_Q = \lambda W$$

III. Poisson Distribution with Parameter m

$$p_n = \frac{e^{-m} m^n}{n!}, \quad n = 0,1,\ldots$$

mean = variance = m

IV. Exponential Distribution with Parameter λ

$$P[\tau \le s] = 1 - e^{-\lambda s}, \quad s \ge 0$$

Density: $p(\tau) = \lambda e^{-\lambda \tau}$

Mean: $= 1 / \lambda$

Variance $= 1 / \lambda^2$

สรุป M/M/1

(a) Utilization Factor (อัตราส่วนของเวลาที่ Server Busy)

$$\rho = \frac{\lambda}{\mu}$$

(b) Probability ที่จะมี n customer ในระบบ

$$p_n = \rho^n (1-\rho), \quad n = 0,1,\ldots$$

(c) จำนวนเฉลี่ยของ Customer ในระบบ

$$N = \frac{\rho}{1-\rho}$$

(d) เวลาเฉลี่ยของ Customer ในระบบ

$$T = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\mu-\lambda}$$

(e) จำนวนเฉลี่ยของ Customer ใน Queue

$$N_Q = \frac{\rho^2}{1-\rho}$$

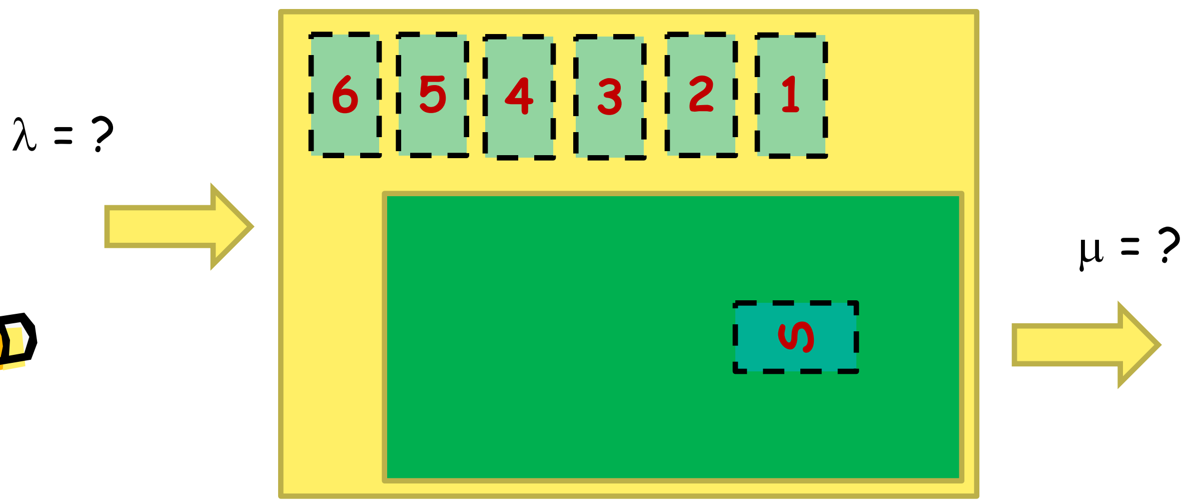
(f) เวลาเฉลี่ยที่ต้องรอใน Queue ของ Customer

$$W = \frac{\rho}{\mu-\lambda}$$

M/M/1 Example 1

Examples (M/M/1)

1. สถานีรับจ้างล้างรถอัตโนมัติแห่งหนึ่ง จากการรวบรวมข้อมูลของรถที่เข้ามารับบริการ พบว่าจำนวนรถที่เข้ามามีการกระจายแบบ Poisson โดยมีอัตราเฉลี่ย 4 คัน ต่อชั่วโมง เวลาที่ใช้ในการล้างรถแต่ละคันพบว่ามีการกระจายแบบ Exponential โดยมีค่าเฉลี่ยเท่ากับ 10 นาทีต่อหนึ่งคัน และการล้างรถมีเพียง 1 Line กล่าวคือจะล้างได้เพียงครั้งละหนึ่งคันเท่านั้น จงหา (1) Probability ที่รถที่เข้ามาใช้บริการจะต้องรอการใช้บริการ (2) ถ้าทางร้านจัดที่จอดรถให้ลูกค้าที่รอการใช้บริการไว้จำนวน 6 คัน จงหา Probability ที่รถที่เข้ามาใช้บริการจะไม่สามารถจอดรถรอได้



M/M/1 Example 1

<u>คำตอบ</u>

- จากที่โจทย์กำหนด กล่าวคือ Arrival เป็น Poisson Distribution, Service เป็น Exponential Distribution และความสามารถที่จะบริการล้างรถได้ทีละ 1 คัน เราสรุปได้ว่าระบบจะเป็น M/M/1 Model
- ในกรณีนี้เราจำเป็นต้องหาค่าเฉลี่ย 2 ตัว และสองตัวเท่านั้น กล่าวคือค่าเฉลี่ยของ Arrival Rate หรือ λ และค่าเฉลี่ยของ Service Rate คือ μ ที่สำคัญในการคิดคือ ต่อหน่วยเวลาจะต้องเท่ากัน ในกรณีของข้อนี้จะคิดเป็นต่อชั่วโมง(กรณีอื่นๆอาจจะเป็น ต่อวินาที, ต่อ 5 นาที หรืออื่นๆตามความเหมาะสม) เมื่อได้ 2 ตัวนี้แล้วก็สามารถจะแทนค่าไปในสูตรอื่นๆที่ให้มาข้างต้นได้หมด
- จากข้อมูลที่โจทย์ให้เราได้ $\lambda = 4$ คันต่อชั่วโมง และ $\mu = 60/10 = 6$ คันต่อชั่วโมง
- เมื่อแทนค่าตามสูตรเราได้ดังนี้
    - Utilization Factor (อัตราส่วนของเวลาที่ Server Busy)

$$\rho = \frac{\lambda}{\mu} = 4/6 = 2/3 = 0.6667$$

M/M/1 Example 1

- Probability ที่จะมี n customer ในระบบ

$$p_n = \rho^n (1-\rho), \quad n = 0,1,\ldots = (2/3)^n (1/3)$$

- จำนวนเฉลี่ยของ Customer ในระบบ

$$N = \frac{\rho}{1-\rho} = \frac{2/3}{1/3} = 2 \text{ คัน}$$

- เวลาเฉลี่ยของ Customer ในระบบ

$$T = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\mu-\lambda} = \frac{1}{6-4} = 1/2$$ ชั่วโมง หรือ 30 นาที(สังเกตหน่วย เป็นชั่วโมง)

M/M/1 Example 1

- จำนวนเฉลี่ยของ Customer ใน Queue

$$N_Q = \frac{\rho^2}{1-\rho} = \frac{4/9}{1/3} = \frac{4}{3} = 1.3333 \text{ คัน}$$

- เวลาเฉลี่ยที่ต้องรอใน Queue ของ Customer

$$W = \frac{\rho}{\mu - \lambda} = \frac{2/3}{6-4} = \frac{1}{3}$$ ชั่วโมง = 20 นาที ค่านี้คือเวลาเฉลี่ยสำหรับเฉพาะใน Queue เท่านั้น คือเวลาที่ลูกค้าต้องรอก่อนที่จะได้รับบริการ ไม่ใช่เวลาเฉลี่ยทั้งหมดในระบบ ซึ่งค่าเฉลี่ยทั้งหมดในระบบต้องบวกกับค่าเฉลี่ยของ Service Time กล่าวคือ 10 นาที เป็น 30 นาที(ค่า T) ซึ่งคำนวณให้ดูก่อนหน้านี้แล้ว

M/M/1 Example 1

- คำตอบคำถามที่ (1) โจทย์ต้องการให้คำนวณ Probability ที่รถเข้ามาใช้บริการจะต้องรอใน Queue กล่าวคือ รถที่เข้ามาจะต้องรอก็ต่อเมื่อมีรถคันอื่นก่อนหน้านี้กำลังใช้บริการอยู่ ถ้าไม่มีรถในระบบเลย รถที่เข้ามาจะไม่ต้องรอและสามารถใช้บริการได้ทันที นั่นคือเราต้องการหา Probability ที่จะมีรถ 1 คัน หรือมากกว่าในระบบ

  หรือ $P[X \geq 1] = \sum_{n=1}^{\infty} P_n$ ในกรณีนี้จะหาได้ง่ายกว่าถ้าเราใช้กฎของ Probability

  โดย $P[X \geq 1] = 1 - P[X < 1] = 1 - P[0] = 1 - P_0 = 1 - (2/3)^0(1/3) = 0.6667$

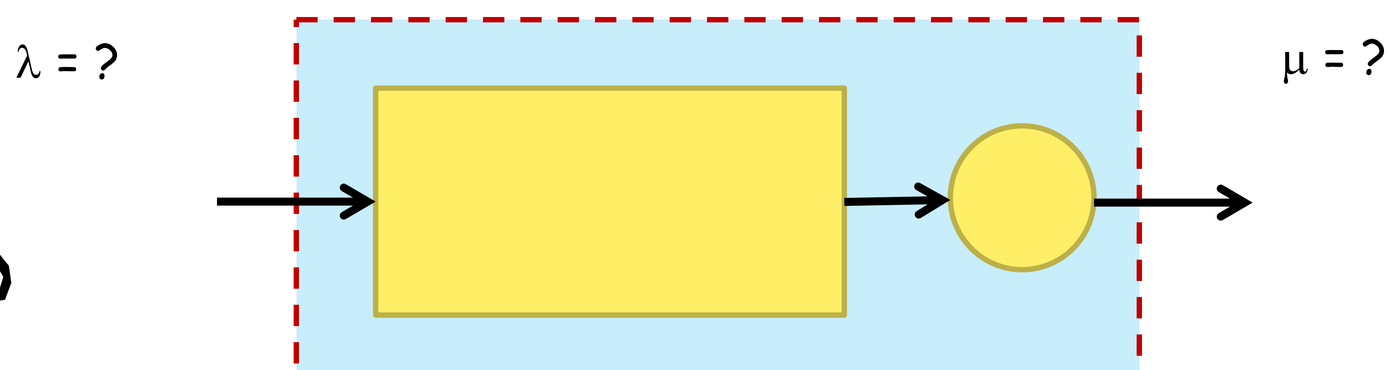
- คำตอบคำถามที่ (2) ในกรณีนี้รถที่รอเป็นคันที่ 7 จะไม่มีที่จอด นั่นก็คือมีรถจอดอยู่แล้ว 6 คัน รวมกับรถที่กำลังให้บริการ 1 คัน ดังนั้นถ้ามีรถอยู่แล้ว 7 คัน หรือ มากกว่า รถที่เข้ามาต่อไปจะไม่มีที่จอด นั่นก็คือโจทย์ต้องการให้เราหา $P[X \geq 7] = 1 - P_0 - P_1 - \ldots - P_6$ ให้ลองใช้ MATLAB คำนวน จะได้ค่าเท่ากับ 1 – 0.94147 = 0.05853

- หมายเหตุ ข้อ(2) อาจจะคิดจาก $P[X \geq 7] = \sum_{n=7}^{\infty} P_n = \sum_{n=7}^{\infty} \rho^n (1-\rho) = (1-\rho)\sum_{n=7}^{\infty} \rho^n$

$$= (1-\rho)\sum_{k=0}^{\infty} \rho^{k+7} = (1-\rho)\rho^7 \sum_{k=0}^{\infty} \rho^k = (1-\rho)\rho^7 \frac{1}{1-\rho} = \rho^7 = (2/3)^7 = 0.05853$$

M/M/1 Example 2

2. สาขาย่อยของธนาคารแห่งหนึ่งตั้งไว้ที่ศูนย์การค้า ประกอบด้วยเจ้าหน้าที่รับบริการ(Teller)หนึ่งคน จากการเก็บข้อมูลลูกค้าที่มาใช้บริการพบว่าเฉลี่ยแล้วมีผู้เข้ามาใช้บริการชั่วโมงละ 10 คน โดยการเข้ามาของลูกค้าเป็น Random และมีการกระจายแบบ Poisson และลูกค้าแต่ละคนเฉลี่ยแล้วจะใช้บริการ เป็นเวลา 5 นาที โดยระยะเวลาการใช้งาน จัดว่าเป็น Random เช่นกัน แต่มีการกระจายแบบ Exponential ธนาคารได้จัดที่นั่งไว้ 3 ที่สำหรับผู้ที่มารอ(รวมถึงผู้ที่กำลังใช้บริการด้วย) ที่เหลือจะต้องยืนรอ (1) จงหา Probability ที่ลูกค้าเมื่อเข้ามาจะได้นั่ง (2) จงหา Probability ที่ลูกค้าที่เข้ามาจะต้องยืนรอ (3) ฉลี่ยแล้วลูกค้าที่เข้ามาจะต้องรอเป็นเวลานานเท่าไรก่อนที่จะได้รับบริการ (4) ธนาคารควรจะต้องจัดที่นั่งไว้เท่าไรเพื่อจะให้แน่ใจว่าอย่างน้อย 80% ของลูกค้าที่เข้ามาใช้บริการจะมีที่นั่งรอได้ทันที



M/M/1 Example 2

<u>คำตอบ</u>

Model นี้เช่นกันจัดได้ว่าเป็น M/M/1(ในชั้นนี้จะเจอแต่ M/M/1, M/M/N, M/M/N/N)

ก่อนอื่นหาค่าที่ต้องการ 2 ตัวคือ λ และ μ

$\lambda = 10$ คนต่อชั่วโมง $\mu = 60/5 = 12$ คนต่อชั่วโมง

จากนั้นคำนวณ $\rho = \mu/\lambda = 5/6$

(1) ถ้ามีลูกค้าอยู่ในระบบ 2 คนหรือน้อยกว่า ลูกค้าที่เข้ามาใหม่จะได้ที่นั่ง ดังนั้น

$$P[X \le 2] = P[0] + P[1] + P[2]] = (1/6)[1 + 5/6 + 25/36] = 0.421$$

(2) จากข้อ (1) เราต้องการ $P[X > 2] = 1 - P[X \le 2] = 1 - .421 = .579$

(3) $W = \dfrac{\rho}{\mu - \lambda} = \dfrac{5/6}{12 - 10} = \dfrac{5}{12}$ ชั่วโมง หรือ 25 นาที

4.ธนาคารต้องจัดที่นั่งกี่ที่(รวมที่นั่งตอนใช้บริการ)เพื่อจะให้แน่ใจว่าอย่างน้อย 80% ของผู้ที่เข้ามาจะได้นั่ง

- สมมุติให้มีที่นั่งทั้งหมด = x ที่
- ถ้าระบบอยู่ที่ State x-1 ลงมา ลูกค้าที่เข้ามาใหม่จะได้นั่ง
- เราต้องการ P[ระบบอยู่ที่ State $\leq$ x-1] > 0.8

  หรือ P[State < x] > 0.8

  หรือ 1-P[State $\geq$ x] > 0.8

  หรือ P[State $\geq$x] $\leq$ 0.2 กล่าวคือ $\rho^{x} \leq 0.2$

  มีวิธีเดียวในการแก้ อสมการนี้คือใช้ Log

4.ธนาคารต้องจัดที่นั่งกี่ที่(รวมที่นั่งตอนใช้บริการ)เพื่อจะให้แน่ใจว่าอย่างน้อย 80% ของผู้ที่เข้ามาจะได้นั่ง

หรือ P[State ≥x] ≤ 0.2 กล่าวคือ $\rho^x \leq 0.2$

มีวิธีเดียวในการแก้ อสมการนี้คือใช้ Log

$$\mathrm{Log}(\rho^x) \geq \log(0.2)$$

Note:เนื่องจากค่าติดลบ ต้องกลับเครื่องหมาย

$$x \ \log \rho \geq \log 0.2$$

ใช้ Natural Log(ความจริงจะใช้ Log ฐานอะไรก็ได้)

$$x \log 5/6 \geq \log 0.2$$

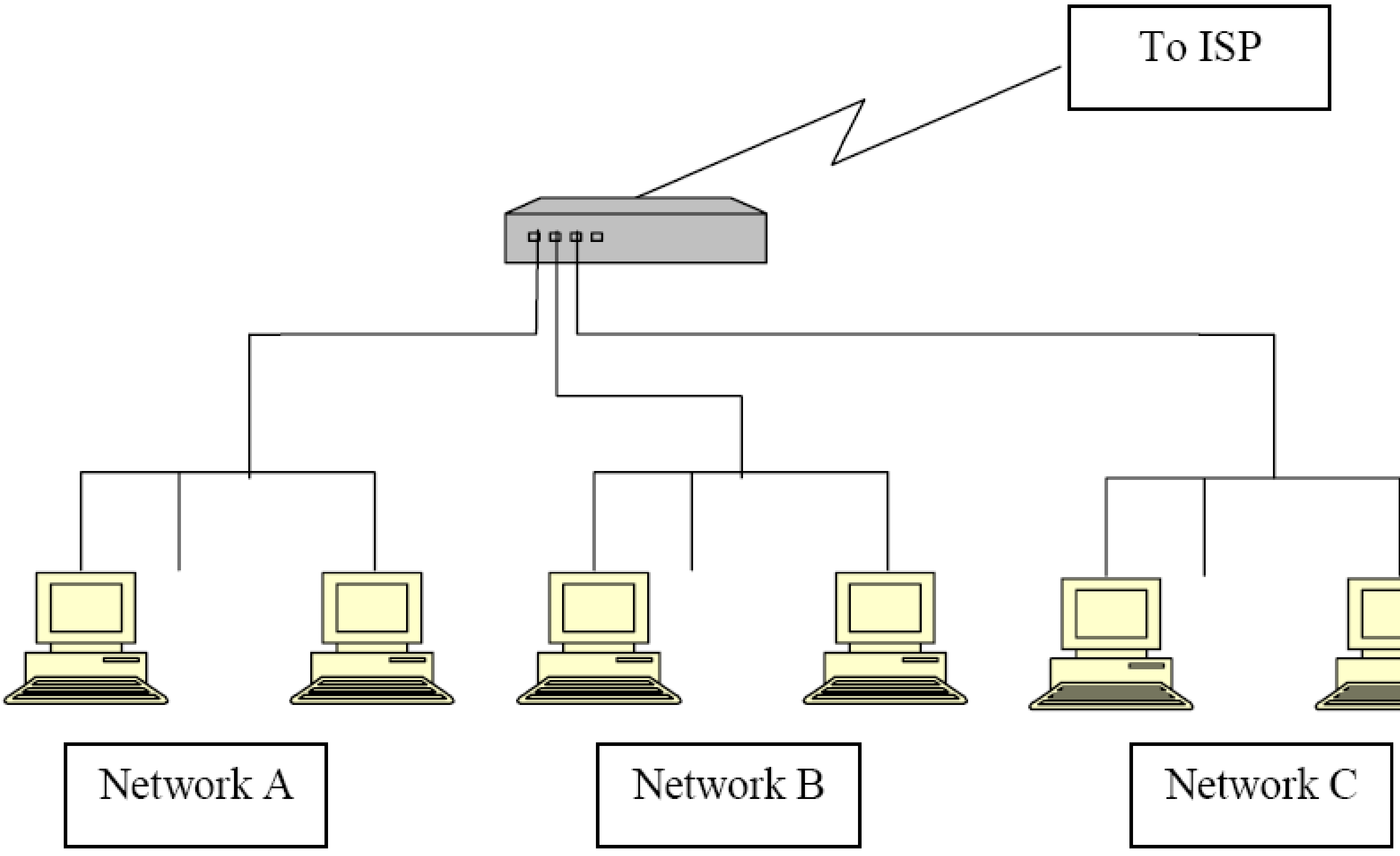
$$x \geq \frac{-1.60944}{-0.18232} = 8.8275$$

เนื่องจากคำตอบต้องเป็นจำนวนเต็ม และจากเครื่องหมายมากกว่าหรือเท่ากับ ดังนั้น $x = 9$

M/M/1 Example 3

3. ในระบบ Network ของสำนักงานแห่งหนึ่ง ทำการแบ่ง Network ออกเป็น 3 Subnet ดังรูป ซึ่ง Network ของสำนักงานแห่งนี้จำเป็นที่จะต้องมีการต่อออก Internet ผ่าน Router เชื่อต่อกับ ISP และจากการใช้ Network Tool จับ Internet Traffic ขาออก พบว่า Network A มีอัตรา Packet เฉลี่ย 300 pps(Packet Per Second), Network B มีอัตราเฉลี่ย 500 pps และ Network C มีอัตราเฉลี่ย 600 pps โดยที่ลักษณะการเข้ามาของ Packet มีการกระจายแบบ Poisson นอกจากนี้แล้ว ยังพบว่าขนาดของแต่ละ Packet มีขนาดไม่แน่นอน แต่สามารถประมาณได้ว่ามีการกระจายแบบ Exponential โดยมีความยาวเฉลี่ย 500 Octet, สาย Lease Line ที่เช่าต่อระหว่าง Router และ ISP มีอัตราความเร็วที่ 8 Mbps และ Router มีการทำงานในระดับ Wire-speed จงคำนวณ (1) Link Utilization(Uplink) ของสาย Lease Line (2) ค่า Delay เฉลี่ยในส่วนของ Uplink (3) สมมุติว่าอุปกรณ์ Router มี Queue ที่มีความสามารถเก็บได้เฉลี่ย 10 Packet จงคำนวณหา % Packet Drop อันเนื่องมาจาก Queue Overflow ที่อุปกรณ์ Router

M/M/1 Example 3



M/M/1 Example 3

<u>คำตอบ</u>

สังเกตุว่าแม้ว่า Output Speed จะมีอัตราคงที่คือ 8 Mbps แต่ขนาด Packet เป็น Exponential Distribution ดังนั้น Service Time จะเป็น Exponential Distribution ด้วย

$\lambda = 300 + 500 + 600 = 1400$ packet ต่อวินาที(pps)

$\mu = \frac{8,000,000}{500 \times 8} = 2000$ packet ต่อวินาที(pps)

1. $\rho = \frac{1400}{2000} = 0.7$ เป็นค่า Utilization ของ Server/
2. Delay เราวัดเริ่มจาก Packet เข้ามาจนกระทั่งออกไ
   ดังนั้น เราใช้สมการ T=1/(μ-λ) = 1/600 = 1.6667
3. Queue มีขนาด 10 Packet รวมกับอีกหนึ่งที่ Serve
   P[X>= 10+1] = 0.70^{11} = 0.019773 = 1.9773%

End of Chapter 6

- HW6 Due Monday Noon
  - ส่งที่พี่หนึ่งเท่านั้น ก่อนเที่ยง จันทร์ 29 ก.พ.
  - ใส่ตะกร้า หน้าโต๊ะ Counter อย่าส่งผิดที่
  - เฉลยจะประกาศวันถัดไปบน Web
- Next
  - Random Number Generator
  - Preparation for Midterm Exam

Topics

- Random Number and Properties
- Random Number Test
- Pseudo Random Number Generator
- **MT Exam Preparation**

Random Number

- Random Number เป็น Set ของตัวเลขที่สุ่มขึ้นมาแบบอิสระ(Random) และไม่ขึ้นต่อกัน (Independent) โดยมีค่าการกระจาย (Distribution) ของชุดตัวเลข ตามที่กำหนดแบบใดแบบหนึ่ง
- เลขแต่ละตัวที่สุ่มจะต้องไม่มีความสัมพันธ์กัน ทดสอบได้จากค่า Correlation

[illegible] Random Number

- Gambling
- Statistical Sampling
- Simulation
- Cryptography
- Computer Games
- Hash Algorithm/Searching Algorithm

Random Number Generation

- เป็นอุปกรณ์ที่ใช้เป็นตัวกำเนิดชุดของตัวเลข Random ที่ต้องการ
  - True Random Number Generator(TRNG)
    - ประกอบด้วยอุปกรณ์ทาง Physic ที่ใช้กำเนิดตัวเลข
    - มักจะได้จากแหล่งกำเนิดของ Noise ที่เกิดตามธรรมชาติ
      - Thermal Noise/Shot Noise/Radioactive
    - หรือใช้ Roulette Wheel
  - Pseudo Random Number Generator(PRNG)
    - Computational Algorithm จากสมการทางคณิตศาสตร์ โดยเริ่มจากตัวเลข “Seed”
    - มีคุณสมบติเหมือนกับเป็น Random แต่มี Period และสามารถคำนวณล่วงหน้าตามสมการได้

PRNG: Pseudo Random Number Generator: Linear Congruential Generator

- เริ่มจาก "Seed", X_0 และคำนวณ Series $X_1, X_2, X_3, \dots$ จากสมการ Recurrence

$$X_{n+1} = (aX_n + b) mod\, m$$

  - *a*, *b* และ *m* เป็น Integer ปกติจะมีขนาดใหญ่; จำนวนเลขสูงสุดที่ไม่ซ้ำกันจะถูกกำหนดด้วยค่า Modulus *m*
    - Sequence ที่ได้จะมีค่าระหว่าง [0,m) หรือ $0 \leq X < m$
    - Seed $X_0; 0 \leq X_0 < m$
    - Multiplier $a;\ 0 < a < m$
    - Increment $b; 0 \leq b < m$
    - ถ้า b = 0 เราเรียก Multiplicative Congruential Generator หรือ Lehmer Generator มิฉะนั้นแล้วเราเรียก Mixed Congruential Generator
  - ตัวเลขที่ได้จะมีการกระจายแบบ Uniform ถ้าต้องการการกระจายแบบอื่น จะใช้การ Transformation
  - Algorithm นี้เรียก LCG หรือ Linear Congruential Generator เป็นวิธีที่ง่ายในการสร้าง PRN

Period of LCG

- Period มีค่าได้สูงสุดคือ m เรียก Full Period
  - บางกรณีจะได้น้อยกว่านั้น ขึ้นอยู่กับการเลือกค่า 'a' และในกรณีที่ b=0
- Generator จะมี Full Period ก็ต่อเมื่อ (Hull-Dobell Theorem)
  - 1. b และ m เป็น Relative Prime
  - 2. a-1 จะต้องสามารถหารได้ด้วยทุกๆ factor ที่เป็นค่า prime ของ m
  - 3. a-1 จะต้องหารได้ด้วย 4 ถ้า m หารได้ด้วย 4

Parameters used

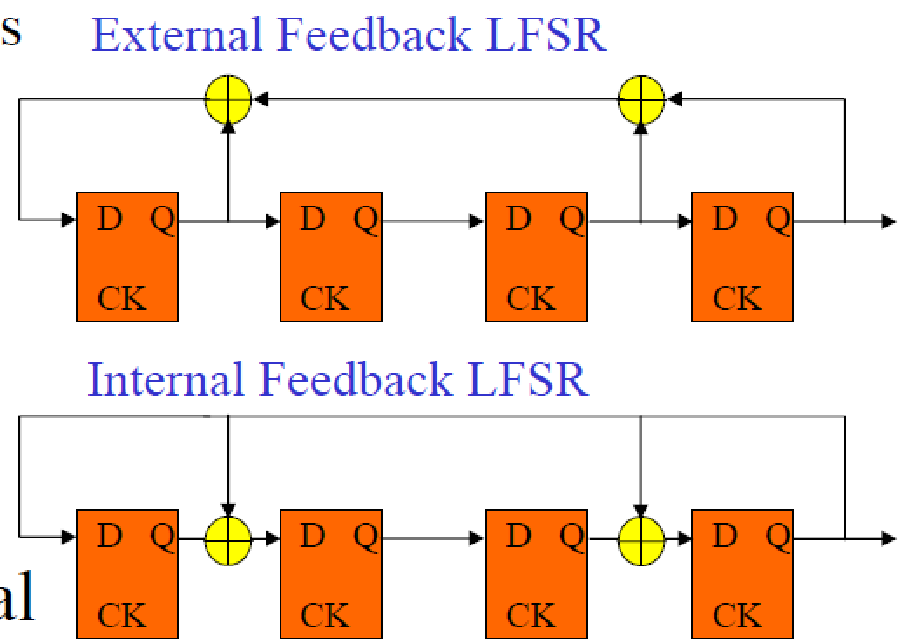
- ส่วนใหญ่จะใช้ LCG ที่ m มีค่าเป็นกำลังของ 2 ที่นิยมมากสุดคือ 2^{32} และ 2^{64}
  - MS Visual C++ ใช้ $m=2^{32}$, $a=214013, b=2531011$
  - MS Visual Basic 6 ใช้ $m=2^{24}$, $a=1140671485, b=12820163$

[illegible] LCG

- มีข้อดีคือคำนวณได้ง่าย แต่ให้ Series ของ Random Number ที่มีคุณสมบัติพอประมาณเท่านั้น เพราะให้ค่า Serial Correlation สูง
  - ไม่เหมาะกับการนำไปใช้ใน Monte Carlo Simulation
  - ไม่เหมาะกับการนำไปใช้ใน Cryptography
- สามารถสร้างได้จากวงจร Linear Feedback Shift Register(LFSR)
  - ปกติวงจรนี้จะผลิต Stream ของ bit
- ได้ Uniform Distributed PRNG

Linear Feedback Shift Registers (LFSRs)

- Efficient design for Test Pattern Generators & Output Response Analyzers (also used in CRC)
  - FFs plus a few XOR gates
  - better than counter
    - fewer gates
    - higher clock frequency
- Two types of LFSRs
  - External Feedback
  - Internal Feedback
    - higher clock frequency
- Characteristic polynomial
  - defined by XOR positions
  - $P(x) = x^4 + x^3 + x + 1$ in both examples



PRNG [illegible]

- Blum Blum Shub
- Wichmann-Hill
- Multiply with Carry
- Mersenne Twister (นิยมมากสุด)
- Park-Miller
- RC4
- Rule 30

Randomness Test

- เพื่อหา Pattern ในชุด หรือ Series ของตัวเลข
  - ซึ่งไม่ควรจะมีอยู่ถ้าชุดตัวเลขเป็น Random และ Independent อย่างแท้จริง
- ใช้ในการทดสอบ และเปรียบเทียบ Algorithm ต่างๆ ที่ใช้ผลิต PRNG โดยใช้พื้นฐานจาก
  - Statistical Test ทดสอบจากคุณสมบัติทางสถิติ
    - **Diehard Tests** ประกอบด้วยชุดของ Test
    - **TestU01** library ประกอบด้วย utilities สำหรับ statistical testing ของ uniform RNG
  - Transform ดูคุณสมบัติเมื่อ Transform
    - Hadamard Transform(Generalized FT)
  - Complexity ความซับซ้อนของตัวเลข
    - Kolmogorov complexity

Midterm Preparation

- เก็บ 35 เปอร์เซ็นต์
- ไม่มีการ Make Up
- สอบ 3 ชั่วโมง บทที่ 1-6
  - Monday 7 March; 13:30 - 16:30
- ต้องใช้เครื่องคิดเลข
  - รุ่นตามที่คณะประกาศเท่านั้น ห้ามใช้อุปกรณ์อื่นๆ
- สมการที่สำคัญจะให้มา ไม่ต้องจำ
- **6 ข้อ บทละ 1 ข้อ รวม 6 ข้อ เลือกทำ 5 ข้อ**
  - **5 ข้อ 50 คะแนน คิดเป็น 35 %**

สมการที่ให้ ในการสอบ MT

ข้อมูลที่น่าสนใจ

สูตรทั่วไป

$$e^{\pm jx} = \cos x \pm j\sin x$$

$$\sin x = \tfrac{1}{2j}(e^{jx} - e^{-jx})$$

$$\cos(a \pm b) = \cos a\cos b \mp \sin a \sin b$$

$$\sin^2 a + \cos^2 a = 1$$

$$\int \sin ax\,dx = -\tfrac{1}{a}\cos ax$$

$$d(uv) = v\,du + u\,dv$$

$$\sum_{n=0}^{\infty} a^n = \frac{1}{1-a}, \quad 0 < a < 1$$

$$\binom{N}{k} = \frac{N!}{k!(N-k)!}$$

Vector/Matrix

$$\mathbf{A}\times\mathbf{B} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 \end{vmatrix},$$

$$\mathbf{A}^{-1} = \frac{\mathrm{adj}\ \mathbf{A}}{|\mathbf{A}|}$$

$$\det(\mathbf{A}) = \sum_{j=1}^{n} (-1)^{j+1} a_{1,j} \det(\mathbf{A}_{1,j})$$ (First Row Expansion)

Characteristic Equation $|\lambda I_n - A| = 0$,

Eigenvector $(\lambda I_n - A)X = 0$

Probability, RV, RP

$$P[E_i \setminus A] = \frac{P[E_i]P[A \setminus E_i]}{P[A]}$$

$$= \frac{P[E_i]P[A \setminus E_i]}{\sum_{j=1}^{n} P[E_j]P[A \setminus E_j]}$$

$$F_X(x) = \int_{-\infty}^{x^+} f_X(x)\,dx,$$

$$f_X(x) = \frac{dF_X(x)}{dx}$$

Binomial: $P[k] = \binom{N}{k} p^k (1-p)^{N-k}$

$$E[g(X)] = \int_{-\infty}^{\infty} g(X) f_X(x)\,dx$$

$$R_{XY}(m) = \sum X(n)Y(n+m)$$

Global Balance Equation:

$$p_j \sum_{i=0, i\neq j}^{\infty} P_{ji} = \sum_{i=0, i\neq j}^{\infty} p_i P_{ij}$$

Detailed Balance Equation:

$$p_i P_{ij} = p_j P_{ji}$$

II. Little's Theorem

$$N = \lambda T$$

$$N_Q = \lambda W$$

III. Poisson Distribution with Parameter m

$$p_n = \frac{e^{-m} m^n}{n!}, \quad n = 0,1,\ldots$$

mean = variance = m

IV. Exponential Distribution with Parameter λ

$$P[\tau \le s] = 1 - e^{-\lambda s}, \quad s \ge 0$$

Density: $p(\tau) = \lambda e^{-\lambda \tau}$

Mean: $= 1/\lambda$

Variance $= 1/\lambda^2$

(a) Utilization Factor (อัตราส่วนของเวลาที่ Server Busy)

$$\rho = \frac{\lambda}{\mu}$$

(b) Probability ที่จะมี n customer ในระบบ

$$p_n = \rho^n (1-\rho), \quad n = 0,1,\ldots$$

(c) จำนวนเฉลี่ยของ Customer ในระบบ

$$N = \frac{\rho}{1-\rho}$$

(d) เวลาเฉลี่ยของ Customer ในระบบ

$$T = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\mu - \lambda}$$

(e) จำนวนเฉลี่ยของ Customer ใน Queue

$$N_Q = \frac{\rho^2}{1-\rho}$$

(f) เวลาเฉลี่ยที่ต้องรอใน Queue ของ Customer

$$W = \frac{\rho}{\mu - \lambda}$$

Review [illegible]

- Chapter 1: Vector
  - Magnitude/Direction/Unit Vector
  - Direction Cosine
  - Component Vector/Position Vector
  - Dot/Cross Product and Properties
  - Equation of Line and Plane, Angle of Vectors
- Chapter 2: Matrices
  - Types of Matrices, Minor, Cofactor, Diagonal
  - Determinant by Expansion
  - Inverse of Matrix
  - Rank/Reduced Matrix (Process of Elimination)
  - Homogeneous/Non Homogeneous Linear Eq.

Review

- Chapter 3: Eigen Value/Vector/Diag
  - Eigenvalues
  - Eigenvectors
  - Diagonalization
  - Symmetric/Orthogonal Matrix
- Chapter 4: Probability
  - Conditional Probability/Bayes Rule
  - Random Variable
  - CDF/PDF/PMF; Poisson/Exponential Distribution
  - Expectation Concept
  - Mean, Mean Square, Variance
  - Joint Random Variable, Correlation/Covariance

Review

- Chapter 5: Random Process
  - Stationary/Ergodic
  - Autocorrelation/Cross Correlation
  - Counting/Poisson Process/Birth and Death Process
  - MarKov Model; Global Balanced Equation
  - Simple MarKov Model; Detail Balanced Equation
- Chapter 6: Queuing System
  - M/M/1 Concept
  - Arrival Rate/Inter Arrival Time
  - Departure Rate/Service Time
  - Utilization, P[X=k], P[X<=k]
  - Average Customer(System/Q), Time(System/Q)

CPE 332 Computer Engineering Mathematics II

Week 10

Part III, Chapter 8

Error, Transcendental Function and Power Series

Taylor Series and Function Approximation

Today Topics

- Numerical Methods
- Errors
- Transcendental Function
- Series
- Power Series
- Taylor Series
- MacLaurin Series
- Function Approximation
- HW VII
- MT Exam and Cumulative Grades

Numerical Methods

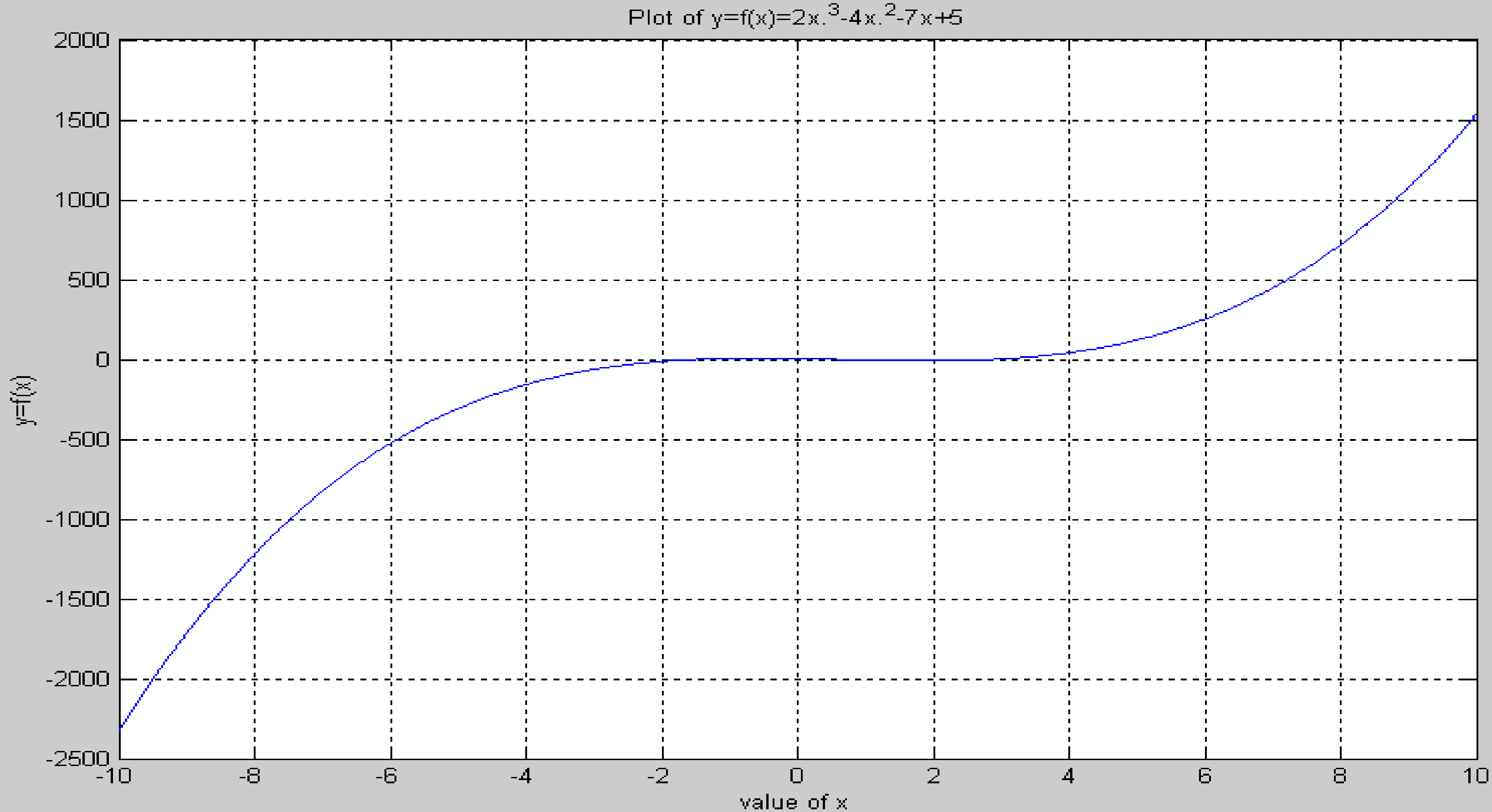
- ในวิชาคณิตศาสตร์ที่เราเรียนที่ผ่านมา การหา Solution จะประกอบด้วยการแก้สมการ ซึ่งรวมถึงการแก้สมการพิคณิต การหา Derivative การ Integrate การแก้สมการ Differential Equation รวมถึงการใช้เครื่องมือทางคณิตศาสตร์เพื่อแก้สมการ เช่น Fourier Transform, Laplace Transform หรือ Z-Transform วิธีการดังกล่าวจัดว่าเป็นการแก้ปัญหาที่เรียก Analytical Method อย่างไรก็ตาม ไม่ใช่ว่าสมการทุกสมการจะหาคำตอบได้(ที่อยู่ในรูปของ Explicit Form) ดังนั้นเราจำเป็นที่จะต้องหาวิธีอื่นในการแก้ปัญหา นั่นก็คือวิธีการที่เรียก Numerical Method

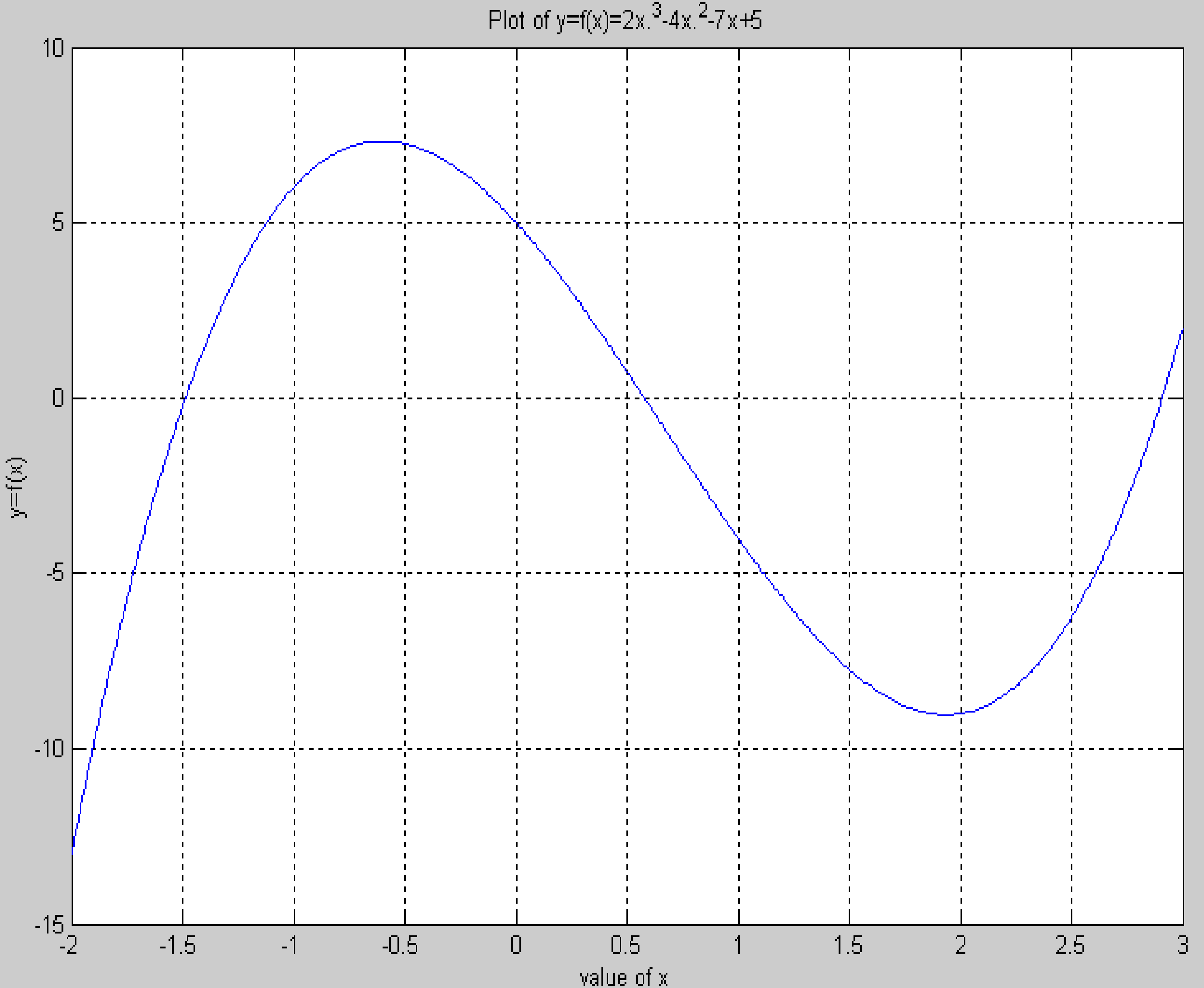
Numerical Methods

- วิธีการทาง Numerical ที่เคยใช้กันมาได้แก่วิธีการ Graphical Method และใช้บวนการ Interpolation แต่เมื่อมีการประดิษฐ์คอมพิวเตอร์ขึ้นมา ก็มีการนำคอมพิวเตอร์เข้ามาใช้ และขยายขอบเขตของวิชานี้ออกเป็นสาขาใหม่ทางคณิตศาสตร์ มีการศึกษา Algorithm และวิธีการเขียนโปรแกรม รวมถึงการวิเคราะห์ค่าความผิดพลาด(Error Analysis) และลักษณะการ Convergence และ Complexity ของ Algorithm

$Y=f(x)=2x^3-4x^2-7x+5$

- หา f(k) เช่น f(0), f(-2), f(7) ง่าย
- แก้สมการ f(x)=k จะยาก เช่น หาค่า x ที่ทำให้ f(x) = 0



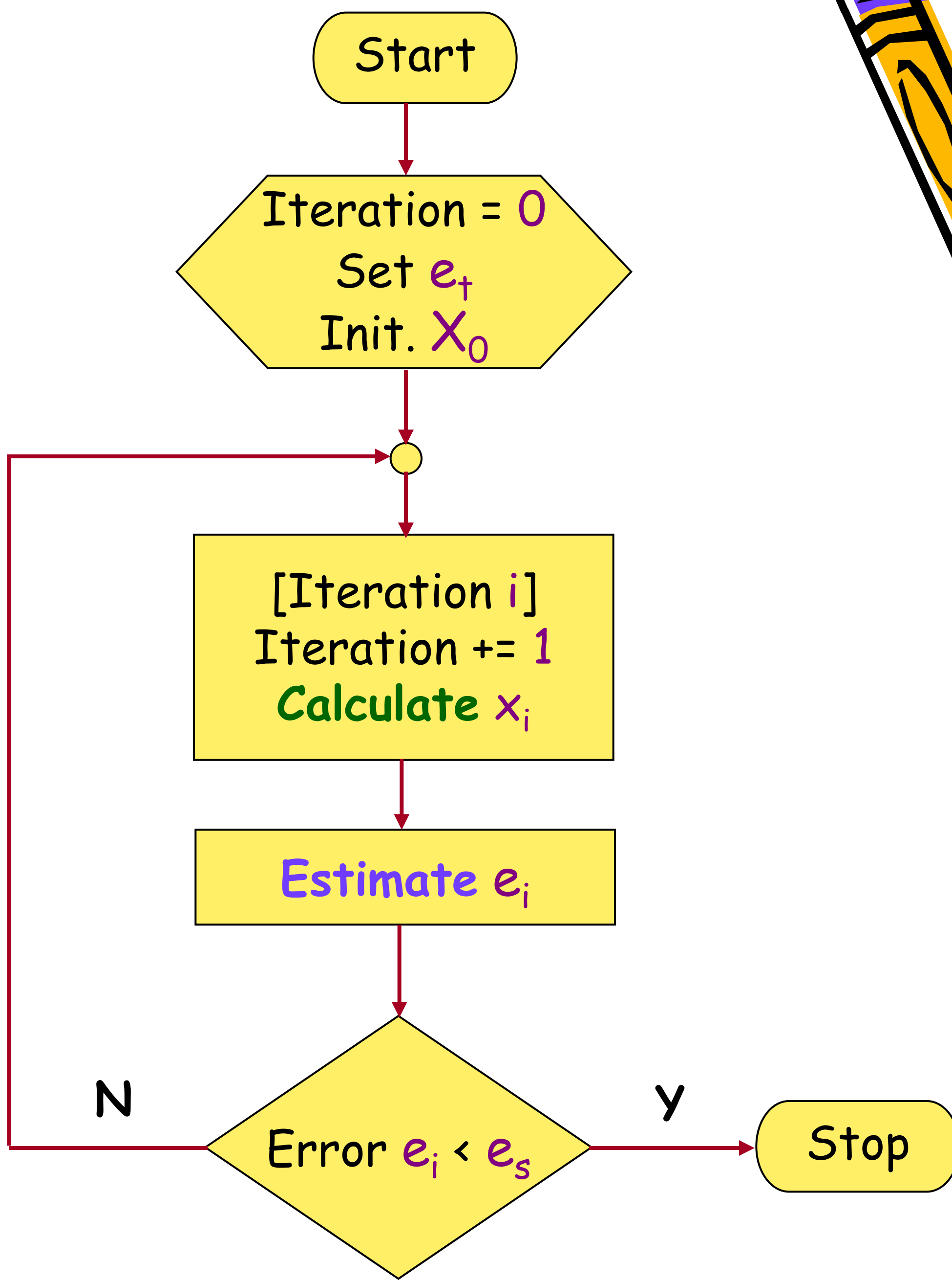

Plot of y=f(x)=2x.^3-4x.^2-7x+5
10
5
0
-5
-10
-15
y=f(x)
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
value of x

Quadratic Equation

- Polynomial Degree = 1, Straight Line
  - $y = f(x) = mx+b$; $x = (y-b)/m$
- Polynomial Degree = 2
- $y = f(x) = ax^2+bx+c$
- แก้สมการ $y = f(x) = 3$ เราได้
  - $y' = f'(x) = ax^2+bx+(c-3) = 0$
  - $x = [-b\pm\sqrt{(b2-4a(c-3))}]/2a$
- Polynimial Degree = n, n > 2
  - ไม่มีสมการโดยตรงในการแก้,
  - แต่มี Algorithm ในวิธีของ Numerical Method
    - *Algorithm จะเป็น Iterative Method*
  - คำตอบของ Numerical Method จะให้แค่ค่าประมาณ
    - *ค่าจะถูกต้องขึ้นเรื่อยๆ ใน Iteration ที่สูงขึ้น (คำนวณนานขึ้น) ถ้า Algorithm Converge*

Iterative Technique

โปรแกรมจะ Converge ถ้า $e_i < e_{i-1}$



Errors

- ในการนำคอมพิวเตอร์มาช่วยแก้ปัญหาทางด้านคณิตศาสตร์นั้น จะต้องเข้าใจก่อนว่าลักษณะการทำงานของคอมพิวเตอร์ หรือการคำนวณจะขึ้นอยู่กับวิธีการที่เราเขียนโปรแกรมและ *Algorithm* ที่ใช้ ในการคำนวณโดยใช้เครื่องคำนวณใดใดจะมี Error เกิดขึ้นเสมอ ซึ่งจะแบ่งได้เป็นสองประเภทคือ ***Round-Off Error*** และ ***Truncation Error***

Errors

- *Round-Off Error เกิดจากการเก็บตัวเลขใน Memory ของคอมพิวเตอร์นั้นจะใช้ขนาดของ Memory ที่จำกัด และขึ้นอยู่กับว่าเราให้ตัวแปรชนิดอะไร เช่น Integer, Long Integer, Float, หรือ Double*

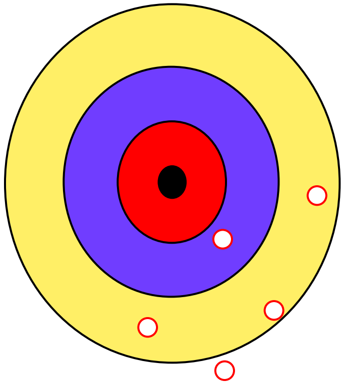
Errors

- *Truncation Error เกิดจาก Algorithm หรือวิธีที่เราให้คอมพิวเตอร์คำนวณ นั้นเป็นแค่การประมาณ จากสมการทางคณิตศาสตร์ที่เราต้องการหาค่าโดยใช้คอมพิวเตอร์เข้าช่วย และเราไม่ได้แก้สมการทางคณิตศาสตร์จริงๆ*
  - *การคำนวณ จะต้องมีการบวกกันของ Infinite Terms จึงจะได้ค่าที่ถูกต้อง ดังนั้นคอมพิวเตอร์จะคำนวณ ค่าประมาณ โดยตัดเทอมท้ายๆทิ้ง*

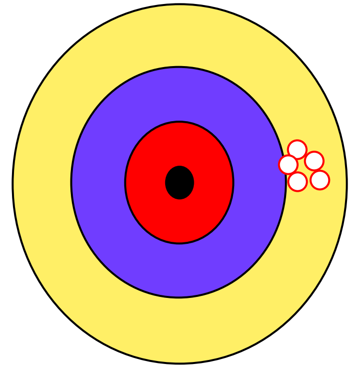
Precision, Accuracy, Significant Digit

- ดังนั้นเราจำเป็นที่จะต้องรู้ว่า Error เรามีมากแค่ไหน และพอจะยอมรับได้หรือไม่ นั้นขึ้นอยู่กับการนำการคำนวณไปใช้งาน ปกติตัวเลขที่จะไปใช้งานทางวิศวกรรมศาสตร์จะกำหนดด้วยค่า ***Significant Digit*** และจะยอมให้ความไม่แน่นอนของตกเลขตกอยู่ที่หลักท้ายเท่านั้น ดังนั้นค่าของ Significant Digit จะบ่งบอกถึงจำนวนหลักของตัวเลขที่จะนำไปใช้ได้อย่างมั่นใจ
- คำว่า ***Accuracy*** หมายถึงตัวเลขที่เราได้นั้นมีค่าใกล้เคียงกับค่าในความเป็นจริงเท่าไร แต่คำว่า ***Precision*** หมายถึงตัวเลขที่ได้มานั้นเกาะกลุ่มกันมากแค่ไหน ดังนั้นค่าของ Precision จะเป็นตัวกำหนดจำนวนของ Significant Digit ที่จะต้องใช้ และค่าของ Error จะเป็นตัวกำหนดความ Accuracy ของตัวเลข

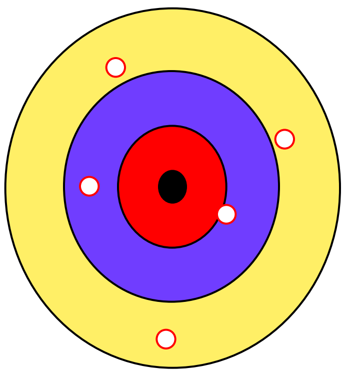
การยิงเป้า



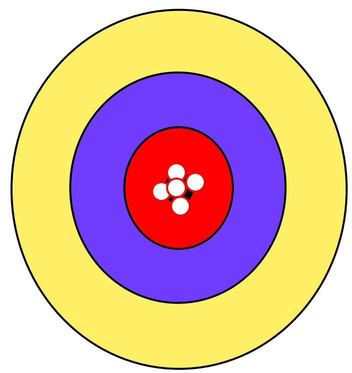
Precision = ต่ำ
Accuracy = ต่ำ



Precision = สูง
Accuracy = ต่ำ



Precision = ต่ำ
Accuracy = สูง



Precision = สูง
Accuracy = สูง

Accuracy วัดจากค่า Error ที่เกิดระหว่างค่าเฉลี่ยของ Data กับค่าจริง

Precision วัดจากค่า Variance หรือ SD ของกลุ่มของ Data

Significant Digits (Figures)

- คือตัวเลขที่มีความหมายในการกำหนดค่า Precision
  - เป็นเลขทุกตัวยกเว้นเลขศูนย์นำหน้า และศูนย์ที่ต่อท้าย
    - จำนวนศูนย์ต่อท้ายที่เป็นส่วนของ s.f. บางทีเรากำหนดโดยใช้เครื่องหมาย 'bar'
- Examples
  - 11.152 = 5 s.f.
  - 0.00879 = 3 s.f.
  - 000125600. = 6 s.f.
  - 0123400$\underline{0}$00 = 7 s.f.
- เลขทางขวามือของ s.f. จะไม่นำมาใส่ ให้ทำการ Round-Off เช่น
  - 456.389864 (6 s.f.) = 456.390
  - 1287563.94 (3 s.f.) = 12$\underline{9}$0000

Error

- คือค่าที่แตกต่างจากค่าที่แท้จริง
- เป็นตัวกำหนดค่า *Accuracy*
- แบ่งเป็น
  - *Absolute* Error, E_t
  - Relative Error (%จากค่าจริง), e_t
- นอกจากนี้ยังมี
  - Estimated Error, e_a
    - ค่าประมาณของ Error (Relative Error)

Absolute Error

$$\text{True Value} = \text{Approximation} + \text{Error}$$

หรือ

$$\text{Error} = E_t = \text{True Value - Approximation}$$

Relative Error

$$e_t = \frac{Error}{\text{True Value}} \times 100\,\% = \frac{\text{True Value - Approximation}}{\text{True Value}} \times 100\,\%$$

Error Estimation

- Convergence
- Divergence

$$e_a = \frac{\text{Approximate Error}}{\text{Approximation}} \times 100\,\%$$

$$e_a = \frac{\text{Present Approximation - Previous Approximation}}{\text{Present Approximation}} \times 100\,\%$$

Mean Square Error(MSE)

- ในการเปรียบเทียบความแตกต่างระหว่างค่าที่แท้จริงกับค่าที่ประมาณได้ สำหรับกลุ่มของตัวอย่าง เรามักจะใช้ค่าเฉลี่ยของ Error ในรูปของค่าเฉลี่ยของกำลังสองของ Error ในแต่ละคู่ เรียก Mean Square Error
  - ถ้าให้ Y เป็นค่าที่แท้จริง และ $\hat{Y}$ เป็นค่าที่ประมาณได้สำหรับคู่ของตัวอย่าง N ตัวอย่าง

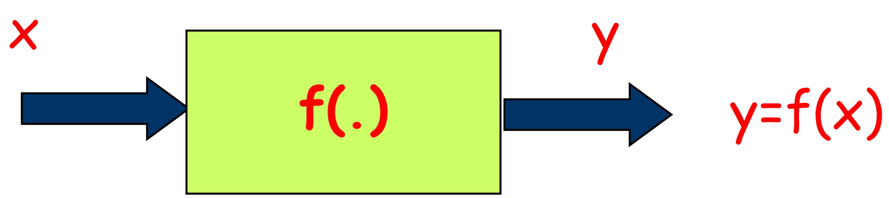
$$MSE = \frac{1}{N}\sum_{i=0}^{N-1}(\hat{Y}_i - Y_i)^2$$

  - ค่า RMSE หรือค่า Root Mean Square Error คือค่า Square Root ของค่า MSE

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(\hat{Y}_i - Y_i)^2}$$

Functions

- เป็นความสัมพันธ์แสดง Set ของ Input และ Set ของ Output
  - Function สามารถรับค่า Input ได้หลาย Set (หลายตัวแปร)
  - การ Mapping เป็นทางเดียว
    - ถ้า Map กลับจะเป็น อีก Function ที่เป็น Inverse Function กับตัวเดิม
    - Inverse อาจจะไม่มีสำหรับบาง Function



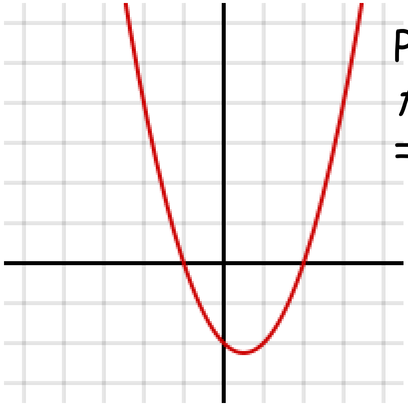
$x=f^{-1}(y)$ อาจจะไม่มี

Polynomial Functions

- เป็นรูปแบบของ Function ที่ง่ายที่สุด
- Polynomial function Degree 'n' จะอยู่ในรูปของ(One Argument)

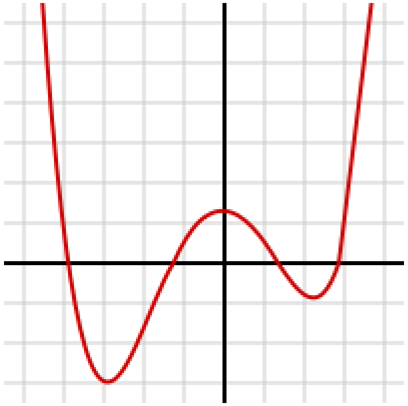
$$y = f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$$

$$= \sum_{i=0}^{n} a_i x^i = a_0 + \sum_{i=1}^{n} a_i x^i$$

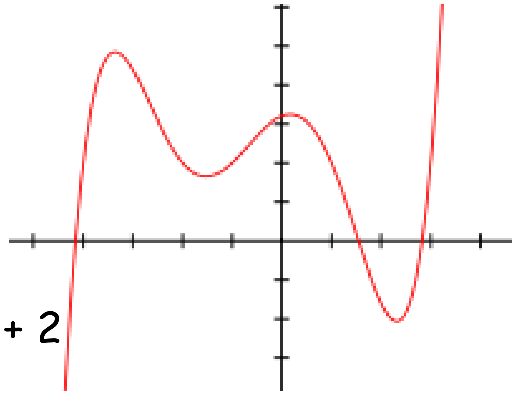


Polynomial of degree 2:
$f(x) = x^2 - x - 2$
$= (x+1)(x-2)$

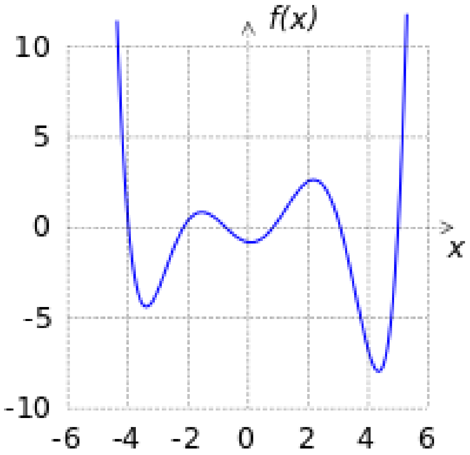
Polynomial of degree 3:
$f(x) = x^3/4 + 3x^2/4 - 3x/2 - 2$
$= 1/4\ (x+4)(x+1)(x-2)$



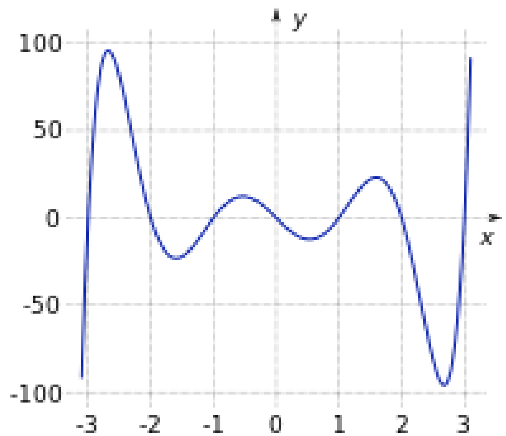
Polynomial of degree 4:
$f(x) = 1/14\ (x+4)(x+1)(x-1)(x-3) + 0.5$



Polynomial of degree 5:
$f(x) = 1/20\ (x+4)(x+2)(x+1)(x-1)(x-3) + 2$



เอามาจาก
http://en.wikipedia.org/wiki/Polynomial



Polynomial of degree 6:
$f(x) = 1/30\ (x+3.5)(x+2)(x+1)(x-1)(x-3)(x-4) + 2$

Polynomial of degree 7:
$f(x) = (x-3)(x-2)(x-1)(x)(x+1)(x+2)(x+3)$

Transcendental Function

- เป็น Function ที่ไม่ใช่ Algebraic
  - Algebraic function คือ Function ที่สามารถนิยามได้จาก Root ของสมการ Polynomial
- Transcendental Function ไม่สามารถเขียนในรูป Solution ของ Polynomial
  - Exponential Function
  - Logarithm
  - Trigonometric Functions
- ด้วยเหตุผลนี้ การหาค่าของ Function อาจจะต้องใช้วิธีการประมาณค่าแทน

Transcendental Examples

$$f_1(x) = x^{\pi}$$

$$f_2(x) = c^{x},\ c \neq 0,1$$

$$f_3(x) = x^{x}$$

$$f_4(x) = x^{\frac{1}{x}}$$

$$f_5(x) = \log_c x,\ c \neq 0,1$$

$$f_6(x) = \sin x$$

การประมาณค่าของ Function

- ในส่วนนี้เราจะ Review เรื่อง
  - Infinite Series
  - Power Series
  - Taylor Series
  - Maclaurin Series
  - การประมาณค่าโดยใช้ Series

Series และ Infinite Series

- Series คือผลรวมของเทอมใน Sequence ในกรณีของ Infinite Series นั้น เทอมที่จะต้องนำมาบวกกันจะไม่มีที่สิ้นสุด
- ถ้าให้แต่ละเทอมใน Sequence เป็น a_n ดังนั้น Series คือ $a_1+a_2+a_3+\ldots$ เช่น

$$S = \sum_{n=1}^{\infty} a_n = a_1 + a_2 + a_3 + \cdots$$

$$a_n = \frac{1}{2^n}; S = \sum_{n=1}^{\infty} \frac{1}{2^n} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots$$

NOTE: บางครั้ง Index ของ Summation อาจจะเริ่มที่ 0 ก็ได้

Convergence of Series

- Series จะ Converge ก็ต่อเมื่อผลรวมของ Term สามารถหาค่าได้
  - กล่าวคือผลรวมของเทอมจนถึง N มี Limit

$$S = \sum_{n=0}^{\infty} a_n = \lim_{N\to\infty} S_N = \lim_{N\to\infty} \sum_{n=0}^{N} a_n$$

- ถ้า Series ไม่ Converge มันจะ Diverge

Power Series

- Power Series เป็น Infinite Series ในรูปของ (c = Constant, a_n = coefficient, x เป็น Variable ที่อยู่รอบๆ c)

$$f(x)=\sum_{n=0}^{\infty}a_n(x-c)^n$$

  - Power Series ที่สำคัญคือ Taylor Series

- ในกรณีที่ c=0 เราได้

$$f(x)=\sum_{n=0}^{\infty}a_nx^n=a_0+a_1x+a_2x^2+a_3x^3+\cdots$$

เช่นในกรณีของ Maclaurin Series

Taylor Series/Maclaurin Series

- ให้ f(x) เป็น Function ที่สามารถหา Derivative ได้อย่างไม่จำกัดในช่วงใกล้เคียงกับ a ดังนั้น Taylor Series ของ f(x) คือ Power Series ในรูปของ

$$f(a)+\frac{f'(a)}{1!}(x-a)+\frac{f''(a)}{2!}(x-a)^2+\frac{f^{(3)}(a)}{3!}(x-a)^3+\cdots=\sum_{n=0}^{\infty}\frac{f^{(n)}(a)}{n!}(x-a)^n$$

- ในกรณีที่ a = 0 บางครั้งเราเรียก Maclaurin Series

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

9.2 Taylor's Theorem

ทฤษฎีบท

ถ้า Function f และค่า $n+1$ Derivative แรกของมันมีความต่อเนื่องในช่วงของ a และ x ดังนั้นค่าของ Function ที่จุด x สามารถแสดงได้โดย

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

โดย R_n เรียก Remainder และให้นิยามว่าเป็น

$$R_n = \int_a^x \frac{(x-t)^n}{n!} f^{(n+1)}(t)\,dt$$

ซึ่งค่าของ Remainder ดังกล่าวยังสามารถเขียนในรูปที่เรียก Derivative Form หรือ Lagrange Form ดังนี้

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-a)^{n+1}$$

สมการข้างบนรู้จักกันในนาม Taylor Series หรือ Taylor's Formula ซึ่งถ้าไม่รวม R_n สมการที่เหลือก็คือค่าประมาณของ $f(x)$ ที่มีลักษณะเป็น Polynomial กล่าวอีกนัยหนึ่งก็คือ ค่าของ Function ใดๆที่มีคุณสมบัติตามที่กำหนด สามารถประมาณได้จากสมการของ Polynomial

การละเทอม R_n ออกจากสมการ จะส่งผลให้การคำนวณค่าประมาณนั้นมี Error นี่คือที่มาของ Truncation Error ที่เรากล่าวในบทที่ 7 ดังนั้นการหาค่าของ Function ขึ้นอยู่ว่าเราต้องการ Significant Digit แค่ไหน ซึ่งจะเป็นตัวจะกำหนดว่าเราต้องใช้กี่เทอมใน Polynomial และจะลงเอยด้วย Degree ของ Polynomial และ Derivative ของ Function ที่จุด a ที่ต้องใช้

ถ้าให้ a เป็นจุดของ Function ที่เรารู้ค่าของมัน และ Derivative ของมัน และสมมุติว่าอยู่ที่ x_i เราสามารถใช้ Taylor Series ประมาณค่าของ Function ที่จุดใหม่ กล่าวคือ x_{i+1} โดยกำหนดขนาดของ Step $h = x_{i+1} - x_i$ ให้มีค่าเท่าๆกัน เช่น

$$Zero-Order\ Approximation: f(x_{i+1}) \cong f(x_i)$$

$$First-Order\ Approximation: f(x_{i+1}) \cong f(x_i) + f'(x_i)h$$

$$Second-Order\ Approximation: f(x_{i+1}) \cong f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2$$

และโดยทั่วไปเราสามารถเขียน

$$n-Order\ Approximation: f(x_{i+1}) \cong f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \cdots + \frac{f^{(n)}(x_i)}{n!}h^n$$

โดยที่ค่า Remainder สามารถแสดงได้เป็น $R_n = \dfrac{f^{(n+1)}(\xi)}{(n+1)!}h^{n+1}$

Example 1

- $$f(x)=f(a)+f'(a)(x-a)+\frac{f''(a)}{2!}(x-a)^2+\frac{f^{(3)}(a)}{3!}(x-a)^3+\frac{f^{(4)}(a)}{4!}(x-a)^4+\dots$$

$$f(x)=f(0)+f'(0)x+\frac{f''(0)}{2!}x^2+\frac{f^{(3)}(0)}{3!}x^3+\frac{f^{(4)}(0)}{4!}x^4+..$$

Suppose $f(x)=e^x,\ f'(x)=e^x,\dots,\ f^{(n)}(x)=e^x \Rightarrow f^{(n)}(0)=1$

We have $f(x)=e^x=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}+\dots=\sum_{k=0}^{\infty}\frac{x^k}{k!}$

Example find $f(2)=e^2$

Second Order Approximation : $e^2\approx 1+2+\frac{4}{2}=5$

Forth Order Approximation : $e^2\approx 1+2+\frac{4}{2}+\frac{8}{6}+\frac{16}{24}=7$

Sixth Order Approximation : $e^2\approx 1+2+\frac{4}{2}+\frac{8}{6}+\frac{16}{24}+\frac{32}{120}+\frac{64}{720}=7.356$

Eighth Order Approximation : $e^2\approx 1+2+\frac{4}{2}+\frac{8}{6}+\frac{16}{24}+\frac{32}{120}+\frac{64}{720}+\frac{128}{5040}+\frac{256}{40320}=7.3873$

Actual Value : $e^2=7.389056099$

นักศึกษาหาค่าประมาณของ e^{-2} ถึง 4 Significant Digit ได้หรือไม่

Example 2:

$$f(x) = f(0) + f'(a)x + \frac{f''(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + \frac{f^{(4)}(0)}{4!}x^4 + ..$$

Suppose $f(x) = \sin x,\ f'(x) = \cos x,\ f''(x) = -\sin x,\ f'''(x) = -\cos x,\ f^{(4)} = \sin(x), ...$

We have $f(x) = \sin x = 0 + x + 0 - \frac{x^3}{3!} + 0 + \frac{x^5}{5!} + 0 - \frac{x^7}{7!} + ... = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$

Example find $f(0.5) = \sin 0.5$ radian

Second Order Approximation : $\sin 0.5 \approx 0.5$

Forth Order Approximation : $\sin 0.5 \approx 0.5 - \frac{0.5^3}{3!} = 0.47916667$

Sixth Order Approximation : $\sin 0.5 \approx 0.5 - \frac{0.5^3}{3!} + \frac{0.5^5}{5!} = 0.47942708$

Eighth Order Approximation : $\sin 0.5 \approx 0.5 - \frac{0.5^3}{3!} + \frac{0.5^5}{5!} - \frac{0.5^7}{7!} = 0.479425533$

Actual Value : $\sin 0.5 = 0.479425538$

นักศึกษาหาค่าประมาณของ cos 0.5 ถึง 8 Significant Digit ได้หรือไม่

Taylor (Maclaurin) Series

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

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

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

$$\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n;\ |x| < 1$$

$$\frac{1-x^{m+1}}{1-x} = \sum_{n=0}^{m} x^n;\ x \ne 1$$

$$\frac{x}{(1-x)^2} = \sum_{n=0}^{\infty} n x^n;\ |x| < 1$$

$$\sqrt{1+x} = \sum_{n=0}^{\infty} \frac{(-1)^n (2n)!}{(1-2n)(n!)^2 (4n)} x^n = 1 + \tfrac{1}{2}x - \tfrac{1}{8}x^2 + \tfrac{1}{16}x^3 - \tfrac{5}{128}x^4 + \cdots;\ |x| < 1$$

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

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

MATLAB Program

- จะสาธิตการเขียน Function โดยใช้ MATLAB
  - เขียน Function ที่รับค่า Vector x และ Vector y return z1 และ z2; x=-3:.1:3; y = -3:.1:3
  - คำนวณค่า z1 = 2sin(x^2+y^2)/(x^2+y^2)
  - คำนวณค่า z2 = xsiny-ycosx
  - mesh (x,y,z1), (x,y,z2), (x,z1), (y,z2) และ (z1,z2)
  - Surf function, shading modes
  - Contour plot
- ศึกษาวิธีการเขียน Function ใน MATLAB โดยดูจาก Tutorial 4-5

END OF WEEK 10

- Download HW 7
- Next Week Chapter 9: Zeros of Functions

MATLAB Program

```
function [z1,z2]=test(x,y)

% function [z1,z2]=test(x,y)
% Test matlab program calculate z1=f(x,y)=sin(x^2+y^2)/(x^2+y^2)
% and z2=f(x,y)=xsiny-ysinx

n=length(x);
m=length(y);
z1=zeros(n,m);
z2=zeros(n,m);
for i = 1:n
    for j= 1:m
        t=x(i)^2+y(j)^2;
        if (t ~= 0)
            z1(i,j)=sin(t)/t;
        else
            z1(i,j)=1;
        end
        z2(i,j)=x(i)*sin(y(j))-y(j)*cos(x(i));
    end
end
```

MATLAB Program

- function view2(x,y,z,az)
- mesh(x,y,z);
- view(0,az);
- for i = 0:10:360
-         view(i,az);
-         pause(0.05);
- end

MATLAB Program

- function view3(az)
- view(0,az);
- for i = 0:10:360
-         view(i,az);
-         pause(0.05);
- end

CPE 332 Computer Engineering Mathematics II

Week 11

Part III, Chapter 9

Roots of Equations

Today Topics

- Roots of Equation
- Bracketing Method
  - Bisection Method
  - False Position Method
- Open Method
  - Simple One-Point Iteration
  - Newton-Ralphson Method
  - Secant Method
  - Multiple Roots Problem
  - Modified Method

Roots of Equations

ในบทนี้เราจะศึกษา Algorithm ที่ใช้ในการหาราก หรือ Root ของ Function โดยที่เราจะกล่าวเฉพาะ Function ที่ประกอบด้วยหนึ่งตัวแปรเท่านั้น ซึ่งเขียนได้ในรูปของ $y = f(x)$ และรากของ Function $f(x)$ สามารถหาได้จากการแก้สมการ $f(x) = 0$

พึงเข้าใจว่ารากของสมการอาจจะมีได้มากกว่าหนึ่งตัว และอาจจะมีรากที่ซ้ำกันได้หรือที่เรียก Multiple Roots อย่างไรก็ตาม ในการศึกษาขั้นต้นนี้ เราจะสมมุติว่า Root ทุกตัวจะมีค่าไม่ซ้ำกัน และเราพอจะรู้ว่ารากที่เราต้องการอยู่ที่บริเวณไหน(การหาค่าประมาณของรากของสมการสามารถทำได้โดยการ Plot Graph และดูที่เมื่อไรเส้น $f(x)$ ตัดกับแกน x หรือมีค่าเท่ากับศูนย์)

Zeroes of Functions

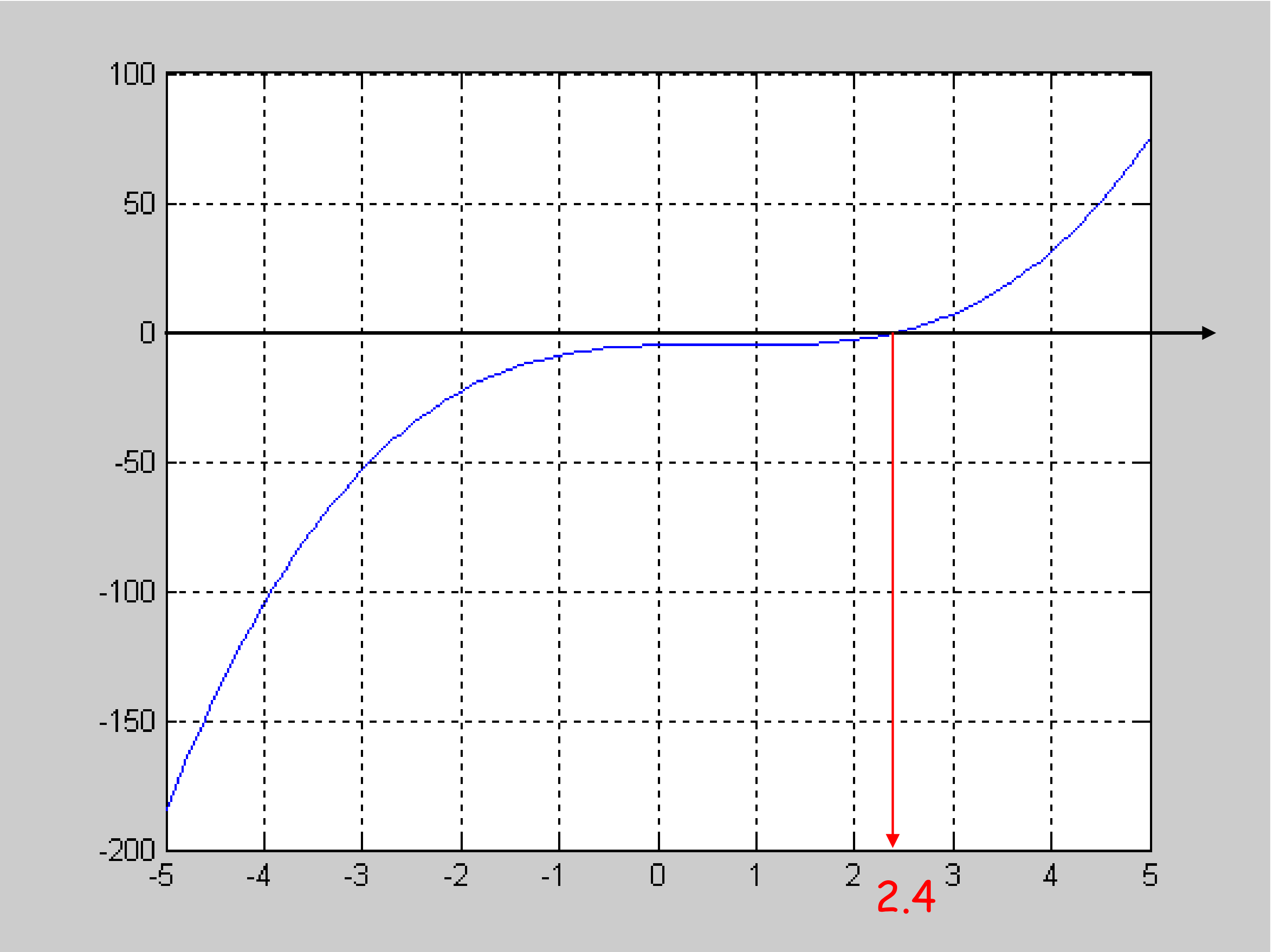
- กำหนด Function y=f(x)
  - Zeroes ของ Function คือค่าของ x ที่ทำให้ f(x)=0 หรือค่า y=0
  - คือราก(root) ของสมการ f(x)=0 นั่นเอง
- ตัวอย่าง $y=x^3-2x^2+x-5$
  - สมการ Polynomial, degree = 3
  - ถ้าเกิน Quadratic (Degree = 2) การหารากจะทำได้ยาก, ไม่มีวิธีทาง *Analytic Method* โดยตรง เหมือน $y=Ax^2+Bx+C=0$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Zeroes of Functions

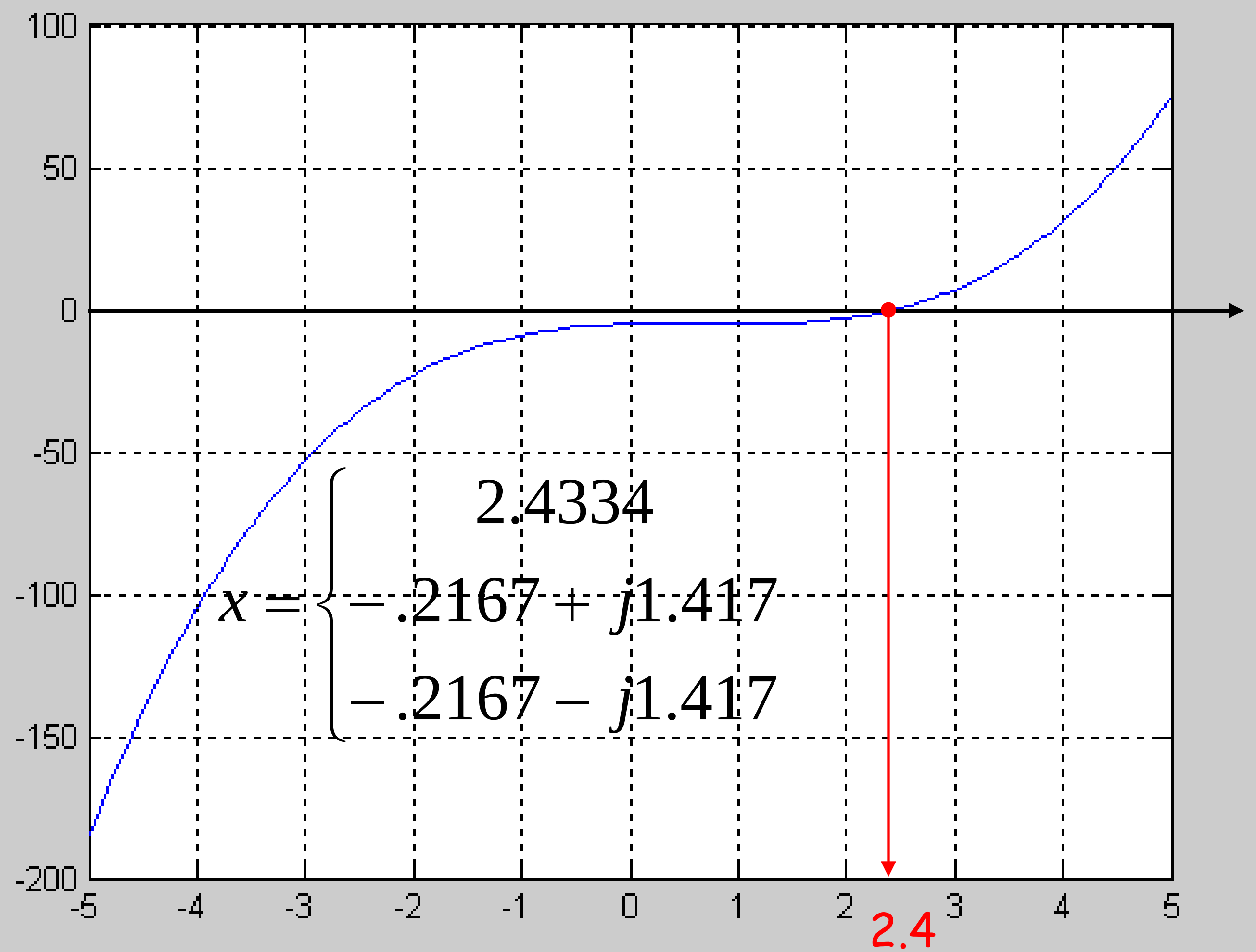
- ตัวอย่าง $y=x^3-2x^2+x-5$
  - วิธีทาง Numerical ที่เราเคยเรียนมาในวิชาเรขาคณิตย์ คือ Plot Graph และหารากของสมการจาก Graph
    - 1. แต่วิธีนี้จะให้ค่าที่หยาบ
    - 2. กรณีที่เป็น Complex Root จะหาไม่ได้ จาก Graph

Zeroes of Functions
$y=x^3-2x^2+x-5$



Zeroes of Functions

$y=x^3-2x^2+x-5$



Roots of Equation

- วิธีที่จะกล่าวต่อไป
  - Numerical Method = Algorithm
  - หา Zero Crossing (รากที่เป็นค่าจริง)
  - สามารถได้คำตอบให้ถูกต้องด้วย Significant Digit มากเท่าที่เราต้องการ
    - ต้อง Run Algorithm นานขึ้น
    - Algorithm จะต้อง Converge
  - เป็น Iterative Method
    - ถ้า Algorithm Converge, แต่ละ Iteration ที่ Run จะให้คำตอบที่ถูกต้องขึ้นเรื่อยๆ
    - จะช้าหรือเร็วขึ้นอยู่กับ Convergence Rate ของแต่ละ Algorithm

Methods

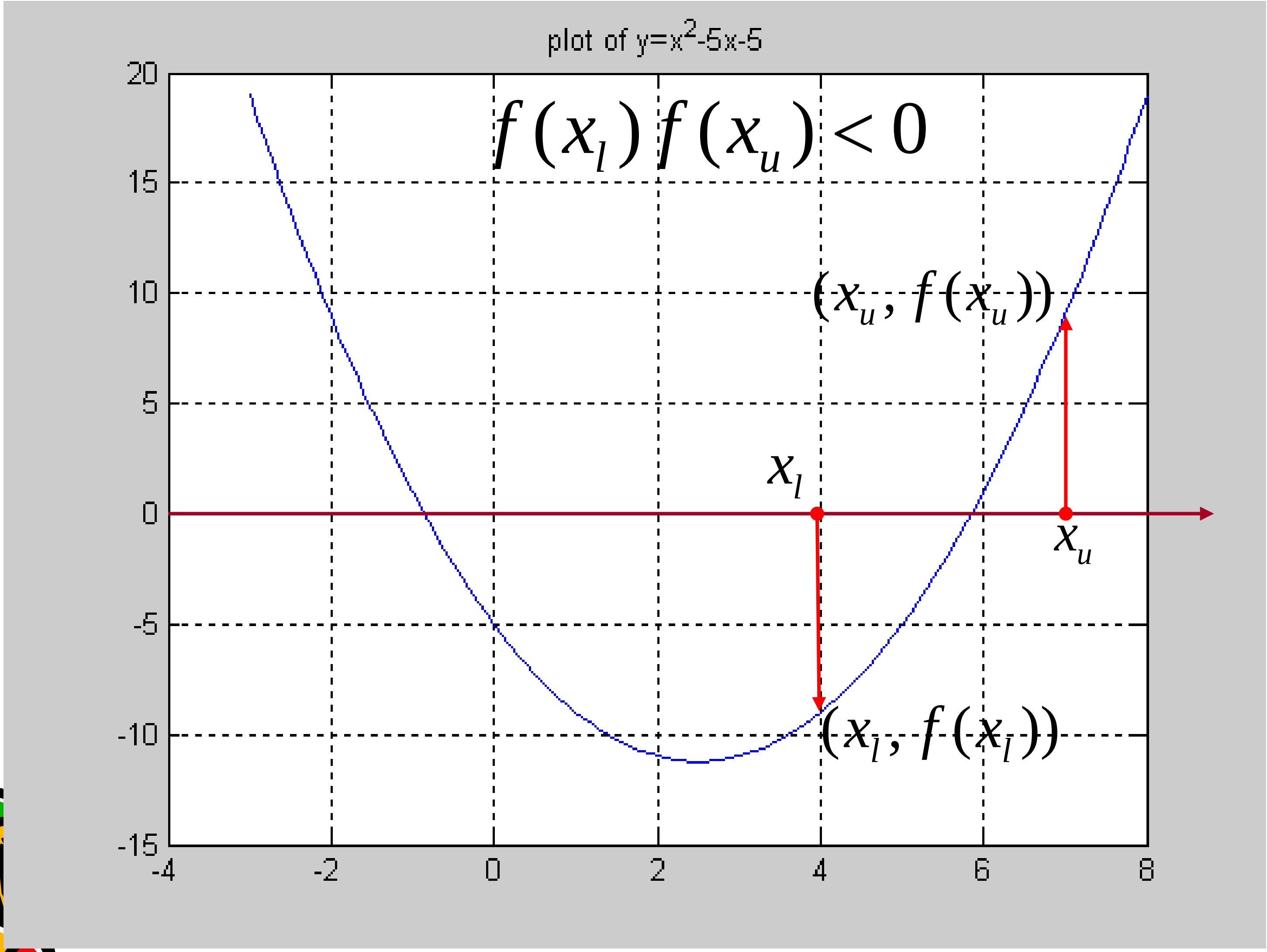
- Bracketing Method
  - Bisection
  - False Position
- Open Method
  - Simple one-point Iteration
  - Newton-Ralphson
  - Secant Method

กรรมวิธีในการหารากของสมการจะแบ่งออกเป็นสองกลุ่มใหญ่ๆ กลุ่มแรกเรียก Bracketing Method เป็นวิธีที่เมื่อเรารู้ว่าคำตอบจะต้องอยู่ในช่วงใดช่วงหนึ่งของค่า x ดังนั้นวิธีนี้จะมีการกำหนดค่าสองค่า คือช่วงที่เราจะหารากของสมการ วิธีนี้เราสามารถแน่ใจได้ว่าโปรแกรมจะ Converge อีกวิธีหนึ่งเรียก Open Method ซึ่งเราไม่จำเป็นจะต้องรู้ล่วงหน้าถึงช่วงที่คำตอบจะต้องอยู่ เราอาจจะสมมุติค่าตั้งต้นขึ้นมาค่าหนึ่งสำหรับ Algorithm และ Algorithm มันจะเริ่มทำงานจากค่าตั้งต้นนี้ จนถึงคำตอบที่ต้องการ อย่างไรก็ตาม วิธีนี้จะไม่ Guarantee ว่าโปรแกรมจะ Converge ขึ้นอยู่กับการสมมุติค่าตั้งต้นของเรา

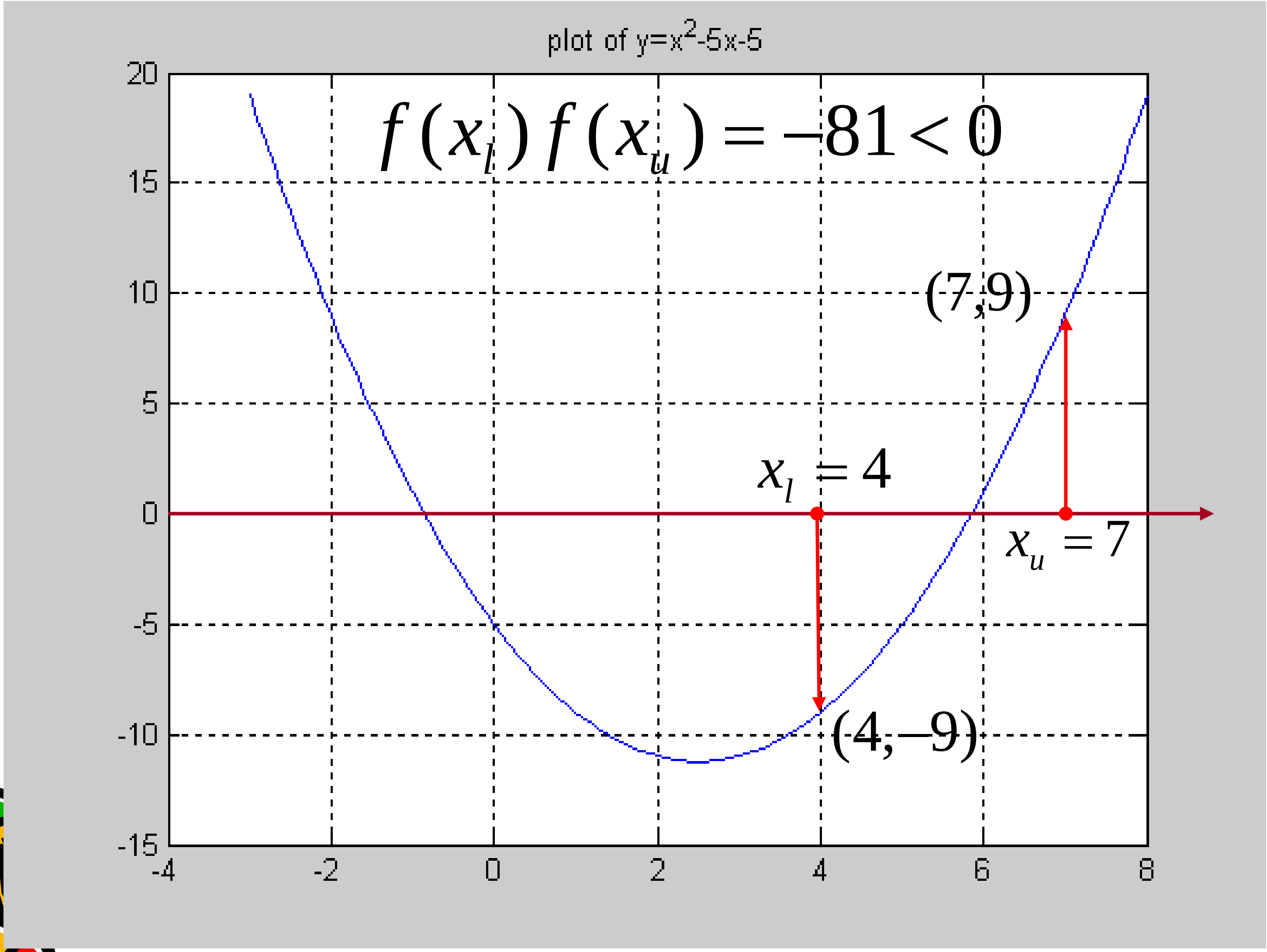
Bracketing Method

- ใช้หลักความจริงที่ว่า เมื่อ $f(x)=0$ มันจะต้องเปลี่ยนเครื่องหมาย
  - ดังนั้น $f(x^-)f(x^+) < 0$ เมื่อ $f(x)=0$
- เราเริ่มจาก สองค่าของ x คือ x_l และ x_u ที่มีคุณสมบัติ $f(x_l)f(x_u) < 0$
  - อย่างน้อยต้องมีคำตอบหนึ่งอยู่ในช่วงนี้
- Algorithm จะ Search หาคำตอบในช่วง Bracket นี้ โดยจะลดขนาดของ Bracket ลงเรื่อยๆ

Bracketing Method



Bracketing Method



Bracketing Method:Bisection

จากที่กล่าวมาแล้ว รากของสมการที่จุด x ที่ค่าของ $f(x)=0$ ดังนั้นที่จุดนี้ $f(x)$ จะมีค่าเปลี่ยนจากบวกเป็นลบ หรือเปลี่ยนจากลบเป็นบวก ถ้าเรากำหนดสองจุดคือ x_l และ x_u จากนั้นคำนวณหา $f(x_l)$ และ $f(x_u)$ และถ้า

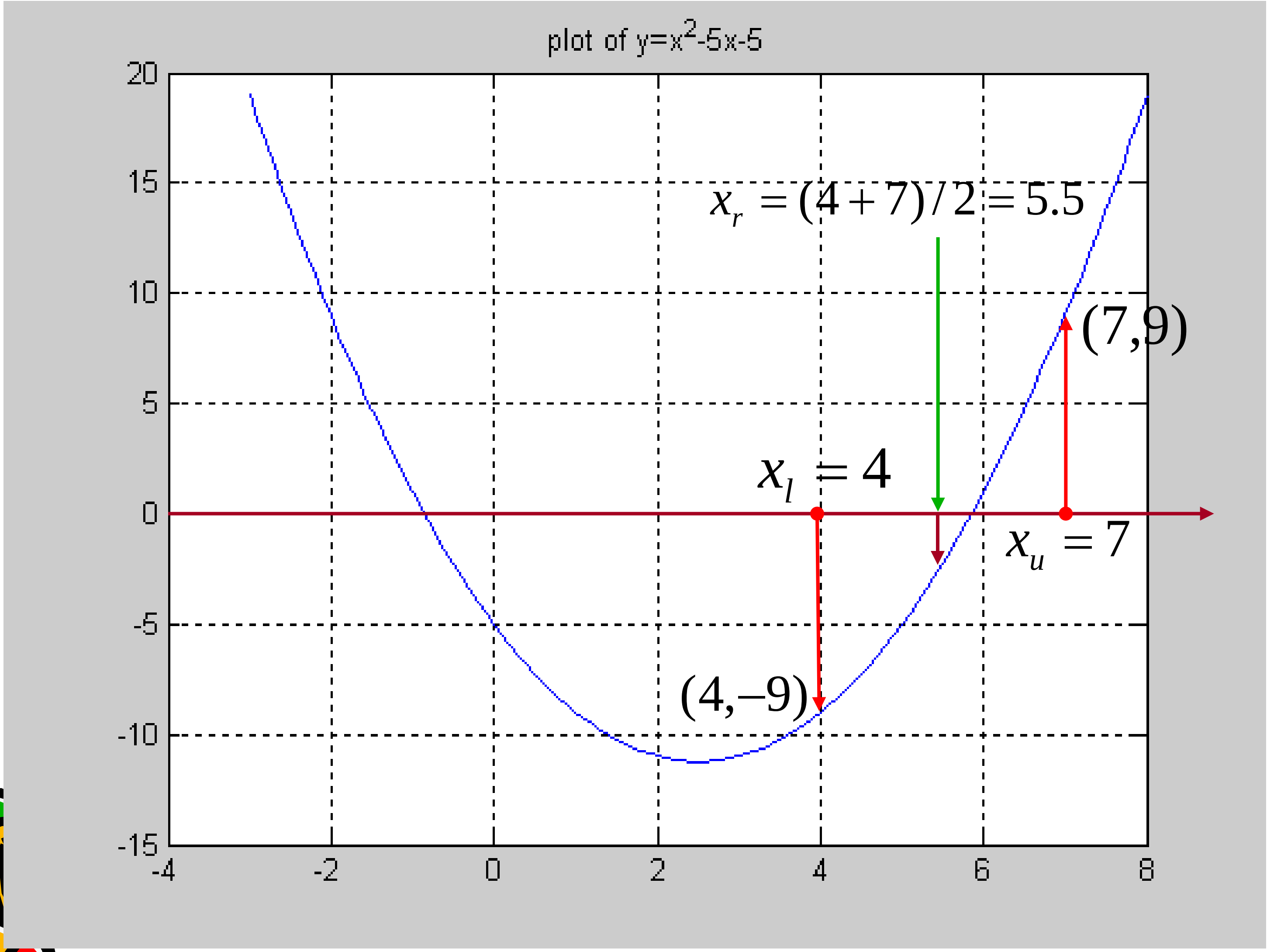
$$f(x_l)f(x_u)<0$$

หมายถึงว่าในช่วง $[x_l, x_u]$ Function จะมีการเปลี่ยนเครื่องหมาย และจะต้องผ่านจุด $f(x)=0$ กล่าวอีกนัยหนึ่งก็คือจะต้องมีอย่างน้อยหนึ่งรากของสมการในช่วงนี้

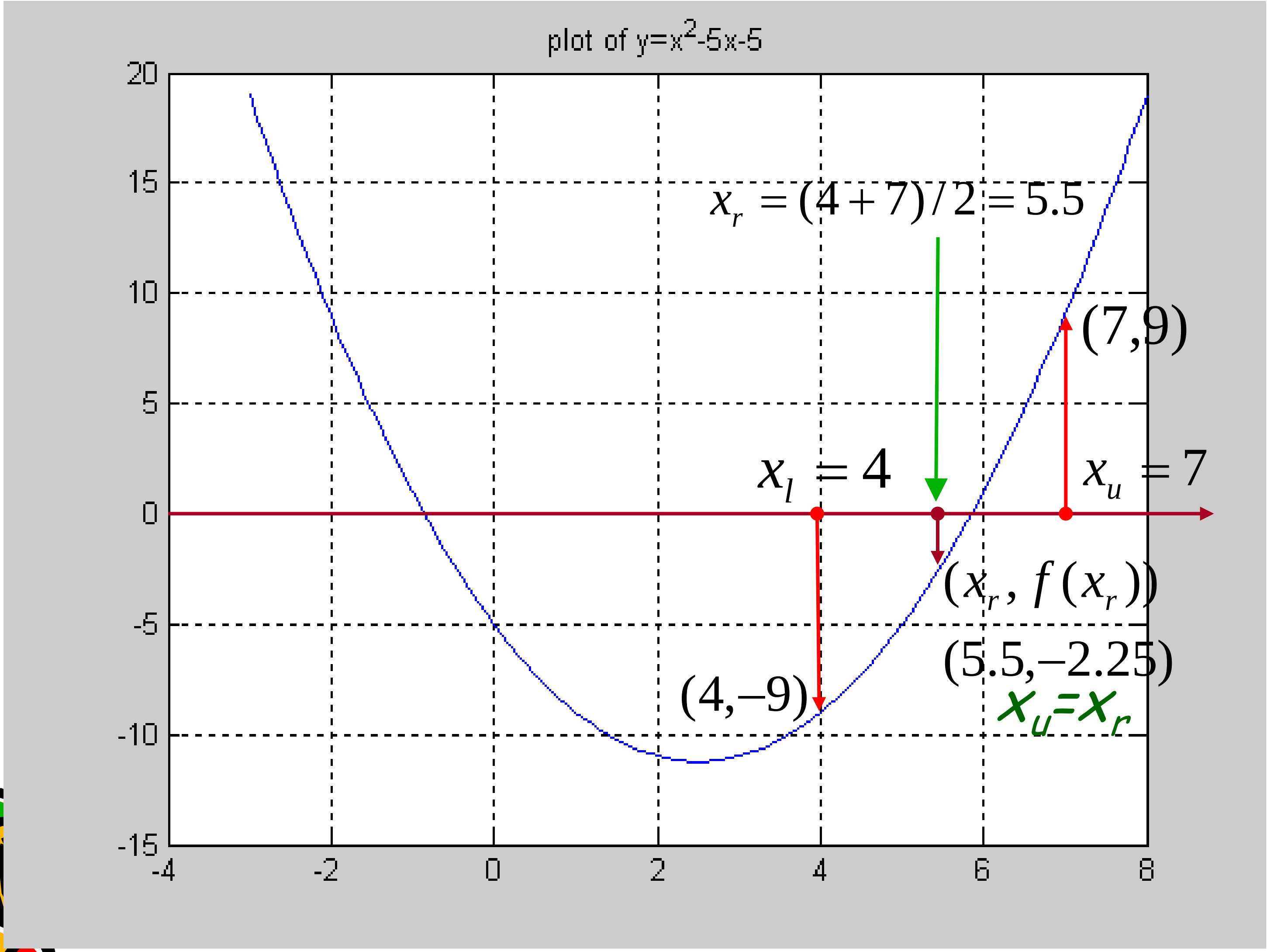
จากนั้นเราลดขนาดช่วงให้เล็กลงโดยแบ่งครึ่งที่จุด $x_r = \dfrac{x_l + x_u}{2}$ และหาว่า Function มีการเปลี่ยนเครื่องหมายในช่วงไหน เราจะลดขนาดของช่วงที่ต้องการหาไปได้ครึ่งหนึ่ง และทำเช่นนี้ต่อไปเรื่อยๆจนได้คำตอบที่พอใจ

หลักการดังกล่าวก็คือหลักการของ Bisection Method หรือ Bolzano's Method และสามารถสรุปเป็น Algorithm ได้สามขั้นตอนดังนี้

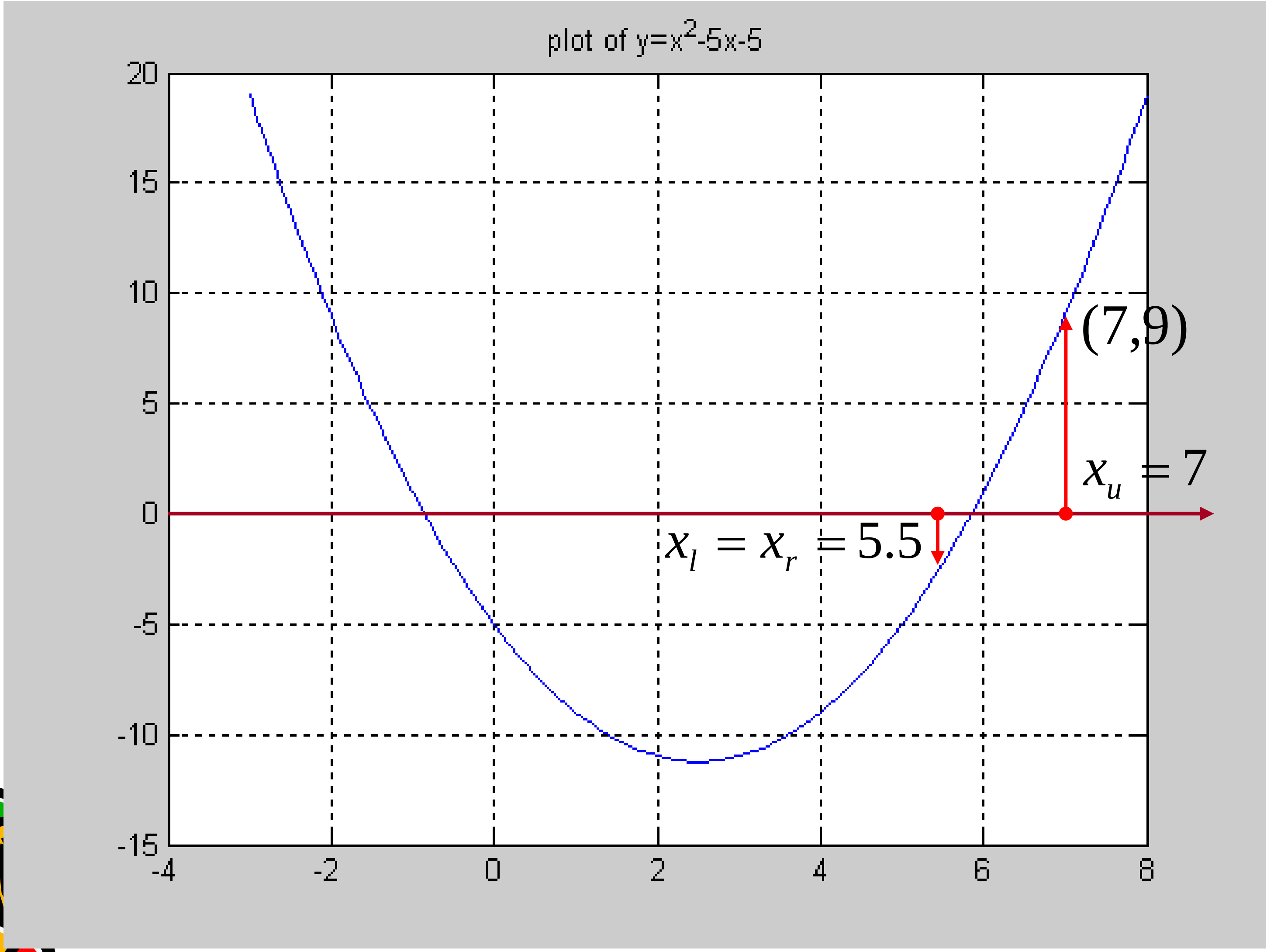
Bracketing Method



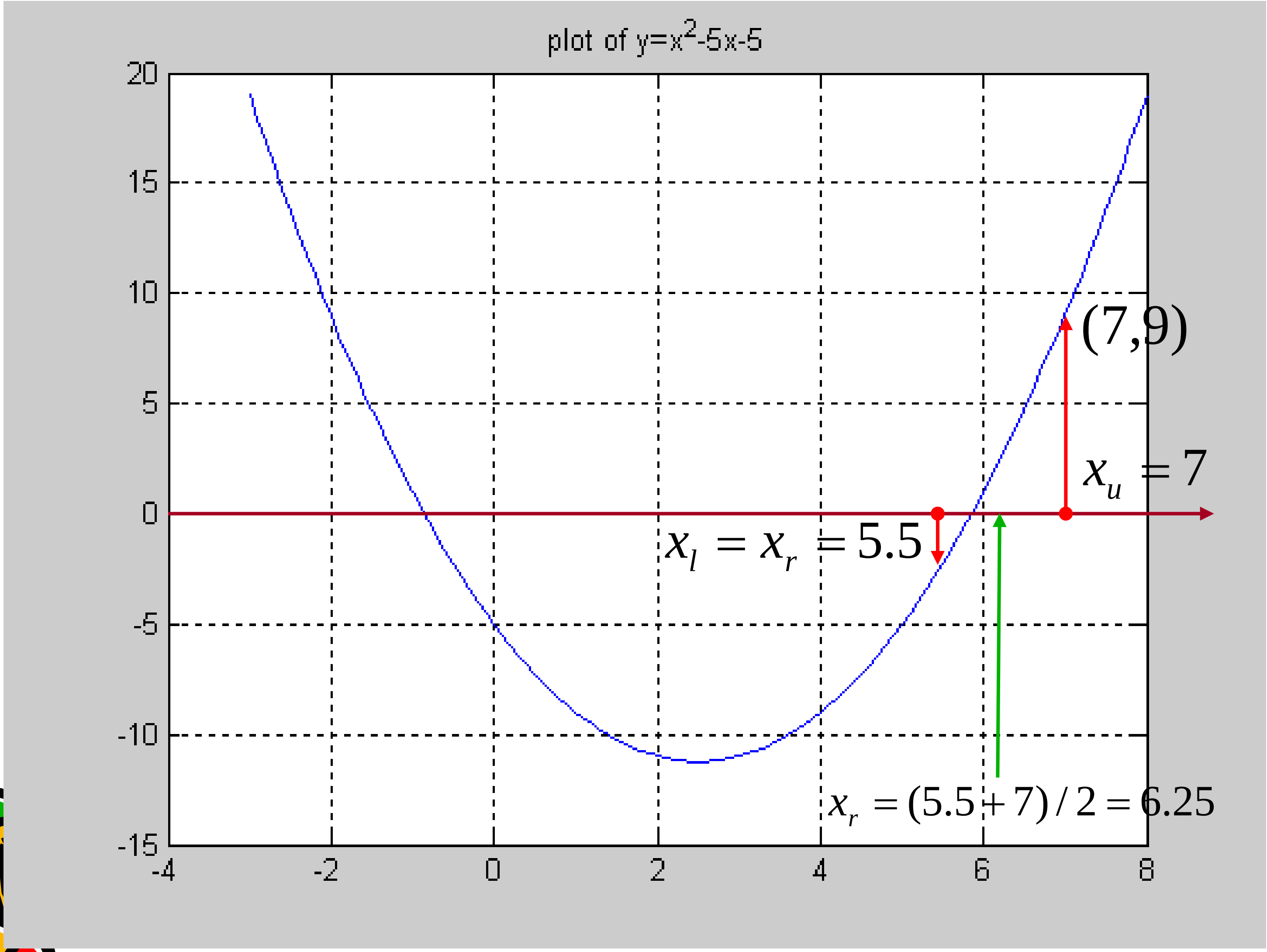
Bracketing Method



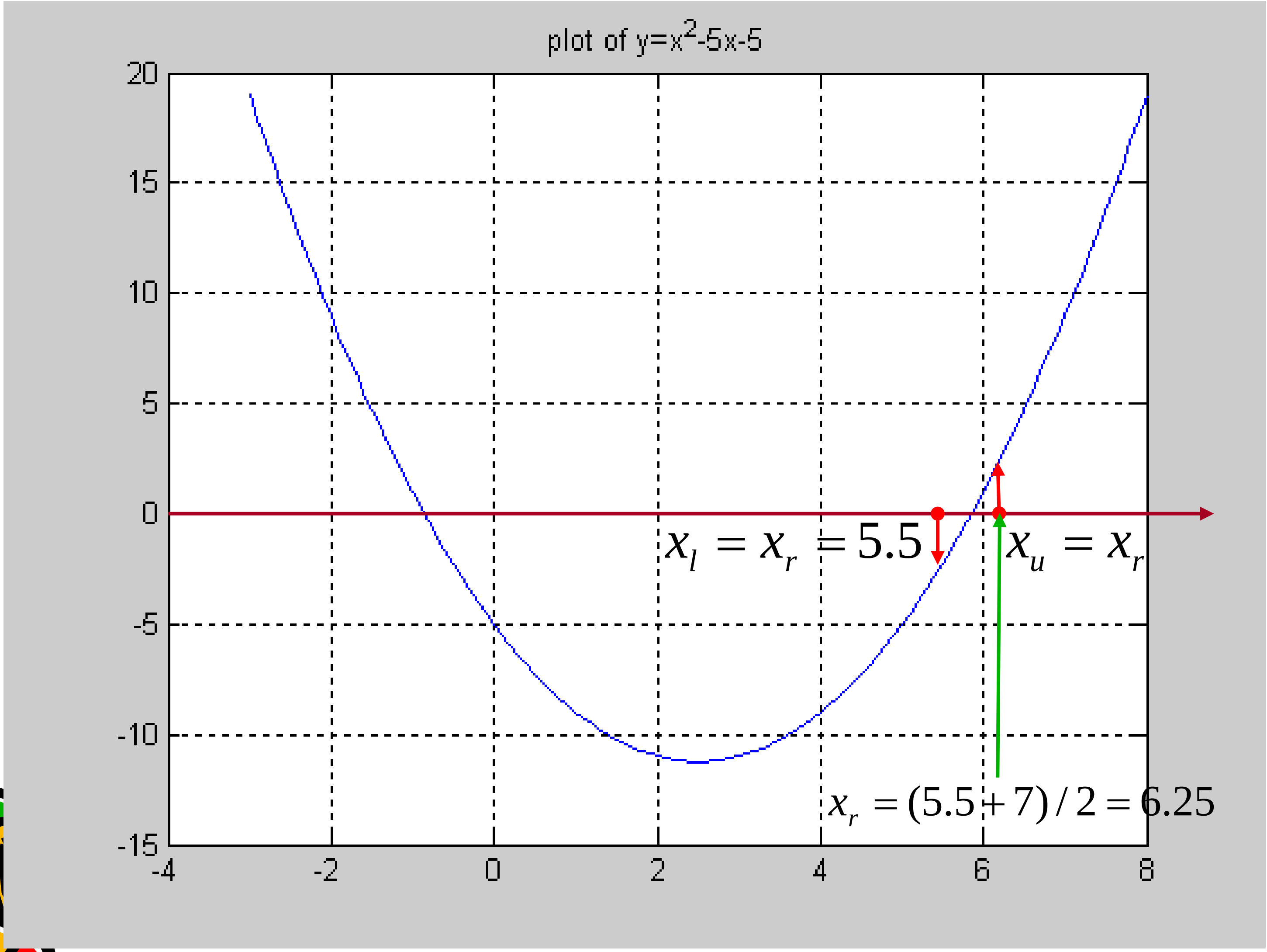
Bracketing Method



Bracketing Method



Bracketing Method



Bracketing Method:Bisection

Algorithm:

Step 1: ทำการเดาจุดสองจุดคือค่า x_l และค่า x_u สมมุติว่าค่า x_l เป็นค่าที่ต่ำกว่า จากนั้นทดสอบว่า $f(x_l)f(x_u) < 0$ ถ้าไม่ใช่ให้หาจุดใหม่ ซึ่งจากขั้นตอนนี้ เรารู้ว่ารากจะต้องอยู่ในช่วงนี้

Step 2: ทำการประมาณค่า Root ที่ต้องการที่จุดกึ่งกลางระหว่างสองค่าใน Step 1 โดยคำนวณ $x_r = \dfrac{x_l + x_u}{2}$

Step 3: หาว่าตอนนี้ Root ที่ต้องการอยู่ในครึ่งไหนดังนี้

3.1 ถ้า $f(x_l)f(x_r) < 0$ แสดงว่า Root ที่ต้องการอยู่ในครึ่งล่าง ให้ตั้ง $x_u = x_r$ และกลับไปทำ Step 2

3.2 ถ้า $f(x_l)f(x_r) > 0$ แสดงว่า Root ที่ต้องการอยู่ในครึ่งบน ให้ตั้ง $x_l = x_r$ และกลับไปทำ Step 2

3.3 ถ้า $f(x_l)f(x_r) = 0$ แสดงว่าคำตอบที่ต้องการเท่ากับ x_r

จากสมการของ Error Approximation เราสามารถหาค่า Error Estimate ของ Bisection Method ได้จาก

$$|e_a| = \left|\frac{x_r^{New} - x_r^{Old}}{x_r^{New}}\right| \times 100\%$$

Bracketing Method:Bisection

พึงสังเกตว่า แม้ว่าวิธีนี้จะให้ค่า $|e_a|$ ที่ลดลงเรื่อยๆ แต่ค่า $|e_t|$ จะมีการแปรผันขึ้นลงได้ ดังนั้นค่า Approximate Error จะไม่สอดคล้องกับ True Error ทีเดียวนัก ตามปกติเวลาเรา Run โปรแกรม เราจำเป็นต้องตั้ง Condition ที่โปรแกรมจะหยุดทำงาน ซึ่งมักจะกำหนดเป็นค่า Threshold Error, e_s โดยที่แต่ละ Iteration จะมีการตรวจสอบค่า e_a และเมื่อใดมันมีค่าต่ำกว่า e_s (ค่า Absolute) โปรแกรมจะหยุดทำงาน และให้คำตอบออกมา อีกอย่างหนึ่งก็คือ $|e_a|$ ที่ได้จะสูงกว่า $|e_t|$ เสมอ

Bracketing Method:Bisection

Example 7.1: สมการสำหรับหาความเร็วของนักโดดร่มสามารถแสดงได้ดังนี้ $v = \frac{gm}{c}[1 - e^{-(c/m)t}]$ โดยที่ v คือความเร็ว เป็น Dependent Variable, t คือเวลา เป็น Independent Variable, g เป็น Gravitational Constant, c เป็นค่าคงที่ของการฉุดของร่ม(Drag Coefficient) และ m คือมวลของนักโดดร่มและร่ม

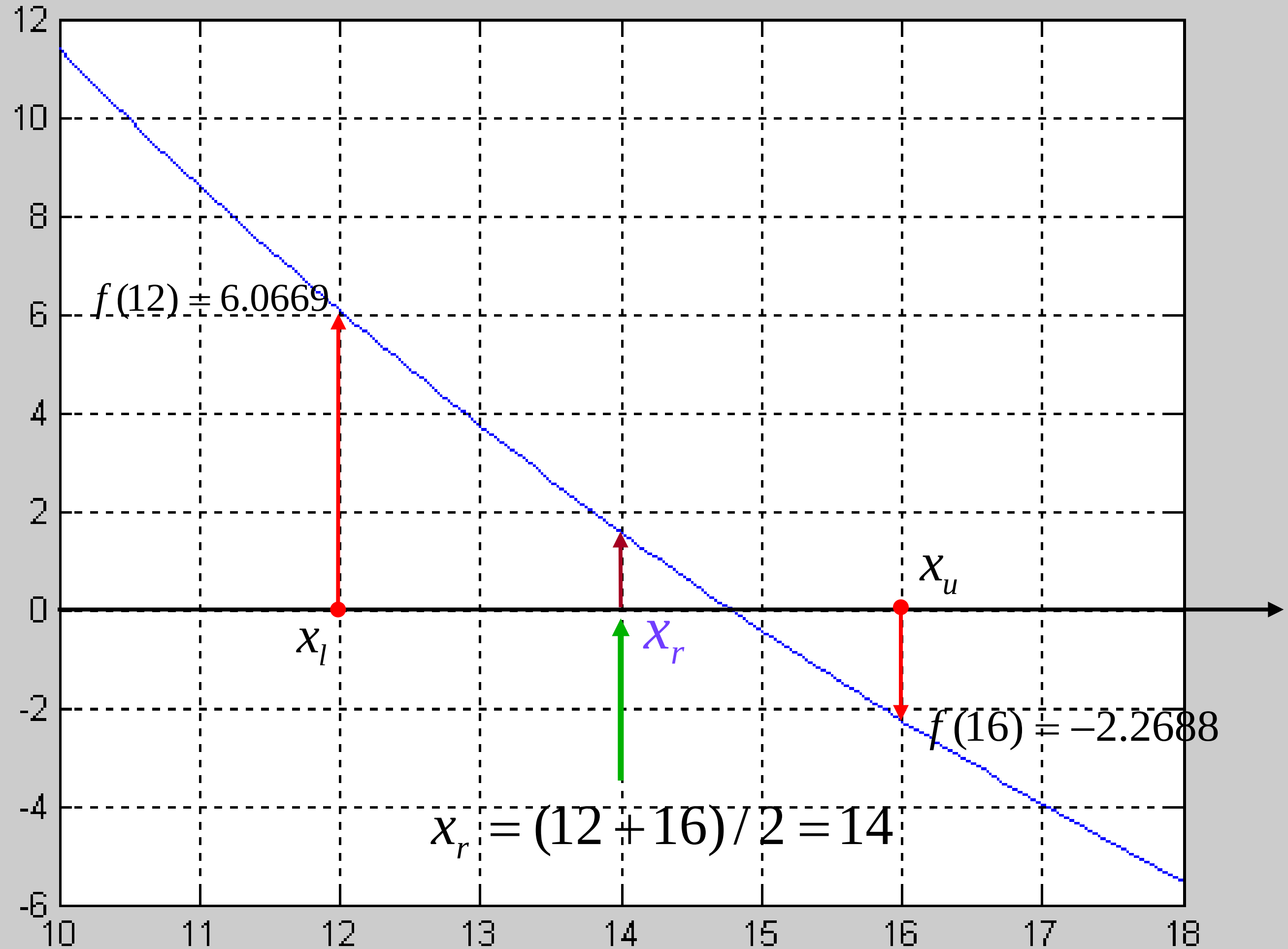
ในการออกแบบร่มชูชีพ เราต้องการหาค่า Drag Coefficient ที่จะคงความเร็วของนักโดดร่มให้ได้ในช่วงเวลาที่กำหนด ซึ่งการแก้สมการหาค่า c จะไม่สามารถทำได้ทางคณิตศาสตร์ ในการนี้เราจะเอาคอมพิวเตอร์เข้ามาช่วย โดยการแก้สมการ

$$f(c) = \frac{gm}{c}[1 - e^{-(c/m)t}] - v = 0$$

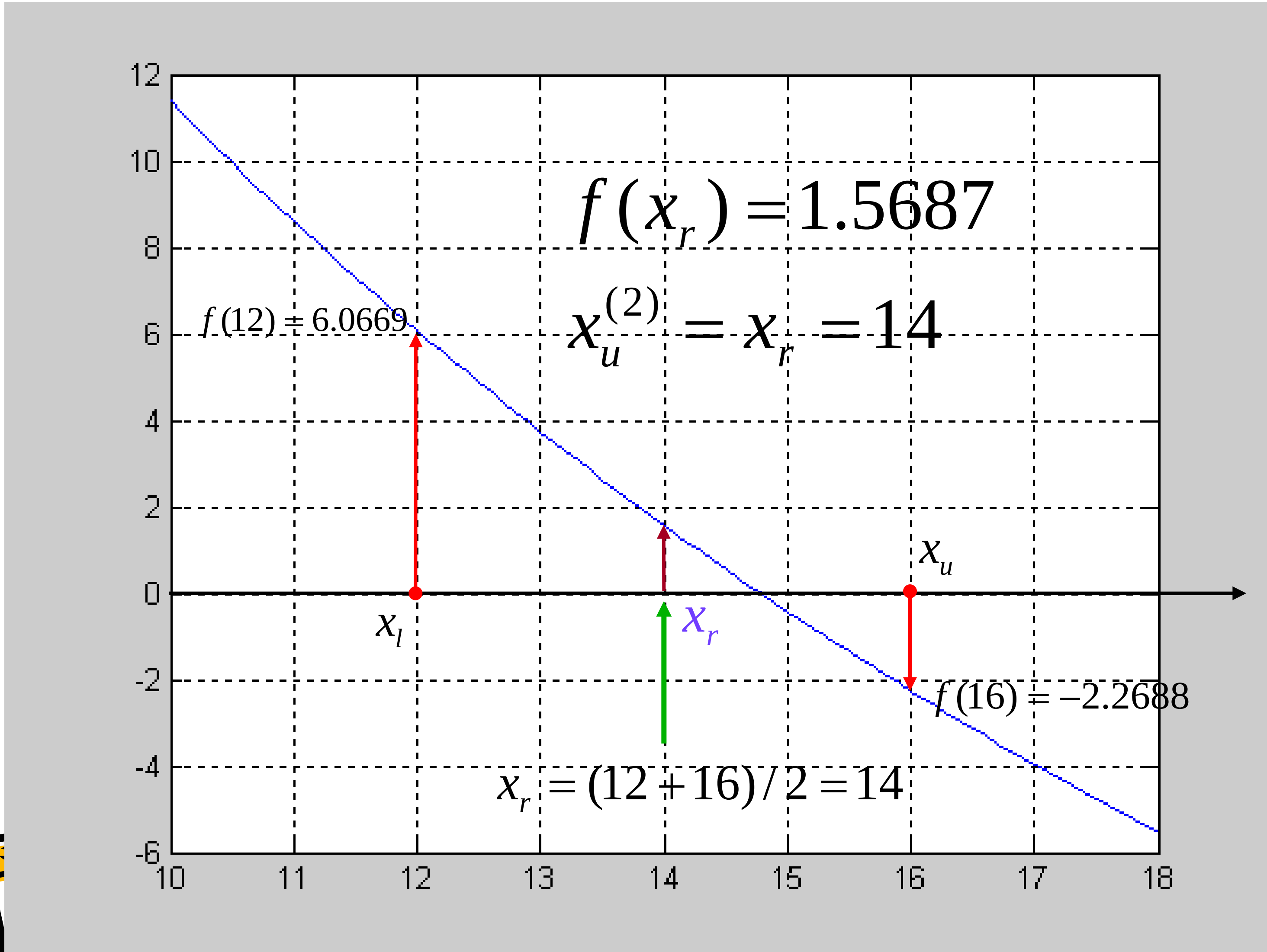
และหาค่า c ออกมา ซึ่งก็คือรากของสมการดังกล่าว

สมมุติว่า $m = 68.1\ kg$, $g = 9.8\ m/s^2$ เราต้องการหาค่า c ที่จะให้นักโดดร่มคงความเร็วอยู่ที่ $40\ m/s$ ในเวลา 10 วินาที และค่า c ควรมีค่าอยู่ระหว่าง 12 และ 16

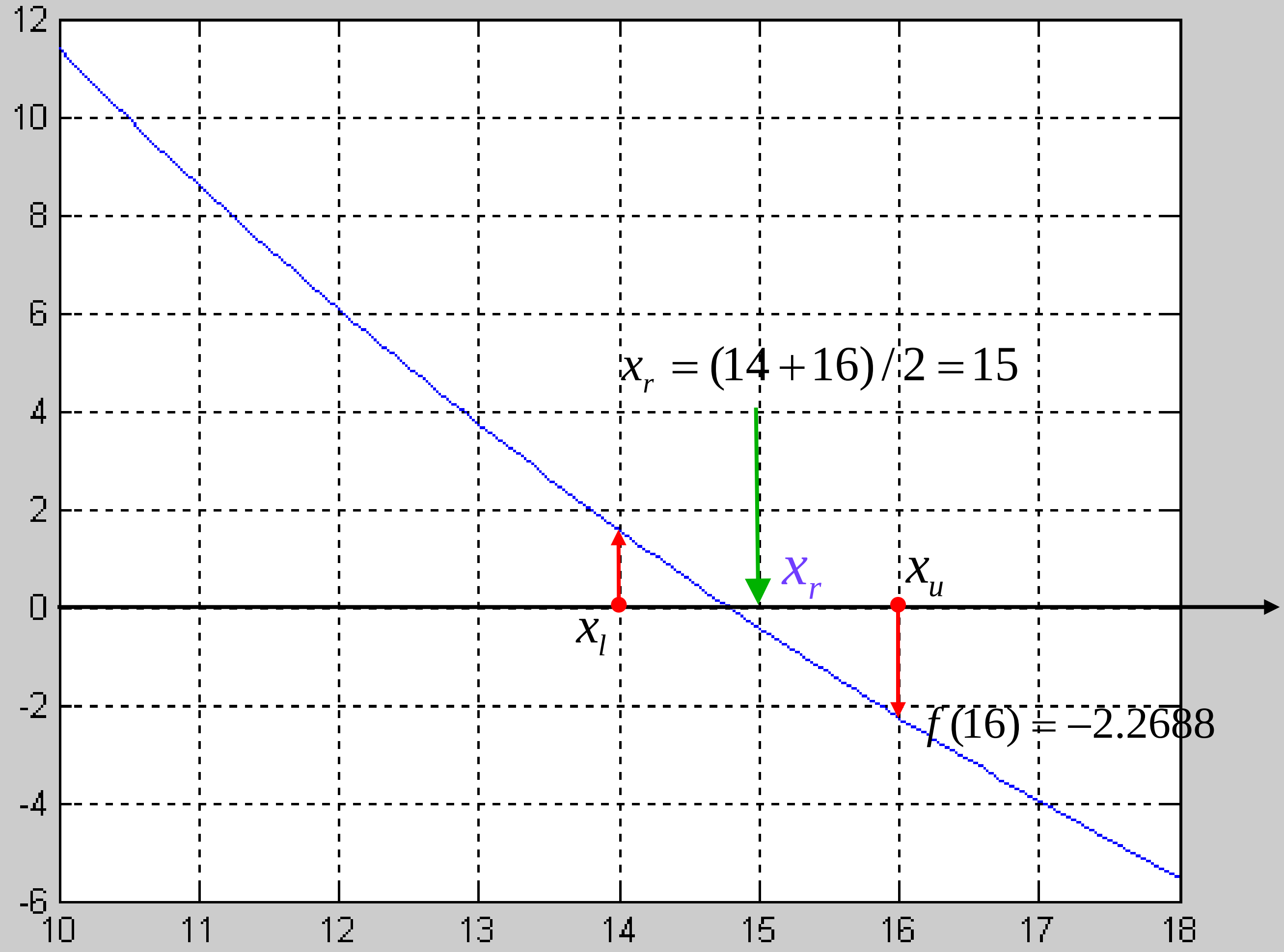
$$f(c) = \frac{667.38}{c}(1 - e^{-10c/68.1}) - 40 = 0$$



12
10
8
6
4
2
0
-2
-4
-6
10
11
12
13
14
15
16
17
18
$f(12) = 6.0669$
x_l
x_r
x_u
$f(16) = -2.2688$
$x_r = (12+16)/2 = 14$



$f(x_r) = 1.5687$
$x_u^{(2)} = x_r = 14$
$f(12) = 6.0669$
x_l
x_r
x_u
$f(16) = -2.2688$
$x_r = (12+16)/2 = 14$
12
10
8
6
4
2
0
-2
-4
-6
10
11
12
13
14
15
16
17
18



12
10
8
6
4
2
0
-2
-4
-6
10
11
12
13
14
15
16
17
18
$x_r = (14+16)/2 = 15$
x_r
x_u
x_l
$f(16) = -2.2688$

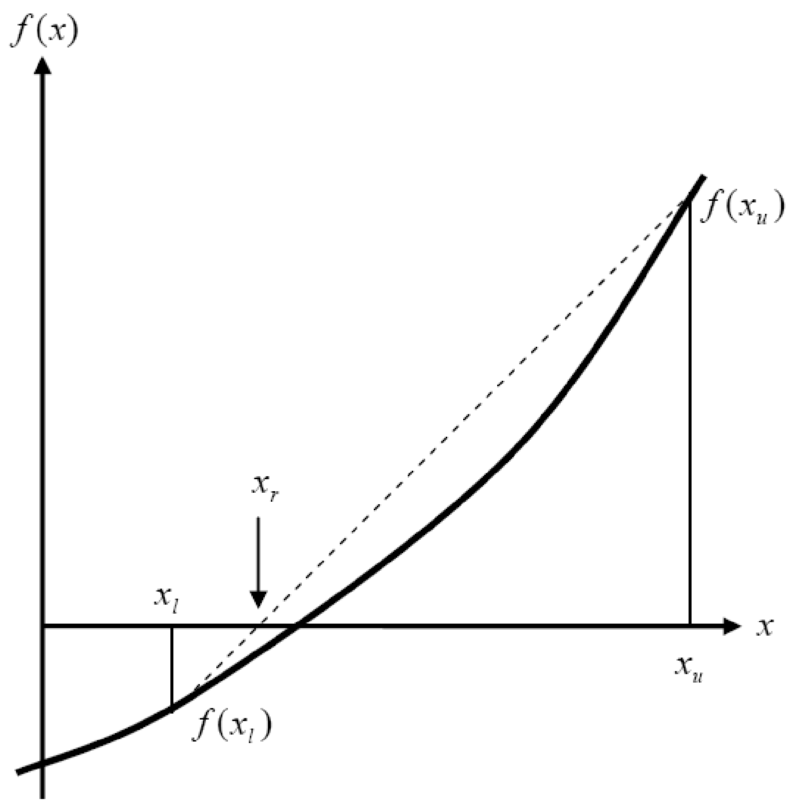
Bracketing Method: Bisection

สังเกตว่า $f(12)f(16) < 0$ และคำตอบรวมทั้งค่า Error สรุปได้ดังตาราง(ค่า e_t คำนวณจากค่า True Value)

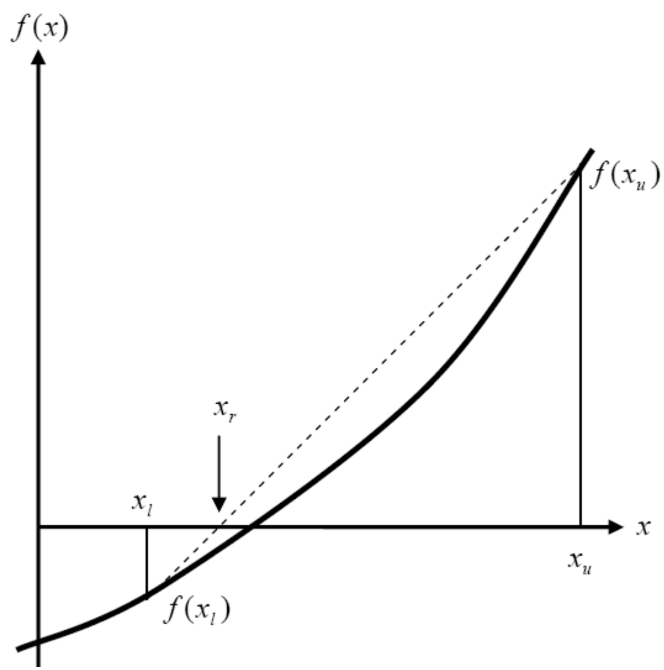
Iteration	x_l	x_u	x_r	$\lvert e_a \rvert$, %	$\lvert e_t \rvert$, %
1	12	16	14		5.279
2	14	16	15	6.667	1.487
3	14	15	14.5	3.448	1.896
4	14.5	15	14.75	1.695	0.204
5	14.75	15	14.875	0.840	0.641
6	14.75	14.875	14.8125	0.422	0.219

False-Position Method

แม้ว่า Bisection Method จะเป็นวิธีการที่ดี แต่ประสิทธิภาพสามารถปรับปรุงได้เพิ่มขึ้น เนื่องจากมันจะ Estimate ค่า Root ใหม่ที่กึ่งกลางช่วงเสมอ ซึ่งค่า Root ที่แท้จริงอาจจะอยู่ใกล้จุดปลายด้านใดด้านหนึ่ง ทำให้วิธีนี้ Converge ช้ากว่าที่ควรจะเป็น ดังนั้นเราควรจะมีวิธีในการประมาณค่า Root ใหม่ วิธีที่ดีกว่าคือใช้สมการเส้นตรง หรือที่เรียก Linear Interpolation(ดูรูป) และกรรมวิธีนี้เรียก False-Position Method หรือ Linear Interpolation Method



False-Position Method



จากสามเหลี่ยมคล้ายในรูป จุดที่เส้นตรงตัดกับแกน x สามารถคำนวณได้จากสมการ

$$\frac{f(x_l)}{x_r - x_l} = \frac{f(x_u)}{x_r - x_u}$$

ดังนั้นเราสามารถแก้สมการหาค่า x_r ได้ดังนี้

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

False-Position Method

Example 7.2: จากตัวอย่างเดิม $f(c)=\frac{gm}{c}[1-e^{-(c/m)t}]-v=0$ ให้จะลองใช้ False-Position Method ทำการแก้ปัญหา

$$x_l=12,\quad x_u=16$$

Iteration 1:

$$x_l=12,\quad x_u=16$$

$$f(x_l)=6.0669,\quad f(x_u)=-2.2688$$

$$x_r=16-\frac{-2.2688(12-16)}{6.0669-(-2.2688)}=14.9113,\quad e_t=0.89\%$$

Iteration 2:

$$f(x_l)f(x_r)=-1.5426\Rightarrow x_u=x_r=14.9113$$

False-Position Method

$$x_l = 12, \quad x_u = 14.9113$$

$$f(x_l) = 6.0669, \quad f(x_u) = -0.2543$$

$$x_r = 14.9113 - \frac{-0.2543(12 - 14.9113)}{6.0669 - (-0.2543)} = 14.7942, \quad e_t = 0.09\%, e_a = 0.79\%$$

จากตัวอย่างข้างบน จะดูเหมือนว่า Convergence Rate ของ False-Position Method จะดีกว่า Bisection Method มาก ทั้งนี้เนื่องจากเราได้ค่า Estimate ของ Solution ที่ใกล้เคียงกับค่าจริงมากกว่า และปกติก็เป็นเช่นนั้น ทำให้วิธีนี้นิยมใช้มากกว่าวิธี ของ Bisection อย่างไรก็ตาม ในบางกรณี วิธีของ False-Position จะ Converge ได้ช้ากว่า Bisection Method มาก ดังตัวอย่างถัดไป

False-Position Method

Example 7.3: จงเปรียบเทียบการทำงานของวิธี Bisection Method และ False-Position Method ในการหารากของสมการ $f(x) = x^{10} - 1$ โดย Root ที่จะหามีค่าอยู่ระหว่าง $[0,1.3]$

วิธีของ Bisection Method จะสรุปเป็นตารางได้ดังนี้

Iteration	x_l	x_u	x_r	$\lvert e_t \rvert$, %	$\lvert e_a \rvert$, %
1	0	1.3	0.65	35	100.0
2	0.65	1.3	0.975	2.5	33.3
3	0.975	1.3	1.1375	13.8	14.3
4	0.975	1.1375	1.05625	5.6	7.7
5	0.975	1.05625	1.015625	1.6	4.0

สังเกตได้ว่า หลังจาก Iteration ที่ 5 ค่า Error จะลดลงไม่ถึง 2%

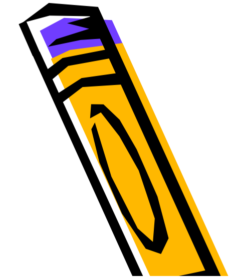
False-Position Method

วิธีของ False-Position Method จะสรุปเป็นตารางได้ดังนี้

Iteration	x_l	x_u	x_r	$\|e_t\|$, %	$\|e_a\|$, %
1	0	1.3	0.09430	90.6	
2	0.09430	1.3	0.18176	81.8	48.1
3	0.18176	1.3	0.26287	73.7	30.9
4	0.26287	1.3	0.33811	66.2	22.3
5	0.33811	1.3	0.40788	59.2	17.1

พึงสังเกตอีกว่าในกรณีนี้ $|e_a| < |e_t|$ ซึ่งกลับกับวิธีก่อน ทำให้การพิจารณาจากค่า e_a ทำให้เราเข้าใจผิดได้ เหตุผลก็คือการ Interpolation Fail เพราะตำแหน่งที่ได้แย่กว่าวิธีของ Bisection Method เนื่องจาก Function มีการเปลี่ยนแปลงอย่างกะทันหันในช่วง Bracket

Open Method: Simple One-Point Iteration



ในกรรมวิธีของ Bracket Method ค่าของ Root จะอยู่ในช่วงที่กำหนดไว้ ดังนั้นในแต่ละ Iteration เราจะทำให้ช่วงนี้แคบลง และผลลัพธ์จะ Converge อย่างไรก็ตาม ใน Open Method จะกำหนดแค่ค่าตั้งต้น หรือช่วงที่ค่าของ Root อาจจะไม่อยู่ในช่วงนี้ก็ได้ ดังนั้นบางครั้งโปรแกรมจะ Diverge แต่ถ้าเราเลือกค่าที่เหมาะ โปรแกรมจะ Converge และปกติจะ Converge ได้รวดเร็วกว่า Bracket Method มาก

วิธีการง่ายที่สุดในการหารากของสมการ ทำได้โดยการจัดเรียงสมการใหม่ในรูป $x = g(x)$ คือยามยามย้ายค่า x บางส่วนมาอยู่ด้านซ้ายของสมการ และทำการประมาณค่าใหม่จากค่าตั้งต้น ทำเช่นนี้เรื่อยไปจนได้คำตอบที่มีค่า Precision ที่ต้องการ สังเกตว่าบางครั้งการจัดเรียงสมการใหม่สามารถกระทำได้หลายวิธี และจะมีผลต่อการ Converge และผลลัพธ์ที่ได้

ยกตัวอย่างสมการ $x^2 - 3x + 3 = 0$ สามารถจัดเรียงใหม่ในรูป $x = g(x)$ ได้เป็น

$$x = \frac{x^2 + 3}{3} \qquad \text{หรือ} \qquad x = \sqrt{3x - 3}$$

ถ้าเราทำ Iteration ของสองสมการนี้ จะได้การ Converge ที่ต่างกัน

บางครั้งสมการสามารถสร้างได้โดยการบวกด้วย x ทั้งสองข้างเช่นจากสมการ $\sin x = 0$ เราได้สมการในรูป $x = g(x)$ คือ $x = \sin x + x$ เป็นต้น

Open Method: Simple One-Point Iteration

การทำงานของ Algorithm จะประมาณค่าใหม่ x_{i+1} จากค่าเดิม x_i ใน Iteration ที่ $i+1$ โดยให้ x_0 เป็นค่าตั้งต้นดังนี้

$$x_{i+1} = g(x_i)$$

ดังนั้นค่า Estimate Error, e_a สามารถคำนวณได้จาก

$$|e_a| = \left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| \times 100\%$$

Open Method: Simple One-Point Iteration

Example 7.4: ใช้ Simple One-Point Iteration หารากของสมการ $f(x) = e^{-x} - x$

Solution: ในกรณีนี้เราเขียนสมการใหม่ดังนี้ $x = e^{-x}$ และกำหนด $x_0 = 0$ จากนั้นทำ Iteration

Iteration	x_i	$\lvert e_t \rvert$, %	$\lvert e_a \rvert$, %
0	0	100	
1	1.000000	76.3	100.0
2	0.367879	35.1	171.8
3	0.692201	22.1	46.9
4	0.500473	11.8	38.3
5	0.606244	6.89	17.4
6	0.545396	3.83	11.2
7	0.579612	2.20	5.90

Open Method: Simple One-Point Iteration

Iteration	x_i	$\lvert e_t \rvert$, %	$\lvert e_a \rvert$, %
8	0.560115	1.24	3.48
9	0.571143	0.705	1.93
10	0.564879	0.399	1.11

โดยค่าที่แท้จริงของ Root คือ 0.56714329

สังเกตว่า Error ที่ได้ในแต่ละ Iteration จะเป็นประมาณ 50-60% เมื่อเทียบกับ Iteration ก่อน และโปรแกรมจะ Converge เข้าสู่ค่าจริงในกรณีนี้ การ Converge เช่นนี้เราเรียกว่าเป็น Linear Convergence

จากที่กล่าวมาแล้วว่าวิธีของ Open Method อาจจะได้โปรแกรมที่ไม่ Converge ยกตัวอย่างสมการที่คล้ายกันในตัวอย่า ถัดไป

Example 7.5: ใช้ Simple One-Point Iteration หารากของสมการ $f(x) = e^{-x} - x/4$

Solution: ในกรณีนี้เราเขียนสมการใหม่ดังนี้ $x = 4e^{-x}$ และกำหนด $x_0 = 0$ จากนั้นทำ Iteration

Iteration	x_i	$\lvert e_t \rvert$, %	$\lvert e_a \rvert$, %
0	0	100	
1	4	232.7322	100.0
2	.0733	93.9058	5.3598e+003
3	3.7174	209.2270	98.0292
4	0.0972	91.9158	3.7251e+003
5	3.6296	201.9171	97.3224
6	0.1061	91.1732	3.3205e+003
7	3.5973	199.2339	97.0502
8	0.1096	90.8839	3.1825e+003
9	3.5848	198.1948	96.9429
10	0.1110	90.7693	3.1305e+003
⋮	⋮	⋮	⋮
26261	3.5766	197.5121	96.8718
26262	0.1119	90.6392	3.0967e+003

Open Method: Simple One-Point Iteration

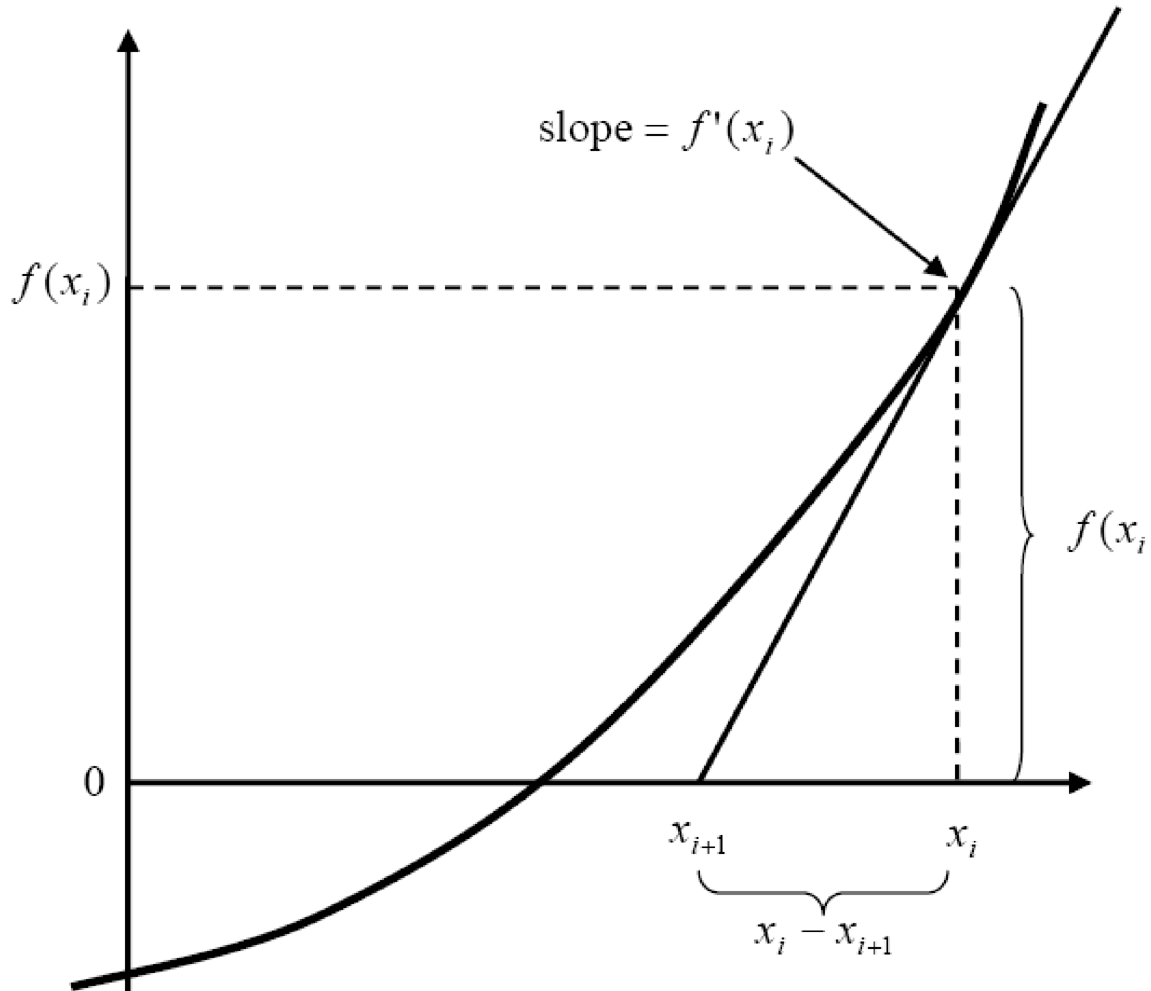
คำตอบที่แท้จริงคือ 1.20216787319704 และในกรณีนี้คำตอบจะ Oscillate และจะไม่ Converge ถ้านักศึกษาลองเปลี่ยนเลข 4 เป็นเลขอื่นที่มีค่ามากกว่านี้ คำตอบอาจจะไม่ Converge เช่นกัน ในกรณีนี้โปรแกรมจะ Diverge และ Algorithm จะ Fail (ในกรณีนี้ ตราบใดที่ค่า Absolute ของ Slope ของ $y_2 = g(x)$ น้อยกว่าค่า Alsolute ของ Slope $y_1 = x$ หรืออีกนัยหนึ่งเมื่อ $|g'(x)| < 1$ โปรแกรมจะ Converge)

Open Method: Newton-Ralphson Method

เป็นวิธีที่นิยมมากที่สุดในการหา Root ของสมการ เนื่องจากรวดเร็ว เราสามารถพิสูจน์ได้(จาก Taylor Series Expansion) ว่า ถ้าโปรแกรม Converge แล้ว Error ใน Iteration ใหม่ จะมีค่าประมาณเท่ากับกำลังสองของ Error ใน Iteration ก่อนหน้านี้ และในกรณีนี้เราเรียกว่าเป็น Quadratic Convergence

Open Method: Newton-Ralphson Method

วิธีการของ Newton-Ralphson สามารถอธิบายได้จากรูป



Open Method: Newton-Ralphson Method

ในกรรมวิธีนี้ เราจะ Estimate ค่า x_{i+1} ที่ดีกว่า โดยใช้ค่า Tangent ที่จุด $[x_i, f(x_i)]$ ตัดกับแกน x ซึ่งเขียนเป็นสมการค่า Tangent ได้ดังนี้

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

และเมื่อจัดเรียงใหม่ เราได้

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

Open Method: Newton-Ralphson Method

Example 7.6: จงใช้กรรมวิธีของ Newton-Ralphson หารากของสมการ $e^{-x}-x$

Solution:

เราได้ $f(x)=e^{-x}-x$ และดังนั้น $f'(x)=-e^{-x}-1$

ดังนั้นสมการของ Newton-Ralphson จะเป็น

$$x_{i+1}=x_i-\frac{e^{-x_i}-x_i}{-e^{-x_i}-1}$$

เริ่มจากค่า $x_0=0$ เราได้ Iteration ดังนี้

Open Method: Newton-Ralphson Method

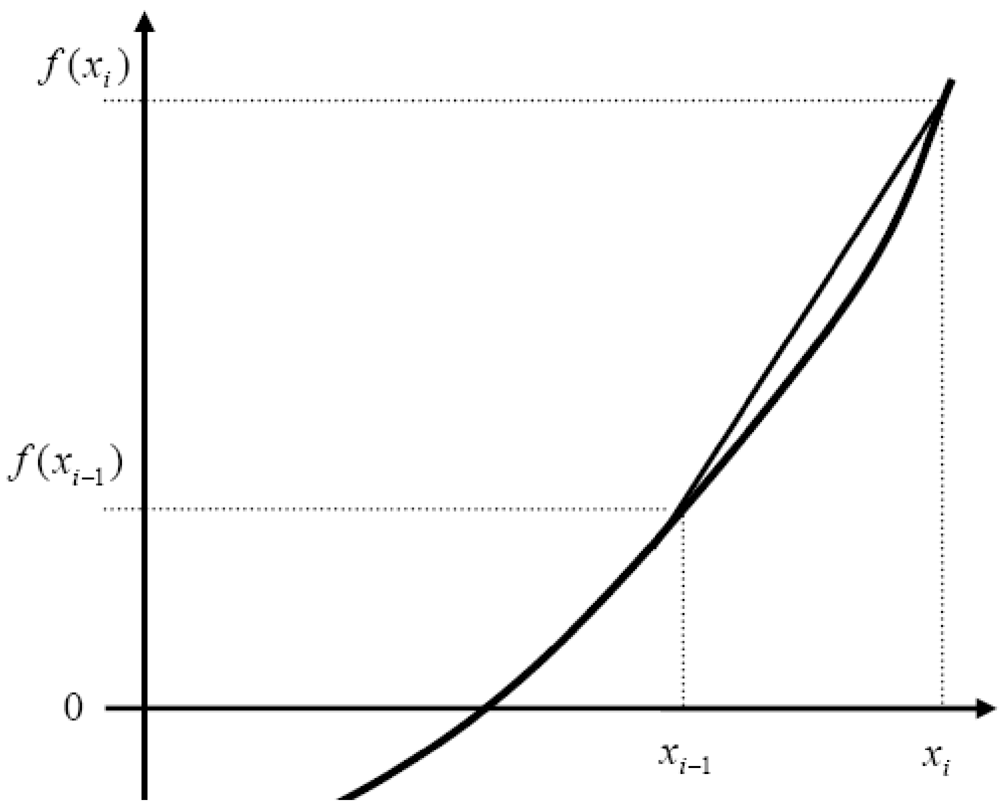
Iteration	x_i	$\lvert e_t \rvert, \%$
0	0	100
1	0.500000000	11.8
2	0.566311003	0.147
3	0.567143165	0.0000220
4	0.567143290	$< 10^{-8}$

จะเห็นได้ว่าการ Converge ของ Newton-Ralphson เร็วกว่าวิธีก่อนมาก เพียงแค่ 2 Iteration ก็ดีกว่าวิธีของ Simple-One Point Iteration ที่ทำ 10 Iteration อย่างไรก็ตาม วิธีของ Newton-Ralphson นั้นมีข้อเสียที่ว่าจะให้ผลลัพธ์ที่แย่ในกรณีของ Multiple Root และแม้แต่ Simple Root บางครั้งก็มีปัญหา เช่นกรณีของ $f(x) = x^{10} - 1$ โดยเริ่มจาก $x_0 = 0.5$ จะพบว่า Converge ได้ช้ามาก และอัตราการ Converge จะขึ้นกับค่า x_0 ที่เลือก

นอกจากนี้แล้ว วิธีการนี้จะต้องใช้การหา Derivative ของ Function ซึ่งบางครั้งไม่สามารถหาได้ง่ายๆ การแก้ไขก็คือใช้วิธีการประมาณค่า Derivative ซึ่งเป็นวิธีการของ Secant Method ดังจะกล่าวในหัวข้อต่อไป

Open Method: Secant Method

จากที่กล่าวมาแล้ว ในการใช้วิธีของ Newton-Ralphson Method นั้น เราจะต้องหา Derivative ของ Function ซึ่งบาง Function จะหาค่า Derivative ได้ยากมาก ในกรณีเช่นนี้ เราอาจจะใช้วิธีการประมาณค่า Derivative ดังนี้(ดูรูป)



Open Method: Secant Method

$$f'(x_i) \cong \frac{f(x_{i-1}) - f(x_i)}{x_{i-1} - x_i}$$

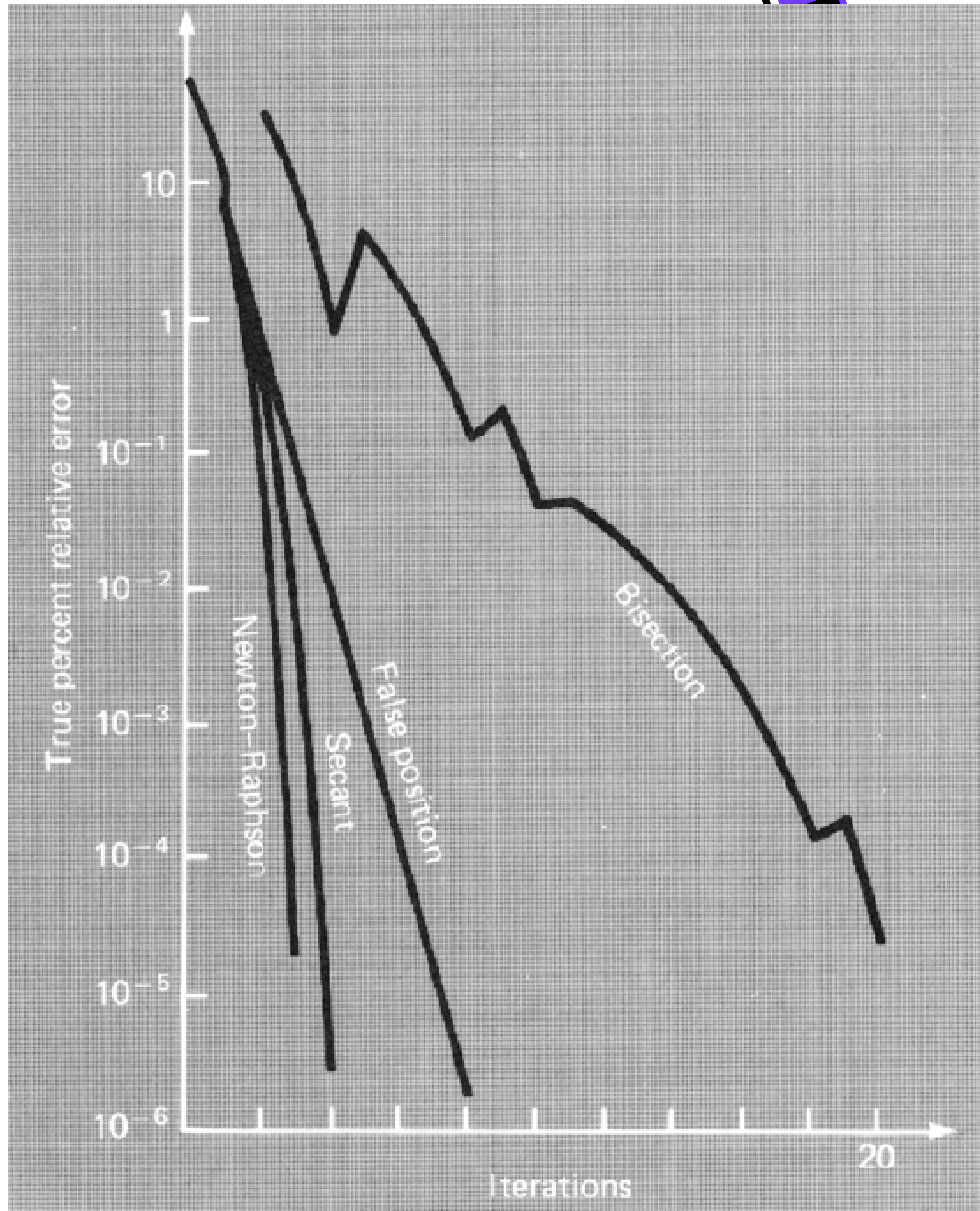
เมื่อนำสมการข้างบนไปแทนค่าในสมการของ Newton-Ralphson เราจะได้

$$x_{i+1} = x_i - \frac{f(x_i)[x_{i-1} - x_i]}{f(x_{i-1}) - f(x_i)} \qquad \text{(Secant Method Formula)}$$

สังเกตว่าวิธีการนี้จำต้องใช้ค่า Estimate ของ x จำนวนสองค่า

Open Method: Secant Method

Secant Method อาจจะไม่ Converge ถ้าเราเลือกสองจุดที่ไม่เหมาะสม แต่ถ้ามัน Converge แล้ว มันจะ Converge ได้เร็วเกือบเท่าๆ Newton-Ralphson Method อย่างไรก็ตาม การ Convergence ขึ้นอยู่กับ Function และจุดเริ่มต้นที่เลือก รูปข้างล่างแสดงการเปรียบเทียบการ Convergence ของ $f(x) = e^{-x} - x$ (Simple One-Point Iteration เป็น Linear Convergence และจะ Converge ช้าสุด ไม่ได้แสดงไว้ ขอให้นักศึกษาลอง Plot เอง โดยใช้ข้อมูลจากตัวอย่างที่ 7.4)



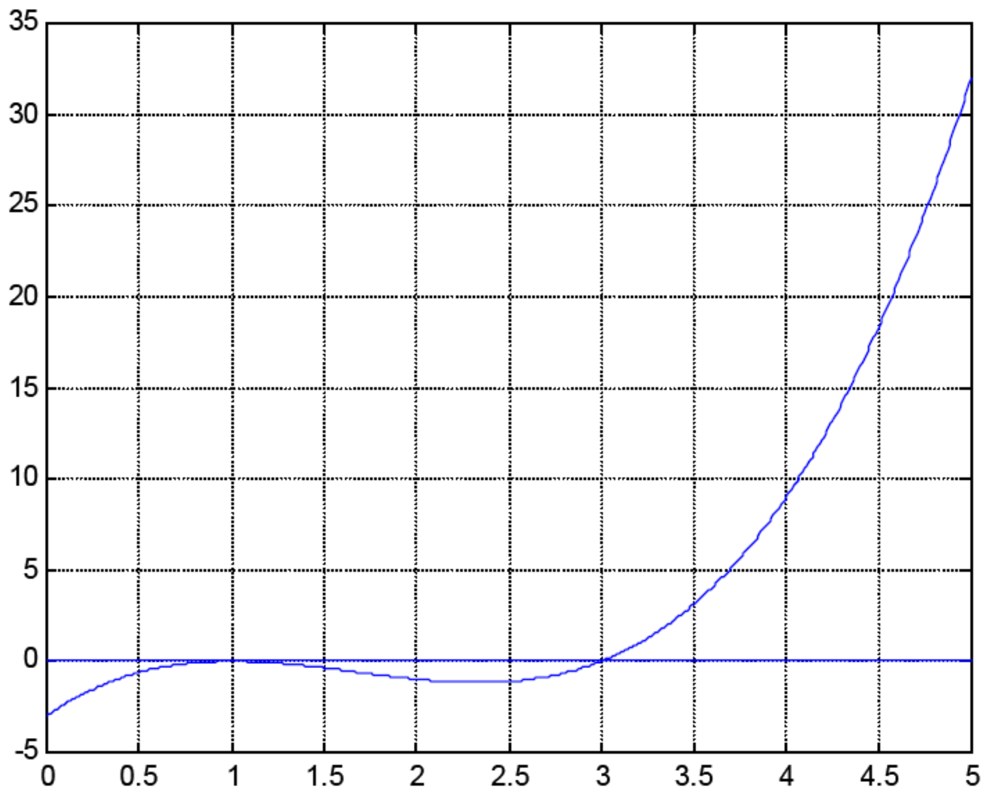
True percent relative error
10
1
10⁻¹
10⁻²
10⁻³
10⁻⁴
10⁻⁵
10⁻⁶
Newton–Raphson
Secant
False position
Bisection
20
Iterations

Multiple Roots

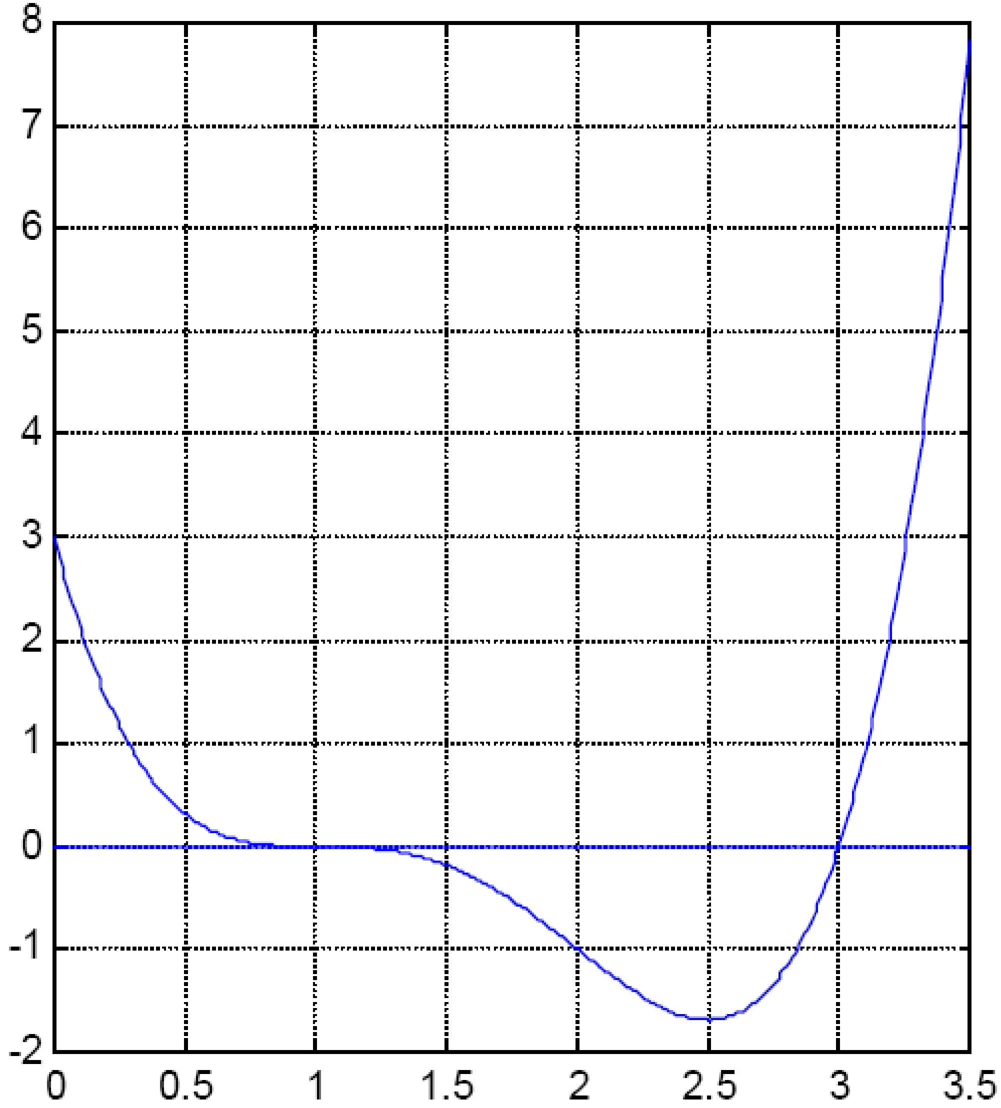
Multiple Root จะเป็นจุดที่ Function สัมผัสกับแกน x กล่าวคือค่า Slope จะเป็นศูนย์ ยกตัวอย่างเช่นสมการ

$$f(x) = x^3 - 5x^2 + 7x - 3 = (x-3)(x-1)(x-1)$$

ในกรณีเช่นนี้เรากล่าวว่า Function มี Double Root ที่ $x = 1$ ดังรูป



หรือในกรณีของ Triple Root เช่น $f(x) = x^4 - 6x^3 + 12x^2 - 10x + 3 = (x-3)(x-1)^3$ ดังรูป



Multiple Roots

กรณีของ Multiple Root จะทำให้วิธีของ Numerical Method ที่กล่าวมาเกิดปัญหา เนื่องจาก Function ไม่มีการเปลี่ยนเครื่องหมายที่จุดของ Root

1. ที่จุดนี้ นอกจาก $f(x)$ จะเท่ากับ ศูนย์แล้ว ค่า $f'(x)$ จะเท่ากับศูนย์ด้วย จะทำให้เกิดปัญหาใน Newton-Ralphson Method และ Secant Method อย่างไรก็ตาม $f(x)$ จะเข้าใกล้ศูนย์ก่อน $f'(x)$ เสมอและเราสามารถจะเพิ่มส่วนของโปรแกรมเพื่อจะตรวจสอบค่า $f(x)$ และหยุดโปรแกรมก่อนที่ค่า $f'(x)$ จะเป็นศูนย์ ซึ่งจะทำให้โปรแกรมเกิด “Divide by Zero Overflow”

ในกรณีของ Multiple Root วิธีของ Newton-Ralphson และ Secant Method จะมีการ Converge แบบ Linear แทนที่จะเป็น Quadratic วิธีการแก้มีหลายวิธี ที่แนะนำคือวิธีที่เสนอโดย Ralston and Rabinowitz(1978) โดยการให้นิยาม Function ใหม่ดังนี้

Multiple Roots

$$u(x) = \frac{f(x)}{f'(x)} \text{ และ } u'(x) = \frac{f'(x)f'(x) - f(x)f''(x)}{[f'(x)]^2}$$

และใช้สมการในการทำ Iteration เป็น

$$x_{i+1} = x_i - \frac{u(x_i)}{u'(x_i)}$$

ดังนั้นเราได้

$$x_{i+1} = x_i - \frac{f(x_i)f'(x_i)}{[f'(x_i)]^2 - f(x_i)f''(x_i)} \qquad \text{(Modified Newton-Ralphson Method)}$$

ซึ่ง Algorithm ข้างบนจะเป็น Quadratic Convergence ทั้ง Simple Root และ Multiple Root แต่จะใช้การคำนวณ มากกว่าวิธีการปกติของ Newton-Ralphson สำหรับแต่ละ Iteration

Multiple Roots

Example 7.7: เปรียบเทียบ Newton-Ralphson และ Modified Newton-Ralphson ในกรณีของการหา Root ของ

$$f(x) = x^3 - 5x^2 + 7x - 3$$

Solution:

$$f'(x) = 3x^2 - 10x + 7$$

$$f''(x) = 6x - 10$$

ดังนั้นเราได้ Newton-Ralphson: $x_{i+1} = x_i - \dfrac{x_i^3 - 5x_i^2 + 7x_i - 3}{3x_i^2 - 10x_i + 7}$

และสำหรับ Modified Newton-Ralphson Method เราได้

Modified Newton-Ralphson: $x_{i+1} = x_i - \dfrac{(x_i^3 - 5x_i^2 + 7x_i - 3)(3x_i^2 - 10x_i + 7)}{(3x_i^2 - 10x_i + 7)^2 - (x_i^3 - 5x_i^2 + 7x_i - 3)(6x_i - 10)}$

เมื่อ Run Iteration เราจะได้คำตอบดังตารางข้างล่าง ($x_0 = 0$)

Multiple Roots

Normal Newton-Ralphson:

Iteration	x_i	$\lvert e_t \rvert$, %
0	0	100
1	0.428571429	57
2	0.685714286	31
3	0.832865400	17
4	0.913328983	8.7
5	0.955783293	4.4
6	0.977655101	2.2

Multiple Roots

Modified Newton-Ralphson:

Iteration	x_i	$\lvert e_t \rvert$, %
0	0	100
1	1.105263158	22
2	1.003081664	0.31
3	1.000002382	0.00024

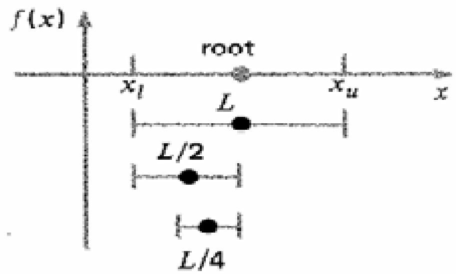
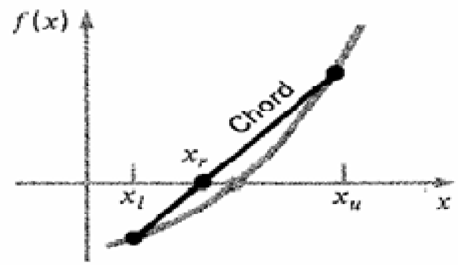
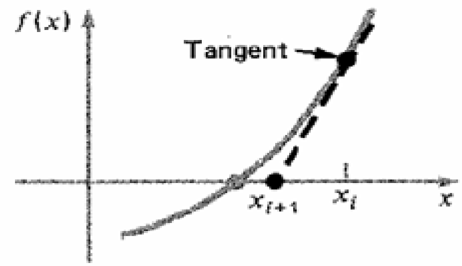
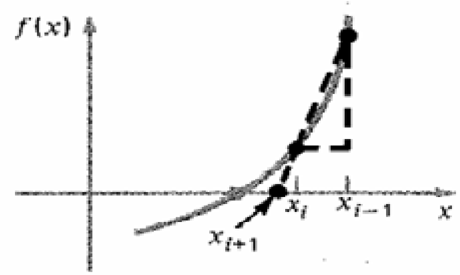
เช่นเดียวกัน เราสามารถปรับปรุงวิธีของ Secant Method ได้เช่นเดียวกัน โดยใช้การ Estimate ของ Function $u(x)$ และ $u'(x)$ แต่ในกรณีนี้จะไม่กล่าวถึง

Comparison

การหา Root ของ Function เป็นเรื่องที่สำคัญสำหรับวิศวกรในการแก้ปัญหาทางคณิตศาสตร์ เนื่องจากการแก้ปัญหาแบบ Analytical Method ไม่สามารถกระทำได้ทุกกรณี เราจึงหันมาใช้วิธีทาง Numerical Method

การนำวิธีทาง Numerical Method จะต้องคำนึงถึง Error เป็นสำคัญ ทั้ง Truncation Error และ Round Off Error นอกจากนี้แล้วจะต้องคำนึงถึงการ Convergence ของ Algorithm ด้วย ดังนั้นการเลือกกรรมวิธีที่จะนำมาใช้จะเป็นสิ่งที่ควรจะพิจารณาเป็นอันดับแรก ตารางข้างล่างเป็นตารางสรุปของกรรมวิธี และข้อดีข้อเสียของแต่ละวิธี

Method	Initial Guest	Rate of Convergence	Stability	Accuracy	Breadth of Application	Program	Comments
Direct	-	-	-	-	Very Limited	-	
Graphical	-	-	-	Poor	General	-	May Take Much Time
Bisection	2	Slow	Always Converges	Good	General	Easy	
False Position	2	Medium	Always Converges	Good	General	Easy	
One-Point Iteration	1	Slow	May Not Converge	Good	General	Easy	
Newton-Ralphson	1	Fast	May Not Converge	Good	Limited if $f'(x)=0$	Easy	Requires Evaluation of $f'(x)$
Modified Newton-Ralphson	1	Fast for Multiple Roots, Medium for Single Roots	May Not Converge	Good	Specifically Designed for Multiple Roots	Easy	Requires Evaluation of $f'(x)$ and $f''(x)$
Secant	2	Medium to Fast	May Not Converge	Good	General	Easy	Initial Guesses Do Not Have to Bracket The Roots

Method	Formulation	Graphical Interpretation	Errors and Stopping Criteria
		Bracketing methods:	
Bisection	$x_r = \dfrac{x_l + x_u}{2}$ If $f(x_l)f(x_r) < 0,\ x_u = x_r$ $f(x_l)f(x_r) > 0,\ x_l = x_r$		Stopping criterion: $\left\| \dfrac{x_r^{new} - x_r^{old}}{x_r^{new}} \right\| 100\% \le \epsilon_s$
False Position	$x_r = x_u - \dfrac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$ If $f(x_l)f(x_r) < 0,\ x_u = x_r$ $f(x_l)f(x_r) > 0,\ x_l = x_r$		Stopping criterion: $\left\| \dfrac{x_r^{new} - x_r^{old}}{x_r^{new}} \right\| 100\% \le \epsilon_s$
		Open methods:	
Newton-Raphson	$x_{i+1} = x_i - \dfrac{f(x_i)}{f'(x_i)}$		Stopping criterion: $\left\| \dfrac{x_{i+1} - x_i}{x_{i+1}} \right\| 100\% \le \epsilon_s$ Error: $E_{i+1} = O(E_i^2)$
Secant	$x_{i+1} = x_i - \dfrac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$		Stopping criterion: $\left\| \dfrac{x_{i+1} - x_i}{x_{i+1}} \right\| 100\% \le \epsilon_s$

Homework 8

- นักศึกษาต้องเขียนโปรแกรมช่วยคำนวณหรือใช้ Spreadsheet (MS Excel) ช่วยคำนวณ
  - แนะนำให้ใช้ MATLAB
    - เขียน Function หรือ Scratch File ก็ได้
    - หรือคำนวณจาก Workspace โดยตรง
- Download คำถามและตอบคำถาม

CPE 332 Computer Engineering Mathematics II

Week 12

Part III, Chapter 10

Linear Equations

Today(Week 13) Topics

- Chapter 9 Linear Equations
  - Gauss Elimination
  - Gauss-Jordan
  - Gauss-Seidel
  - LU Decomposition
  - Crout Decomposition
- HW(9) Ch 9 Due Next Week

MATLAB Programming

- เราสามารถเขียน Function การคำนวณโดยใช้ MATLAB Editor และบันทึกเป็น ‘.m’ File
  - ขึ้นบันทัดแรกของ Function ด้วย

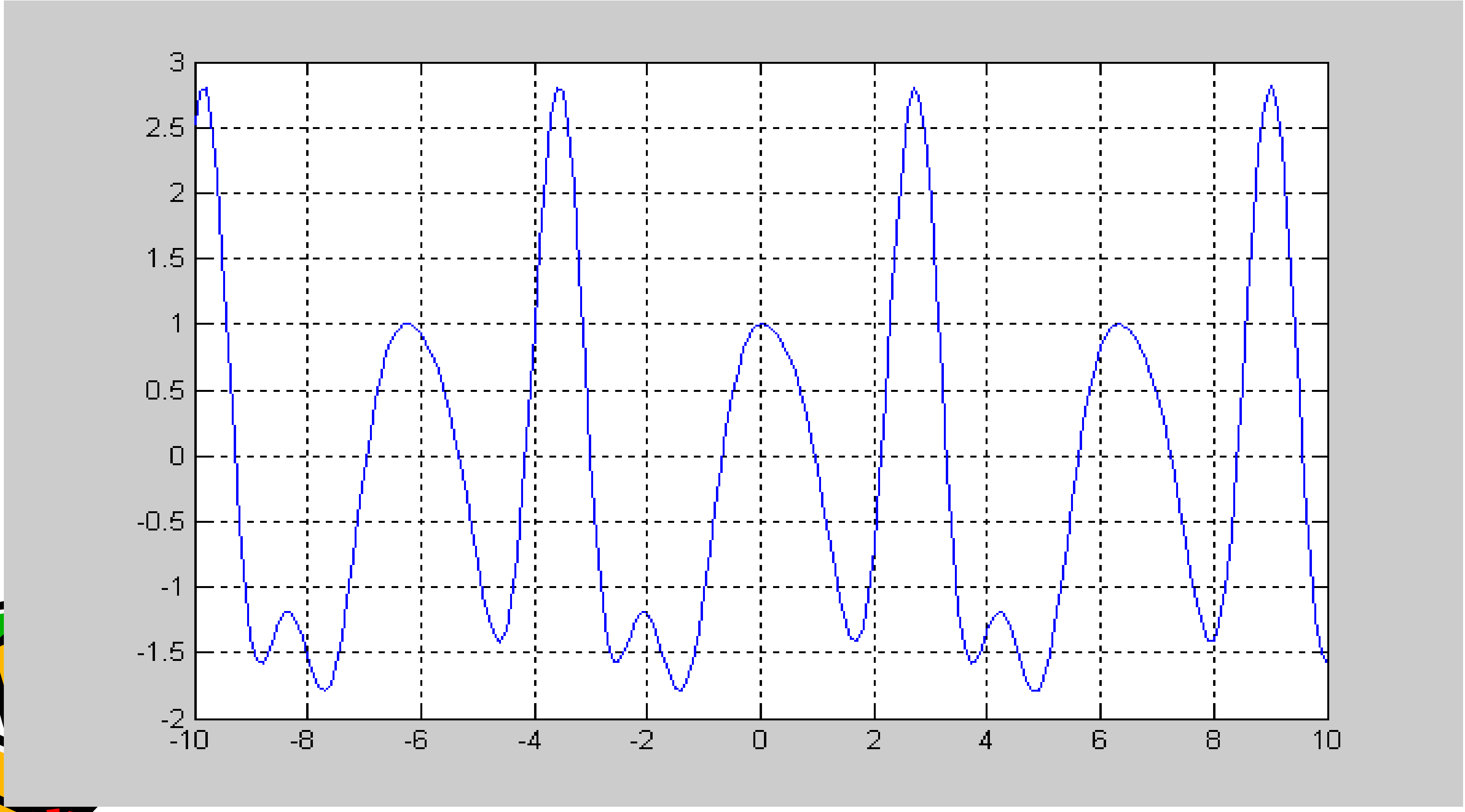
  function [List ของค่าที่ส่งคืน]=fname(List ของ Parameter)

  function [x,y,z]=find123(a,b,c)

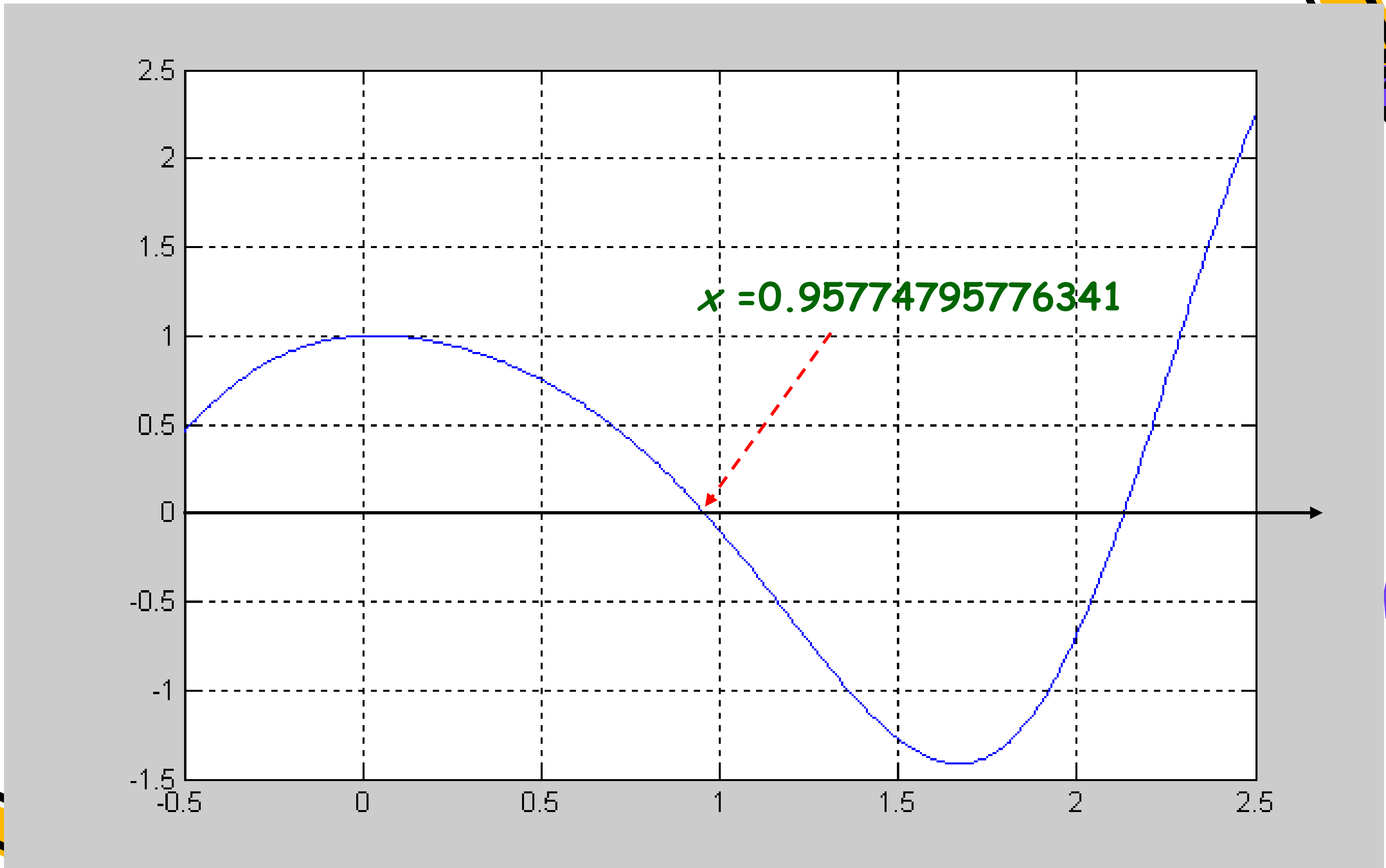
  - ภายใน Function สามารถใช้ Loop, Branch ได้เหมือนการเขียนโปรแกรม, สามารถกำหนด Local Variable ภายในได้เช่นกัน
  - อย่าลืมว่า พื้นฐาน Variable จะเป็น Matrix
- Function นี้สามารถเรียกใช้งานได้ใน MATLAB
- ดูรายละเอียดใน Tutorial 4-5 ของ MATLAB

Ex: หารากของ $f(x) = \sin 3x \cdot e^{-\cos x} + \cos 2x \cdot e^{-\sin x}$

- `x=-10:.1:10;`
- `y=sin(3*x).*exp(-cos(x))+cos(2*x).*exp(-sin(x));`
- `plot(x,y)`



เราจะหาคำตอบในช่วง [0, 2]



MATLAB: Bisection Mtd.

```
function [x]=example91a(es)
% Calculate using Bisection Method between [0,2]
ea = inf;
xr = inf;
it=0;
xl=0;
xu=2;
while(ea > es)
    it = it+1;
    pxr=xr;
    fxl=sin(3*xl).*exp(-cos(xl))+cos(2*xl).*exp(-sin(xl));
    fxu=sin(3*xu).*exp(-cos(xu))+cos(2*xu).*exp(-sin(xu));
    xr=(xl+xu)/2;
    fxr=sin(3*xr).*exp(-cos(xr))+cos(2*xr).*exp(-sin(xr));
    ea = abs((xr-pxr)/xr)*100;
    x=[it xl fxl xu fxu xr fxr ea]
    if(fxl*fxr > 0.0)
        xl=xr;
    elseif (fxl*fxr < 0.0)
        xu=xr;
    else
        ea=0.0;
    end
end
```

Bisection Results:>> example91a(0.01)

	Iter	xl	fxl	xu	fxu	xr	fxr	ea
x =	1.0000	0	1.0000	2.0000	-0.6869	1.0000	-0.0972	Inf
x =	2.0000	0	1.0000	1.0000	-0.0972	0.5000	0.7493	100.0000
x =	3.0000	0.5000	0.7493	1.0000	-0.0972	0.7500	0.4101	33.3333
x =	4.0000	0.7500	0.4101	1.0000	-0.0972	0.8750	0.1774	14.2857
x =	5.0000	0.8750	0.1774	1.0000	-0.0972	0.9375	0.0451	6.6667
x =	6.0000	0.9375	0.0451	1.0000	-0.0972	0.9688	-0.0249	3.2258
x =	7.0000	0.9375	0.0451	0.9688	-0.0249	0.9531	0.0104	1.6393
x =	8.0000	0.9531	0.0104	0.9688	-0.0249	0.9609	-0.0072	0.8130
x =	9.0000	0.9531	0.0104	0.9609	-0.0072	0.9570	0.0016	0.4082
x =	10.0000	0.9570	0.0016	0.9609	-0.0072	0.9590	-0.0028	0.2037
x =	11.0000	0.9570	0.0016	0.9590	-0.0028	0.9580	-0.0006	0.1019
x =	12.0000	0.9570	0.0016	0.9580	-0.0006	0.9575	0.0005	0.0510
x =	13.0000	0.9575	0.0005	0.9580	-0.0006	0.9578	-0.0000	0.0255
x =	14.0000	0.9575	0.0005	0.9578	-0.0000	0.9576	0.0002	0.0127
x =	15.0000	0.9576	0.0002	0.9578	-0.0000	0.9577	0.0001	0.0064

- ans =
- 15.00000000000000 0.95776367187500 -0.00003535871565 0.95770263671875 0.00023929892750 **0.95770263671875 0.00010197464576 0.00637308010962**

***x* =0.95774795776341**
True error = 0.004732%

Other Results: xt= **0.95774795776341**

- Es = 0.01%
  - It = 15;xr=0.95770263671875
  - Ea=0.006373%, et = 0.004732%
- Es = Es = 0.001%
  - It = 18;xr=0.95774078369141
  - Ea=0.0007966%, et = 0.0007491%
- Es = 0.0001%
  - It = 21;xr=0.95774745941162
  - Ea=0.00009957%, et = 0.00005203%
- Es = 0.000001%
  - It = 28;xr=0.95774795860052
  - Ea=0.0000007779%, et = 8.740e-008%

MATLAB: False-Position

```
function [x]=example91b(es)
% Calculate using False-Position Method between [0,2]
ea = inf;
xr = inf;
it=0;
xl=0;
xu=2;
while(ea > es)
    it = it+1;
    pxr=xr;
    fxl=sin(3*xl).*exp(-cos(xl))+cos(2*xl).*exp(-sin(xl));
    fxu=sin(3*xu).*exp(-cos(xu))+cos(2*xu).*exp(-sin(xu));
    % xr=(xl+xu)/2;
    xr=xu-((fxu*(xl-xu))/(fxl-fxu));
    fxr=sin(3*xr).*exp(-cos(xr))+cos(2*xr).*exp(-sin(xr));
    ea = abs((xr-pxr)/xr)*100;
    x=[it xl fxl xu fxu xr fxr ea]
    if(fxl*fxr > 0.0)
        xl=xr;
    elseif (fxl*fxr < 0.0)
        xu=xr;
    else
        ea=0.0;
    end
end
```

FP Results:>> example91b(0.01)

	Iter	xl	fxl	xu	fxu	xr	fxr	ea
x =	1.0000	0	1.0000	2.0000	-0.6869	1.1856	-0.5611	Inf
x =	2.0000	0	1.0000	1.1856	-0.5611	0.7595	0.3940	56.1096
x =	3.0000	0.7595	0.3940	1.1856	-0.5611	0.9353	0.0500	18.7964
x =	4.0000	0.9353	0.0500	1.1856	-0.5611	0.9557	0.0045	2.1423
x =	5.0000	0.9557	0.0045	1.1856	-0.5611	0.9576	0.0004	0.1922
x =	6.0000	0.9576	0.0004	1.1856	-0.5611	0.9577	0.0000	0.0166
x =	7.0000	0.9577	0.0000	1.1856	-0.5611	0.9577	0.0000	0.0014

- ans =
- 7.00000000000000 0.95773289766706 0.00003388653487 1.18559512875289 -0.56109590391892 0.95774665822935 0.00000292408716 0.00143676432376

x **=0.95774795776341**
True error = 0.0001357 %

Other Results: x_t= **0.95774795776341**

- สีน้ำเงินได้จาก Bisection Method
- สีเขียวได้จาก False-Position Method
- Es = 0.01%
  - It = 15;xr=0.95770263671875 Ea=0.006373%, et = 0.004732%
  - It = 7;xr=0.95774665822935 Ea=0.001437%, et = 0.0001357%
- Es = Es = 0.001%
  - It = 18;xr=0.95774078369141 Ea=0.0007966%, et = 0.0007491%
  - It = 8;xr=0.95774784562942 Ea=0.0001240%, et = 0.00001171%
- Es = 0.0001%
  - It = 21;xr=0.95774745941162 Ea=0.00009957%, et = 0.00005203%
  - It = 9;xr=0.95774794808763 Ea=0.00001070%, et = 0.000001010%
- Es = 0.000001%
  - It = 28;xr=0.95774795860052 Ea=0.0000007779%, et = 8.740e-008%
  - It = 10;xr=0.95774795692851 Ea=0.0000009231%, et = 8.717e-008%

Newton-Ralphson Method

$$f(x) = \sin 3x \cdot e^{-\cos x} + \cos 2x \cdot e^{-\sin x}$$

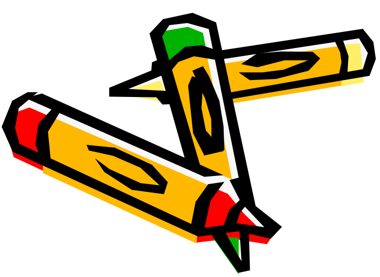
$$f'(x) = \sin 3x \cdot d\frac{e^{-\cos x}}{dx} + e^{-\cos x} d\frac{\sin 3x}{dx} + \cos 2x \cdot d\frac{e^{-\sin x}}{dx} + e^{-\sin x} d\frac{\cos 2x}{dx}$$

$$= \sin 3x \cdot e^{-\cos x} \cdot \sin x + e^{-\cos x} \cdot 3\cos 3x + \cos 2x \cdot e^{-\sin x}(-\cos x) + e^{-\sin x} \cdot (-2\sin 2x)$$

$$= e^{-\cos x}[\sin 3x \sin x + 3\cos 3x] - e^{-\sin x}[\cos 2x \cos x + 2\sin 2x]$$

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

$$x_{i+1} = x_i - \frac{\sin 3x_i \cdot e^{-\cos x_i} + \cos 2x_i \cdot e^{-\sin x_i}}{e^{-\cos x_i}[\sin 3x_i \sin x_i + 3\cos 3x_i] - e^{-\sin x_i}[\cos 2x_i \cos x_i + 2\sin 2x_i]}$$



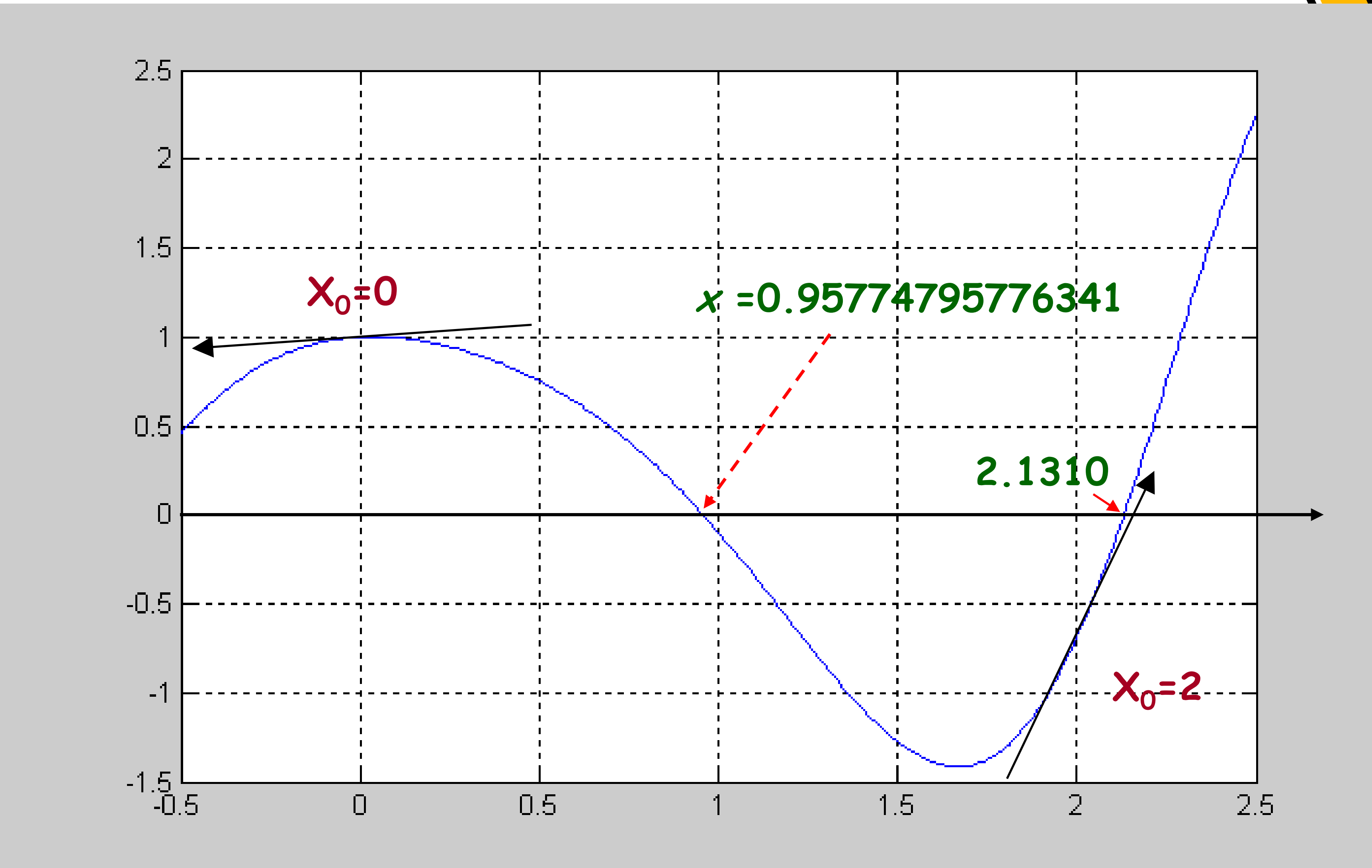
MATLAB Program:

```
function [x]=example91c(es,x0)

% Calculate solution using Newton-Ralphson, x0=initial;

it=0;
xi=x0;
ea=inf;
while (ea > es)
    it = it+1;
    fxi=sin(3*xi)*exp(-cos(xi))+cos(2*xi)*exp(-sin(xi));
    dfxi=exp(-cos(xi))*(sin(3*xi)*sin(xi)+3*cos(3*xi))...
        -exp(-sin(xi))*(cos(2*xi)*cos(xi)+2*sin(2*xi));
    pxi=xi;
    xi=pxi-fxi/dfxi;
    ea=abs((xi-pxi)/xi)*100;
    x=[it pxi fxi dfxi xi ea]
end
```

Result: ea=0.01, x_o=?

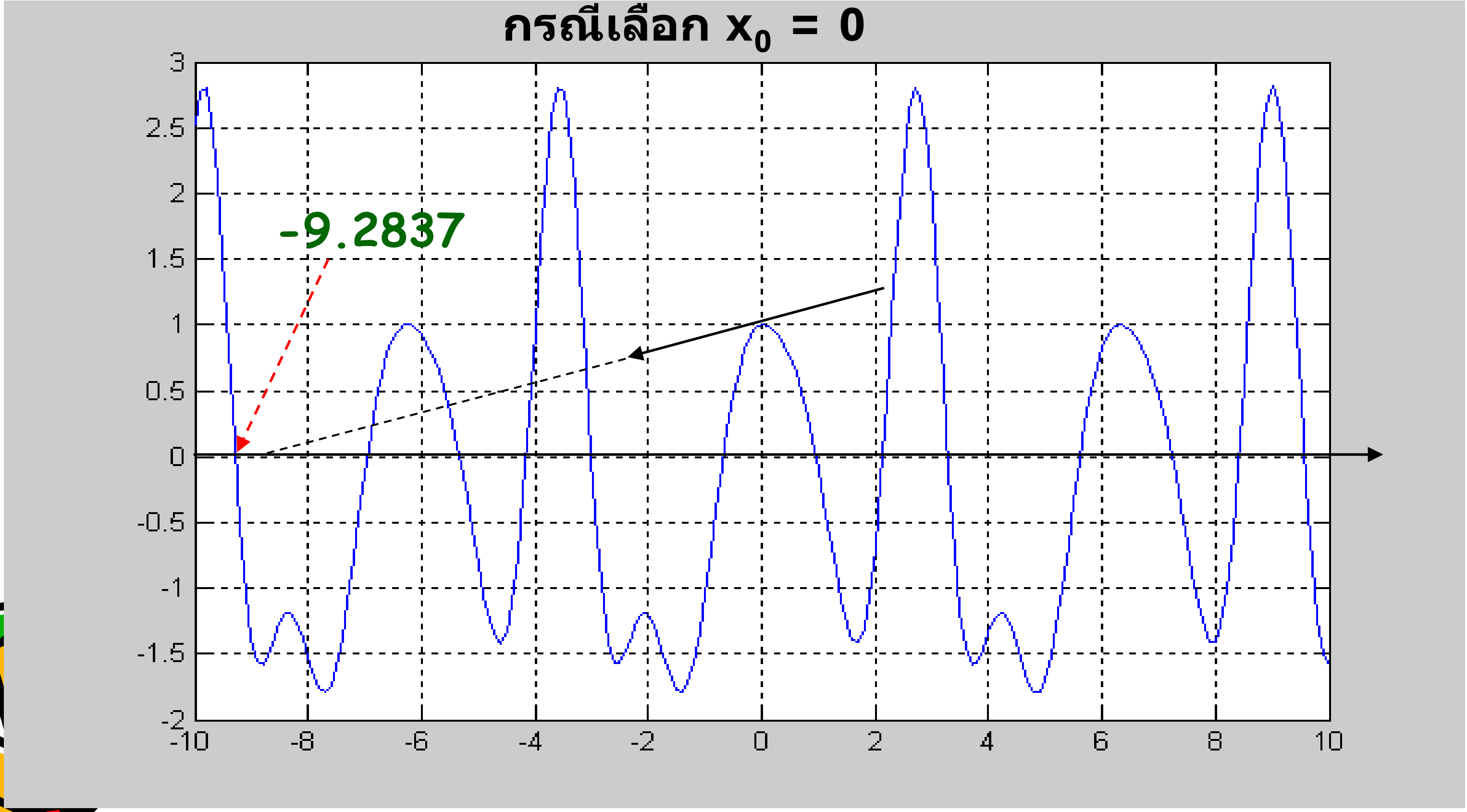
- X_0=0 โปรแกรมจะ Converge เข้าสู่จุดอื่นด้านซ้าย
- X_0=2 โปรแกรมจะ Converge เข้าสู่จุดอื่นด้านขวา
- ดูรูป
- ถ้า x_0 = 0.5 หรือ 1.5 โปรแกรมจะ Converge เข้าจุดที่ต้องการอย่างรวดเร็วมาก
- เป็นไปได้ที่เราเลือกจุดที่โปรแกรมไม่ Converge
- เราอาจจะใช้ Bisection Method ก่อนเพื่อหาจุด x_0 ที่ดี จากนั้นต่อด้วย Newton-Ralphson เพื่อให้ได้คำตอบอย่างรวดเร็ว

เราจะหาคำตอบในช่วง [0, 2]



Ex: หารากของ $f(x) = \sin 3x \cdot e^{-\cos x} + \cos 2x \cdot e^{-\sin x}$

- `x=-10:.1:10;`
- `y=sin(3*x).*exp(-cos(x))+cos(2*x).*exp(-sin(x));`
- `plot(x,y)`



Result: x_0=0.5, es = 0.01

	Iter	xi	fxi	dfxi	xi+1	ea
• x =	1.0000	0.5000	0.7493	-1.0485	1.2146	58.8351
• x =	2.0000	1.2146	-0.6362	-2.5824	0.9683	25.4413
• x =	3.0000	0.9683	-0.0238	-2.2755	0.9578	1.0939
• x =	4.0000	0.9578	-0.0001	-2.2502	0.9577	0.0061

- x = 4 0.95780651884294 -0.00013177280 -2.2502485811352 0.9577479596203 0.0061142623199 8

x =0.9577479577634 1

True error = 0.0000001939 %

Result: x_0=0.5, es = 0.000001

	Iter	xi	fxi	dfxi	xi+1	ea
x =	1.0000	0.5000	0.7493	-1.0485	1.2146	58.8351
x =	2.0000	1.2146	-0.6362	-2.5824	0.9683	25.4413
x =	3.0000	0.9683	-0.0238	-2.2755	0.9578	1.0939
x =	4.0000	0.9578	-0.0001	-2.2502	0.9577	0.0061
x =	5.0000	0.9577	-0.0000	-2.2501	0.9577	0.0000

- x = 5 0.95774795962033 -0.00000000417826 -2.25010587635298 0.95774795776341 0.00000019388349

x =0.95774795776341

True error = 0.000000000000000 %

Compare : *xt* =0.95774795776341

- Es = 0.01%
  - It = 15;xr=0.95770263671875 Ea=0.006373%, et = 0.004732%
  - It = 7;xr=0.95774665822935 Ea=0.001437%, et = 0.0001357%
  - It = 4;xi= 0.95774795962033 Ea=0.006114%, et = 0.0000001939 %
- Es = Es = 0.001%
  - It = 18;xr=0.95774078369141 Ea=0.0007966%, et = 0.0007491%
  - It = 8;xr=0.95774784562942 Ea=0.0001240%, et = 0.00001171%
  - It = 5;xi= 0.95774795776341 Ea=0.0000001939%, et < 1.0e-15 %
- Es = 0.0001%
  - It = 21;xr=0.95774745941162 Ea=0.00009957%, et = 0.00005203%
  - It = 9;xr=0.95774794808763 Ea=0.00001070%, et = 0.000001010%
  - It = 5;xi= 0.95774795776341 Ea=0.0000001939%, et < 1.0e-15 %
- Es = 0.000001%
  - It = 28;xr=0.95774795860052 Ea=0.0000007779%, et = 8.740e-008%
  - It = 10;xr=0.95774795692851 Ea=0.0000009231%, et = 8.717e-008%
  - It = 5;xi= 0.95774795776341 Ea=0.0000001939%, et < 1.0e-15 %

เพียง 5 iteration วิธีของ Newton-Ralphson
ให้ Error น้อยจน Double Precision วัดไม่ได้
แต่ข้อเสียคือจุด x_0 จะต้องเลือกให้ดี

Chapter 10: System of Linear Eq.

- จะ Limit อยู่ที่สมการ AX=B โดย A เป็น Square Matrix
  - N สมการ N Unknown
  - จะมีคำตอบที่ Unique
  - คำตอบจะมีได้ต่อเมื่อ A ไม่เป็น Singular
    - Determinant ไม่เท่ากับ 0
    - A หา Inverse ได้ และ $X = A^{-1}B$
  - ในกรณีที่ Determinant A ใกล้ศูนย์ แต่ไม่ใช่ศูนย์ คำตอบจะ Sensitive กับ Error การคำนวณเมื่อมีการปัดเศษจะต้องระวัง
    - กรณีนี้ เราเรียกว่ามันเป็น 'Ill-Conditioned System'

System of Linear Equations

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= c_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= c_2 \\
\cdot \qquad\quad \cdot \qquad\quad \cdot \quad & \\
\cdot \qquad\quad \cdot \qquad\quad \cdot \quad & \\
\cdot \qquad\quad \cdot \qquad\quad \cdot \quad & \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= c_n
\end{aligned}
=
\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}
=
\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}
= \mathbf{AX} = \mathbf{C}
$$

Krammer's Rule

$$x_1 = \frac{\begin{vmatrix} c_1 & a_{12} & a_{13} \\ c_2 & a_{22} & a_{23} \\ c_3 & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}$$

Solution ของ $AX=C$

- $A^{-1}AX=A^{-1}C$
- $X=A^{-1}C$
- Inverse หาได้ยาก แม้จะใช้ Computer คำนวณ เพราะเป็น $O(n^4)$

Solution by Elimination

วิธีการ Elimination ของ Unknown ยังสามารถนำมาใช้ได้ในกรณีที่เรามี Unknown มากกว่า 3 ตัว ด้วยการหา Factor ร่วมของคู่ของสมการ และกำจัด หนึ่ง Unknown ออกไป ทำให้เราเหลือระบบที่มี $n-1$ Unknown และ $n-1$ สมการ และเราก็สามารถทำเป็น Iteration(Loop) ต่อไปจนเหลือแค่ Unknown เดียว ซึ่งเราจะได้คำตอบสำหรับ Unknown นั้น จากนั้นเราสามารถที่จะแทนค่าย้อนกลับหา Unknown ตัวที่สอง สาม จนถึงตัวสุดท้าย ซึ่งวิธีการเช่นนี้ แม้ว่าจะใช้การคำนวณหลายขั้นตอน แต่เป็นวิธีการที่เหมาะสมในการเขียนโปรแกรมคอมพิวเตอร์ เพราะว่าจะได้ Source Code ที่สั้น วิธีการดังกล่าวก็คือ Gauss Elimination Method ที่จะกล่าวในหัวข้อต่อไป

Gauss Elimination

8.2.1 หลักการของ Gauss Elimination

1. ใน Elimination Step จาก AX=C เราพยายามทำให้ Matrix A อยู่ในรูป Upper Diagonal Matrix ด้วยขบวนการ Elimination คือการบวกและลบแต่ละแถวเข้าด้วยกัน และค่า C จะถูกบวกลบตามไปด้วย

2. เมื่อ A เป็น Upper Diagonal แล้ว การแก้สมการสามารถทำได้ง่าย โดยเราหา x_n ก่อนในแถวสุดท้ายของสมการ จากนั้นนำ x_n ที่หาได้มาแทนค่า เพื่อหา x_{n-1} ในแถวรองสุดท้าย เนื่องจากเป็นการแทนค่าเพื่อหา Unknown ย้อนหลัง เราจึงเรียก Back-Substitution

Gauss Elimination

- จาก

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= c_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= c_2 \\
\cdot \qquad\qquad \cdot \qquad\qquad &\cdot \\
\cdot \qquad\qquad \cdot \qquad\qquad &\cdot \\
\cdot \qquad\qquad \cdot \qquad\qquad &\cdot \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= c_n
\end{aligned}
$$

$$\mathbf{AX} = \mathbf{C}$$

Gauss Elimination

8.2.1 หลักการของ Gauss Elimination

$$\mathbf{AX} = \mathbf{C}$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} \hat{a}_{1,1} & \hat{a}_{1,2} & \cdots & \hat{a}_{1,n} \\ a_{2,1} & \hat{a}_{2,2} & \cdots & \hat{a}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{a}_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \vdots \\ \hat{c}_n \end{bmatrix}$$

Gauss Elimination

8.2.1 หลักการของ Gauss Elimination

$$
\begin{bmatrix}
a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\
a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\
a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\
a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} \\
a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5}
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}
=
\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}
$$

$$\Downarrow$$

$$
\begin{bmatrix}
a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\
0 & a'_{2,2} & a'_{2,3} & a'_{2,4} & a'_{2,5} \\
0 & a'_{3,2} & a'_{3,3} & a'_{3,4} & a'_{3,5} \\
0 & a'_{4,2} & a'_{4,3} & a'_{4,4} & a'_{4,5} \\
0 & a'_{5,2} & a'_{5,3} & a'_{5,4} & a'_{5,5}
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}
=
\begin{bmatrix} c_1 \\ c'_2 \\ c'_3 \\ c'_4 \\ c'_5 \end{bmatrix}
$$

Gauss Elimination

8.2.1 หลักการของ Gauss Elimination

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ 0 & a_{2,2}^{/} & a_{2,3}^{/} & a_{2,4}^{/} & a_{2,5}^{/} \\ 0 & a_{3,2}^{/} & a_{3,3}^{/} & a_{3,4}^{/} & a_{3,5}^{/} \\ 0 & a_{4,2}^{/} & a_{4,3}^{/} & a_{4,4}^{/} & a_{4,5}^{/} \\ 0 & a_{5,2}^{/} & a_{5,3}^{/} & a_{5,4}^{/} & a_{5,5}^{/} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} c_1 \\ c'_2 \\ c'_3 \\ c'_4 \\ c'_5 \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ 0 & a_{2,2}^{/} & a_{2,3}^{/} & a_{2,4}^{/} & a_{2,5}^{/} \\ 0 & 0 & a_{3,3}^{//} & a_{3,4}^{//} & a_{3,5}^{//} \\ 0 & 0 & a_{4,3}^{//} & a_{4,4}^{//} & a_{4,5}^{//} \\ 0 & 0 & a_{5,3}^{//} & a_{5,4}^{//} & a_{5,5}^{//} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} c_1 \\ c'_2 \\ c''_3 \\ c''_4 \\ c''_5 \end{bmatrix}$$

Gauss Elimination

8.2.1 หลักการของ Gauss Elimination

$$
\begin{bmatrix}
a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\
0 & a'_{2,2} & a'_{2,3} & a'_{2,4} & a'_{2,5} \\
0 & 0 & a''_{3,3} & a''_{3,4} & a''_{3,5} \\
0 & 0 & a''_{4,3} & a''_{4,4} & a''_{4,5} \\
0 & 0 & a''_{5,3} & a''_{5,4} & a''_{5,5}
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}
=
\begin{bmatrix} c_1 \\ c'_2 \\ c''_3 \\ c''_4 \\ c''_5 \end{bmatrix}
$$

$$\Downarrow$$

$$
\begin{bmatrix}
a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\
0 & a'_{2,2} & a'_{2,3} & a'_{2,4} & a'_{2,5} \\
0 & 0 & a''_{3,3} & a''_{3,4} & a''_{3,5} \\
0 & 0 & 0 & a'''_{4,4} & a'''_{4,5} \\
0 & 0 & 0 & a'''_{5,4} & a'''_{5,5}
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}
=
\begin{bmatrix} c_1 \\ c'_2 \\ c''_3 \\ c'''_4 \\ c'''_5 \end{bmatrix}
$$

Gauss Elimination

8.2.1 หลักการของ Gauss Elimination

$$
\begin{bmatrix}
a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\
0 & a'_{2,2} & a'_{2,3} & a'_{2,4} & a'_{2,5} \\
0 & 0 & a''_{3,3} & a''_{3,4} & a''_{3,5} \\
0 & 0 & 0 & a'''_{4,4} & a'''_{4,5} \\
0 & 0 & 0 & a'''_{5,4} & a'''_{5,5}
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}
=
\begin{bmatrix} c_1 \\ c'_2 \\ c''_3 \\ c'''_4 \\ c'''_5 \end{bmatrix}
$$

$$\Downarrow$$

$$
\begin{bmatrix}
a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\
0 & a'_{2,2} & a'_{2,3} & a'_{2,4} & a'_{2,5} \\
0 & 0 & a''_{3,3} & a''_{3,4} & a''_{3,5} \\
0 & 0 & 0 & a^{(3)}_{4,4} & a^{(3)}_{4,5} \\
0 & 0 & 0 & 0 & a^{(4)}_{5,5}
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}
=
\begin{bmatrix} c_1 \\ c'_2 \\ c''_3 \\ c^{(3)}_4 \\ c^{(4)}_5 \end{bmatrix}
$$

Back Substitution

8.2.1 หลักการของ Gauss Elimination

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ 0 & a'_{2,2} & a'_{2,3} & a'_{2,4} & a'_{2,5} \\ 0 & 0 & a''_{3,3} & a''_{3,4} & a''_{3,5} \\ 0 & 0 & 0 & a^{(3)}_{4,4} & a^{(3)}_{4,5} \\ 0 & 0 & 0 & 0 & a^{(4)}_{5,5} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} c_1 \\ c'_2 \\ c''_3 \\ c^{(3)}_4 \\ c^{(4)}_5 \end{bmatrix}$$

$$\Downarrow$$

$$a_{1,1}x_1 + a_{1,2}x_2 + a_{1,3}x_3 + a_{1,4}x_4 + a_{1,5}x_5 = c_1$$

$$a'_{2,2}x_2 + a'_{2,3}x_3 + a'_{2,4}x_4 + a'_{2,5}x_5 = c'_2$$

$$a''_{3,3}x_3 + a''_{3,4}x_4 + a''_{3,5}x_5 = c''_3$$

$$a^{(3)}_{4,4}x_4 + a^{(3)}_{4,5}x_5 = c^{(3)}_4$$

$$a^{(4)}_{5,5}x_5 = c^{(4)}_5$$

Back Substitution

$$x_i = \frac{c_i^{(i-1)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j}{a_{ii}^{(i-1)}}$$

$$\begin{aligned}
a_{1,1}x_1 + a_{1,2}x_2 + a_{1,3}x_3 + a_{1,4}x_4 + a_{1,5}x_5 &= c_1 \\
a_{2,2}^{/}x_2 + a_{2,3}^{/}x_3 + a_{2,4}^{/}x_4 + a_{2,5}^{/}x_5 &= c'_2 \\
a_{3,3}^{//}x_3 + a_{3,4}^{//}x_4 + a_{3,5}^{//}x_5 &= c''_3 \\
a_{4,4}^{(3)}x_4 + a_{4,5}^{(3)}x_5 &= c_4^{(3)} \\
a_{5,5}^{(4)}x_5 &= c_5^{(4)}
\end{aligned}$$

$$\Downarrow$$

$$x_5 = \frac{c_5^{(4)}}{a_{5,5}^{(4)}}$$

$$x_4 = \frac{c_4^{(3)} - a_{4,5}^{(3)}x_5}{a_{4,4}^{(3)}}$$

$$x_3 = \frac{c''_3 - a_{3,4}^{//}x_4 - a_{3,5}^{//}x_5}{a_{3,3}^{//}}$$

$$x_2 = \frac{c'_2 - a_{2,3}^{/}x_3 - a_{2,4}^{/}x_4 - a_{2,5}^{/}x_5}{a_{2,2}^{/}}$$

$$x_1 = \frac{c_1 - a_{1,2}x_2 - a_{1,3}x_3 - a_{1,4}x_4 - a_{1,5}x_5}{a_{1,1}}$$

Gauss Elimination

การคูณสมการบรรทัดที่หนึ่ง ด้วยค่า a_{21}/a_{11} และได้

$$a_{21}x_1 + \frac{a_{21}}{a_{11}}a_{12}x_2 + \cdots + \frac{a_{21}}{a_{11}}a_{1n}x_n = \frac{a_{21}}{a_{11}}c_1$$

จากนั้นนำค่าที่ได้ หักลบออกจากสมการในบรรทัดที่สอง ทำให้ x_1 หายไป และได้สมการในบรรทัดที่สองใหม่เป็น

$$(a_{22} - \frac{a_{21}}{a_{11}}a_{12})x_2 + \cdots + (a_{2n} - \frac{a_{21}}{a_{11}}a_{1n})x_n = c_2 - \frac{a_{21}}{a_{11}}c_1$$

ด้วยการเปลี่ยน Variable สมการสามารถเขียนในรูปของ

$$a'_{22}\,x_2 + \cdots + a'_{2n}\,x_n = c'_2$$

Gauss Elimination

จากนั้นทำการกำจัด Unknown ตัวแรกของแถวที่ 3, 4 จนถึงแถวสุดท้าย โดยการคูณด้วย Factor $a_{31}/a_{11}, a_{41}/a_{11},\ldots.$ ในแถวแรก และนำมาหักออกจากแถวที่ต้องการ เราจะได้

$$
\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = c_1 \\
0 + a'_{22}\,x_2 + \cdots + a'_{2n}\,x_n = c'_2 \\
\qquad\quad . \qquad\quad . \qquad\quad . \\
\qquad\quad . \qquad\quad . \qquad\quad . \\
\qquad\quad . \qquad\quad . \qquad\quad . \\
0 + a'_{n2}\,x_2 + \cdots + a'_{nn}\,x_n = c'_n
\end{array}
$$

Gauss Elimination

ขบวนการสามารถกระทำซ้ำ โดยการกำจัด Unknown ที่เหลือทีละตัว โดยเริ่มจากแถวถัดไป (กำจัด x_1 เริ่มจากแถวที่สอง , กำจัด x_2 เริ่มจากแถวที่สาม จนถึงกำจัด x_{n-1} ออกจากแถวที่ n) สุดท้ายเราจะเหลือแค่ Upper Triangular Matrix

$$
\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = c_1 \\
0 + a'_{22}\,x_2 + a'_{23}\,x_3 + \cdots + a'_{2n}\,x_n = c'_2 \\
0 + \quad 0 \quad + a''_{33}\,x_3 + \cdots + a''_{2n}\,x_n = c''_3 \\
\qquad \cdot \qquad \cdot \qquad \cdot \\
\qquad \cdot \qquad \cdot \qquad \cdot \\
\qquad \cdot \qquad \cdot \qquad \cdot \\
0 + \quad 0 \quad + \cdots + \quad 0 \quad + a_{nn}^{(n-1)}x_n = c_n^{(n-1)}
\end{array}
$$

Gauss Elimination

ยกตัวอย่างกรณีของ 3 Unknown Equation จาก

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c_1 \\ c'_2 \\ c''_3 \end{bmatrix}$$

Gauss Elimination

เมื่อได้ดังนี้แล้ว เราสามารถใช้ขบวนการ Back-Substitution แทนค่าย้อนกลับตั้งแต่ x_3 ดังนี้

$$x_3 = \frac{c''_3}{a''_{33}}$$

$$x_2 = \frac{c'_2 - a'_{23}\,x_3}{a'_{22}}$$

$$x_1 = \frac{c_1 - a_{12}x_2 - a_{13}x_3}{a_{11}}$$

Gauss Elimination Alg

- Elimination by Forward Substitution

Forward Substitution Pseudo Code: (FORTRAN STYLE)

```
DOFOR k = 1 to n-1
   DOFOR i = k+1 to n
      factor = a(i,k)/a(k,k)
      DOFOR j = k+1 to n
         a(i,j) = a(i,j)-factor*a(k,j)
      ENDDO
      c(i) = c(i)-factor*c(k)
   ENDDO
ENDDO
```

Gauss Elimination Alg

- Back-Substitution

สำหรับขั้นตอนของ Back-Substitution เราเริ่มจาก

$$x_n = \frac{c_n^{(n-1)}}{a_{nn}^{(n-1)}}$$

และเราสามารถพิสูจน์ได้ว่า Unknown ที่เหลือสามารถหาได้จากสมการในรูป

$$x_i = \frac{c_i^{(i-1)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j}{a_{ii}^{(i-1)}}$$

Gauss Elimination Alg

- Back-Substitution

$$x_i = \frac{c_i^{(i-1)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j}{a_{ii}^{(i-1)}}$$

Backward-Substitution Pseudo Code:

```
x(n) = c(n)/a(n,n)
DOFOR i = n-1 to 1 step -1
   sum = 0
   DOFOR j = i+1 to n
      sum = sum + a(i,j)*x(j)
   ENDDO
   x(i) = (c(i)-sum)/a(i,i)
ENDDO
```

Example 8.1

Example 8.1 จงใช้ Gauss Elimination เพื่อแก้สมการ

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

ต่อไปนี้เป็นผลลัพธ์จากการ Run Program

```
factor = 0.0333
a =
    3.0000   -0.1000   -0.2000
         0    7.0033   -0.2933
         0   -0.1900   10.0200
c =
    7.8500
  -19.5617
   70.6150
```

Example 8.1

```
factor = 0.0333
a =
    3.0000   -0.1000   -0.2000
         0    7.0033   -0.2933
         0   -0.1900   10.0200
c =
    7.8500
  -19.5617
   70.6150

factor =  -0.0271
a =
    3.0000   -0.1000   -0.2000
         0    7.0033   -0.2933
         0         0   10.0120
c =
    7.8500
  -19.5617
   70.0843
```

Example 8.1

- Back Substitution

```
x =
         0
         0
    7.0000
x =
         0
   -2.5000
    7.0000
x =
    3.0000
   -2.5000
    7.0000
```

ปัญหาของ Gauss Elimination

ปัญหาเรื่อง Divide By Zero:

ในการกำจัด Unknown ออกนั้น เราต้องหาค่า Factor ที่มีการหารด้วย a_{ii} และในกรณีที่สมการดั้งเดิม ค่า a_{ii} มีค่าเป็นศูนย์ เราจะเจอปัญหาของ Divide by Zero ยกตัวอย่างเช่นในกรณีของ

$$2x_2 + 3x_3 = 8$$
$$4x_1 + 6x_2 + 7x_3 = -3$$
$$2x_1 + x_2 + 6x_3 = 5$$

แม้ว่าบางครั้งเราไม่เกิดปัญหา Divide by Zero แต่ถ้าค่า Coefficient เหล่านี้มีค่าใกล้ศูนย์มาก จะมีผลต่อ Accuracy ของคำตอบ วิธีการแก้ไขคือวิธีการที่เรียก Pivoting(รายละเอียดจะไม่กล่าวถึง) โดยสลับแถวของสมการ เช่น เปลี่ยนสมการในรูป

$$4x_1 + 6x_2 + 7x_3 = -3$$
$$2x_2 + 3x_3 = 8$$
$$2x_1 + x_2 + 6x_3 = 5$$

ปัญหาของ Gauss Elimination

ปัญหา Round-Off Error:

เนื่องจากการคำนวณเป็น Iteration และจะเกิด Error จากการปัดเศษของเลขหลักสุดท้ายในแต่ละขั้นตอน และลักษณะการทำงานของ Algorithm จะทำให้เกิดการสะสมของ Error เกิดขึ้น โดยเฉพาะสมการที่มีขนาดใหญ่ วิธีการแก้ไขคือการเพิ่ม Significant Digit ในการคำนวณ แต่วิธีที่ดีกว่าคือใช้คณิตศาสตร์ในการคำนวณที่ไม่ต้องมีการปัดเศษ คือการคำนวณแบบใช้เศษส่วน อย่างไรก็ตาม รายละเอียดเราจะไม่กล่าวถึง

ปัญหาของ Gauss Elimination

Ill-Conditioned system เกิดจากเมื่อมีการเปลี่ยนแปลงเล็กน้อยในค่า Coefficient ของระบบ จะทำให้ Solution ของระบบเปลี่ยนแปลงไปมาก ดังนั้นเมื่อมี Round-Off Error เพียงเล็กน้อย จะมีผลให้ Solution เปลี่ยนแปลงไป เนื่องจาก Round-Off Error ที่เกิด จะเปลี่ยนแปลงค่าที่แท้จริงของ Coefficient ที่คำนวณได้ ยกตัวอย่างสมการ

$$x_1 + 2x_2 = 10$$
$$1.1x_1 + 2x_2 = 10.4$$

ซึ่งมี Solution $x_1 = 4, x_2 = 3$ ถ้า Coefficient ของ x_1 ในสมการที่สองเปลี่ยนเพียงเล็กน้อย จาก 1.1 เป็น 1.05 ในสมการ

$$x_1 + 2x_2 = 10$$
$$1.05x_1 + 2x_2 = 10.4$$

เราจะได้ Solution ที่เปลี่ยนแปลงไปมากคือ $x_1 = 8, x_2 = 1$ สังเกตว่า ค่าที่เปลี่ยนไปของ Coefficient มีเพียงเล็กน้อยเท่านั้น ซึ่งอาจจะเกิดจาก Round-Off Error หรือการเก็บข้อมูลที่ไม่แน่นอน

ปัญหาของ Gauss Elimination

เป็นการยากที่เราจะตรวจสอบว่าระบบเป็น Ill-Conditioned หรือไม่ สิ่งหนึ่งที่ดูได้คือการหาค่า Determinant ของ Matrix **A** ที่มีค่าใกล้เคียงศูนย์ แต่ว่ามันควรจะใกล้เคียงศูนย์แค่ไหนถึงจะเป็น Ill-Conditioned นั้นยากที่จะบอกได้ เพราะเราสามารถคูณสมการทั้งหมดด้วยค่า Constant ซึ่งค่า Determinant ก็จะมีค่าเป็นจำนวนเท่าตาม Factor ที่มาคูณด้วย แต่สมการที่ได้ยังคงเป็นสมการเดิม และมีคำตอบเดิม

Gauss-Jordan Method

Gauss-Jordan Method เป็นวิธีที่ปรับมาจากวิธีของ Gauss Elimination ซึ่งจะทำการกำจัด Unknown ให้เหลือเพียงตัวเดียวในแต่ละสมการ ทำให้เราได้คำตอบทันที ดังนั้นในขบวนการ Elimination เราจะเหลือ Identity Matrix และไม่ต้องมี Back-Substitution เช่นจาก

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

เมื่อใช้ Gauss-Jordan Method เราจะได้

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c_1^n \\ c_2^n \\ c_3^n \end{bmatrix} \Rightarrow x_1 = c_1^n, x_2 = c_2^n, x_3 = c_3^n$$

Gauss-Jordan Method

วิธีการก็คือ ในแต่ละ Step เมื่อเรากำจัด Unknown ออกแล้ว เราทำการ Scale ค่า a_{ii} ให้เท่ากับ 1 ในสมการที่ i จากนั้นจะนำสมการที่ $i+1$ มาทำการ Scale และไปลบออกจากสมการที่ i เพื่อกำจัด Unknown ในสมการ i ออกด้วย (ดูตัวอย่าง 7.2)

Algorithm สามารถเขียนเป็น Pseudo Code ได้ดังนี้(สมมุติ Matrix มีขนาด $n \times n+1$ โดย Column สุดท้ายคือ $\mathbf{C}$)

```
DOFOR k = 1 to n
   dummy = a(k,k)
   DOFOR j = 1 to n+1
      a(k,j) = a(k,j)/dummy
   ENDDO
   DOFOR i = 1 to n
      IF (i <> k)
         dummy = a(i,k)
         DOFOR j = 1 to n+1
            a(i,j) = a(i,j) - dummy*a(k,j)
         ENDDO
      ENDIF
   ENDDO
ENDDO
```

Example 8.2

Example 8.2 จงใช้ Gauss-Jordan เพื่อแก้สมการ

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$

$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$

$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

Example 8.2

$$\begin{bmatrix} 3 & -0.1 & -0.2 & | & 7.85 \\ 0.1 & 7 & -0.3 & | & -19.3 \\ 0.3 & -0.2 & 10 & | & 71.4 \end{bmatrix}$$

$$(R2)-(R1)\times .1/3 \rightarrow (R2)$$

$$\begin{bmatrix} 3 & -0.1 & -0.2 & | & 7.85 \\ 0 & 7.0033 & -0.2933 & | & -19.5617 \\ 0.3 & -0.2 & 10 & | & 71.4 \end{bmatrix}$$

$$(R3)-(R1)\times .3/3 \rightarrow (R3)$$

$$\begin{bmatrix} 3 & -0.1 & -0.2 & | & 7.85 \\ 0 & 7.0033 & -0.2933 & | & -19.5617 \\ 0 & -0.1900 & 10.0200 & | & 70.6150 \end{bmatrix}$$

$$(R1) = (R1)/3$$

$$\begin{bmatrix} 1 & -0.0333 & -0.0667 & | & 2.6167 \\ 0 & 7.0033 & -0.2933 & | & -19.5617 \\ 0 & -0.1900 & 10.0200 & | & 70.6150 \end{bmatrix}$$

Example 8.2

$$\begin{bmatrix} 1.0000 & -0.0333 & -0.0667 & | & 2.6167 \\ 0 & 7.0033 & -0.2933 & | & -19.5617 \\ 0 & -0.1900 & 10.0200 & | & 70.6150 \end{bmatrix}$$

$$(R3) = (R3) - (R2) \times (-.19) / 7.0033$$

$$\begin{bmatrix} 1.0000 & -0.0333 & -0.0667 & | & 2.6167 \\ 0 & 7.0033 & -0.2933 & | & -19.5617 \\ 0 & 0 & 10.0120 & | & 70.0843 \end{bmatrix}$$

$$(R2) = (R2) / 7.0033$$

$$\begin{bmatrix} 1.0000 & -0.0333 & -0.0667 & | & 2.6167 \\ 0 & 1 & -0.0419 & | & -2.7932 \\ 0 & 0 & 10.0120 & | & 70.0843 \end{bmatrix}$$

$$(R1) = (R1) - (R2) \times (-.0333)$$

$$\begin{bmatrix} 1.0000 & 0 & -0.0681 & | & 2.5236 \\ 0 & 1.0000 & -0.0419 & | & -2.7932 \\ 0 & 0 & 10.0120 & | & 70.0843 \end{bmatrix}$$

Example 8.2

$$\begin{bmatrix} 1.0000 & 0 & -0.0681 & | & 2.5236 \\ 0 & 1.0000 & -0.0419 & | & -2.7932 \\ 0 & 0 & 10.0120 & | & 70.0843 \end{bmatrix}$$

$$(R3) = (R3)/10.012$$

$$\begin{bmatrix} 1.0000 & 0 & -0.0681 & | & 2.5236 \\ 0 & 1.0000 & -0.0419 & | & -2.7932 \\ 0 & 0 & 1.0000 & | & 7.0000 \end{bmatrix}$$

$$(R2) = (R2) - (R3) * (-.0419)$$

$$\begin{bmatrix} 1.0000 & 0 & -0.0681 & | & 2.5236 \\ 0 & 1.0000 & 0 & | & -2.5000 \\ 0 & 0 & 1.0000 & | & 7.0000 \end{bmatrix}$$

$$(R1) = (R1) - (R3) * (-.0681)$$

$$\begin{bmatrix} 1.0000 & 0 & 0 & | & 3.0000 \\ 0 & 1.0000 & 0 & | & -2.5000 \\ 0 & 0 & 1.0000 & | & 7.0000 \end{bmatrix}$$

Example 8.2

$$\begin{bmatrix} 1.0000 & 0 & 0 & | & 3.0000 \\ 0 & 1.0000 & 0 & | & -2.5000 \\ 0 & 0 & 1.0000 & | & 7.0000 \end{bmatrix}$$

$$\mathbf{A'X} = \mathbf{C'}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -2.5 \\ 7 \end{bmatrix}$$

$$\begin{aligned} x_1 \qquad\qquad &= 3 \\ x_2 \qquad &= -2.5 \\ x_3 &= 7 \end{aligned}$$

Example 8.2

```
k = 0
a | c =

     3.0000   -0.1000   -0.2000  |   7.8500
     0.1000    7.0000   -0.3000  | -19.3000
     0.3000   -0.2000   10.0000  |  71.4000
k =1
a | c =

     1.0000   -0.0333   -0.0667  |   2.6167
          0    7.0033   -0.2933  | -19.5617
          0   -0.1900   10.0200  |  70.6150

k = 2
a | c =

     1.0000         0   -0.0681  |  2.5236
          0    1.0000   -0.0419  | -2.7932
          0         0   10.0120  | 70.0843

k = 3
a | c =

     1.0000         0         0  |  3.0000
          0    1.0000         0  | -2.5000
          0         0    1.0000  |  7.0000
```

การหา Matrix Inverse ด้วย GJ

ปกติแล้วการหา Inverse ของ Matrix คือการหา Solution ของสมการ $\mathbf{AX} = \mathbf{I}$ เช่นในกรณีของ 3×3 Matrix เราได้

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ซึ่งที่จริงแล้วเราได้ $\mathbf{X} = \mathbf{A}^{-1}$ นั่นเอง และในกรณีนี้ Unknown $\mathbf{X}$ เป็น Matrix ไม่ใช่ Vector

ถ้าพิจารณาให้ดีจากสมการข้างบน โดยใช้หลักการคูณกันของ Matrix การหา Inverse ก็คือการแก้สมการ Linear n สมการพร้อมๆกัน ในตัวอย่างข้างบนเราแก้สมการ 3 สมการดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

ซึ่งคำตอบของสมการนี้ เราจะได้ Column แรกของ Inverse ของ Matrix จากนั้นเราหา Solution ของ

การหา Matrix Inverse ด้วย GJ

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

และคำตอบจะได้ Column ที่สองของ Inverse ของ Matrix สุดท้ายเราแก้สมการ

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

เราจะได้ Column สุดท้ายของ Matrix Inverse

การหา Matrix Inverse ด้วย GJ

จากหลักการที่อธิบายข้างบน เราสามารถใช้ Algorithm ใดๆก็ได้ที่ใช้ในการแก้สมการ Linear Equation มาดัดแปลงหา Inverse ของ Matrix อย่างไรก็ตาม เราจะต้องกระทำถึง n ครั้ง เพื่อจะได้คำตอบ และจะใช้การคำนวณมากสำหรับการหา Inverse ของ Matrix ขนาดใหญ่

ถ้าเรามาดูวิธีของ Gauss-Elimination เมื่อนำมาหา Matrix Inverse จะพบว่าส่วนของ Forward Substitution สามารถกระทำไปพร้อมกันได้หมด โดยเราตั้งสมการดังนี้(ใช้ตัวอย่างเดิม)

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ซึ่ง Step ของ Forward Elimination สามารถกระทำทีเดียวพร้อมๆกัน และเราจะได้

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix}\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ c'_{21} & c'_{22} & c'_{23} \\ c''_{31} & c''_{32} & c''_{33} \end{bmatrix}$$

การหา Matrix Inverse ด้วย GJ

เทียบสัดส่วนของการคำนวณแล้ว ส่วนของ Forward Elimination จะใช้คิดเป็นอัตราส่วนมากกว่า 90 % แต่เมื่อนำมาใช้ในการหา Inverse แล้ว จำนวนของ Operation จะมีมากเพิ่มขึ้นในส่วนของ Back-Substitution เนื่องจากเราต้องทำถึง n ครั้ง ทำให้ไม่เหมาะสมสำหรับ Matrix ขนาดใหญ่

ในกรรมวิธีของ Gauss-Jordan นั้น เนื่องจากไม่มีการทำ Back-Substitution ทำให้การดัดแปลง Algorithm เมื่อมาใช้หา Inverse ของ Matrix สามารถกระทำไปพร้อมๆกันภายในขั้นตอนเดียว ซึ่งเราจะได้โปรแกรมที่รวบรัด และรวดเร็ว ซึ่งเป็นวิธีที่นิยมมากอันหนึ่งในการหา Inverse ของ Matrix

ในการนำกรรมวิธีของ Gauss-Jordan มาใช้หา Inverse ของ Matrix นั้น เราเขียน **C** ให้เป็น Identity Matrix แทนที่จะเป็น Column Vector ในรูปของ

การหา Matrix Inverse ด้วย GJ

$$[\mathbf{A}:\mathbf{C}] = \left[\begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{array}\right]$$

จากนั้นเมื่อเราใช้ Gauss-Jordan เพื่อลดรูป Matrix $\mathbf{A}$ ให้อยู่ในรูป Identity Matrix พร้อมๆกันนั้น เราจะทำการเปลี่ยน Identity Matrix เดิมให้อยู่ในรูปของ Inverse ของ Matrix $\mathbf{A}$ กล่าวคือ

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & a_{11}^{-1} & a_{12}^{-1} & a_{13}^{-1} \\ 0 & 1 & 0 & a_{21}^{-1} & a_{22}^{-1} & a_{23}^{-1} \\ 0 & 0 & 1 & a_{31}^{-1} & a_{32}^{-1} & a_{33}^{-1} \end{array}\right]$$

ซึ่งจะเห็นได้ว่าวิธีของ Gauss-Jordan มีความรวบรัดมากกว่า

Example 8.3

Example 8.3 จงใช้ Gauss-Jordan เพื่อหา Inverse ของ

$$\mathbf{A} = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}$$

Example 8.3

```
k = 0
    3.0000   -0.1000   -0.2000    1.0000         0         0
    0.1000    7.0000   -0.3000         0    1.0000         0
    0.3000   -0.2000   10.0000         0         0    1.0000
k = 1
ans =

    1.0000   -0.0333   -0.0667    0.3333         0         0
         0    7.0033   -0.2933   -0.0333    1.0000         0
         0   -0.1900   10.0200   -0.1000         0    1.0000
k = 2
ans =
    1.0000         0   -0.0681    0.3332    0.0048         0
         0    1.0000   -0.0419   -0.0048    0.1428         0
         0         0   10.0120   -0.1009    0.0271    1.0000
k =  3
ans =
    1.0000         0         0    0.3325    0.0049    0.0068
         0    1.0000         0   -0.0052    0.1429    0.0042
         0         0    1.0000   -0.0101    0.0027    0.0999
```

Example 8.3

```
ans =
    1.0000         0         0    0.3325    0.0049    0.0068
         0    1.0000         0   -0.0052    0.1429    0.0042
         0         0    1.0000   -0.0101    0.0027    0.0999
```

$$\begin{bmatrix} 3 & -.1 & -.2 \\ .1 & 7 & -.3 \\ .3 & -.2 & 10 \end{bmatrix}^{-1} = \begin{bmatrix} .3325 & .0049 & .0068 \\ -.0052 & .1429 & .0042 \\ -.0101 & .0027 & .0999 \end{bmatrix}$$

จำนวนการคำนวณจะใช้น้อยกว่าวิธีทาง Analytic Method มาก

$$\mathbf{A}^{-1} = \frac{Adj\mathbf{A}}{|\mathbf{A}|}$$

Iterative Method and Gauss-Seidel

Gauss-Jordan สามารถใช้หา Inverse ของ Matrix หรือแก้สมการที่มีถึง 100 Unknown ถ้าเราตั้ง Significant Digit ให้เหมาะสม และถ้าระบบไม่เป็น Ill-Conditioned อย่างไรก็ตาม ในระบบที่มีขนาดใหญ่กว่านี้ จะเกิดปัญหาของ Round-Off Error และวิธีการแบบ Elimination จะไม่เหมาะสม เราจำเป็นจะต้องพึ่งวิธีทาง Iterative หรือวิธี Approximate แทน เนื่องจากวิธีทาง Iterative นั้นสามารถกระทำต่อได้เรื่อยๆ และจะหยุดเมื่อได้ Estimate Error น้อยกว่าค่า e_s ที่กำหนด(ดูบทที่ 7) ผิดกับวิธีของ Elimination ที่จะมี Loop ของการคำนวณที่แน่นอน ตามขนาดของระบบ อย่างไรก็ตามพึงเข้าใจก่อนว่าวิธีทาง Iterative นั้นจัดว่าเป็นวิธี Approximate เพราะเราจะไม่ได้คำตอบที่แท้จริง เพียงแต่ว่าคำตอบจะเข้าใกล้ค่าที่แท้จริงเรื่อยๆ ถ้าระบบ Converge และในทางทฤษฎีแล้ว เราจะต้อง Run ถึง Infinity Iteration ถึงจะได้คำตอบที่แท้จริง

วิธีของ Gauss-Seidel เป็นวิธีทาง Iterative สำหรับการแก้ปัญหาของระบบของสมการที่นิยมใช้กันมากที่สุด โดยทำการหา Solution จากระบบสมการเดิมด้วยการหาค่า x_1 ในสมการที่หนึ่ง ค่า x_2 ในสมการที่สอง จนถึง x_n ในสมการสุดท้าย จากนั้นจึงทำ Iteration ซึ่งความจริงก็คือวิธีของ Simple One-Point Iteration สำหรับหลาย Unknown และมีรายละเอียดต่อไปนี้

Gauss-Seidel

จากสมการ ของระบบ Linear Algebraic Equation ที่มี n Unknown $x_1, x_2, \ldots, x_n$ สามารถเขียนในรูป Matrix

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= c_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= c_2 \\
\cdot \qquad\quad \cdot \qquad\quad \cdot & \\
\cdot \qquad\quad \cdot \qquad\quad \cdot & \\
\cdot \qquad\quad \cdot \qquad\quad \cdot & \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= c_n
\end{aligned}
$$

$$
= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \mathbf{AX} = \mathbf{C}
$$

Gauss-Seidel

เมื่อเราแก้สมการหาค่า x_i จากแถวที่ i เราได้

$$x_1 = \frac{c_1 - a_{12}x_2 - a_{13}x_3 - \cdots - a_{1n}x_n}{a_{11}}$$

$$x_2 = \frac{c_2 - a_{21}x_1 - a_{23}x_3 - \cdots - a_{2n}x_n}{a_{22}}$$

$$x_3 = \frac{c_3 - a_{31}x_1 - a_{32}x_2 - \cdots - a_{3n}x_n}{a_{33}}$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$x_n = \frac{c_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{nn-1}x_{n-1}}{a_{nn}}$$

Gauss-Seidel

จากนั้นเราเริ่มขบวนการโดยเดาค่า x_i เริ่มต้น ทั้งหมด n ค่า ซึ่งปกติจะเริ่มจากให้ x_i เริ่มต้นเป็นศูนย์ทั้งหมด และใน Iteration แรก เราคำนวณค่าใหม่ของ $x_1^1, x_2^1, \cdots, x_n^1$ โดยตัวเลข Superscript หมายถึงค่า x_i^j ที่ Estimate ใหม่ใน Iteration ที่ j และโปรแกรมจะ Converge ถ้าค่า Estimate Error สำหรับทุกๆ Unknown มีค่าลดลง โปรแกรมจะหยุดเมื่อเราได้ Estimate Error ทุกตัวน้อยกว่าค่า e_s ที่ตั้งไว้

ค่า Estimate Error สำหรับ Unknown x_i^j ที่ Iteration j สามารถคำนวณได้จาก

$$\left|e_{a,j}\right| = \left|\frac{x_i^j - x_i^{j-1}}{x_i^j}\right| \times 100\%$$

เนื่องจาก Gauss-Seidel Algorithm อาจจะไม่ Converge หรือ Converge ช้าเราสามารถปรับปรุงวิธีการให้ Converge เร็วขึ้น โดยแทนที่จะใช้ค่าใหม่ที่ Estimate ได้โดยตรง แต่เราใช้ Weight Sum ของค่าใหม่กับค่าเดิม ซึ่งเราเรียกว่าเป็นการทำ Relaxation ดังนี้

Gauss-Seidel

$$x_i^{New} = \lambda x_i^{New} + (1-\lambda)x_i^{Old}$$

โดยค่า λ มีค่าอยู่ระหว่าง ศูนย์และสอง ถ้า $\lambda = 1$ เราจะได้วิธีปกติของ Gauss-Seidel ที่อธิบายมาแล้ว แต่ถ้า λ มีค่าอยู่ระหว่าง 0 และ 1 เราเรียกว่าเป็น ***Underrelaxation*** และจะใช้ในกรณีที่จะทำให้ระบบที่ไม่ Converge เป็นระบบที่สามารถ Converge หรือ ในกรณีที่คำตอบ Converge ช้าเนื่องจากมีการ Oscillation ของคำตอบกลับไปกลับมา และถ้าค่า λ มีค่าอยู่ระหว่าง 1 ถึง 2 เราเรียกว่าเป็น Overrelaxation เพราะเป็นการเพิ่มค่าน้ำหนักให้กับค่าใหม่ที่มากกว่าเดิม กรณีนี้จะใช้กับระบบที่มีการ Converge อยู่แล้ว แต่จะช่วยให้มัน Converge ได้เร็วยิ่งขึ้น

Algorithm ของ Gauss Seidel เขียนเป็น Pseudo Code ได้ดังนี้ (รวมถึงการ Relaxation ด้วย)

```
DOFOR i = 1 to n
   dummy = a(i,i)
   DOFOR j = 1 to n
      a(i,j) = a(i,j)/dummy
   ENDDO
   c(i) = c(i)/dummy
ENDDO
sentinel = 0
iter = 0
DOWHILE (iter < maxit) and (sentinel = 0)
   sentinel = 1
   iter = iter + 1
   DOFOR i = 1 to n
      old = x(i)
      sum = c(i)
      DOFOR j = 1 to n
         IF i <> j
            sum = sum - a(i,j)*x(j)
         ENDIF
      ENDDO
      x(i) = Lambda*sum + (1-Lambda)*old
      IF (sentinel = 1)and (x(i) <> 0)
         ea = abs((x(i)-old)/x(i))*100
         IF ea > es
            sentinel = 0
         ENDIF
      ENDIF
   ENDDO
ENDDO
```

ในภาคผนวกของบทนี้แสดงตัวอย่างของ MATLAB Program ที่มีการ Relaxation

Gauss-Seidel: Ex 8.4

Example 8.4 จงใช้ Gauss-Seidel เพื่อแก้สมการ

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$

$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$

$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

เมื่อเรียงสมการใหม่เพื่อหาค่า Unknown เราได้

$$x_1 = \frac{7.85 + 0.1x_2 + 0.2x_3}{3}$$

$$x_2 = \frac{-19.3 - 0.1x_1 + 0.3x_3}{7}$$

$$x_3 = \frac{71.4 - 0.3x_1 + 0.2x_2}{10}$$

Iteration ที่ 1

Gauss-Seidel: Ex 8.4

เริ่มจาก $x_2 = 0, x_3 = 0$ เราคำนวณหา x_1 จากสมการแรก เราได้ $x_1 = 7.85/3 = 2.61667$

จากค่า $x_1 = 2.61667, x_3 = 0$ เราคำนวณ x_2 จากสมการที่ 2 เราได้ $x_2 = -2.79452$

และจาก $x_1 = 2.61667, x_2 = -2.79452$ เราคำนวณค่า x_3 จากสมการสุดท้าย ได้ $x_3 = 7.00561$

Iteration ที่ 2, 3, …

คำนวณเหมือนเดิม แต่ใช้ค่า x_1, x_2, x_3 ที่ได้ใหม่ล่าสุด แทนค่าลงในสมการที่หนึ่ง สอง และ สาม

ข้างล่างเป็นผลจากการ Run Program โดยตั้ง $e_s = 0.01\%$ และไม่มี Relaxation (โดยการตั้งค่า $\lambda = 1$)

Gauss-Seidel: Ex 8.4

$$x_1^0 = x_2^0 = x_3^0 = 0$$

$$x_1^1 = \frac{7.85 + 0.1x_2^0 + 0.2x_3^0}{3} = \frac{7.85 + 0.1(0) + 0.2(0)}{3} = \frac{7.85}{3} = 2.61667$$

$$x_2^1 = \frac{-19.3 - 0.1x_1^1 + 0.3x_3^0}{7} = \frac{-19.3 - 0.1(2.61667) + 0.3(0)}{7} = \frac{-19.56167}{7} = -2.79452$$

$$x_3^1 = \frac{71.4 - 0.3x_1^1 + 0.2x_2^1}{10} = \frac{71.4 - 0.3(2.61667) + 0.2(-2.79452)}{10} = \frac{70.056095}{10} = 7.00561$$

$$e_{a,1} = \left| \frac{2.61667 - 0}{2.61667} \times 100 \right| = 100\%$$

$$e_{a,1} = \left| \frac{-2.79452 - 0}{-2.79452} \times 100 \right| = 100\%$$

$$e_{a,1} = \left| \frac{7.00561 - 0}{7.00561} \times 100 \right| = 100\%$$

Gauss-Seidel: Ex 8.4

$$x_1^1 = 2.61667,\ x_2^1 = -2.79452,\ x_3^1 = 7.00561$$

$$x_1^2 = \frac{7.85 + 0.1x_2^1 + 0.2x_3^1}{3} = \frac{7.85 + 0.1(-2.79452) + 0.2(7.00561)}{3} = \frac{8.9716968}{3} = 2.99056$$

$$x_2^2 = \frac{-19.3 - 0.1x_1^2 + 0.3x_3^1}{7} = \frac{-19.3 - 0.1(2.99056) + 0.3(7.00561)}{7} = \frac{-17.49737}{7} = -2.49962$$

$$x_3^2 = \frac{71.4 - 0.3x_1^2 + 0.2x_2^2}{10} = \frac{71.4 - 0.3(2.99056) + 0.2(-2.49962)}{10} = \frac{70.00291}{10} = 7.00029$$

$$e_{a,2} = \left|\frac{2.99056 - 2.61667}{2.99056} \times 100\right| = 12.50\%$$

$$e_{a,2} = \left|\frac{-2.49962 - (-2.79452)}{-2.49962} \times 100\right| = 11.80\%$$

$$e_{a,2} = \left|\frac{7.00029 - 7.00561}{7.00029} \times 100\right| = 0.076\%$$

Gauss-Seidel: Ex 8.4

$$x_1^2 = 2.99056,\ x_2^2 = -2.49962,\ x_3^2 = 7.00029$$

$$x_1^3 = \frac{7.85 + 0.1x_2^2 + 0.2x_3^2}{3} = \frac{7.85 + 0.1(-2.49962) + 0.2(7.00029)}{3} = \frac{9.0001}{3} = 3.00003$$

$$x_2^3 = \frac{-19.3 - 0.1x_1^3 + 0.3x_3^2}{7} = \frac{-19.3 - 0.1(3.00003) + 0.3(7.00029)}{7} = \frac{-17.499916}{7} = -2.499988$$

$$x_3^3 = \frac{71.4 - 0.3x_1^3 + 0.2x_2^3}{10} = \frac{71.4 - 0.3(3.00003) + 0.2(-2.499988)}{10} = \frac{69.99999}{10} = 6.999999$$

$$e_{a,3} = \left|\frac{3.00003 - 2.99056}{3.00003} \times 100\right| = 0.3157\%$$

$$e_{a,3} = \left|\frac{-2.499988 - (-2.49962)}{-2.499988} \times 100\right| = 0.01472\%$$

$$e_{a,3} = \left|\frac{6.999999 - 7.00029}{6.999999} \times 100\right| = 0.004157\%$$

Gauss-Seidel: Ex 8.4

```
iter =  0

x =
      0
      0
      0

iter =  1
x =
    2.6167
   -2.7945
    7.0056
ea =
   100
   100
   100
iter =  2
x =
    2.9906
   -2.4996
    7.0003
ea =
   12.5023
   11.7977
    0.0760
```

Gauss-Seidel: Ex 8.4

```
iter =  3
x =
    3.0000
   -2.5000
    7.0000
ea =
    0.3158
    0.0145
    0.0042
iter =  4
x =
    3.0000
   -2.5000
    7.0000
ea =
    0.0011
    0.0005
    0.0000
```

Jacobi Method

เราจะจบวิธีของ Iterative Method ด้วยการกล่าวถึง Variation ของ Gauss-Seidel ที่ชื่อ Jacobi Method ซึ่งในวิธีของ Gauss-Seidel ที่กล่าวมานั้น เราใช้ค่า x_i^j และ x_i^{j-1} ที่คำนวณได้ล่าสุดในการหา x_{i+1}^j ดังนั้นถ้าระบบมีการ Converge เราจะได้ค่า Estimate ที่ดีที่สุดของ Unknown ตัวต่อไป และจะทำให้เกิดการ Converge ได้อย่างรวดเร็ว อย่างไรก็ตามถ้าเราใช้ชุดของ x_i^j ทั้งหมดมาทำการ Estimate ค่าชุดใหม่ของ x_i^{j+1} เราจะได้กรรมวิธีที่ชื่อ Jacobi Method

ที่จริงแล้ว Gauss-Seidel Method เป็นวิธีที่ปรับปรุงมาจาก Jacobi Method และปกติจะให้การ Converge ที่รวดเร็วกว่าและปกติเป็นวิธีที่นิยมใช้กัน แต่ก็มีบางกรณีที่วิธีของ Jacobi กลับให้ผลที่รวดเร็วกว่า

Convergence of Iterative Method

เมื่อพูดถึงการ Convergence ซึ่งเป็นสิ่งที่สำคัญสำหรับ Iterative Method เพราะวิธีนี้เป็นวิธีที่ปรับปรุงมาจากวิธีของ Simple One-Point Iteration สำหรับชุดของสมการ Linear ที่มี n Dimension เราสามารถพิสูจน์ได้ว่า Sufficient Condition ในการ Converge (แต่ไม่ใช่ Necessary Condition) นั้นคือถ้าระบบเป็น ***Diagonally Dominant*** กล่าวคือค่าของ Coefficient ที่จุด Pivot ทุกตัว(ค่า a_{ii}) ในระบบสมการมีค่ามากกว่าผลรวมของค่า Absolute ของ Coefficient อื่นในแถวเดียวกัน ดังนี้

$$\left|a_{ii}\right| > \sum_{i \neq j} \left|a_{ij}\right|$$

อย่างไรก็ตาม ในปัญหาทั่วไปแล้วปกติเป็นการยากที่จะสลับแถวของสมการหรือมีการบวกลบแถว เพื่อทำให้ระบบเป็น Diagonally Dominant รายละเอียดเราจะไม่กล่าวถึง

อีกวิธีหนึ่งที่จะทำให้ระบบ Converge ก็คือใช้การ Relaxation ซึ่งได้กล่าวมาแล้ว และการเลือกค่า λ ที่ถูกต้องถือว่าเป็นเรื่องสำคัญ รายละเอียดจะเกินเนื้อหาของวิชานี้

Break

- After Break
  - LU Decomposition

LU Decomposition

สรุปจากที่กล่าวมาแล้ว เรามีวิธีในการแก้ปัญหาระบบ Linear Algebraic Equation 2 แบบ คือวิธี Elimination Method (Gauss Elimination, Gauss-Jordan) และวิธี Iterative Method (Jacobi, Gauss-Seidel) ในส่วนนี้เราจะมากล่าวต่อในเรื่องของ Elimination Method ที่เรียกเทคนิคการทำ LU Decomposition

วิธีของ LU Decomposition จะมีประสิทธิภาพสูงกว่า เพราะจะปรับปรุงในส่วนของ Elimination Step และจะใช้ได้ดีกว่า Gauss-Jordan ในการหา Inverse ของ Matrix ทำให้วิธีนี้เป็นวิธีที่ใช้กันมากที่สุดในการแก้ปัญหาของระบบของ Linear Algebraic Equations

LU Decomposition

LU Decomposition จะเจอปัญหาเรื่อง Divide by Zero เช่นเดียวกับ Gauss Elimination และจะต้องมีการทำ Pivoting เพื่อป้องกัน อย่างไรก็ตามที่จะอธิบายต่อไปนี้จะถือว่า Coefficient Matrix มีค่า a_{ii} ทุกตัวไม่เท่ากับศูนย์

จากระบบของสมการ Linear เขียนในรูป Matrix ได้เป็น

$$\mathbf{AX} = \mathbf{C}$$

และสามารถจัดเรียงใหม่ได้เป็น

$$\mathbf{AX} - \mathbf{C} = \mathbf{0}$$

ถ้าสมมุติเราใช้ Gauss Elimination เปลี่ยน Coefficient Matrix ให้เป็น Upper Diagonal และทำการ Scale ให้ค่า Coefficient ในส่วน Diagonal มีค่าเท่ากับหนึ่ง ยกตัวอย่างในระบบที่มี 4 สมการ ดังนี้

LU Decomposition

$$\begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix}$$

เมื่อเขียนในรูปแบบของ Matrix เราจะได้

$$\mathbf{UX} - \mathbf{D} = \mathbf{0}$$

LU Decomposition

คราวนี้สมมุติว่าเรามี Lower Diagonal Matrix $\mathbf{L}$ ดังนี้

$$\mathbf{L} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix}$$

โดยที่ Lower Diagonal Matrix ดังกล่าวมีคุณสมบัติที่เมื่อคูณกับสมการข้างบนของ Upper Diagonal Matrix แล้วได้สมการเดิมกลับคืนมา นั่นคือ

$$\mathbf{L}[\mathbf{UX} - \mathbf{D}] = \mathbf{AX} - \mathbf{C}$$

ซึ่งถ้าสมการเป็นจริง เราสามารถ Equate สองส่วนของสมการ จากทฤษฎีในการคูณกันของ Matrix ได้ดังนี้

$\mathbf{LU} = \mathbf{A}$ และ $\mathbf{LD} = \mathbf{C}$

LU Decomposition

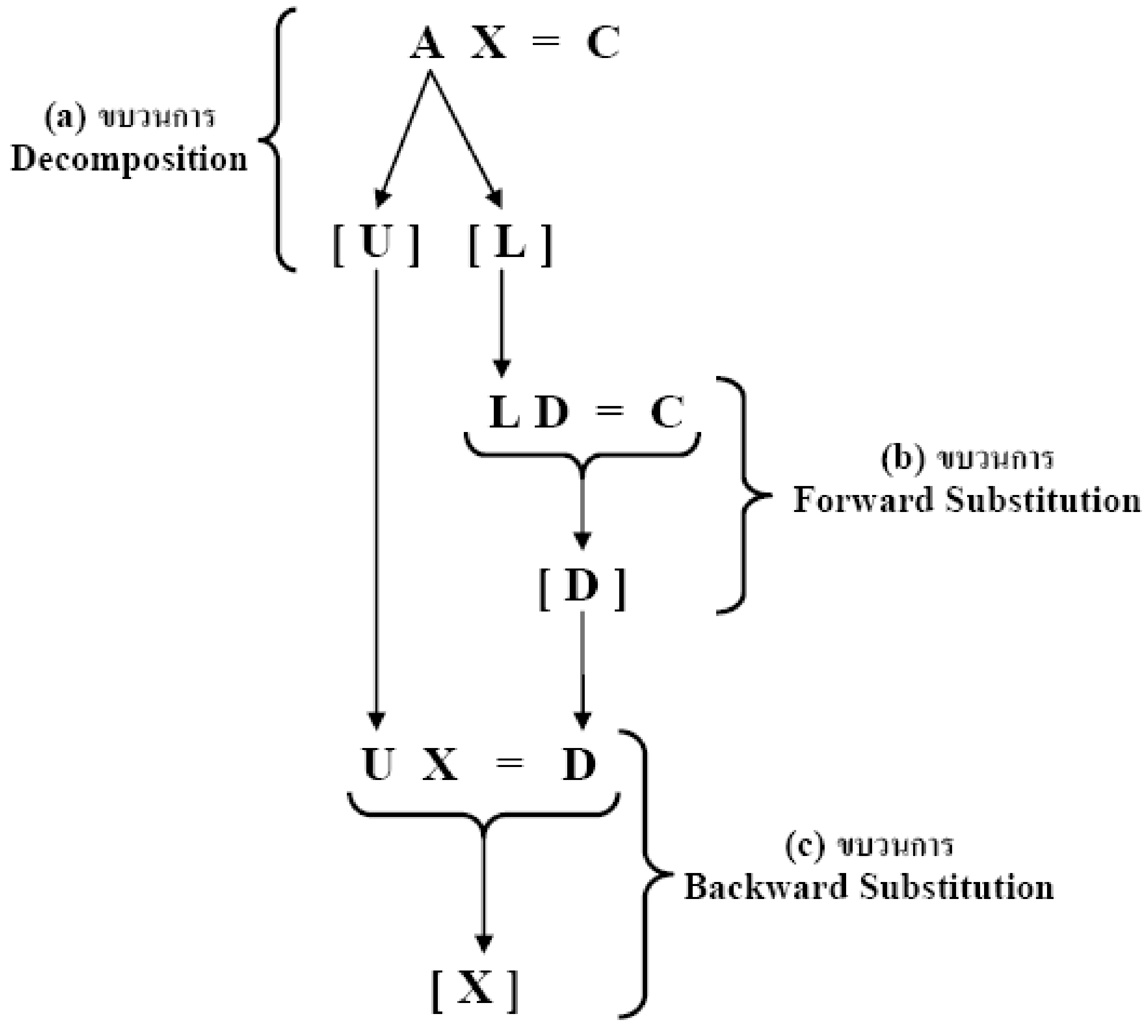
$$\mathbf{L[UX - D] = AX - C}$$

ซึ่งถ้าสมการเป็นจริง เราสามารถ Equate สองส่วนของสมการ จากทฤษฎีในการคูณกันของ Matrix ได้ดังนี้

$$\mathbf{LU = A} \text{ และ } \mathbf{LD = C}$$

สมการแรกข้างบนเรียกว่าเป็น LU Decomposition ของ Matrix $\mathbf{A}$ และถ้าเราสามารถหาได้ เราสามารถหา Solution ของสมการได้อย่างมีประสิทธิภาพ ด้วยขบวนการ 2 Step Substitution (Forward Substitution และ Backward Substitution) โดยใช้สมการ $\mathbf{UX - D = 0}$ และสมการ $\mathbf{LD = C}$ ดัง Diagram ในรูป

LU Decomposition



LU Decomposition

8.5.2 การดัดแปลง Gauss Elimination มาใช้ใน LU Decomposition

ในขบวนการ Gauss Elimination นั้น เราสร้าง Upper Diagonal Matrix ในรูป (ยกตัวอย่างกรณี 3×3 Matrix)

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix}$$

จากสมการดั้งเดิมคือ

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

LU Decomposition

ซึ่งขั้นตอนแรกนั้นเราทำการคูณแถวที่ 1 ด้วย Factor $f_{21} = a_{21} / a_{11}$ จากนั้นนำผลที่ได้มาลบออกจากสมการในแถวที่สองเพื่อกำจัด a_{21} จากนั้นเราทำการกำจัด a_{31} ออก ด้วยการคูณแถวที่ 1 เช่นกันด้วย Factor $f_{31} = a_{31} / a_{11}$ และหักลบจากสมการในแถวที่ 3

ในรอบที่สอง เรากำจัด a'_{32} ออกด้วยการคูณสมการแถวที่สองที่ได้จากรอบแรกด้วย Factor $f_{32} = a'_{32} / a'_{22}$ ซึ่งที่จริงแล้วค่า Factor ที่ใช้ดังกล่าวคือค่า Coefficient ของ Lower Diagonal Matrix

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ f_{21} & 1 & 0 \\ f_{31} & f_{32} & 1 \end{bmatrix}$$

ขอให้นักศึกษาลองทดสอบดูว่าความจริงแล้ว เราได้ $\mathbf{LU} = \mathbf{A}$ สังเกตอีกอย่างหนึ่งว่า Matrix ที่เราได้จาก Gauss Elimination จะต่างจากที่เราอธิบายในตอนต้นเล็กน้อย ซึ่งมีค่าในส่วนของ Diagonal เป็นหนึ่งใน $\mathbf{L}$ Matrix แทนที่จะเป็น $\mathbf{U}$ Matrix ซึ่งเราเรียกว่าเป็น ***Doolittle Decomposition*** หรือ ***Factorization*** ในขณะที่แบบที่อธิบายในตอนต้นเราเรียก ***Crout Decomposition***

LU Decomposition

Example 8.5 จงใช้ Gauss Elimination เพื่อทำ LU Decomposition ของ Matrix

$$\mathbf{A} = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}$$

ต่อไปนี้เป็นผลที่ได้จากการ Run Program ที่แสดงในภาคผนวก โดยที่ 3 Column แรกคือ Matrix $\mathbf{L}$ และสาม Column ถัดมาคือ Matrix $\mathbf{U}$

LU Decomposition

```
k = 1

ans =

    1.0000         0         0    3.0000   -0.1000   -0.2000
    0.0333    1.0000         0         0    7.0033   -0.2933
    0.1000         0    1.0000         0   -0.1900   10.0200

k = 2

ans =

    1.0000         0         0    3.0000   -0.1000   -0.2000
    0.0333    1.0000         0         0    7.0033   -0.2933
    0.1000   -0.0271    1.0000         0         0   10.0120
```

Crout Decomposition

8.5.3 Crout Decomposition

ในขบวนการของ Gauss Elimination นั้น การคำนวณส่วนใหญ่จะอยู่ที่ Forward Elimination โดยเฉพาะอย่างยิ่งถ้าปัญหามีขนาดใหญ่ ดังนั้นจึงมีความพยายามที่จะปรับปรุงขบวนการให้ดีขึ้น และแยกส่วนการคำนวณด้านซ้ายออกมา

วิธีที่มีประสิทธิภาพมากที่สุดอันหนึ่งก็คือ Crout Decomposition ซึ่งจะแยก Matrix ออกเป็น Upper Triagular และ Lower Triangular ยกตัวอย่างเช่น Matrix ขนาด 4×4

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{23} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Crout Decomposition

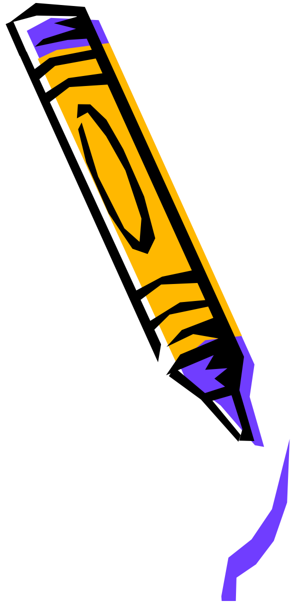
$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{23} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Crout Decomposition จะเริ่มจากการคูณกันของ Matrix ด้านซ้ายของสมการ จากนั้น Equate ผลที่ได้ให้เท่ากับด้านขวาของสมการ ซึ่งขั้นแรกเราได้ว่า

$$l_{11} = a_{11}, \quad l_{21} = a_{21}, \quad l_{31} = a_{31}, \quad l_{41} = a_{41}$$

กล่าวคือ Column แรกของ $\mathbf{L}$ จะเท่ากับ Column แรกของ $\mathbf{A}$ หรือ $l_{i1} = a_{i1}, i = 1,2,\ldots,n$

Crout Decomposition



$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{23} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

จากนั้นพิจารณาจากแถวแรกของ **L** ที่ต้องคูณกับแต่ละ Column ของ **U** และแก้สมการหาแถวแรกของ **U** ออกมา เราได้

$$l_{11} = a_{11}, \quad l_{11}u_{12} = a_{12}, \quad l_{11}u_{13} = a_{13}, \quad l_{11}u_{14} = a_{14}$$

และ

$$u_{12} = \frac{a_{12}}{l_{11}}, \quad u_{13} = \frac{a_{13}}{l_{11}}, \quad u_{14} = \frac{a_{14}}{l_{11}}$$

เราสรุปได้ว่า $u_{ij} = \dfrac{a_{1j}}{l_{11}}, \; j = 2,3,\ldots,n$

เมื่อถึงขั้นนี้ เราได้ Column แรกของ **L** และแถวแรกของ **U** จากนั้นขบวนการจะทำซ้ำสำหรับ Column ที่สอง และแถวที่สองของ **L** และ **U** ตามลำดับ ซึ่งเราจะได้

Crout Decomposition

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix}\begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{23} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

เมื่อถึงขั้นนี้ เราได้ Column แรกของ **L** และแถวแรกของ **U** จากนั้นขบวนการจะทำซ้ำสำหรับ Column ที่สอง และแถวที่สองของ **L** และ **U** ตามลำดับ ซึ่งเราจะได้

$$l_{i2} = a_{i2} - l_{i1}u_{12}, \; i = 2,3,\ldots,n$$

$$u_{2j} = \frac{a_{2j} - l_{21}u_{1j}}{l_{22}}, \; j = 3,4,\ldots,n$$

นักศึกษาสามารถพิจารณาเองและหาคำตอบในขั้นต่อไปได้ดังนี้

$$l_{i3} = a_i - l_{i1}u_{13} - l_{i2}u_{23}, \;\; i = 3,4,\ldots,n$$

$$u_{3j} = \frac{a_{3j} - l_{31}u_{1j} - l_{32}u_{2j}}{l_{33}}, \; j = 4,5,\ldots,n$$

$$
\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix}
\begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}
=
\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{23} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}
$$

ด้วยการตรวจสอบในผลลัพธ์ที่ได้ เราสามารถสรุปเป็นชุดสมการในการคำนวณได้ดังนี้

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

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

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

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

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

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

Crout Decomposition

และ Pseudo Code สามารถเขียนได้เป็น

```
DOFOR j = 2 to n
    a(1,j)=a(1,j)/a(1,1)
ENDDO
DOFOR j = 2 to n-1
    DOFOR i = j to n
        sum=0;
        DOFOR k = 1 to j-1
            sum=sum+a(i,k)*a(k,j)
        ENDDO
        a(i,j)=a(i,j)-sum
    ENDDO
    DOFOR k = j+1 to n
        sum=0;
        DOFOR i = 1 to j-1
            sum=sum+a(j,i)*a(i,k)
        ENDDO
        a(j,k)=(a(j,k)-sum)/a(j,j)
    ENDDO
ENDDO
sum=0
DOFOR k = 1 to n-1
    sum=sum+a(n,k)*a(k,n)
ENDDO
a(n,n)=a(n,n)-sum
```

Crout Decomposition

สังเกตว่า Algorithm ประกอบด้วย Loop ที่กะทัดรัด นอกเหนือจากนั้นแล้วลักษณะของ Algorithm จะเป็น In-place กล่าวคือทั้ง Upper Diagonal และ Lower Diagonal Matrix สามารถบรรจุอยู่ใน Matrix เดียวกันและเข้าไปแทนที่ Element ใน Matrix เดิม เนื่องจากค่า Coefficient ของ Matrix **A** จะถูกใช้เพียงแค่ครั้งเดียว ดังนั้นนอกจากจะรวดเร็วแล้ว Algorithm ยังใช้หน่วยความจำน้อย ภาคผนวกแสดงโปรแกรม MATLAB แต่ในกรณีนี้จะแยกแต่ละ Matrix ออกจากกัน เพื่อความสะดวกในการศึกษาการทำงาน

Example 8.6

Example 8.6 จงใช้ Crout Decomposition เพื่อทำ LU Decomposition ของสมการ

$$\begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ -8 \\ 16 \end{bmatrix}$$

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

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

ผลการ Run Program ได้ดังนี้

```
ans =
    3.0000         0         0    1.0000   -0.0333   -0.0667
    0.1000         0         0         0         0         0
    0.3000         0         0         0         0         0

ans =
    3.0000         0         0    1.0000   -0.0333   -0.0667
    0.1000    7.0033         0         0    1.0000   -0.0419
    0.3000   -0.1900         0         0         0         0

ans =
    3.0000         0         0    1.0000   -0.0333   -0.0667
    0.1000    7.0033         0         0    1.0000   -0.0419
    0.3000   -0.1900   10.0120         0         0    1.0000
```

Crout Decomposition

ในการนำ Crout Decomposition ไปแก้สมการ Linear Equation จะต้องมีขบวนการ Substitution และในกรณีนี้เราจะต้องคำนวณ Vector $\mathbf{D}$ จาก Vector $\mathbf{C}$ ด้วย หลังจากเราทำ LU Decomposition แล้ว

จากสมการ $\mathbf{LD} = \mathbf{C}$ ที่กล่าวในหัวข้อก่อน หรือในกรณีของสมการ 4 Unknown

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}$$

เราสามารถสรุปจากการคูณกันของ Matrix ได้คำตอบดังนี้

$$d_1 = \frac{c_1}{l_{11}}; \quad d_i = \frac{c_i - \sum_{j=1}^{i-1} l_{ij} d_j}{l_{ii}}, \quad i = 2,3,\ldots,n$$

Crout Decomposition

และค่าของ x_i สามารถคำนวณได้จากวิธีการ Back Substitution จากสมการ $\mathbf{UX} = \mathbf{D}$ หรือ

$$\begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix}$$

ได้ดังนี้

$$x_n = d_n; \quad x_i = d_i - \sum_{j=i+1}^{n} u_{ij} x_j, \quad i = n-1, n-2, \ldots, 1$$

Example 8.7

Example 8.7 จงใช้ Crout Decomposition เพื่อแก้สมการ

$$\begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ -8 \\ 16 \end{bmatrix}$$

จากโปรแกรมในภาคผนวก เราได้ดังนี้

Example 8.7

```
Calculate LU Decomposition, Row&Col = 1 ; ans = [L U]
     3.0000          0          0     1.0000    -0.0333    -0.0667
     0.1000          0          0          0          0          0
     0.3000          0          0          0          0          0
Calculate LU Decomposition, Row&Col = 2 ; ans = [L U]
     3.0000          0          0     1.0000    -0.0333    -0.0667
     0.1000     7.0033          0          0     1.0000    -0.0419
     0.3000    -0.1900          0          0          0          0
Calculate LU Decomposition, Row&Col = 3 ; ans = [L U]
     3.0000          0          0     1.0000    -0.0333    -0.0667
     0.1000     7.0033          0          0     1.0000    -0.0419
     0.3000    -0.1900    10.0120          0          0     1.0000
Find D from L and C ; ans = [L D C]
     3.0000          0          0     4.0000    12.0000
     0.1000     7.0033          0    -1.1994    -8.0000
     0.3000    -0.1900    10.0120     1.4555    16.0000
Find X from U and D ; ans = [U X D]
     1.0000    -0.0333    -0.0667     4.0591     4.0000
          0     1.0000    -0.0419    -1.1385    -1.1994
          0          0     1.0000     1.4555     1.4555
```

ในการนำ Crout Algorithm มาดัดแปลงหา Inverse ของ Matrix เราสามารถทำได้โดยใช้วิธีที่กล่าวมาในตอนต้น คือทำการหาทีละ Column ในการนี้เราทำการหา Upper และ Lower Diagonal Matrix ครั้งเดียว แต่การหา **D** นั้นจะต้องกระทำแยกและค่อยทำ Back-Substitution หาค่า **X** ที่เป็น Solution ของแต่ละ Column ของ Inverse Matrix

กรรมวิธีที่กล่าวมาทั้งหมด สามารถนำมาปรับปรุงให้มีประสิทธิภาพเพิ่มขึ้นได้อีก ถ้า Matrix มีคุณสมบัติพิเศษ อย่างเช่นในกรณีของ Sparse Matrix หรือใน Banded System ที่ Matrix เป็น Multi-diagonal Matrix ซึ่งเรามี Algorithm ที่มีประสิทธิภาพสูงเฉพาะสำหรับ Matrix ประเภทนี้ หรือใน Matrix ที่เป็น Symmetric Matrix ซึ่งในกรณีหลังนี้เรามี Algorithm ที่ชื่อ Cholsky Decomposition ที่รวดเร็วกว่า Crout Decomposition อย่างไรก็ตาม รายละเอียดของ Algorithm เหล่านี้จะเกินเนื้อหาของวิชานี้

Method	จำนวนสูงสุดของ Unknown	Stability	Precision	Breadth of Application	Program	Comments
Graphical	2	-	Poor	Limited	-	อาจใช้เวลามาก เมื่อเทียบกับวิธี Numerical
Cramer's Rule	3	-	ได้รับผลจาก Round-Off Error	Limited	-	ใช้การคำนวณมากเกินไป ถ้ามีมากกว่า 3 สมการ
Algebraic Elimination	3	-	ได้รับผลจาก Round-Off Error	Limited	-	
Gauss Elimination with Pivoting	100	-	ได้รับผลจาก Round-Off Error	General	ยากปานกลาง	
Gauss-Jordan with Partial Pivoting	100	-	ได้รับผลจาก Round-Off Error	General	ยากปานกลาง	สามารถนำมาใช้คำนวณ Matrix Inverse ได้ดี
LU Decomposition	100	-	ได้รับผลจาก Round-Off Error	General	ยากปานกลาง	เป็นวิธี Elimination ที่นิยมมากที่สุด
Gauss-Seidel	1000	อาจจะไม่ Converge ถ้าไม่เป็น Diagonally Dominant	Excellent	เหมาะสมเฉพาะกับระบบที่เป็น Diagonally Dominant	ง่าย	

Homework 9, Chapter 10

- DOWNLOAD
- คำนวณแนะนำให้เขียนโปรแกรม หรือใช้ MATLAB หรือใช้ Spreadsheet(Excel)
- หยุด 2 สัปดาห์
  - การบ้าน ส่งต้นชั่วโมง พุธ 20 เมษายน

End of Chapter 10

- Next Week (Wk 13)
  - No Class วันจักรี
- Following Week (Wk 14)
  - No Class วันสงกรานต์
- Week 15; Chapter 11
  - April 20
  - Numerical Differentiation and Integration

CPE 332
Computer Engineering Mathematics II

Part III, Chapter 11

Numerical Differentiation and Integration

Today Topics

- Chapter 11 Numerical Differentiation and Integration
  - Derivative Approximation
    - Forward, Backward, Centered Difference
    - High Order Derivative
    - High Accuracy Approximation
  - Integral Approximation
    - Polynomial
      - Zero Order
      - First Order (Trapezoidal)
      - Second Order (Simpson 1/3)
      - Third Order (Simpson 3/8)
    - More Accurate Method
      - Richardson Extrapolation

9.2 Taylor's Theorem

ทฤษฎีบท

ถ้า Function f และค่า $n+1$ Derivative แรกของมันมีความต่อเนื่องในช่วงของ a และ x ดังนั้นค่าของ Function ที่จุด x สามารถแสดงได้โดย

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

โดย R_n เรียก Remainder และให้นิยามว่าเป็น

$$R_n = \int_a^x \frac{(x-t)^n}{n!} f^{(n+1)}(t)dt$$

ซึ่งค่าของ Remainder ดังกล่าวยังสามารถเขียนในรูปที่เรียก Derivative Form หรือ Lagrange Form ดังนี้

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-a)^{n+1}$$

สมการข้างบนรู้จักกันในนาม Taylor Series หรือ Taylor's Formula ซึ่งถ้าไม่รวม R_n สมการที่เหลือก็คือค่าประมาณของ $f(x)$ ที่มีลักษณะเป็น Polynomial กล่าวอีกนัยหนึ่งก็คือ ค่าของ Function ใดๆที่มีคุณสมบัติตามที่กำหนด สามารถประมาณได้จากสมการของ Polynomial

การละเทอม R_n ออกจากสมการ จะส่งผลให้การคำนวณค่าประมาณนั้นมี Error นี่คือที่มาของ Truncation Error ที่เรากล่าวในบทที่ 7 ดังนั้นการหาค่าของ Function ขึ้นอยู่ว่าเราต้องการ Significant Digit แค่ไหน ซึ่งจะเป็นตัวจะกำหนดว่าเราต้องใช้กี่เทอมใน Polynomial และจะลงเอยด้วย Degree ของ Polynomial และ Derivative ของ Function ที่จุด a ที่ต้องใช้

ถ้าให้ a เป็นจุดของ Function ที่เรารู้ค่าของมัน และ Derivative ของมัน และสมมุติว่าอยู่ที่ x_i เราสามารถใช้ Taylor Series ประมาณค่าของ Function ที่จุดใหม่ กล่าวคือ x_{i+1} โดยกำหนดขนาดของ Step $h = x_{i+1} - x_i$ ให้มีค่าเท่าๆกัน เช่น

$$Zero-Order\ Approximation: f(x_{i+1}) \cong f(x_i)$$

$$First-Order\ Approximation: f(x_{i+1}) \cong f(x_i) + f'(x_i)h$$

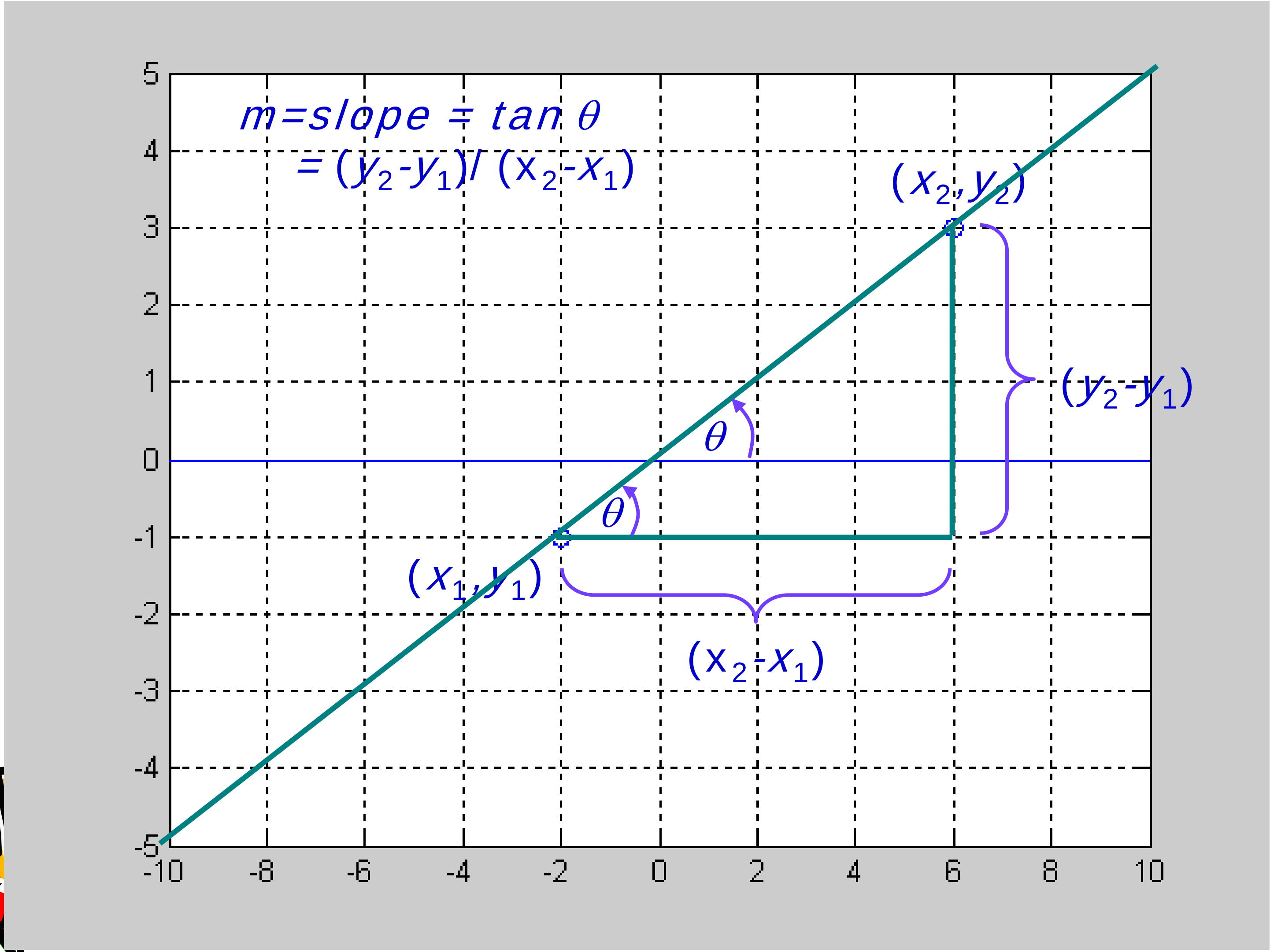
$$Second-Order\ Approximation: f(x_{i+1}) \cong f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2$$

และโดยทั่วไปเราสามารถเขียน

$$n-Order\ Approximation: f(x_{i+1}) \cong f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \cdots + \frac{f^{(n)}(x_i)}{n!}h^n$$

โดยที่ค่า Remainder สามารถแสดงได้เป็น $R_n = \dfrac{f^{(n+1)}(\xi)}{(n+1)!}h^{n+1}$

Slope of Line



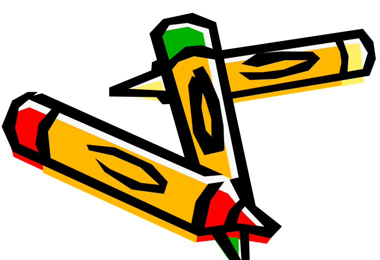
Definition of Derivative

- Derivative of *f*(*x*) at any point *x* is the slope of the tangent line at that point (*x*,*f*(*x*))
- Mathematically

$$\text{Derivative of } f(x) \equiv \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

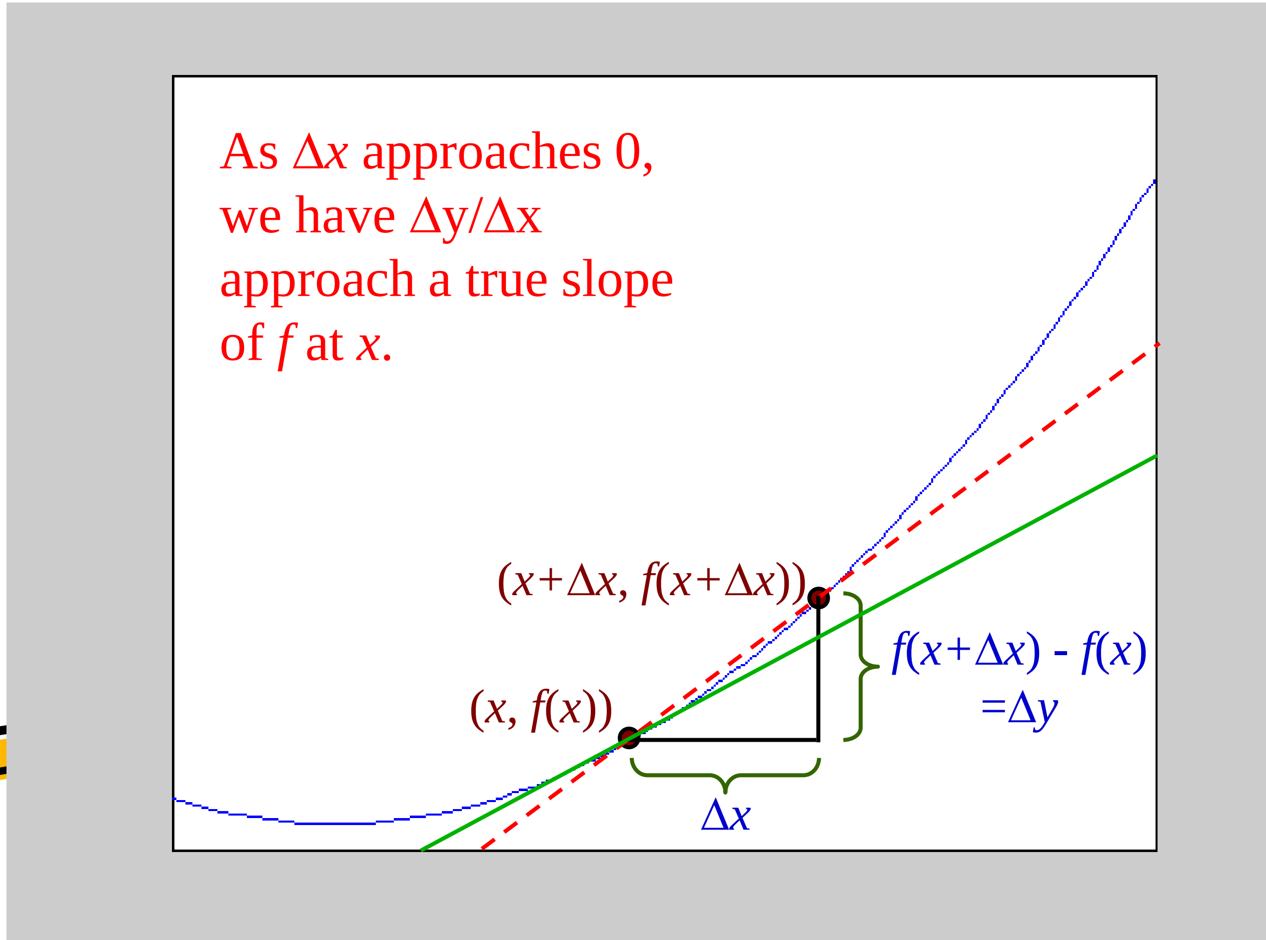
- For function *y*=*f*(*x*), derivative of function is written in many forms

$$f'(x),\ y',\ \frac{dy}{dx}\ (\text{read 'dee - y dee - x'})\ \text{or}\ \dot{y}$$



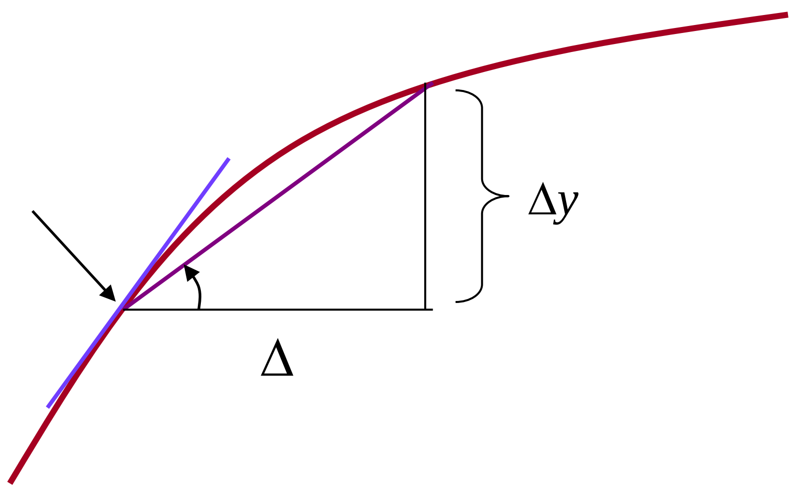
Approximation of slope at point *x* using secant line

- Slope at 'x' $\cong [f(x+\Delta x) - f(x)] / \Delta x = \Delta y / \Delta x$



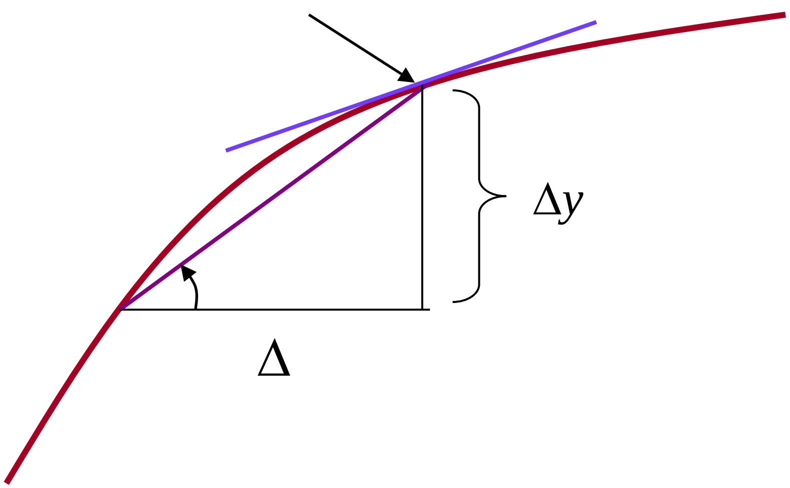
Derivative(Forward)

$$f'(x) = \lim_{\Delta \to 0} \frac{f(x+\Delta) - f(x)}{\Delta}$$



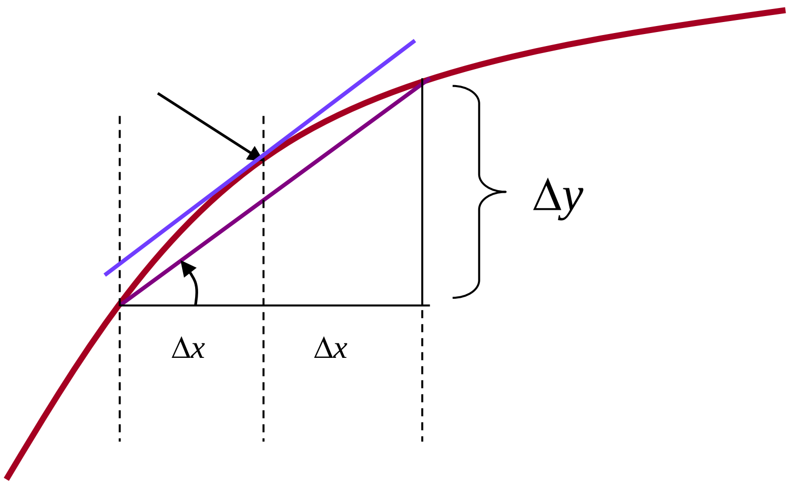
Derivative(Backward)

$$f'(x) = \lim_{\Delta \to 0} \frac{f(x) - f(x - \Delta)}{\Delta}$$



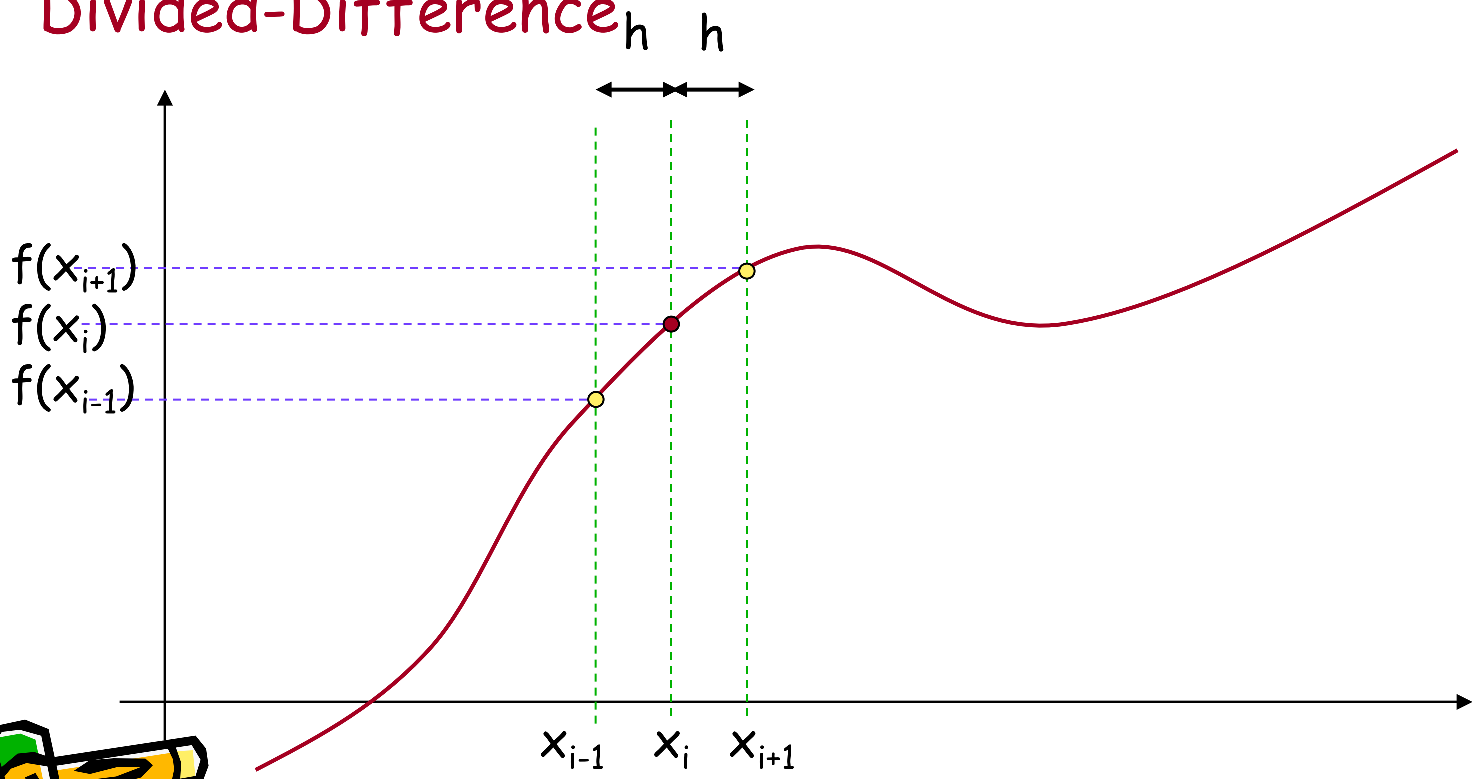
Derivative(Central)

$$f'(x) = \lim_{\Delta \to 0} \frac{f(x+\Delta) - f(x-\Delta)}{2\Delta}$$



Introduction

- ในทางปฏิบัติ เราได้ Sample ของ Data เป็นจุด การหา Derivative ก็คือการลบค่าของจุด Data ที่อยู่ติดกันและหารด้วยระยะห่างระหว่างจุด นี่คือวิธีการของ Finite Divided-Difference



Finite Divided-Difference

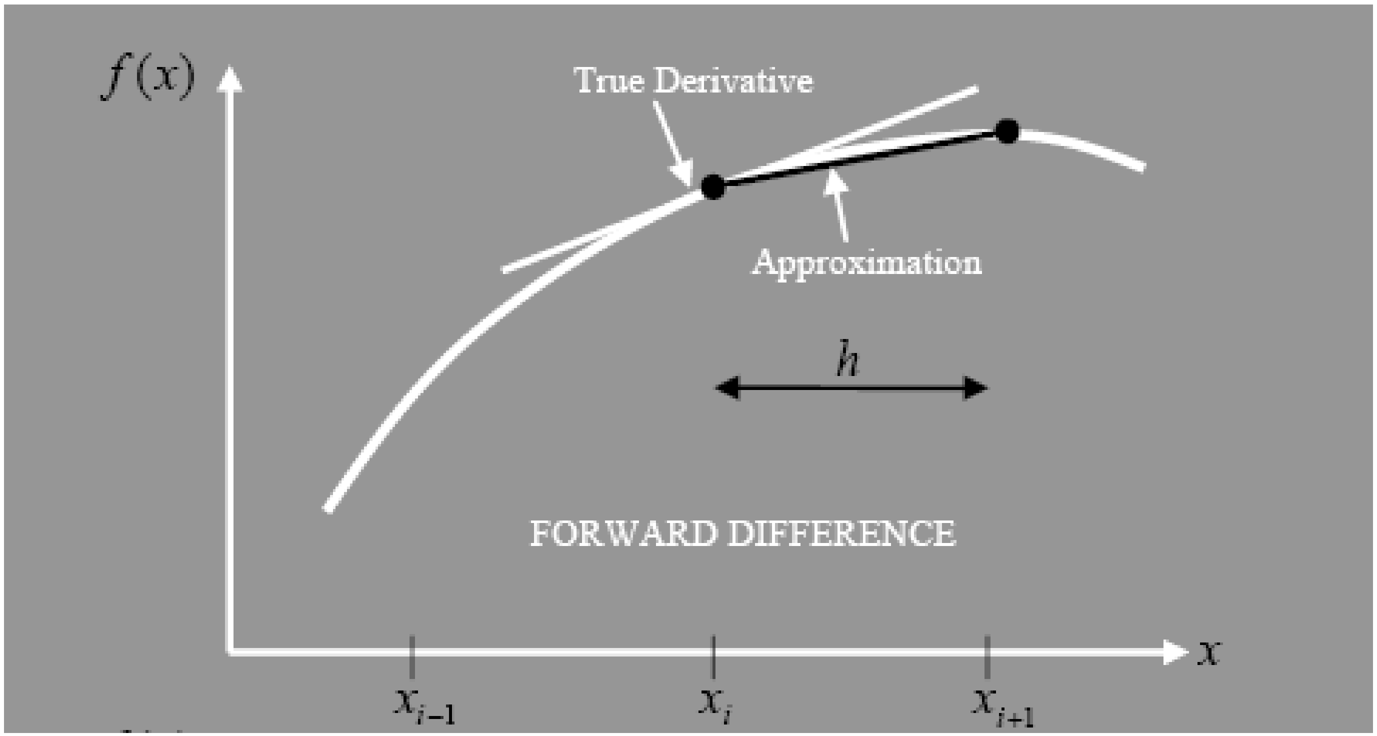
$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \cdots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n$$

ในกรณีของ First Order เราได้

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + R_1$$

และถ้าเรียงสมการใหม่ หาค่า $f'(x_i)$ เราได้

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{R_1}{h}$$ **: First Forward Difference**


$f(x)$
True Derivative
Approximation
h
FORWARD DIFFERENCE
x
x_{i-1}
x_i
x_{i+1}

Finite Divided-Difference

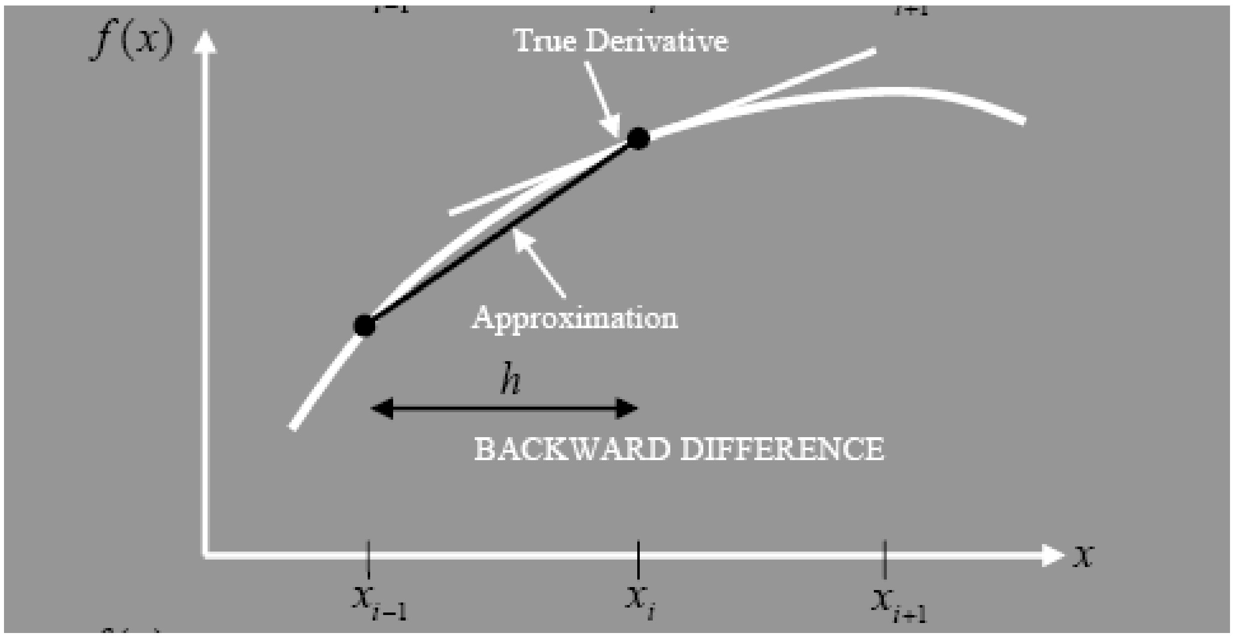
ทำนองเดียวกัน Taylor Series Expansion สามารถ Expand ย้อนหลัง และเขียนได้ในรูปของ

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2}h^2 + \cdots$$

สำหรับ First Order Expansion เมื่อจัดเรียงใหม่เราได้

$$f'(x_i) \cong \frac{f(x_i) - f(x_{i-1})}{h} = \frac{\nabla f_i}{h} \qquad \textbf{: First Backward Difference}$$

ซึ่งเราเรียก First Backward Difference เพราะเราใช้ค่าก่อนหน้านี้ของ $f(x_{i-1})$ มาคำนวณ



$f(x)$
True Derivative
Approximation
h
BACKWARD DIFFERENCE
x
x_{i-1}
x_i
x_{i+1}

Finite Divided-Difference

วิธีที่สามในการประมาณค่า Derivative เรียก Centered หรือ Central Difference คือใช้สมการ Taylor Series ใน Backward Expansion นำไปหักลบออกจากสมการของ Forward Expansion และจัดเรียงสมการเราได้

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \cdots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n$$

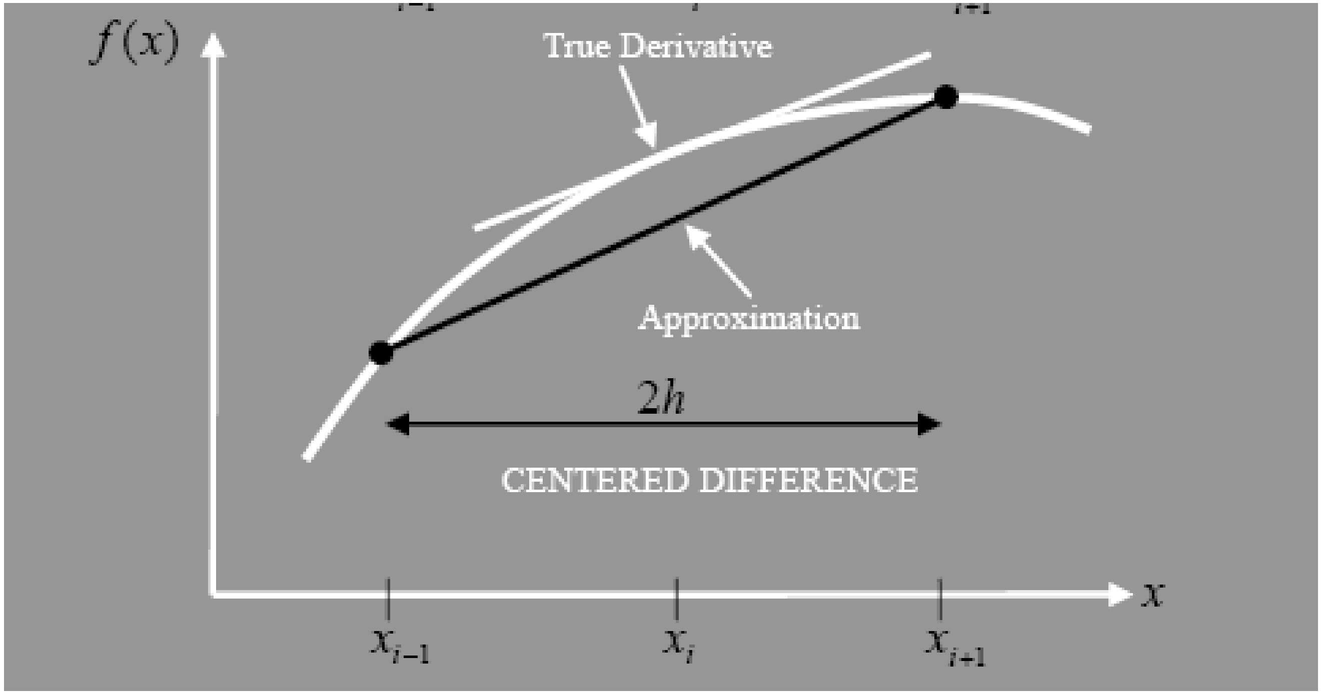
$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2}h^2 + \cdots$$

$$f(x_{i+1}) - f(x_{i-1}) = 2f'(x_i)h + 2\frac{f'''(x_i)h^3}{3!} + \ldots$$

หรือ

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O(h^2)$$ **: First Central Difference**

จะเห็นได้ว่า Central Difference จะมี Truncation Error เป็น $O(h^2)$ ในขณะที่ Forward และ Backward Difference จะมี Error เป็น Order ของ $O(h)$ ซึ่งจะให้คำตอบที่ถูกต้องกว่า



f(x)
True Derivative
Approximation
2h
CENTERED DIFFERENCE
x
x_{i−1}
x_i
x_{i+1}

Second Derivative

$$f(x_{i+2}) = f(x_i) + f'(x_i)(2h) + \frac{f''(x_i)}{2!}(2h)^2 + \cdots +$$

และ

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \cdots$$

เมื่อคูณสมการล่างด้วย 2 และหักลบออกจากสมการบนเราได้

$$f(x_{i+2}) - 2f(x_{i+1}) = -f(x_i) + f''(x_i)h^2 + \cdots$$

และเมื่อจัดเรียงใหม่เราได้

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} + O(h)$$

Second Forward Finite Difference

Second Derivative

ทำนองเดียวกันสำหรับ Backward Difference และ Central Difference โดยใช้วิธีคล้ายกับที่กล่าวมา ซึ่งขอให้นักศึกษาลองทำเป็นการบ้าน เราได้

$$f''(x_i) = \frac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2} + O(h)$$

Second Backward Finite Difference

และ

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2} + O(h^2)$$

Second Central Finite Difference

High Accuracy Finite Divided-Difference

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + R_2$$

เมื่อจัดเรียงใหม่เราได้

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(x_i)}{2}h + O(h^2)$$

คราวนี้ถ้าเราแทนค่า Second Derivative ด้วนค่าประมาณที่เราหาได้ก่อนหน้านี้คือ

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} + O(h)$$

High Accuracy Finite Divided-Difference

เราได้ $$f'(x_i)=\frac{f(x_{i+1})-f(x_i)}{h}-\frac{f(x_{i+2})-2f(x_{i+1})+f(x_i)}{2h}+O(h^2)$$

เมื่อจัดเรียงสมการใหม่ เราได้

$$f'(x_i)=\frac{-f(x_{i+2})+4f(x_{i+1})-3f(x_i)}{2h}+O(h^2)$$

สังเกตว่าการเพิ่มเทอมที่เป็น Second Derivative จะเพิ่มความถูกต้องของคำตอบเป็น $O(h^2)$ แต่เราต้องใช้จุด สามจุด ของ $f(x_i), f(x_{i+1}), f(x_{i+2})$ และเราสามารถทำได้คล้ายๆกันนี้ในกรณีของ Backward Difference และ Central Difference ในตารางถัดไปเป็นการสรุปการหา Derivative ตั้งแต่ First Order จนถึง Forth Order Derivative โดยเพิ่มจำนวนเทอมของ Taylor Series Expansion ในการคำนวณ นั่นก็คือเพิ่มจุดที่ใช้ในการคำนวณค่าประมาณ

Summary

Type	Equation	Error
Forward	$f'(x_i) \cong \dfrac{f(x_{i+1}) - f(x_i)}{h}$	$O(h)$
Forward	$f'(x_i) \cong \dfrac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h}$	$O(h^2)$
Forward	$f''(x_i) \cong \dfrac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2}$	$O(h)$
Forward	$f''(x_i) \cong \dfrac{-f(x_{i+3}) + 4f(x_{i+2}) - 5f(x_{i+1}) + 2f(x_i)}{h^2}$	$O(h^2)$
Forward	$f'''(x_i) \cong \dfrac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{h^3}$	$O(h)$
Forward	$f'''(x_i) \cong \dfrac{-3f(x_{i+4}) + 14f(x_{i+3}) - 24f(x_{i+2}) + 18f(x_{i+1}) - 5f(x_i)}{h^3}$	$O(h^2)$
Forward	$f^{(4)}(x_i) \cong \dfrac{f(x_{i+4}) - 4f(x_{i+3}) + 6f(x_{i+2}) - 4f(x_{i+1}) + f(x_i)}{h^4}$	$O(h)$
Forward	$f^{(4)}(x_i) \cong \dfrac{-2f(x_{i+5}) + 11f(x_{i+4}) - 24f(x_{i+3}) + 26f(x_{i+2}) - 14f(x_{i+1}) + 3f(x_i)}{h^4}$	$O(h^2)$

Summary

Backward	$f'(x_i) \cong \dfrac{f(x_i) - f(x_{i-1})}{h}$	$O(h)$
Backward	$f'(x_i) = \dfrac{3f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{2h}$	$O(h^2)$
Backward	$f''(x_i) \cong \dfrac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2}$	$O(h)$
Backward	$f''(x_i) \cong \dfrac{2f(x_i) - 5f(x_{i-1}) + 4f(x_{i-2}) - f(x_{i-3})}{h^2}$	$O(h^2)$
Backward	$f'''(x_i) \cong \dfrac{f(x_i) - 3f(x_{i-1}) + 3f(x_{i-2}) - f(x_{i-3})}{h^3}$	$O(h)$
Backward	$f'''(x_i) \cong \dfrac{5f(x_i) - 18f(x_{i-1}) + 24f(x_{i-2}) - 14f(x_{i-3}) + 3f(x_{i-4})}{h^3}$	$O(h^2)$
Backward	$f^{(4)}(x_i) \cong \dfrac{f(x_i) - 4f(x_{i-1}) + 6f(x_{i-2}) - 4f(x_{i-3}) + f(x_{i-4})}{h^4}$	$O(h)$
Backward	$f^{(4)}(x_i) \cong \dfrac{3f(x_i) - 14f(x_{i-1}) + 26f(x_{i-2}) - 24f(x_{i-3}) + 11f(x_{i-4}) - 2f(x_{i-5})}{h^4}$	$O(h^2)$

Summary

Central	$f'(x_i) \cong \dfrac{f(x_{i+1}) - f(x_{i-1})}{2h}$	$O(h^2)$
Central	$f'(x_i) = \dfrac{-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})}{12h}$	$O(h^4)$
Central	$f''(x_i) \cong \dfrac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}$	$O(h^2)$
Central	$f''(x_i) = \dfrac{-f(x_{i+2}) + 16f(x_{i+1}) - 30f(x_i) + 16f(x_{i-1}) - f(x_{i-2})}{12h^2}$	$O(h^4)$
Central	$f'''(x_i) = \dfrac{f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2})}{2h^3}$	$O(h^2)$
Central	$f'''(x_i) = \dfrac{-f(x_{i+3}) + 8f(x_{i+2}) - 13f(x_{i+1}) + 13f(x_{i-1}) - 8f(x_{i-2}) + f(x_{i-3})}{8h^3}$	$O(h^4)$
Central	$f^{(4)}(x_i) = \dfrac{f(x_{i+2}) - 4f(x_{i+1}) + 6f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{h^4}$	$O(h^2)$
Central	$f^{(4)}(x_i) = \dfrac{-f(x_{i+3}) + 12f(x_{i+2}) - 39f(x_{i+1}) + 56f(x_i) - 39f(x_{i-1}) + 12f(x_{i-2}) - f(x_{i-3})}{6h^4}$	$O(h^4)$

Numerical Integration

- Newton-Cotes Integration Formula
- Zero-Order Approximation
- First-Order Approximation
  - Trapezoidal Rule
- Second-Order Approximation
  - Simpson 1/3 rule
- Third-Order Approximation
  - Simpson 3/8 rule
- Romberg Integration
  - Richardson Extrapolation
  - Romberg Integration Algorithm

9.4 Newton-Cotes Integration Formulas

สมการของ Newton-Cotes เป็นรูปแบบที่ใช้กันมากที่สุดในการหาค่า Integral แบบ Numerical หลักการคือการเปลี่ยนรูปสมการที่สลับซับซ้อนให้อยู่ในรูปสมการทางคณิตศาสตร์ที่ง่ายในการหาค่า Integrate ในรูปของ Polynomial ดังนี้

$$I = \int_a^b f(x)dx \cong \int_a^b f_n(x)dx$$

ซึ่ง $f_n(x)$ เป็น Polynomial มี Order เท่ากับ n และอยู่ในรูป

$$f_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n$$

ก่อนที่จะพูดในรายละเอียดต่อไป ขอให้เราเข้าใจก่อนว่า สำหรับ Variable แค่ตัวเดียว การหาค่า Integrate จากจุด a ถึงจุด b ของ $f(x)$ ความจริงแล้วคือการหาค่าพื้นที่จากจุด $x = a$ ของ Function ที่กวาดบนแกน x จนถึงจุด $x = b$ โดยพื้นที่ที่อยู่เหนือแกน x จะมีค่าเป็นบวก และพื้นที่ที่อยู่ใต้แกน x จะมีค่าเป็นลบ

การนำ Polynomial ไปใช้ เราจำเป็นต้องหาค่า Coefficient โดยการแทนค่าด้วยจุด $[x_i, f(x_i)]$ ที่รู้ และแก้สมการ Linear Equation ออกมา ซึ่งถ้าเราใช้ Polynomial Order สูง การคำนวณจะสลับซับซ้อน๖ช(ดูหัวข้อในบทที่ 8) วิธีที่ง่ายกว่าคือใช้สมการของ Polynomial ที่เขียนในรูปแบบที่ง่ายกว่า ที่เรียก ***Lagrange Interpolating Polynomial*** ดังนี้

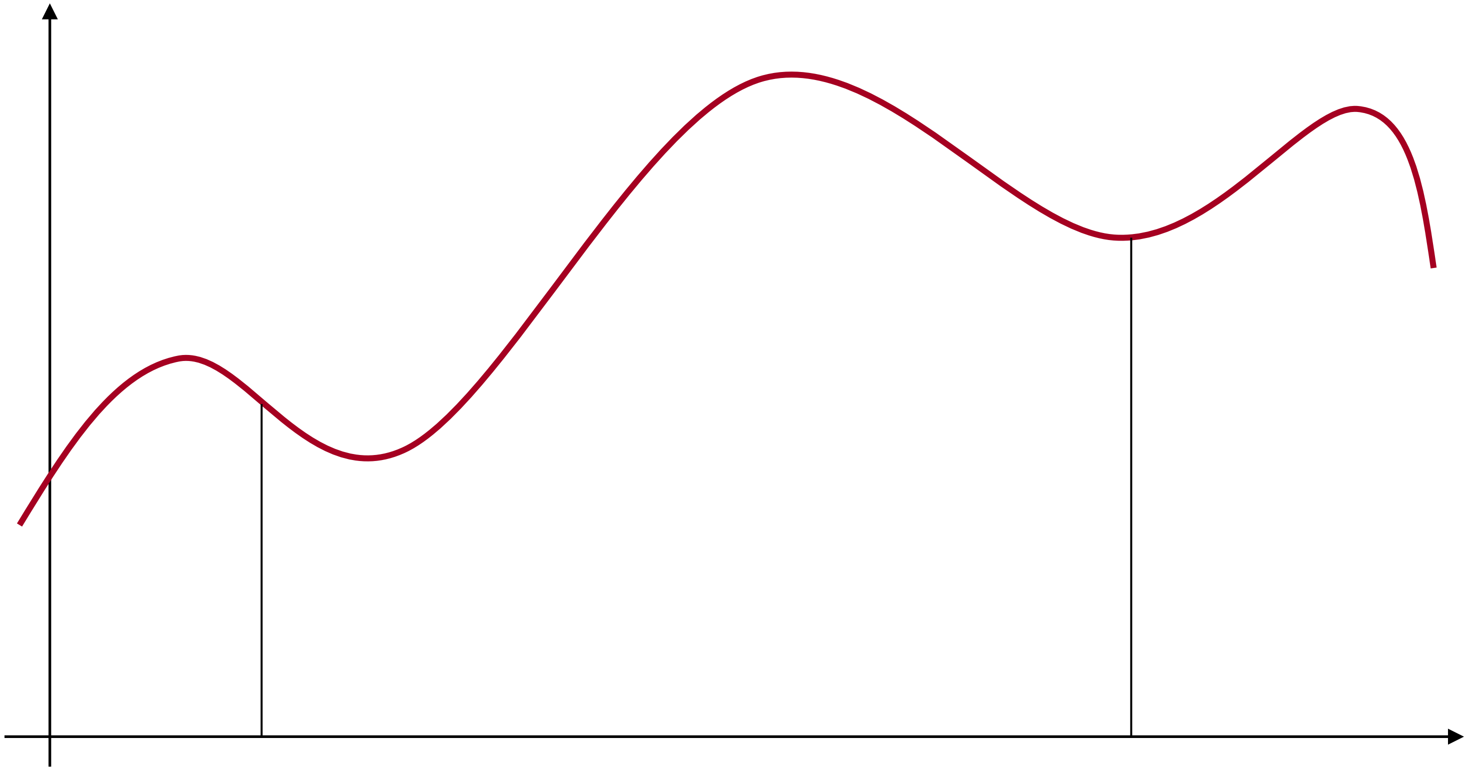
$$f_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i), \quad L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

สมการข้างบน แม้ว่าดูจะสลับซับซ้อน แต่จะมีประโยชน์ในการสร้าง Polynomial ที่จะนำไปใช้ โดยไม่ต้องมีการคำนวณค่า Coefficient ของ Polynomial ล่วงหน้า และจะประหยัดการคำนวณได้มาก เราจะนำไปใช้เมื่อเราเริ่มพูดถึง Second Order Polynimial

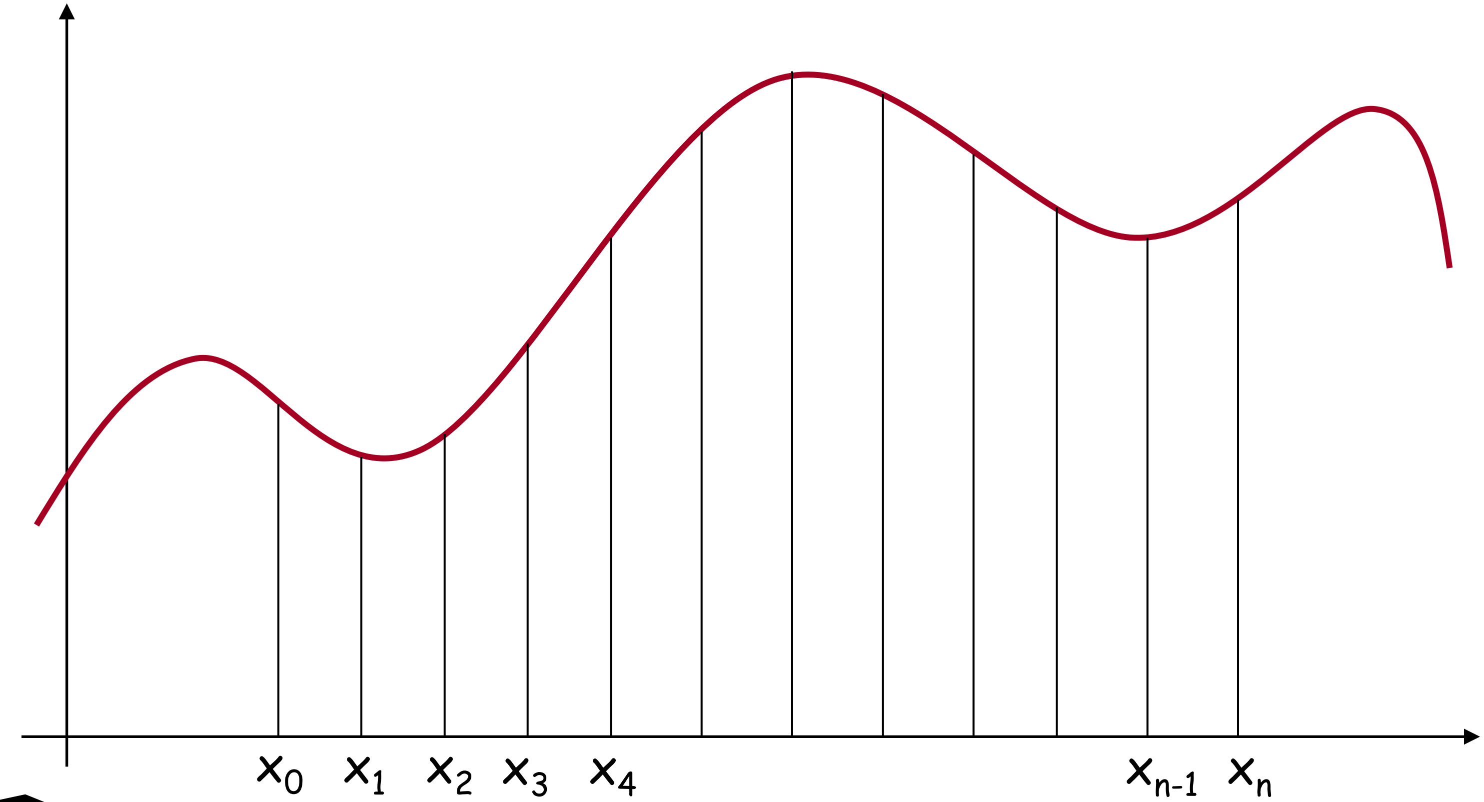
เนื่องจากเป็นการประมาณค่า และ Error ที่เกิดขึ้นจะขึ้นอยู่กับระยะห่างของสองจุด a และ b วิธีการที่จะลด Error ก็คือแบ่งการ Integrate ออกเป็นส่วนที่มีขนาดเล็กหลายๆส่วน ดังนี้

$$I = \int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \cdots + \int_{x_{n-1}}^{x_n} f(x)dx = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)dx;\ x_0 = a, x_n = b$$

Integral Approximation



Integral Approximation



Zero-Order Approximation

9.4.1 Zero-Order Approximation

วิธีนี้เป็นวิธีการประมาณค่า Integrate ตั้งแต่สมัยก่อนที่จะมีคอมพิวเตอร์ ซึ่งเราได้ $f_0(x) = a_0$ ดังนั้นค่า Integral ที่ประมาณได้จะเท่ากับ

$$I = \int_a^b f(x)dx \cong \int_a^b a_0 dx = a_0[b-a]$$

ค่า a_0 อาจจะใช้ค่าเท่ากับ $f(a)$ หรือ $f(b)$ แต่ที่ให้ค่า Error ต่ำสุดจะหาได้จากจุดของ Function ที่อยู่กึ่งกลางระหว่าง a และ b กล่าวคือ

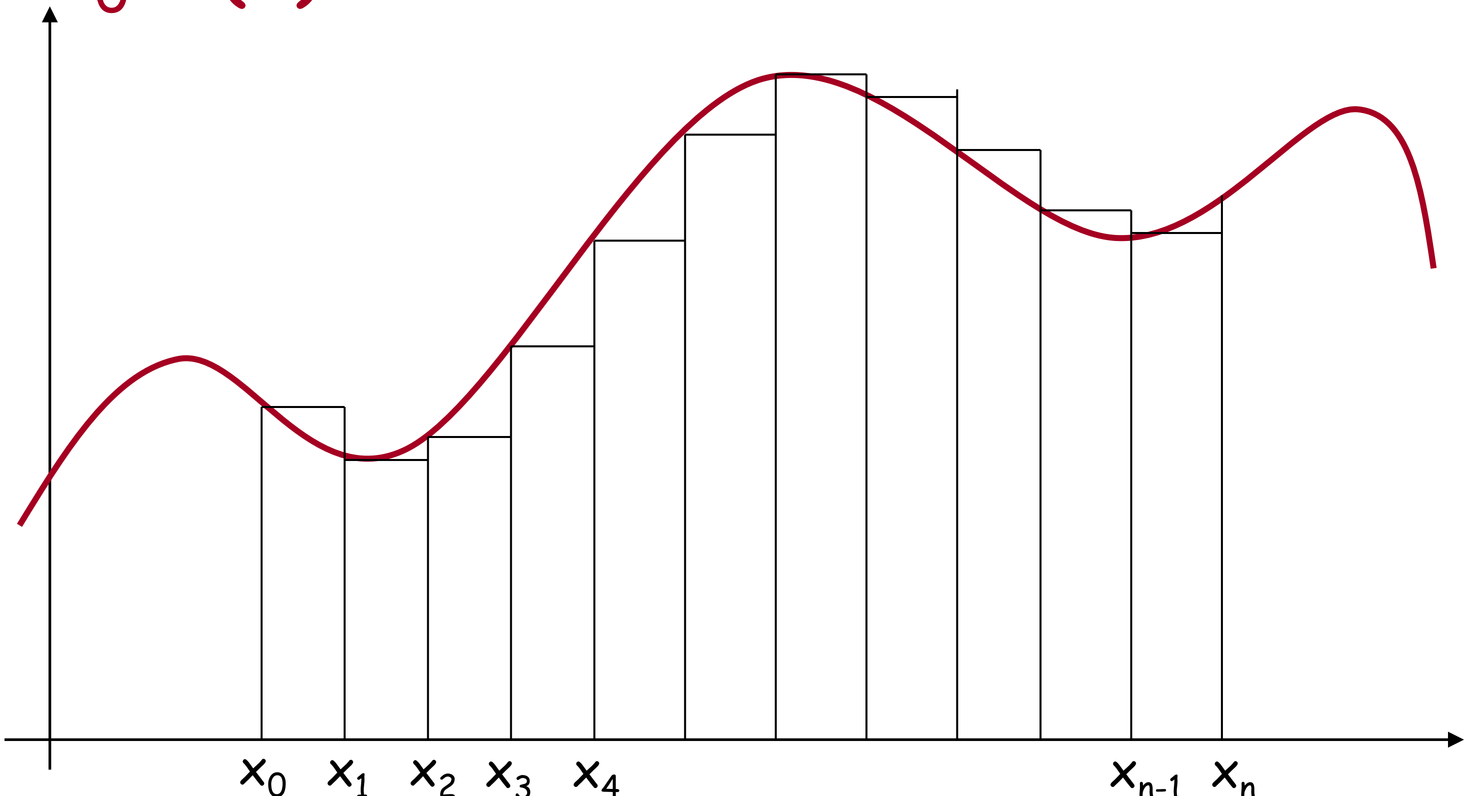
$$a_0 = f(\tfrac{a+b}{2})$$

รูปข้างล่างแสดงถึงวิธีการหาค่า Integrate ด้วยวิธีนี้ และรูปถัดไปแสดงการประมาณค่า Integral ของ Function โดยเพิ่มความถูกต้องด้วยการแบ่งการคำนวณออกเป็นส่วนที่เล็กกว่าและมี Error น้อยกว่า

วิธีการนี้เรียกอีกอย่างหนึ่งว่า Strip Method คือเราแบ่งพื้นที่ออกเป็นแถบของสี่เหลี่ยมผืนผ้า และประมาณค่าพื้นที่โดยคำนวณจากพื้นที่ของสี่เหลี่ยมผืนผ้าที่ได้

Zero-Order Approximation

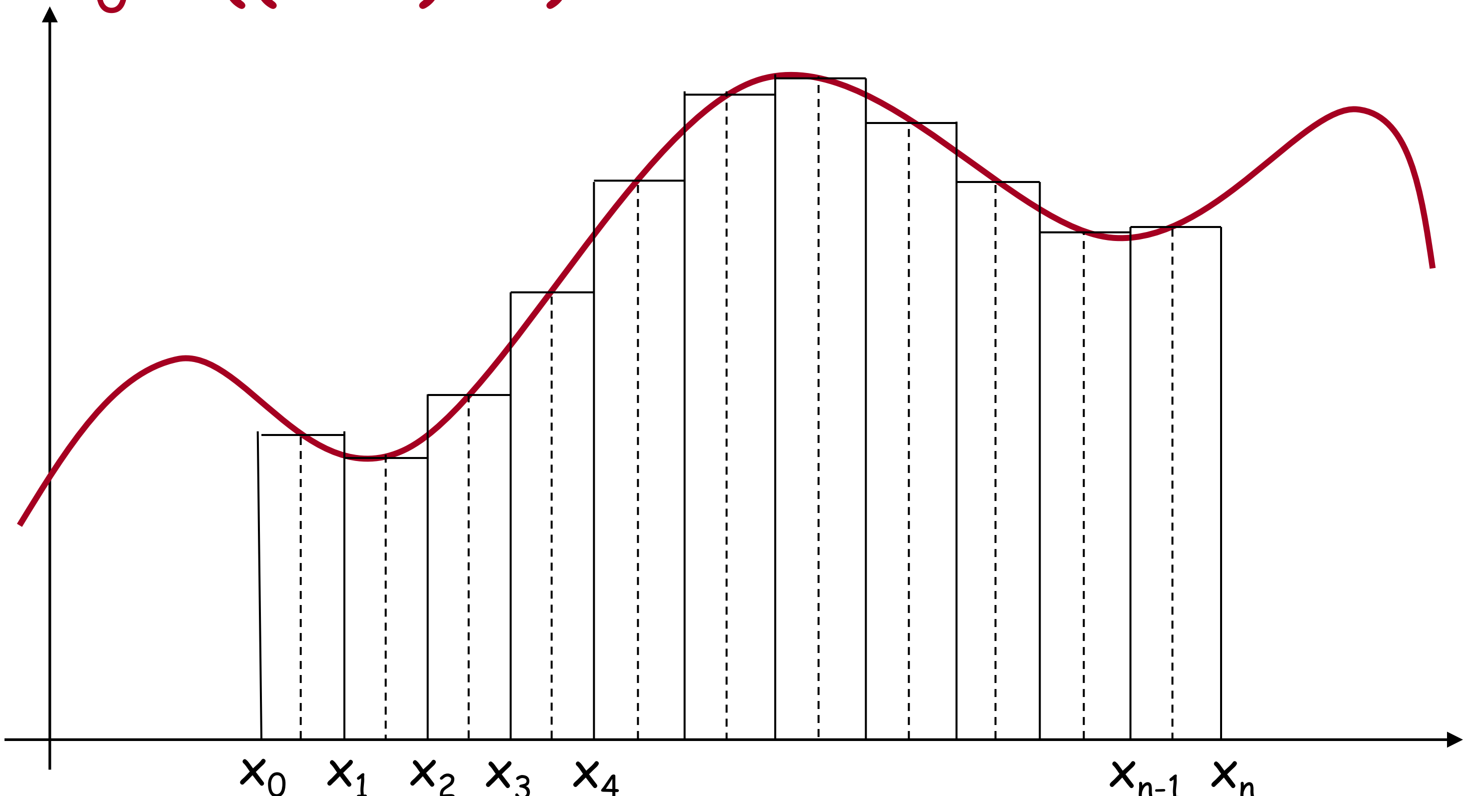
- $a_0 = f(a)$



$$I = \int_a^b f(x)dx \approx \int_a^b a_0 dx = a_0[b-a] = f(a)[b-a]$$

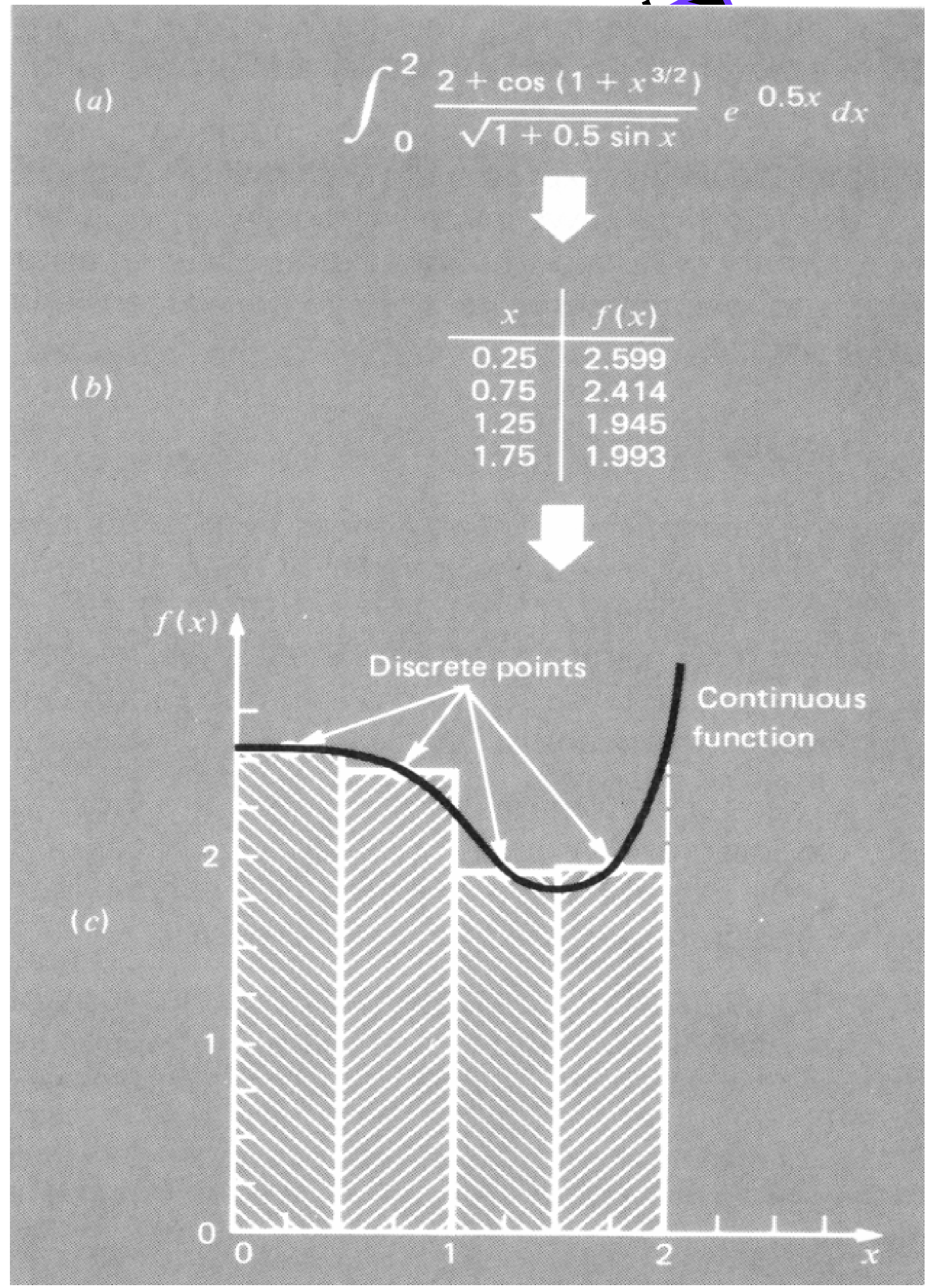
Zero-Order Approximation

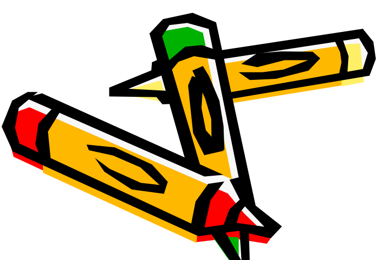
- $a_0 = f((a+b)/2)$



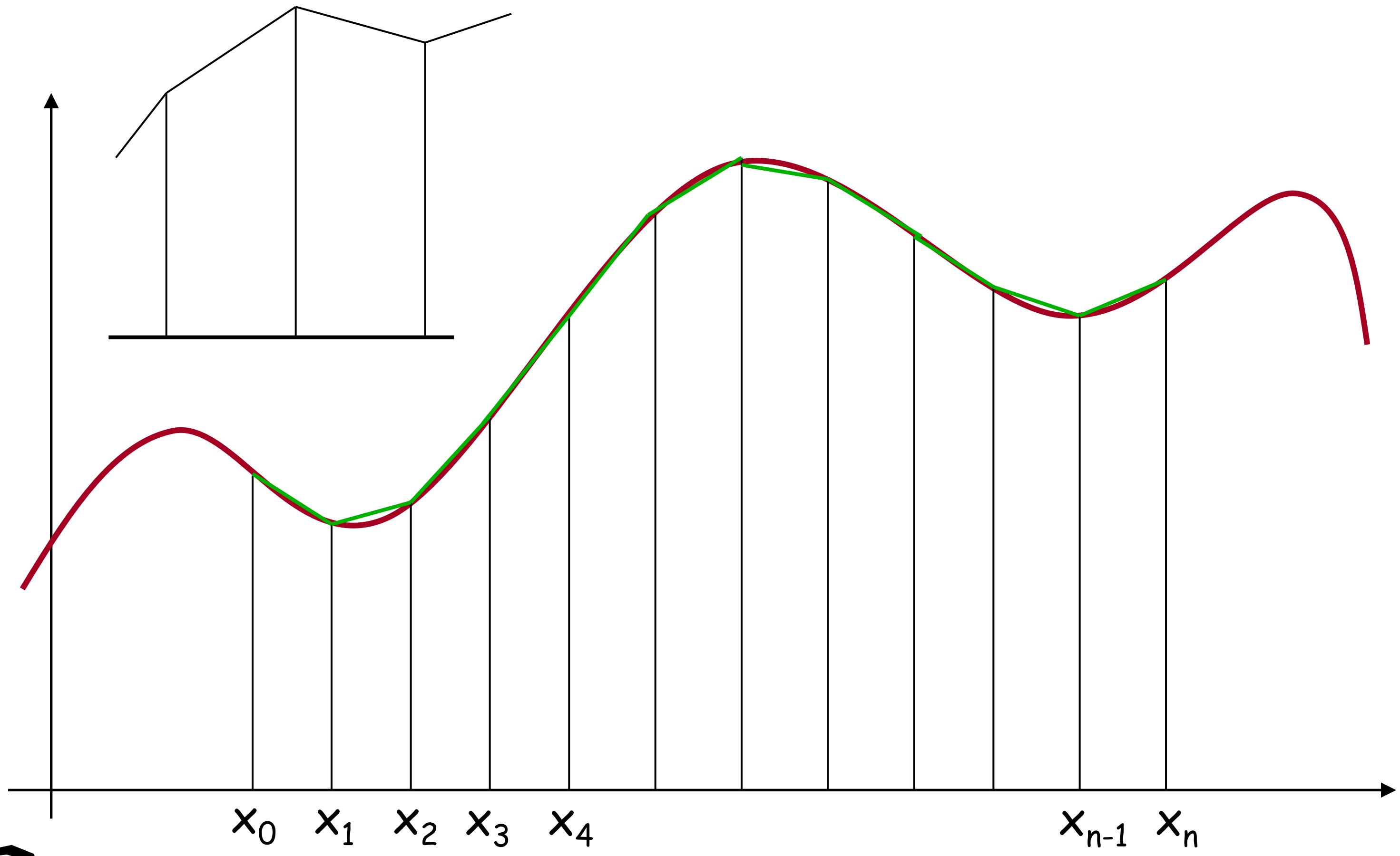
$$I = \int_a^b f(x)dx \approx \int_a^b a_0 dx = a_0[b-a] = f(\frac{a+b}{2})[b-a]$$

Zero-Order Approximation





First-Order Approximation



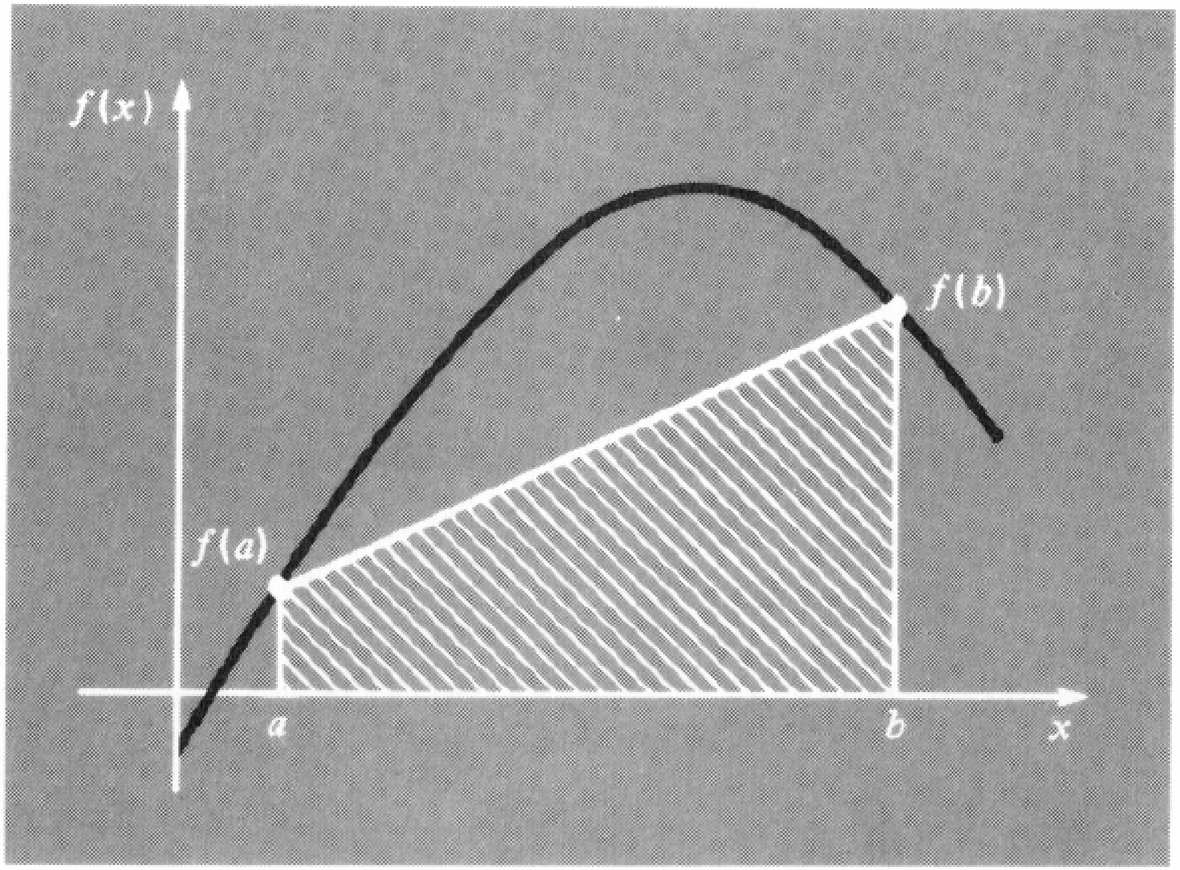
First-Order Approximation

9.4.2 First-Order Approximation: Trapezoidal Rule

ในกรณีที่ Polynomial เป็น First-Order ในรูปของ $f_1(x) = a_0 + a_1 x$ ซึ่งคือสมการเส้นตรง สมการเส้นตรงที่เหมาะสมคือเส้นตรงที่ลากจากจุด a ไปยังจุด b ดังนั้นสมการสามารถเขียนได้เป็น(ดูรูป และจากสมการเส้นตรง $y = a + bx$)

ดังนั้นสมการประมาณค่าจะได้เป็น

First-Order Approximation



$$I = \int_a^b f(x)dx \cong \int_a^b f_1(x)dx = \int_a^b \left[f(a) + \frac{f(b) - f(a)}{b-a}(x-a) \right] dx$$

$$= (b-a)\left[\frac{f(a)+f(b)}{2} \right]$$

: **Trapezoidal Rule**

First-Order Approximation

การนำสมการเส้นตรงมาประมาณค่าของ Function จะยังผลให้เกิด Truncation Error ซึ่งเราสามารถพิสูจน์ได้จาก Newton-Gregory Interpolating Polynomial(รายละเอียดจะไม่กล่าวถึง) ว่าค่าของ Error สำหรับ Trapezoidal Rule จะอยู่ในรูป

$$E_t = -\frac{1}{12} f''(\xi)(b-a)^3$$

โดยที่ค่า ξ จะมีค่าอยู่ระหว่าง a และ b

Example 9.5 ตัวอย่างของ Single-Application ของ Trapezoidal Rule จงทำการประมาณค่า Integral ของ

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

ในช่วงจาก $a = 0$ จนถึง $b = 0.8$ โดยใช้ Trapezoidal Rule จากนั้นคำนวณค่า True Error

Answer:

เราได้ $\int_0^{0.8} f(x)dx = \left[0.2x + 25x^2/2 - 200x^3/3 + 675x^4/4 - 900x^5/5 + 400x^6/6\right]_0^{0.8}$

และคำตอบที่แท้จริงเท่ากับ 1.64053334

เราหาค่า $f(0) = 0.2, \quad f(0.8) = 0.232$

ดังนั้น จาก Trapezoidal Rule เราได้ $I \cong 0.8\dfrac{0.2 + 0.232}{2} = 0.1728$

และ $E_t = 1.64053334 - 0.1728 = 1.46773334, \quad e_t = 89.5\%$

Trapezoidal Rule

จากที่กล่าวมาแล้ว ค่าของ Error สามารถทำให้ลดลงได้ โดยการแบ่งการ Integration เป็นส่วนเล็กๆที่ต่อเนื่องกัน ถ้าเราแบ่งช่วงการ Integration ออกเป็น $n+1$ จุดที่มีระยะห่างเท่ากับ เท่ากับ h และเราจะได้ n Segment ดังนั้น

$$h = \frac{b-a}{n}$$

โดยทั้ง $n+1$ จุดเริ่มนับจาก $x_0, x_1, \ldots, x_n$ ซึ่งจุดเริ่มต้นเราให้ $x_0 = a$ และจุดสุดท้าย $x_n = b$ เราจะได้สมการของ Trapezoidal Rule เป็น

$$I \cong h\frac{f(x_0)+f(x_1)}{2} + h\frac{f(x_1)+f(x_2)}{2} + \cdots + h\frac{f(x_{n-1})+f(x_n)}{2}$$

สรุปแล้ว เราจะได้

$$I \cong \frac{h}{2}\left[f(x_0)+f(x_n)+2\sum_{i=1}^{n-1} f(x_i)\right]$$ **: Trapezoidal Rule**

Trapezoidal Rule

$$I \cong \frac{h}{2}\left[f(x_0) + f(x_n) + 2\sum_{i=1}^{n-1} f(x_i) \right] \qquad \textbf{: Trapezoidal Rule}$$

และค่า Estimate Error สามารถแสดงได้ว่าอยู่ในรูป

$$E_a = -\frac{(b-a)^3}{2n^2}\overline{f''}$$

โดยที่ $\overline{f''}$ คือค่าเฉลี่ยของ Second Derivative ของ Function ในช่วง a ถึง b

Example 9.6 ตัวอย่างของ Multiple-Application ของ Trapezoidal Rule จงทำการประมาณค่า Integral ของ

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

ในช่วงจาก $a = 0$ จนถึง $b = 0.8$ โดยใช้การแบ่งเป็นสอง Segment จากนั้นคำนวณค่า True Error

Answer:

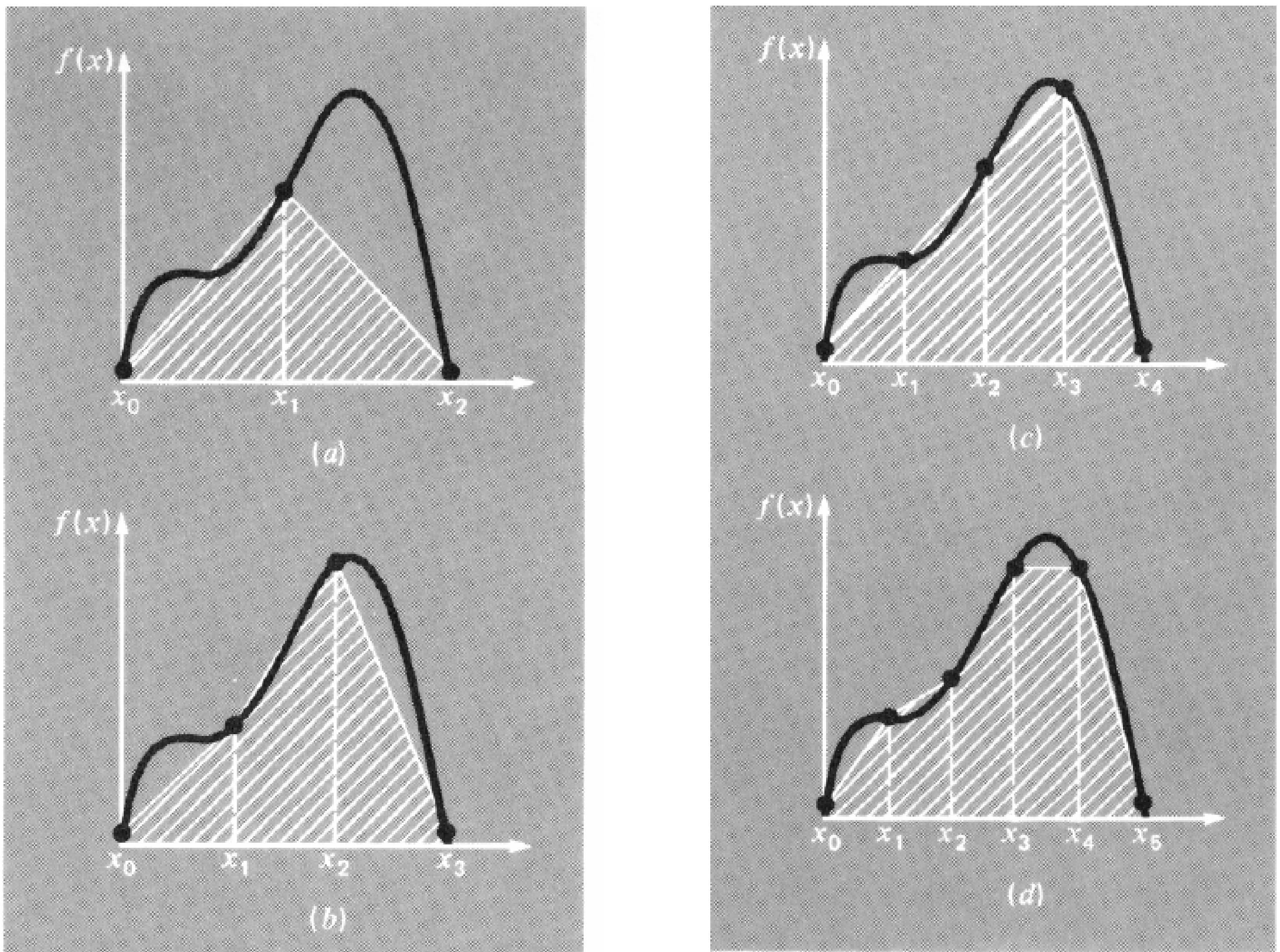
เราหาค่า $f(0) = 0.2, \quad f(0.4) = 2.456, \quad f(0.8) = 0.232$

ดังนั้น $I \cong \dfrac{0.4}{2}[0.2 + 2(2.456) + 0.232] = 1.0688$

เราได้ $E_t = 1.64053334 - 1.0688 = 0.57173,\ e_t = 34.9\%,\ E_a = -\dfrac{0.8^3}{12(2)^2}(-60) = 0.64$

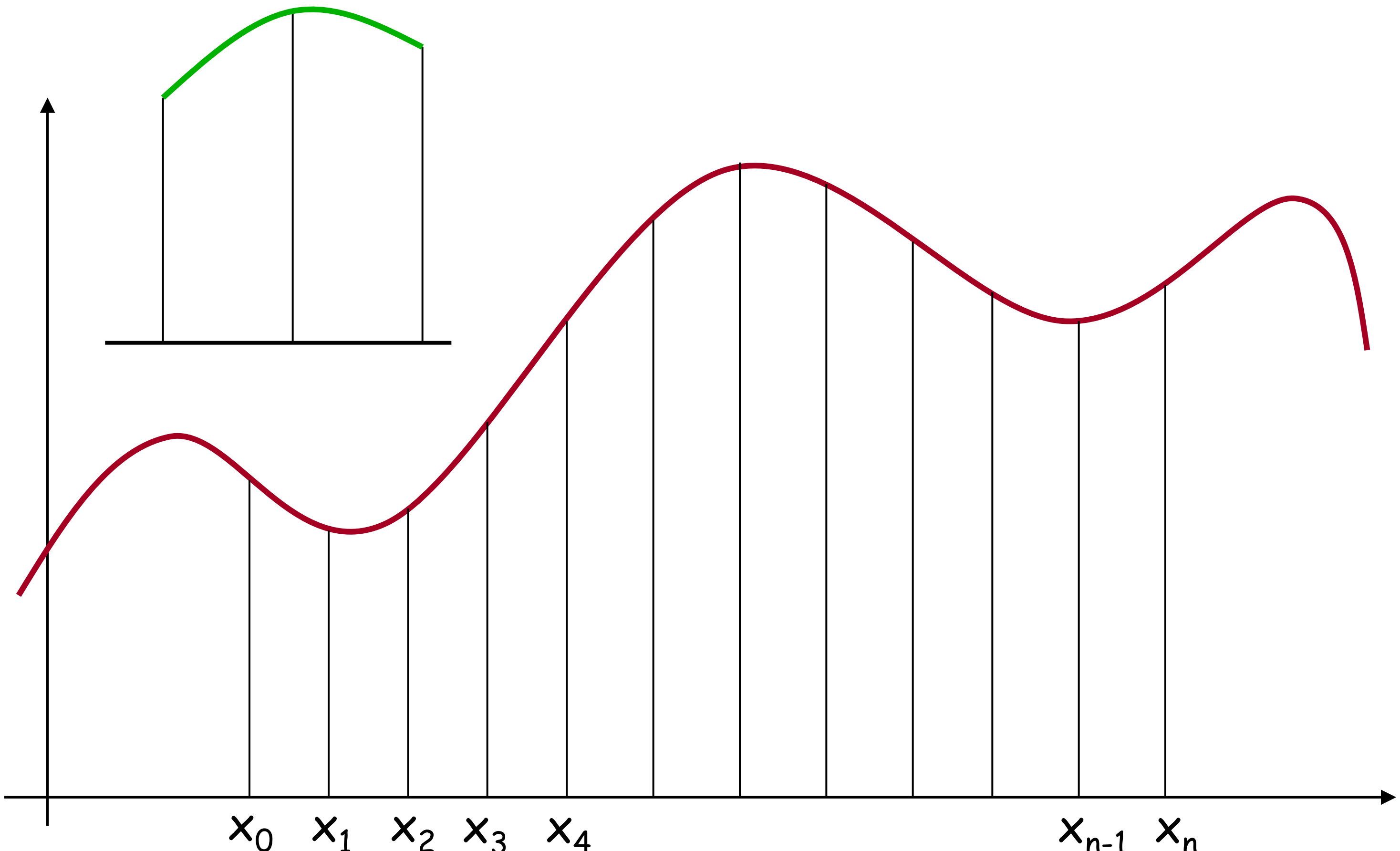
ค่า -60 คือค่าเฉลี่ยของ Second Derivative หาได้จากการ Integrate ของ Second Derivative ในช่วง [0,0.8] และหารด้วยช่วงที่ Integrate หรือ $\overline{f''} = \frac{1}{(0.8-0)}\int_0^{0.8} f''(x)dx$ ขอให้นักศึกษาลองทำดูเอง

รูปข้างล่างสาธิต Error ที่เกิดจากการ Approximation โดยแบ่งเป็น 2, 3, 4 และ 5 Segment และตารางข้างล่างแสดงค่า Error ที่ลดลงของ Function นี้ เมื่อเราเพิ่มจำนวน Segment ซึ่งจะสังเกตได้ว่า ถ้าเราเพิ่ม Segment เป็นสองเท่า Error จะลดลงเป็นหนึ่งในสี่โดยประมาณ



n	h	I	e_t, %
1	0.8	0.1728	89.5
2	0.4	1.0688	34.9
3	0.2667	1.3695	16.5
4	0.2	1.4848	9.5
5	0.16	1.5399	6.1
6	0.1333	1.5703	4.3
7	0.1143	1.5887	3.2
8	0.1	1.6008	2.4
9	0.0889	1.6091	1.9
10	0.08	1.6150	1.6

Second-Order Approximation



Simpson's 1/3 Rule

Second-Order Approximation

Simpson's 1/3 Rule ได้จากการประมาณค่าโดยใช้ Second Order Polynomial หรือสมการ Parabola ซึ่งการสร้างสมการ Parabola $f_2(x) = a_0 + a_1x + a_2x^2$ นั้น เราจะต้องรู้ค่า Coefficient 3 ตัว ดังนั้นเราจะต้องป้อนจุดของค่า $f(x)$ สามจุดและแก้สมการจึงจะหา a_0, a_1, a_2 นอกจากนี้สมการที่ได้ควรจะเริ่มจาก $f(a)$ และจบลงที่ $f(b)$ ซึ่งจุด a และ b คือสองจุดที่เราต้องใช้ และจุดที่สามปกติจะใช้ที่จุดกึ่งกลาง คือ $(a+b)/2$

ในการนี้ เราต้องคำนวณหา Coefficient ทั้งสามของ a_0, a_1, a_2 สำหรับทุก Segment ที่คำนวณ ซึ่งจะทำให้การประมาณค่าใช้การคำนวณที่มาก วิธีที่ดีกว่าคือเราใช้ Polynomial ในรูปของ Lagrange Interpolating Polynomial ดังที่ได้กล่าวในตอนต้นของบท ซึ่งจะหาได้โดยตรงจากการแทนค่าของ Function 3 จุดในสมการ ดังนี้

$$f_2(x) = \sum_{i=0}^{2} L_i(x) f(x_i), \quad L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{2} \frac{x - x_j}{x_i - x_j}$$

Second-Order Approximation

สมมุติเราแบ่ง Segment ออกเป็น n Segment และจุดเริ่มจาก $x_0, x_1, \ldots, x_n$ โดยที่ระยะห่างแต่ละจุดมีค่าเท่ากันคือเท่ากับ h ซึ่งสมการ Parabola แรกจะผ่านสามจุดแรกของ Function คือ $[x_0, f(x_0)], [x_1, f(x_1)], [x_2, f(x_2)]$ ดูรูปประกอบ

ดังนั้น สมการของ Lagrange Interpolating Polynomial เราจะได้

$$
\begin{aligned}
f_2(x) &= L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) \\
&= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2)
\end{aligned}
$$

และการประมาณค่า Integration ของช่วงแรก จาก x_0 ถึง x_2 จะเป็น

$$
I \cong \int_{x_0}^{x_2} \left[\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2) \right] dx
$$

เมื่อทำ Integration ในสมการข้างบน และจัดเรียงเทอมใหม่ ซึ่งขอให้นักศึกษาลองทำเป็นการบ้าน เราจะได้

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

: **Simson's 1/3 Rule**

Second-Order Approximation

ในกรณีของ Multiple Application โดยเราแบ่งออกเป็น n Segment เริ่มจาก $x_0, x_1, \ldots, x_n$ และค่าประมาณของ Integration จะได้จากผลรวมของค่าประมาณในแต่ละ Segment ดังนี้

$$I = \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx + \cdots + \int_{x_{n-2}}^{x_n} f(x)dx$$

$$\cong 2h\frac{f(x_0)+4f(x_1)+f(x_2)}{6} + 2h\frac{f(x_2)+4f(x_3)+f(x_4)}{6}$$

$$+\ldots+2h\frac{f(x_{n-2})+4f(x_{n-1})+f(x_n)}{6}$$

เมื่อรวมเทอมเข้าด้วยกันเราได้

$$I \cong (b-a)\frac{f(x_0)+4\sum_{i=1,3,5,\ldots}^{n-1} f(x_i)+2\sum_{i=2,4,6,\ldots}^{n-2} f(x_i)+f(x_n)}{3n}$$

: Simson's 1/3 Rule

Second-Order Approximation

ในกรณีของ Single Segment นั้น ค่า Error ของ Simson's 1/3 Rule จะดีกว่าที่ควรจะเป็น และพบว่าอยู่ใน $O(h^5)$ แทนที่จะเป็น $O(h^4)$ และเมื่อเทียบกับ $O(h^3)$ ของ Trapezoidal Rule จะดีกว่ามาก ซึ่งค่า Error จะอยู่ในรูปของ(รายละเอียดการคำนวณจะไม่กล่าว ผู้สนใจขอให้ดูจาก Textbook) (Note:สำหรับ Multiple Segment ค่า Error จะเป็น $O(h^4)$ สำหรับ Simpson's Rule เทียบกับ $O(h^2)$ สำหรับ Trapezoidal Rule ถ้ากำหนดช่วง $[a,b]$ ให้คงที่)

$$E_t = -\frac{(b-a)^5}{2880} f^{(4)}(\xi) = -\frac{h}{90} f^{(4)}(\xi); \quad h = \frac{(b-a)}{2}$$

โดยที่ ξ มีค่าอยู่ระหว่าง a และ b และในกรณีของ Multiple Application Simpson's Rule ค่า Error หาได้จากการรวมค่า Error ในแต่ละ Segment ทั้งหมด n Segment และค่าเฉลี่ยของ Forth Derivative ในช่วงนั้น ดังนั้น

$$E_a = -\frac{(b-a)^5}{180n^4} \overline{f}^{(4)}$$

เนื่องจาก Simson's 1/3 Rule แม้ว่าจะเป็นแค่ Second Order Estimate เท่านั้น แต่มี Error ที่เทียบเท่า Third Degree Estimate(Simpson's 3/8 Rule ที่จะกล่าวต่อไป) ดังนั้นจะพบว่าวิธีนี้เป็นวิธีที่นิยมใช้มากที่สุด

Example 9.7 ใช้ Simson's 1/3 Rule ด้วยค่า $n = 4$ ทำการประมาณค่า Integral ของ

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

ในช่วงจาก $a = 0$ จนถึง $b = 0.8$ จากนั้นคำนวณค่า True Error

Answer:

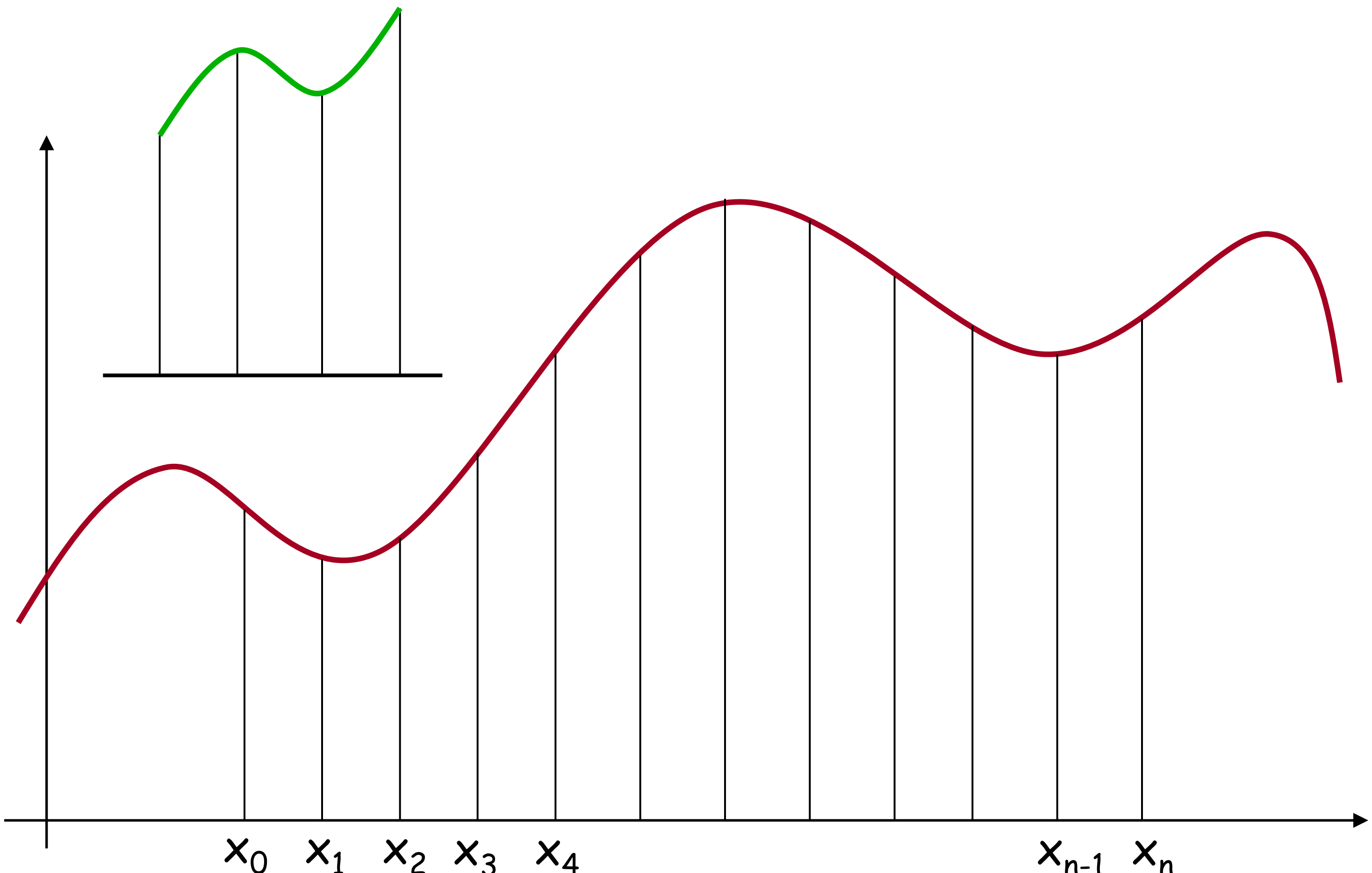
เราได้ $n = 4, h = 0.2$

$$f(0) = 0.2, \quad f(0.2) = 1.288, \quad f(0.4) = 2.456, \quad f(0.6) = 3.464, \quad f(0.8) = 0.232$$

$$I = 0.8\frac{0.2 + 4(1.288 + 3.464) + 2(2.456) + 0.232}{12} = 1.62346667$$

$$E_t = 1.64053334 - 1.62346667 = 0.01706667, \quad e_t = 1.04\%$$

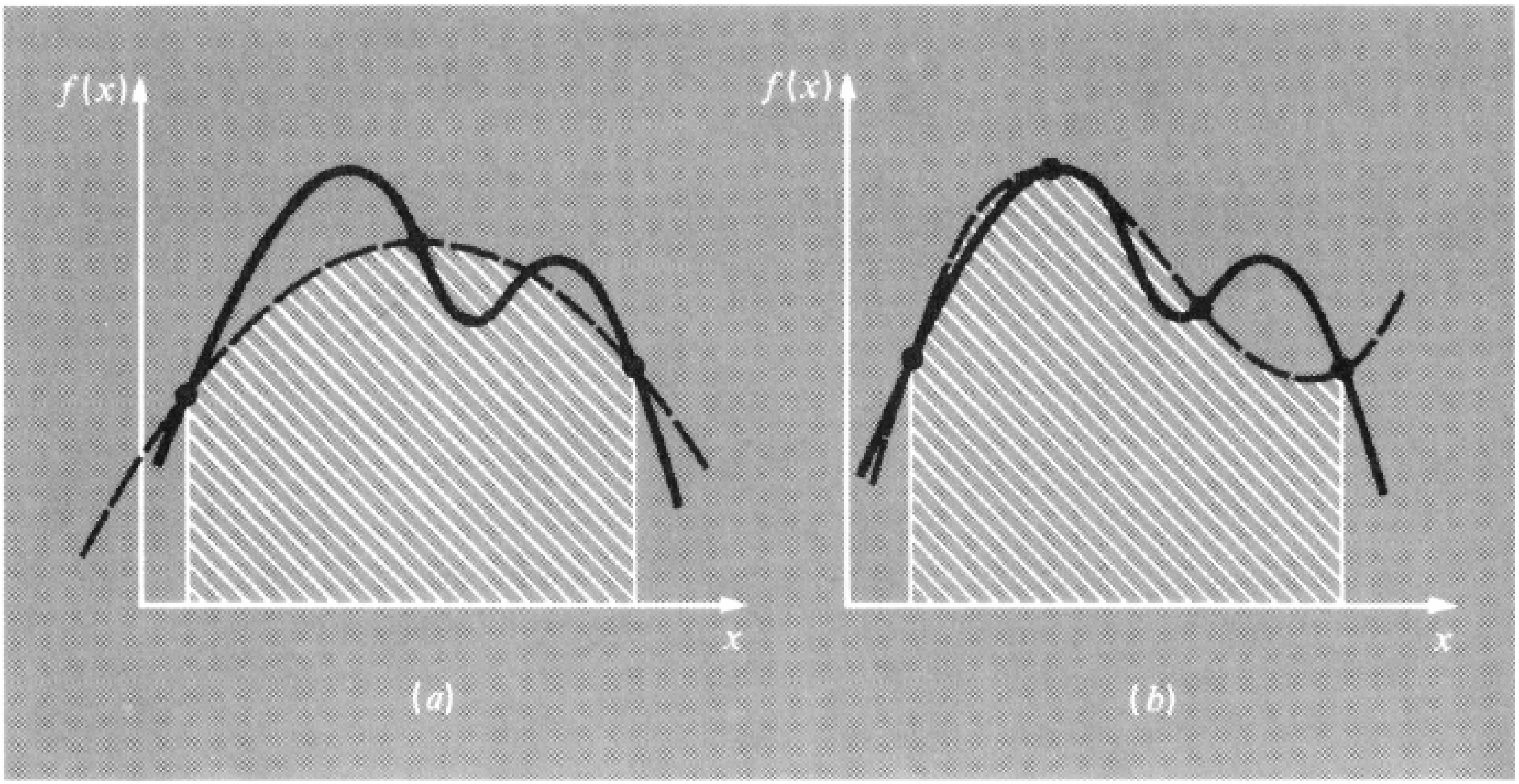
Third-Order Approximation



Simpson's 3/8 Rule

Third Degree Approximation

รูปข้างล่างสาธิตการใช้ Parabola หรือ Second Degree Polynomial มาทำการประมาณค่า Integral ใน Simpson's 1/3 Rule(รูปซ้าย) เปรียบเทียบกับการใช้ Third Degree Polynomial ใน Simpson's 3/8 Rule(รูปขวา)



Simson's 3/8 Rule

9.4.4 Third-Order Approximation: Simpson's 3/8 Rule

ในกรณีนี้ เราใช้ Third Order Lagrange Polynomial มาประมาณค่า Integration

$$I = \int_a^b f(x)dx \cong \int_a^b f_3(x)dx$$

และด้วยวิธีการคล้ายกันกับที่กล่าวในหัวข้อก่อน เราสามารถแสดงได้ว่าสมการข้างบนอยู่ในรูป

$$I \cong \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right] \qquad \text{: } \textbf{Simson's 3/8 Rule}$$

Example 9.8 ใช้ Simson's 3/8 Rule ทำการประมาณค่า Integral ของ

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

ในช่วงจาก $a = 0$ จนถึง $b = 0.8$ จากนั้นคำนวณค่า True Error จากนั้นเปรียบเทียบกับการใช้ Simson's 1/3 Rule ตามด้วย Simson's 3/8 Rule สำหรับในกรณีที่เราแบ่งเป็น 5 Segment

Answer:

กรณีของ Simson's 3/8 Rule เราแบ่งเป็น 4 Segment และได้

$$f(0) = 0.2, \quad f(0.2667) = 1.43272428, \quad f(0.5333) = 3.48717696, \quad f(0.8) = 0.232$$

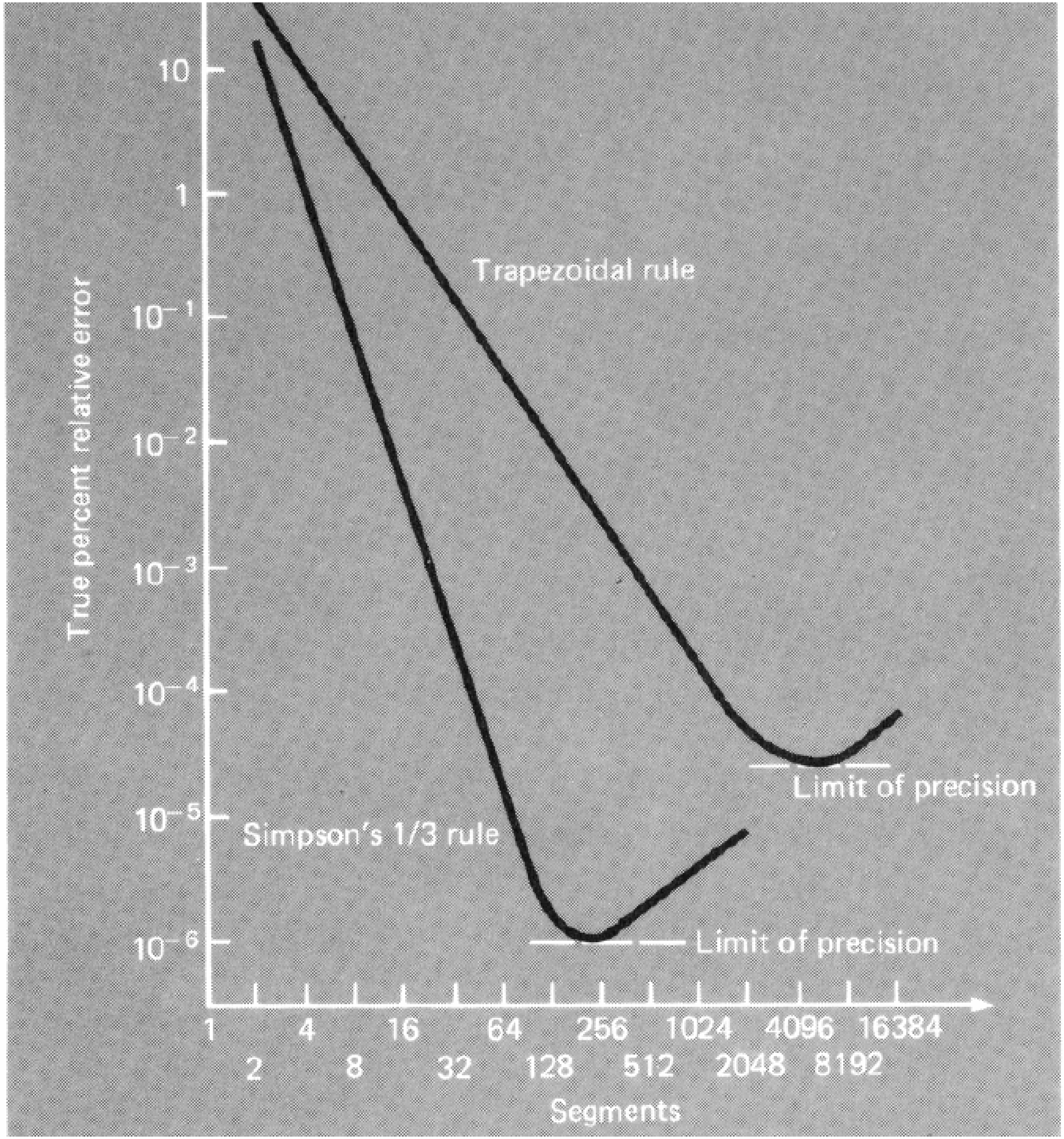
$$I \cong 0.8 \frac{0.2 + 3(1.43272428) + 3(3.48717696) + 0.232}{8} = 1.51917037$$

$$E_t = 1.64053334 - 1.51917037 = 0.12136297, \quad e_t = 7.4\%$$

Romberg Integration

ในกรณีที่เรารู้ Function ที่เราต้องการหาค่า Integration เราสามารถนำ Trapezoidal Rule หรือ Simson's Rule มาใช้ได้ โดยการแบ่งช่วงที่ต้องการตามที่ต้องการ ยิ่งช่วงมีขนาดเล็ก หรือจำนวน Segment มีมาก ค่าของ Integral ที่ได้ก็จะยิ่งถูกต้องมากขึ้นเท่านั้น อย่างไรก็ตามถ้าเราแบ่งช่วงให้เล็กลงไปเรื่อยๆ จนถึงจุดหนึ่งที่ค่า Error กลับจะสูงขึ้น นั่นคือ Limit ที่ถูกกำหนดสำหรับแต่ละวิธี ซึ่งเป็นผลมาจาก Round Off Error

จากรูปแสดงกราฟเปรียบเทียบค่า Error ที่เกิดกับจำนวนของ Segment ที่ใช้ ระหว่าง Trapezoidal Rule และ Simpson's 1/3 Rule ของ Function $f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$ ในช่วงตั้งแต่ $x = 0$ จนถึง $x = 0.8$ จะสังเกตได้ว่าค่า Limit ของ Precision ที่เกิดในแต่ละวิธี และจำนวน Segment ที่ดีที่สุดของแต่ละวิธี อันเนื่องมาจาก Round Off Error นั้นไม่เท่ากัน



True percent relative error
10
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
Trapezoidal rule
Simpson's 1/3 rule
Limit of precision
Limit of precision
1
4
16
64
256
1024
4096
16384
2
8
32
128
512
2048
8192
Segments

Romberg Integration

ในกรณีเช่นนี้ ถ้าเราต้องการความถูกต้องสูงๆ เราจำเป็นที่จะต้องใช้วิธีอื่นในการประมาณค่า ซึ่ง ***Romberg Integration*** เป็นเทคนิคหนึ่งที่นำมาใช้ได้ดี โดยใช้การทำ Trapezoidal Rule หลายๆครั้งติดต่อกันเขียนเป็น Algorithm การคำนวณจะใช้พื้นฐานมาจากวิธีของ ***Richardson Extrapolation*** ในการกำจัด Error สำหรับแต่ละครั้งที่ทำการคำนวณ

อีกวิธีหนึ่งคือวิธีที่เรียก ***Gauss Quadrature*** ซึ่งเป็นวิธีที่ปรับมาจากวิธีของ Newton-Cotes Formula เช่นกัน แต่ใช้วิธีการคำนวณสำหรับการหาค่าแต่ละช่วงที่ดีกว่า โดยมีการเลือกตำแหน่งของจุดที่เหมาะสมกว่า กรรมวิธีของ Gauss Quadrature จะไม่กล่าวในชั้นนี้ ผู้ที่สนใจสามารถศึกษาได้จากตำราของ Numerical Method ทั่วไป

Romberg Integration

9.5.1 Richardson's Extrapolation

วิธีการของ Richardson's Extrapolation จะใช้วิธีการของการ Estimate ค่า Integral สองอัน และมาเปรียบเทียบกับอันที่สาม เพื่อมาแก้ไข Error ที่เกิดขึ้น ซึ่งค่า Error ที่เกิดจากการทำ Multiple Application ของ Trapezoidal Rule สามารถเขียนในรูป

$$I = I(h) + E(h)$$

โดยที่ I คือค่า Integral ที่แท้จริง, $I(h)$ คือค่าประมาณของ Integral ที่ได้จากการทำ n Segment โดย $h = (b-a)/n$ และ $E(h)$ เป็น Truncation Error ที่เกิด ดังนั้นถ้าเราทำการ Estimate สองครั้งโดยใช้ช่วงห่างไม่เท่ากัน คือ h_1 และ h_2 เราสามารถเขียน

$$I = I(h_1) + E(h_1) = I(h_2) + E(h_2)$$

จากที่กล่าวมาแล้วว่าค่า Estimate Error จากการทำ Multiple Application ของ Trapezoidal Rule จะอยู่ในรูป

$$E_a = -\frac{(b-a)^3}{12n^2}\bar{f}''$$

Romberg Integration

เมื่อแทนค่า $n = (b-a)/h$ เราได้ค่า Estimate Error เป็น

$$E_a = -\frac{b-a}{12}h^2\bar{f}''$$

ดังนั้น เราสรุปได้ว่า อัตราส่วนระหว่าง Error ทั้งสอง Estimate จะมีค่าประมาณเป็น

$$\frac{E(h_1)}{E(h_2)} \approx \frac{h_1^2}{h_2^2}$$

หรือ

$$E(h_1) \approx E(h_2)\left(\frac{h_1}{h_2}\right)^2$$

และเมื่อแทนค่ากลับลงในสมการเดิม เราได้

$$I = I(h_2) + E(h_2) \approx I(h_1) + E(h_2)\left(\frac{h_1}{h_2}\right)^2$$

Romberg Integration

และเมื่อจัดเรียงสมการใหม่ เพื่อหาค่า $E(h_2)$ เราจะได้

$$E(h_2) \approx \frac{I(h_1) - I(h_2)}{1 - (h_1 / h_2)^2}$$

และท้ายสุด เราสามารถนำค่าดังกล่าวมาปรับปรุงการ Estimate ในครั้งที่สองได้เป็น

$$I = I(h_2) + E(h_2) \approx I(h_2) + \left[\frac{1}{(h_1 / h_2)^2 - 1}\right]\left[I(h_2) - I(h_1)\right]$$

เราสามารถแสดงให้เห็นได้ว่า ค่า Error ที่ได้จากการ Estimate นี้มี $O(h^4)$ ซึ่งได้จากการรวม Trapezoidal Rule สองอัน แต่ละอันมี $O(h^2)$ เข้าด้วยกัน ในกรณีที่เราใช้ระยะห่างของแต่ละจุดเท่าๆกัน จุดที่เหมาะสมคือที่ $h_2 = h_1 / 2$ และสมการข้างบนจะเป็น

$$I \approx I(h_2) + \frac{1}{2^2 - 1}\left[I(h_2) - I(h_1)\right] = \frac{4}{3} I(h_2) - \frac{1}{3} I(h_1)$$

Example 9.9 จาก Function เดิม เปรียบเทียบการทำ Trapezoidal Rule โดยใช้วิธีของ Error Correction กับวิธีปกติ

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

ในช่วงจาก $a = 0$ จนถึง $b = 0.8$

Answer:

ในตัวอย่างก่อนๆ เราทำ Single Application และ Multiple Application โดยใช้ Trapezoidal Rule ของ Function นี้มาแล้ว ซึ่งผลสรุปในตารางข้างล่าง

Segment	h	Integral	e_t, %
1	0.8	0.1728	89.5
2	0.4	1.0688	34.9
4	0.2	1.4848	9.5

ในการ Estimate จาก หนึ่งและสอง Segment เราได้

$$I \approx \frac{4}{3}(1.0688) - \frac{1}{3}(0.1728) = 1.36746667$$

$$E_t = 1.64053334 - 1.36746667 = 0.27306667, \quad e_t = 16.6\%$$

Romberg Integration

ในการ Estimate จาก หนึ่งและสอง Segment เราได้

$$I \approx \frac{4}{3}(1.0688) - \frac{1}{3}(0.1728) = 1.36746667$$

$$E_t = 1.64053334 - 1.36746667 = 0.27306667, \quad e_t = 16.6\%$$

สำหรับการ Estimate จาก สองและสี่ Segment เราได้

$$I \approx \frac{4}{3}(1.4848) - \frac{1}{3}(1.0688) = 1.62346667$$

$$E_t = 1.64053334 - 1.62346667 = 0.01706667, \quad e_t = 1.0\%$$

Romberg Integration

วิธีการของ Richardson's Extrapolation สามารถนำมาทำต่อ โดยใช้การ Estimate ที่ได้ใน $O(h^4)$ ที่กล่าวมาแล้วสองอัน มาทำการ Estimate ใหม่อีกหน ซึ่งจะให้ความถูกต้องยิ่งขึ้นไปอีกใน $O(h^6)$ ในกรณีที่เราใช้ค่าระยะห่างแต่ละจุดเท่าๆกัน ซึ่งการ Estimate แต่ละครั้งเราใช้ขนาดของ Step เป็นครึ่งหนึ่งของอันเดิม สมการของ $O(h^6)$ สามารถพิสูจน์ได้ว่าอยู่ในรูป

$$I \approx \frac{16}{15} I_m - \frac{1}{15} I_l$$

โดยที่ I_m คือค่าการ Estimate ที่ถูกต้องมากกว่า และ I_l คือการ Estimate ที่ถูกต้องน้อยกว่า(ใช้ Step Size ขนาดใหญ่กว่าเป็นสองเท่า) ในทำนองเดียวกันถ้าเราทำต่อไปอีก โดยการรวมสอง Estimate ที่มี $O(h^6)$ เราจะได้ค่า Estimate ที่มีความถูกต้องใน $O(h^8)$ และสามารถแสดงเป็นสมการสำหรับกรณีที่ Step Size เท่ากัน เป็น

$$I \approx \frac{64}{63} I_m - \frac{1}{63} I_l$$

Romberg Integration

Example 9.10 ทดลองใช้การ Estimate ใน $O(h^6)$ สำหรับข้อมูลในตัวอย่างที่ 9.9 และเปรียบเทียบ Error

Answer:

เราได้ $I \approx \frac{16}{15}I_m - \frac{1}{15}I_l = \frac{16}{15}(1.62346667) - \frac{1}{15}(1.36746667) = 1.64053334$

สังเกตว่าในกรณีนี้เราได้คำตอบที่ถูกต้องถึงตัวเลขนัยสำคัญ 9 หลัก

Romberg Integration

จากที่กล่าวในหัวข้อก่อน ในการปรับปรุงค่าของ Integration ให้ดีขึ้นเรื่อยๆสำหรับแต่ละ Step สำหรับการใช้ระยะห่างแต่ละจุดที่เท่าๆกัน และเป็นครึ่งหนึ่งของ Step ก่อนหน้านี้ ซึ่งเราสามารถเขียนเป็นสมการทั่วไปได้ในรูปของ

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

โดยที่ $I_{j+1,k-1}$ และ $I_{j,k-1}$ คือค่า Integration ที่ถูกต้องมากกว่า และถูกต้องน้อยกว่า ตามลำดับ ที่ได้ก่อนหน้านี้ ส่วนค่า $I_{j,k}$ คือค่า Estimate อันใหม่ โดยที่เมื่อ $k = 1$ เราได้ Trapezoidal Rule เดิม, ส่วน $k = 2$ หมายถึงการ Estimate อันใหม่ที่เป็น $O(h^4)$ และ $k = 3$ คือการ Estimate ที่เป็น $O(h^6)$ และต่อไปเรื่อยๆ สำหรับค่า j ใช้เพื่อแสดงถึงความแตกต่างของการ Estimate สองครั้งมี่มีความถูกต้องมากกว่า $(j+1)$ และน้อยกว่า (j) ของการใช้ Trapezoidal Rule เริ่มจาก Single Segment Application, 2 Segment, 4 Segment และต่อไปเรื่อยๆ

Romberg Integration

	$O(h^2)$	$O(h^4)$	$O(h^6)$	$O(h^8)$
(IT =1)	0.17280000	1.36746667		
	1.06880000			
(IT =2)	0.17280000	1.36746667	1.64053334	
	1.06880000	1.62346667		
	1.48480000			
(IT =3)	0.17280000	1.36746667	1.64053334	1.64053334
	1.06880000	1.62346667	1.64053334	
	1.48480000	1.63946667		
	1.60080000			

จากรูปข้างบน สังเกตว่า $I_{4,1}$ ได้ค่าที่ถูกต้องแล้วในระดับ 9 Significant Digit และเมื่อเปรียบเทียบกับวิธีการเดิม จากรูปกราฟที่แสดงให้ดูก่อนหน้านี้ พบว่า Simpson's 1/3 Rule ต้องใช้ถึง 256 Segment ที่จะได้ค่า Estimate เป็น 1.64053332 และเราจะไม่ได้ค่าที่ดีกว่านี้เนื่องมาจาก Round Off Error ในขณะที่ Romberg Integration ได้ค่าที่ถูกต้อง(ถึง 9 หลัก) โดยใช้พื้นฐานจากการรวมกันของ 1, 2, 4, และ 8-Segment Trapezoidal Rule

Summary

Method	Data Points Required for One Application	Data Points Required for n Application	Truncation Error	Application	Programming Effort	Comments
Trapezoidal Rule	2	$n+1$	$\cong h^3 f''(\xi)$	Wide	Easy	
Simpson's 1/3 Rule	3	$2n+1$	$\cong h^5 f^{(4)}(\xi)$	Wide	Easy	
Simpson's 1/3 and 3/8 Rule	3 or 4	≥ 3	$\cong h^5 f^{(4)}(\xi)$	Wide	Easy	
Higher-Order Newton-Cotes	5 or More	N/A	$\geq h^7 f^{(6)}(\xi)$	Rare	Easy	
Romberg Integration	3			Requires $f(x)$ Be Known	Moderate	ไม่เหมาะกับ Data ที่เป็นตาราง
Gauss Quadrature	2 or More	N/A		Requires $f(x)$ Be Known	Easy	ไม่เหมาะกับ Data ที่เป็นตาราง

Homework 10: Ch 11

- Download HW
- Next Week
  - ส่งการบ้าน HW 10
  - Chapter 11 Solutions of ODE
  - HW 11 (Option)

CPE 332
Computer Engineering Mathematics II

Part III,
Chapter 12 ODE

Today Topics

- Chapter 12 Ordinary Differential Equation
  - HW 10 Due Today
  - HW 11 Due Wed 4 May, 12.00 Noon
    - ส่งห้องเลขาภาค
- Course Ends This Week
  - No Chapter 13(Curve Fitting)

Differential Equation

- เป็นสมการทางคณิตศาสตร์ ที่ประกอบด้วย Derivative (หรือ Integral)
  - Solution ของสมการคือ Function ที่ไม่มี Derivative หรือ Integral ปรากฏอยู่
- รูปแบบทั่วไปคือ (First Order)
  - $dy/dx = f(x,y)$
  - กรณีพิเศษที่ $dy/dx = f(x)$ เราสามารถแก้สมการโดยใช้การ Integrate
    - $dy = f(x)dx$
    - $y = \int f(x)dx = F(x) + C$: ค่า C สามารถหาได้โดยการกำหนด Initial condition
  - ปกติการแก้สมการทั่วไปไม่สามารถใช้ Integrate ได้

Example 1

- Solve for $\dfrac{dy}{dx} = 3x^2; \quad x = 2, y = 4$

Example 1

- Solve for $\dfrac{dy}{dx} = 3x^2; \quad x = 2, y = 4$

$$dy = 3x^2 dx$$

$$y = \int 3x^2 dx$$

$$y = x^3 + C \text{ (General Solution)}$$

Example 1

- Solve for $\dfrac{dy}{dx} = 3x^2; \quad x = 2, y = 4$

$$dy = 3x^2 dx$$

$$y = \int 3x^2 dx$$

$$y = x^3 + C \text{ (General Solution)}$$

$$\mathrm{x} = 2, \mathrm{y} = 4$$

$$4 = 2^3 + C$$

$$C = -4$$

$$y = x^3 - 4$$

Differential Equation

- ค่า Order สูงสุดของ Derivative จะกำหนด Order ของ Differential Equation
  - จำนวน Initial Condition ที่จะต้องใช้จะเท่ากับค่าของ Order ของสมการ

Example 2

$$F(x,y): xy^3 + 2y + x^2 = 4 \quad \text{Find } f(x,y) = \frac{d}{dx}F(x,y)$$

$$\frac{d}{dx}xy^3 + 2y + x^2 = \frac{d}{dx}4$$

$$[x\frac{d}{dx}y^3 + y^3\frac{dx}{dx}] + 2\frac{dy}{dx} + 2x\frac{dx}{dx} = 0$$

$$[3xy^2\frac{dy}{dx} + y^3] + 2\frac{dy}{dx} + 2x = 0$$

$$3xy^2\frac{dy}{dx} + 2\frac{dy}{dx} = -2x - y^3$$

$$\frac{dy}{dx}[3xy^2 + 2] = -2x - y^3$$

$$\frac{dy}{dx} = \frac{-2x - y^3}{[3xy^2 + 2]}$$

Example 2

- แก้สมการ $\dfrac{dy}{dx} = f(x,y) = \dfrac{-2x - y^3}{[3xy^2 + 2]}$

$$dy = \frac{-2x - y^3}{[3xy^2 + 2]} dx$$

$$y = \int \frac{-2x - y^3}{[3xy^2 + 2]} dx$$

Integrate ไม่ได้

Differential Equation

- Tools ทางคณิตศาสตร์ที่สำคัญสำหรับใช้ในการแก้ปัญหา Differential Equation
  - Laplace Transform (one side/two side)
  - Fourier Transform ใช้ได้เช่นกัน
  - Z-Transform ใช้สำหรับแก้ปัญหา Discrete Version ของ Differential Equation
    - คือ Differential Equation ที่ได้จากการสุ่มตัวอย่างของตัวแปร
    - กรณีนี้เราเรียก Difference Equation

ทั้งหมดนี้อยู่ในเนื้อหาวิชา *CPE 308*

Differential Equation

- บทนี้เราจะมาดู Numerical Method สำหรับใช้ในการแก้สมการ Differential Equation
  - เราจะจำกัดอยู่ที่ First Order และ Initial Condition ที่จุดตั้งต้น
  - เป็นสมการของ Ordinary Differential Equation
    - One Independent Variable
  - สามารถดัดแปลงสำหรับ Higher Order ได้
  - กรณีที่เป็น Boundary Condition จะต้องใช้วิธีอื่น
    - ศึกษาเพิ่มเติมได้จาก Reference

Chapter 10

- Ordinary Differential Equation

บทนี้จะเป็นบทที่ 4 และเป็นบทสุดท้ายที่เราจะกล่าวใน Part ที่ 3 นี้ โดยจะเป็นหัวข้อเกี่ยวกับการนำคอมพิวเตอร์มาแก้สมการของ Differential Equation ซึ่งเนื้อหาจะครอบคลุมเฉพาะสมการที่ประกอบด้วยหนึ่ง Independent Variable เท่านั้น หรือเป็นสมการที่เราเรียก Ordinary Differential Equation(ODE) ซึ่งถ้าสมการมีมากกว่าหนึ่ง Independent Variable เราจะเรียก Partial Differential Equation(PDE)

ปกติการจัดสมการของ Differential Equation จะบ่งบอกเป็น Order ของสมการ ซึ่งคือค่า Order ของ Derivative ที่สูงที่สุดในสมการนั้น นอกจากนี้แล้ว สมการของ Differential Equation ยังจัดเป็นได้อีกหลายแบบ ที่สำคัญคือ Linear Constant Coefficient Differential Equation ซึ่งรายละเอียดและคำจำกัดความเหล่านี้ นักศึกษาควรจะได้เรียนมาบ้างแล้วในวิชา Calculus

ODE

Solution ของสมการ Ordinary Differential Equation นั้นคือ Function ของ Independent Variable และค่า Parameter ที่ทำให้สมการเป็นจริง ยกตัวอย่าง Function ข้างล่าง

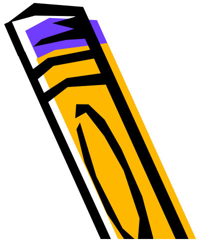
$$y = -0.5x^4 + 4x^3 - 10x^2 + 8.5x + 1$$

ซึ่งเป็น Polynomial ที่มี Order เท่ากับสี่ ถ้าเราทำการ Differentiate สมการดังกล่าว เราจะได้ Ordinary Differential Equation

$$\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5$$

สมการที่ได้จะอธิบายพฤติกรรมของ Polynomial ข้างบน แต่ในมุมมองที่ต่างกัน ซึ่งแทนที่สมการจะแสดงความสัมพันธ์ในค่าของ y ต่อค่าของ x แต่มันจะแสดงอัตราการเปลี่ยนแปลงของค่าของ y ต่อการเปลี่ยนแปลงของ x (ซึ่งก็คือค่า Slope) ที่จุดต่างๆของ x รูปข้างล่างแสดง Plot ของ Function และ Derivative ของมัน

ODE



จาก Differential Equation ที่ได้ ถ้าเราพยายามหา Solution ของมัน คือ Function เดิม ในกรณีนี้ เราทำได้โดยการ Integration ดังนี้

$$y = \int \left[-2x^3 + 12x^2 - 20x + 8.5\right] dx$$

$$= -0.5x^4 + 4x^3 - 10x^2 + 8.5x + C$$

ซึ่งจะอยู่ในรูปแบบที่เหมือนสมการดั้งเดิม ยกเว้นที่จุดเดียว คือในการทำการ Differentiate และ Integrate กลับมานั้น เราได้สูญเสียข้อมูลของค่า Constant ไป ซึ่งถ้าขาดค่านี้แล้วจะทำให้ Solution ที่ได้มีจำนวนที่ไม่จำกัด ขึ้นอยู่กับค่า C ที่เป็นไปได้ ซึ่งไม่จำกัดเช่นกัน(ดูรูปถัดไป)

อย่างไรก็ตาม คำตอบที่ถูกต้องจะมีคำตอบเดียว และในการที่จะหาคำตอบดังกล่าว เราต้องกำหนดสภาวะช่วย (Auxiliary Condition) สำหรับสมการของ First-Order ODE สภาวะช่วยดังกล่าวก็คือ Initial Condition ที่จะใช้ในการหาค่า Constant ยกตัวอย่างเช่นถ้าเรากำหนด Initial Condition ที่ $x = 0, y = 1$ เมื่อแทนค่าในสมการที่ได้

$$1 = -0.5(0)^4 + 4(0)^3 - 10(0)^2 + 8.5(0) + C$$

เราสามารถแก้สมการและหาค่า $C = 1$ และเราจะได้ Unique Solution และเมื่อแทนค่าที่ได้กลับในสมการของ Solution เราจะได้ คำตอบของ Differential Equation ที่ต้องการ

ODE

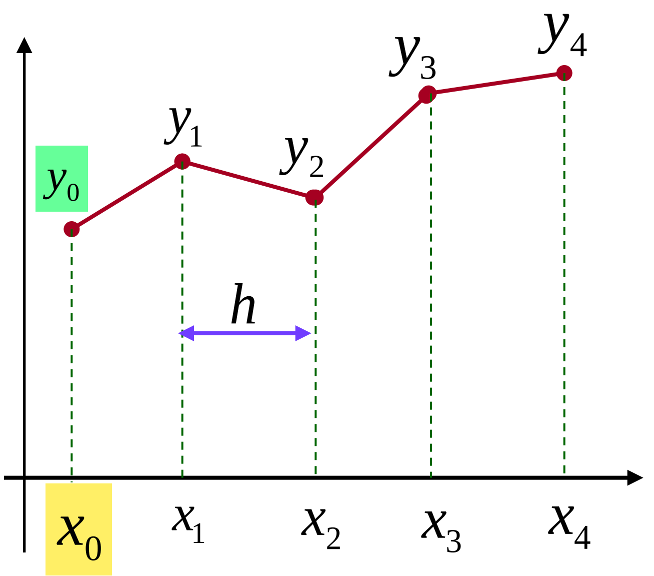
การหา Solution ของ N-Order Differential Equation นั้น เราจะต้องการ Condition ทั้งหมด N ตัวในการหา Unique Solution ถ้าทุกๆ Condition ที่ได้ ถูกกำหนดไว้ที่ค่าของ Independent Variable ตัวเดียวกัน การแก้ปัญหาของเราจะเรียก Initial-Value Problem แต่ถ้าค่า Condition ถูกกำหนดที่ค่าต่างกันของ Independent Variable ปัญหานี้จะเรียก Boundary-Value Proble

ในบทนี้ จะคำนึงถึงปัญหาประเภท Initial-Value Problem เท่านั้น

ODE: One Step Method

เราจะมุ่งความสนใจสำหรับการแก้ปัญหาสมการ Ordinary Differential Equation ที่อยู่ในรูป

$$\frac{dy}{dx} = f(x,y)$$



ซึ่งการใช้วิธีการของ Numerical มาแก้ปัญหานั้น สมการที่ใช้จะอยู่ในรูปของ

ค่าใหม่ = ค่าเก่า + Slope X Step Size

หรือเขียนในลักษณะของสมการทางคณิตศาสตร์

$$y_{i+1} = y_i + \phi h$$

ซึ่งจากสมการ ค่า Estimate ของ Slope ϕ จะถูกใช้ในการ Extrapolate ในการหาค่าใหม่ y_{i+1} จากค่าเก่า y_i ในช่วงของระยะทาง h ซึ่งสมการสามารถกระทำทีละ Step ในการคำนวณหาเส้นทางของ Solution ที่ต้องการ

ODE: One Step Method

$$y_{i+1} = y_i + \phi h$$

ซึ่งจากสมการ ค่า Estimate ของ Slope ϕ จะถูกใช้ในการ Extrapolate ในการหาค่าใหม่ y_{i+1} จากค่าเก่า y_i ในช่วงของระยะทาง h ซึ่งสมการสามารถกระทำทีละ Step ในการคำนวณหาเส้นทางของ Solution ที่ต้องการ

วิธีการดังกล่าว เรียกว่า One-Step Method ซึ่งในกรรมวิธีที่จะกล่าวต่อไป จะอยู่ในรูปของสมการนี้ เพียงแต่ความแตกต่างจะอยู่ในส่วนของการ Estimate ค่า Slope ซึ่งวิธีการที่ง่ายที่สุดก็คือการ Estimate จาก First Derivative ที่จุด x_i โดยใช้วิธีที่กล่าวมาในบทก่อน วิธีนี้เราเรียก Euler's Method จะเป็นวิธีแรกที่เราจะศึกษา

วิธีอื่นๆที่เราจะกล่าวต่อไปนั้น จะแตกต่างที่การปรับปรุงการหาค่า Slope ให้ดียิ่งขึ้น ได้แก่ Heun's Method และ Polygon Method ในส่วนสุดท้ายของบท เราจะกล่าวถึงวิธีที่นิยมใช้กันมากที่สุด คือ Runge-Kutta Method

Euler's Method

10.3 Euler's Method (Euler's Cauchy หรือ Point-Slope Method)

ค่า First Derivative ที่ได้จากการ Estimate โดยตรงของ Slope ที่จุด x_i จะหาได้จาก

$$\phi = f(x_i, y_i)$$

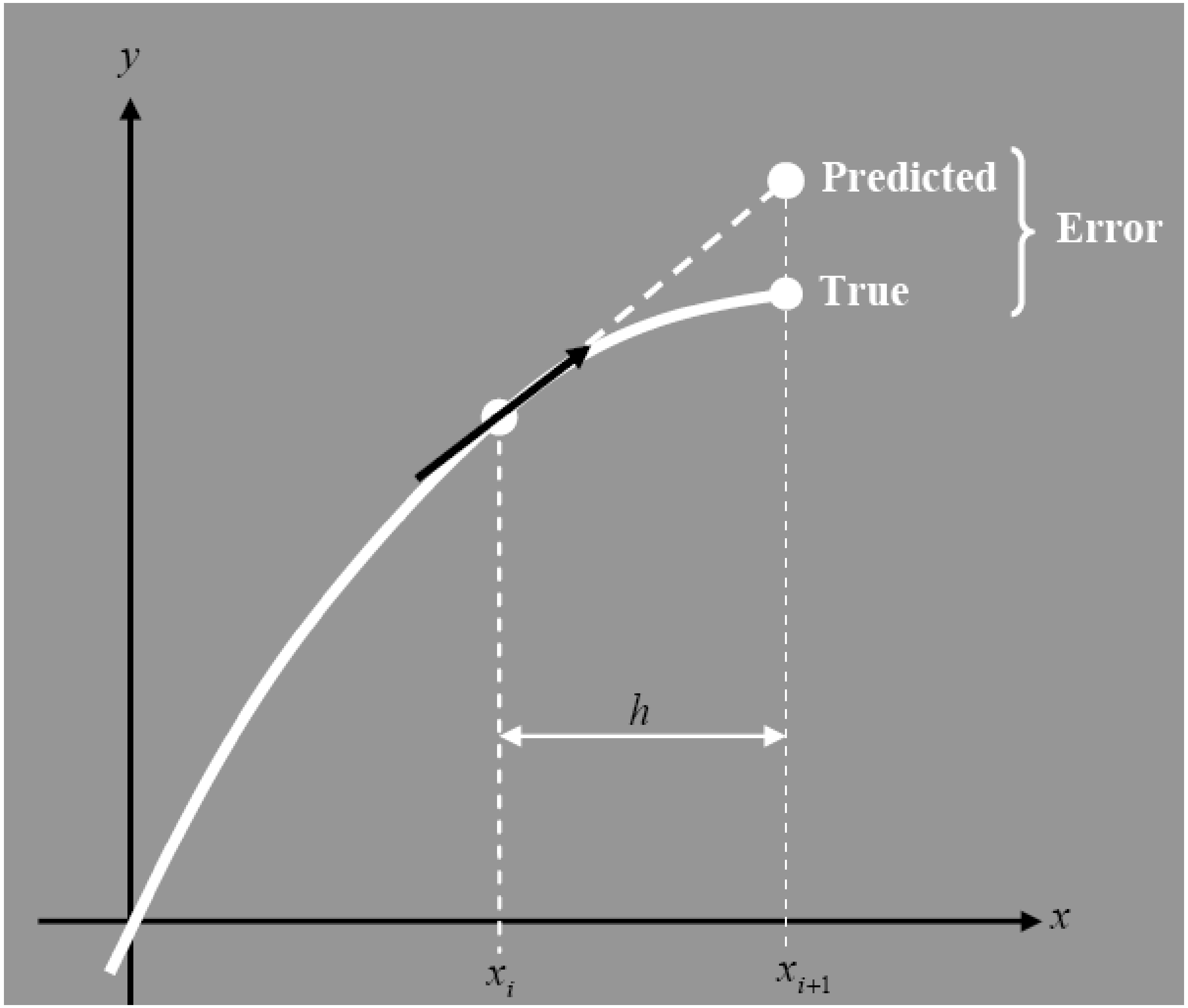
โดยที่ $f(x_i, y_i)$ คือค่าของสมการ Differential Equation ที่จุด x_i และ y_i ดังนั้นสมการของ Solution จะเป็น

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

สมการที่เขียนข้างบนรู้จักกันในนามของ Euler's Method (หรือ Euler-Cauchy Method หรือ Point-Slope Method) โดยที่ค่า y ค่าใหม่หาได้จากการทำนายโดยใช้ค่า Slope จากค่าเก่า(ดูรูป) ทำการ Extrapolate ด้วย Step Size เท่ากับ h

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

สมการที่เขียนข้างบนรู้จักกันในนามของ Euler's Method (หรือ Euler-Cauchy Method หรือ Point-Slope Method) โดยที่ค่า y ค่าใหม่หาได้จากการทำนายโดยใช้ค่า Slope จากค่าเก่า(ดูรูป) ทำการ Extrapolate ด้วย Step Size เท่ากับ h



Euler's Method

Example 10.1 จงใช้ Euler's Method ทำการหาค่า Integrate ของสมการ

$$f(x,y) = -2x^3 + 12x^2 - 20x + 8.5$$

จากจุด $x = 0$ ถึงจุด $x = 4$ โดยใช้ Step Size 0.5 ด้วยค่า Initial Condition $x = 0,\ y = 1$ หรือ $y(0) = 1$ ซึ่งคำตอบที่แท้จริงคือสมการที่แสดงก่อนหน้านี้ คือ $y = -0.5x^4 + 4x^3 - 10x^2 + 8.5x + 1$

Answer:

จาก $y(0) = 1$ เราได้ $y(0.5) = y(0) + f(0,1)0.5 = 5.25$

ซึ่ง Solution ที่แท้จริงของจุดนี้คือ $y(0.5) = -0.5(0.5)^4 + 4(0.5)^3 - 10(0.5)^2 + 8.5(0.5) + 1 = 3.21875$

เราได้ $E_t = 3.21875 - 5.25 = -2.03125,\quad e_t = -63.1\%$

จาก $y(0.5) = 5.25$ เราได้ $y(1.0) = y(0.5) + f(0.5, 5.25)0.5 = 5.875$

True Solution $y(1) = 3.0$ และ $e_t = -95.8\%$

Euler's Method

จากนั้นเราคำนวณต่อที่ $y(1.5), y(2), y(2.5), y(3), y(3.5), y(4)$ ผลลัพธ์ที่ได้แสดงดังตารางข้างล่าง ซึ่งจะสังเกตได้ว่า แม้ว่า Solution ที่ได้จะมีลักษณะการเปลี่ยนแปลงตาม True Solution แต่ Error ที่เกิด ค่อนข้างจะมาก

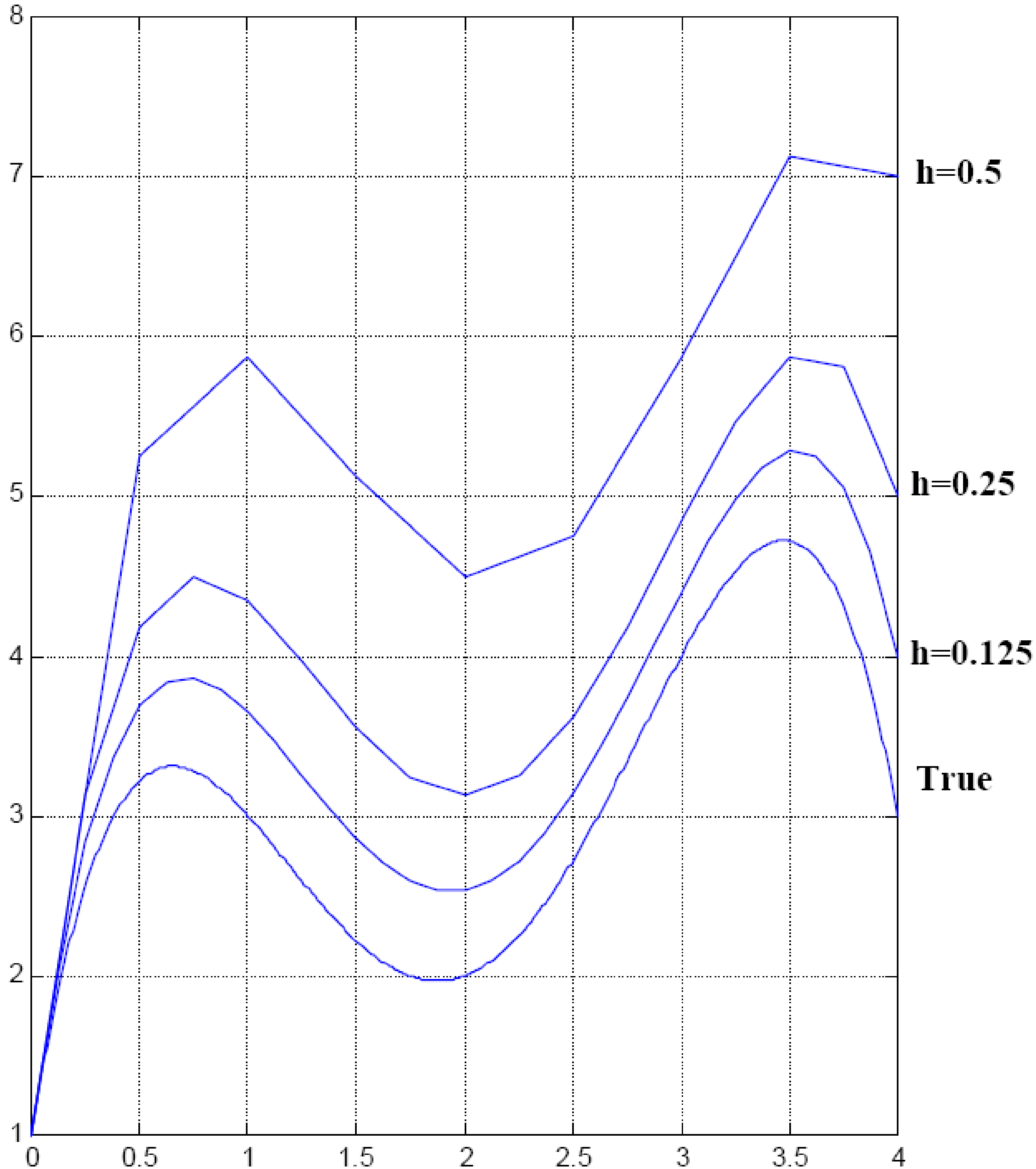
			Relative Error, e_t, %	
x	y_{true}	y_{Euler}	Global	Local(ดูต่อข้างล่าง)
0.0	1.0000	1.0000	0.0000	0.0000
0.5	3.2188	5.2500	-63.1068	-63.1068
1.0	3.0000	5.8750	-95.8333	-28.1250
1.5	2.2188	5.1250	-130.9859	-1.4085
2.0	2.0000	4.5000	-125.0000	20.3125
2.5	2.7188	4.7500	-74.7126	17.2414
3.0	4.0000	5.8750	-46.8750	3.9063
3.5	4.7188	7.1250	-50.9934	-11.2583
4.0	3.0000	7.0000	-133.3333	-53.1250

Euler's Method

ในวิธีของ Euler's Method นั้น เราทำการ Estimate จนถึงเทอม $f(x,y)h = y'h$ ดังนั้นค่าของ Error จะอยู่ใน $O(h^2)$ และเราสามารถแสดงได้ว่าค่า Estimate Local Truncation Error จะอยู่ในรูป

$$E_a = \frac{f'(x,y)}{2}h^2 = O(h^2)$$

ค่า Truncation Error ที่เกิดสามารถลดลงได้โดยการลดขนาดของ Step Size ซึ่งถ้าเราลดขนาด Step Size ลงครึ่งหนึ่ง ค่าของ Local และ Global Error จะเหลือประมาณหนึ่งในสี่ รูปข้างบนแสดง Solution ที่ได้จากการลดของ Step Size ลง อย่างไร ก็ตามแม้ว่าเราจะลด Step ลงถึง $h = 0.001$ ซึ่งเราต้องใช้การคำนวณทั้งหมด 4000 Step จาก $x = 0$ ถึง $x = 4$ ในกรณีนี้ ค่าของ Error ยังคงไม่ได้ถึง 0.1% ดังนั้นวิธีการของ Euler's Method จะใช้การคำนวณที่มากเกินไปถ้าเราต้องการ Error ต่ำๆ และเราจำเป็นต้องหาวิธีอื่นที่ดีกว่า อย่างไรก็ตาม วิธีของ Euler's Method นั้นง่าย และน่าสนใจสำหรับการหาค่าประมาณของ Solution ที่ต้องการ



h=0.5
h=0.25
h=0.125
True

Heun and Polygon Method

10.4 การปรับปรุงวิธีการของ Euler's Method: Heun's Method และ Polygon Method

ส่วนใหญ่ Error ที่เกิดจาก Euler's Method นั้นเป็นผลมาจากการหา Derivative ในส่วนต้นของช่วง และเป็น Derivative ที่จะใช้ตลอดช่วงนั้น วิธีการง่ายๆที่จะแก้ไขความผิดพลาดที่เกิด คือการปรับการหาค่า Derivative ซึ่งในส่วนนี้จะกล่าวสองวิธีง่ายๆที่ใช้การปรับปรุงจากการพิจารณากราฟของ Function ซึ่งทั้งสองวิธีนี้ ความจริงแล้วเป็นวิธีที่อยู่ในเทคนิคการหา Solution ที่เรียก Runge-Kutta Methods อย่างไรก็ตามรายละเอียดของ Runge-Kutta Methods นั้นจะขอกล่าวในส่วนหน้า

Heun Method

10.4.1 Heun's Method

วิธีนี้จะพิจารณาค่า Derivative สองค่า ค่าแรกคือที่จุดเริ่มต้นของช่วง และค่าที่สองคือค่าที่จุดปลายของช่วง และค่า Derivative ทั้งสองจะถูกเฉลี่ย และนำมาใช้สำหรับช่วงนั้น ซึ่งวิธีนี้เราเรียก Heun's Method

จากที่กล่าวใน Euler's Method ค่า Slope ที่จุดเริ่มต้นของ Interval จะเป็น

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

และเรานำมาใช้ในการ Extrapolate หาค่า y_{i+1}

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

ซึ่งสมการนี้เราเรียก ***Predictor Equation*** สังเกตว่าเราใช้ Notation y^0_{i+1} ซึ่งเป็น Solution ใน Euler's Method

ในการ Estimate ค่า Slope ที่ปลายของ Interval เราใช้ค่า y^0_{i+1} ทำการ Estimate และเราได้

$$y'_{i+1} = f(x_{i+1}, y^0_{i+1})$$

Heun Method

เมื่อเรารวมค่า Slope ทั้งสอง และหาค่าเฉลี่ยของ Slope ในช่วงนั้น เราจะได้

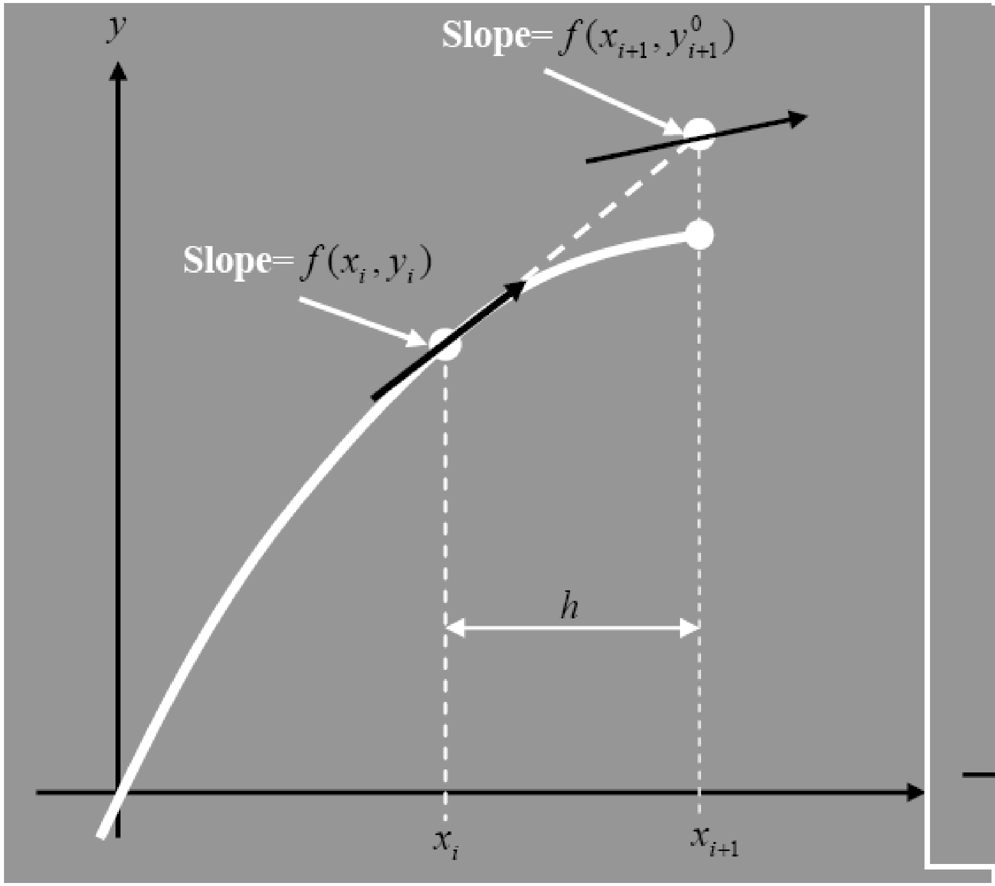
$$\bar{y}' = \frac{y_i' + y_{i+1}'}{2} = \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)}{2}$$

ซึ่งค่าดังกล่าวจะถูกใช้ในการ Extrapolate จาก y_i ไปยัง y_{i+1} โดยใช้สมการเส้นตรง ด้วยวิธีเดิมของ Euler's Method

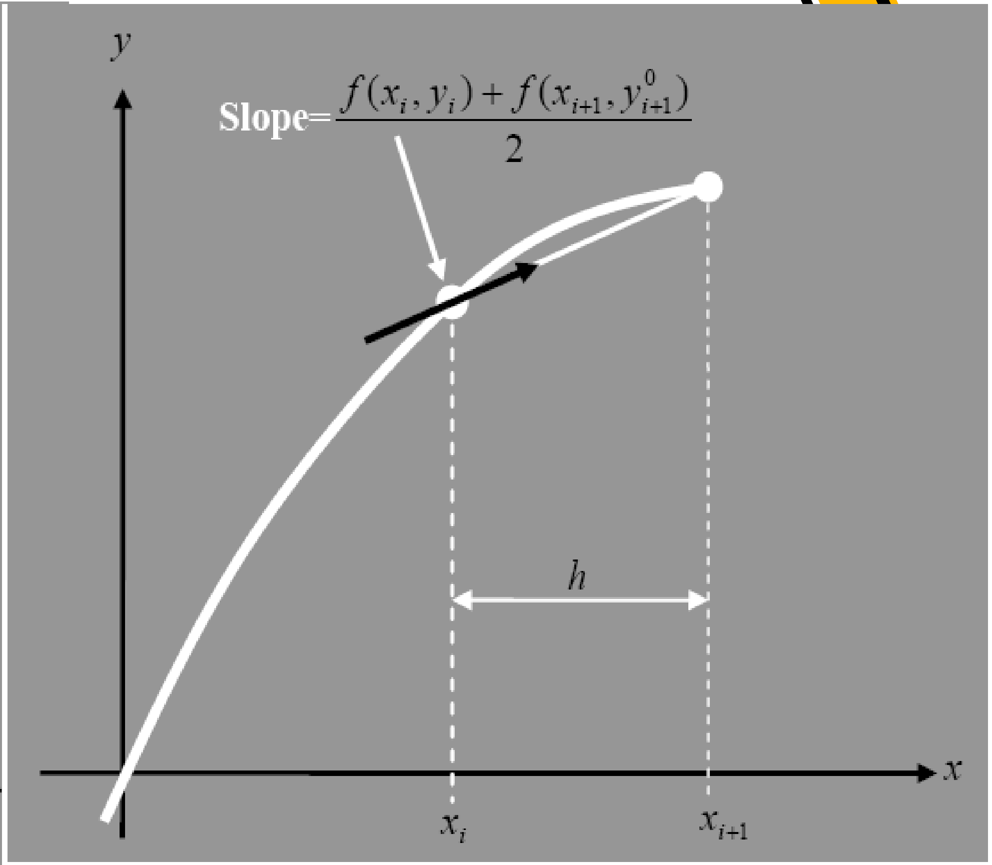
$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)}{2} h$$

และสมการข้างบนเราเรียก ***Corrector Equation***

Heun Method



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



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

Heun and Polygon Method

Heun’s Method จัดว่าเป็นวิธีที่เรียก Predictor-Corrector Approach และยังเป็น One-Step Method ซึ่งใน Class ของ Multistep Method จะใช้หลักการนี้ทั้งนั้น อย่างไรก็ตาม วิธีของ Multistep Method จะไม่กล่าวในชั้นนี้ ดังนั้น Heun’s Method จะเป็นวิธีเดียวของ Predictor-Corrector ที่เราจะกล่าวถึง และสรุปได้ดังนี้

Predictor: $y^0_{i+1} = y_i + f(x_i, y_i)h$

Corrector: $y_{i+1} = y_i + \dfrac{f(x_i, y_i) + f(x_{i+1}, y^0_{i+1})}{2}h$

สมการของ Corrector สามารถทำเป็น Iteration ได้ โดยนำผลที่ได้กลับมาป้อนสมการเดิมเพื่อที่จะได้ Solution ที่ดีขึ้น ดังนี้

Iterative Corrector: $y^j_{i+1} = y_i + \dfrac{f(x_i, y_i) + f(x_{i+1}, y^{j-1}_{i+1})}{2}h$

Heun Method

โดยที่ y_{i+1}^{j} และ y_{i+1}^{j-1} เป็นค่าของ Corrector ที่ Iteration j และ $j-1$ ตามลำดับ และดังนั้นเราสามารถกำหนด Termination Criteria ของแต่ละ Step โดยใช้ Error Estimate ดังนี้

$$\textbf{Estimate Error: } |e_a| = \left| \frac{y_{i+1}^{j} - y_{i+1}^{j-1}}{y_{i+1}^{j}} \right| 100\%$$

ค่าของ Error ที่เกิดใน Heun's Method จะดีกว่า Euler's Method โดยมีค่า Local Error อยู่ที่ $O(h^3)$ และค่า Global Error อยู่ที่ $O(h^2)$ ในขณะที่ Global Truncation Error ของ Euler's Method อยู่ที่ $O(h)$

Example 10.2 จงใช้ Heun's Method ทำการหาค่า Integrate ของสมการ $y' = 4e^{0.8x} - 0.5y$ จาก $x = 0$ ถึง 4 โดยใช้ขนาดของ Step Size = 1 ค่า Initial Condition คือ $f(0) = 2$

Answer:

คำตอบที่แท้จริงของสมการ(ผู้สนใจการแก้สมการสามารถศึกษาได้ในวิชา Calculus(Differential Equation)) คือ

$$y = \frac{4}{1.3}(e^{0.8x} - e^{-0.5x}) + 2e^{-0.5x}$$

การคำนวณใน Step ที่ 1 : Iteration ที่ 1

ค่าของ Predictor ที่ $x = 1$ หาได้จาก $y_1^0 = 2 + [4e^0 - 0.5(2)] \times 1 = 5$ ซึ่งคือค่าที่ได้จาก Euler's Method

ค่า Slope ที่ (x_0, y_0) คำนวณได้เท่ากับ $y'_0 = 4e^0 - 0.5(2) = 3$

โดยที่ ค่า Estimate $y'_1 = f(x_1, y_1^0) = -4e^{0.8(1)} - 0.5(5) = 6.40216371$

และดังนั้นค่าเฉลี่ยของ Slope จะเป็น $\dfrac{3 + 6.40216371}{2} = 4.70108186$

ท้ายสุด สมการ Corrector สามารถคำนวณได้เป็น $y_1^1 = 2 + (4.70108186)(1) = 6.70108186$

เมื่อเทียบกับคำตอบที่แท้จริงคือ 6.19463138 เราได้ $e_t = -8.18\%$

Heun Method

การคำนวณใน Step ที่ 1 : Iteration ที่ 2 และ 3 เราทำดังนี้

เราใช้ค่า y_1^1 ป้อนกลับในสมการเดิม และได้

$$y_1^2 = 2 + \frac{[3 + 4e^{0.8(1)} - 0.5(6.070108186)]}{2} \times 1 = 6.27581139\,;\, |e_t| = 1.31\%$$

$$y_1^3 = 2 + \frac{[3 + 4e^{0.8(1)} - 0.5(6.27581139)]}{2} \times 1 = 6.38212901\,;\, |e_t| = 3.03\%$$

สังเกตว่าใน Iteration ที่ 3 Error จะเพิ่มขึ้น นั่นหมายความว่าขบวนการ Iterative อาจจะไม่ Converge สำหรับ True Error แต่อย่างไรก็ตาม มันจะ Converge สำหรับ Estimate Error ตารางข้างล่างแสดงผลการ Run แต่ละ Step เปรียบเทียบการ Run 1 Iteration และ 15 Iteration สำหรับแต่ละ Step และภาคผนวกแสดง Program MATLAB สำหรับคำนวณ

Heun Method

Result Of 1 and 15 Iterations and Error

		1 Iteration		15 Iterations	
x	y_{true}	y_{heun}	$\lvert e_t \rvert$, %	y_{heun}	$\lvert e_t \rvert$, %
0	2.00000000000000	2.00000000000000	0.00	2.00000000000000	0.00
1	6.19463137720937	6.70108185698494	8.17563546458700	6.36086548685535	2.68351899448886
2	14.8439219076464 9	16.3197819378982 8	9.94252084748262	15.3022366597318 7	3.08755836184566
3	33.6771717679681 7	37.1992488968647 5	10.4583518864448 7	34.7432760906779 2	3.16565871402471
4	75.3389626091585 7	83.3377673354007 7	10.6170890190487 2	77.7350961939616 1	3.18047063805967

Improved Polygon Method

10.4.2 The Improved Polygon Method (Modified Euler Method)

วิธีการง่ายๆอีกวิธีหนึ่งที่สามารถนำมาปรับปรุงวิธีการของ Euler's Method เดิม คือการใช้วิธีของ Euler's Method แต่เราทำการ Predict ค่าของ y ที่จุดกึ่งกลางของ Interval แทนที่จะเป็นจุดเริ่มต้น ดังแสดงในรูป ซึ่งวิธีการนี้ เราเรียก Improved Polygon หรือ Modified Euler Method

จากรูป เราใช้ Euler's Method ทำการ Predict ค่าของ y ที่จุดกึ่งกลางของ Interval ได้เป็น

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

ซึ่งค่าที่ Predict ได้จะถูกใช้ในการ Estimate ค่า Slope ที่จุดกึ่งกลางดังนี้

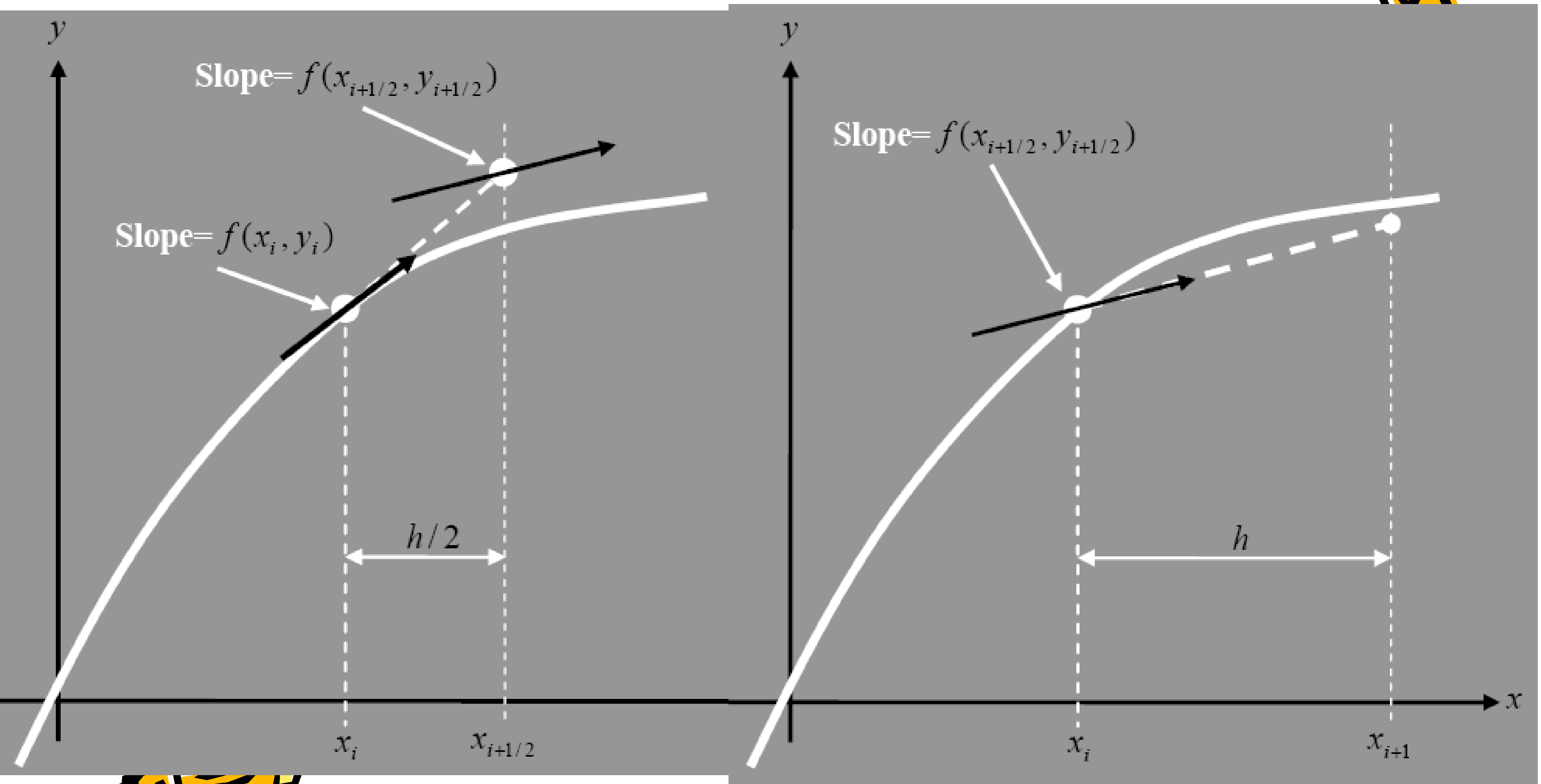
$$y'_{i+1/2} = f(x_{i+1/2}, y_{i+1/2})$$

จากนั้นค่า Slope ที่ได้จะถูกนำมาใช้ในการ Extrapolate แบบ Linear จาก x_i ไปยัง x_{i+1} ด้วย Euler's Method

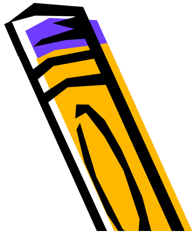
$$y_{i+1} = y_i + f(x_{i+1/2}, y_{i+1/2})h$$

ค่า Local Error และ Global Error ของ Improved Polygon Method พบว่าอยู่ใน $O(h^3)$ และ $O(h^2)$ ตามลำดับ

Improved Polygon Method



Runge-Kutta Method



กรรมวิธีของ Runge-Kutta(RK Method) เป็นวิธีที่ให้ความถูกต้องโดยไม่ต้องมีการคำนวณค่า Derivative ในระดับที่สูงๆ ซึ่งวิธีนี้สามารถเขียนเป็นรูปแบบทั่วไปได้ดังนี้

$$y_{i+1} = y_i + \phi(x_i, y_i, h)h$$

โดยที่ $\phi(x_i, y_i, h)$ เรียก ***Increment Function*** ซึ่งเป็นค่าที่แสดงถึงค่า Slope ของ Function โดย Increment Function นี้สามารถเขียนได้ในรูปของ

$$\phi = a_1k_1 + a_2k_2 + \cdots + a_nk$$

ค่า a_i เป็นค่าคงที่ และค่า k_i กำหนดดังนี้

$$\begin{aligned}
k_1 &= f(x_i, y_i) \\
k_2 &= f(x_i + p_1h,\ y_i + q_{11}k_1h) \\
k_3 &= f(x_i + p_2h,\ y_i + q_{21}k_1h + q_{22}k_2h) \\
&\ \ \vdots \\
k_n &= f(x_i + p_{n-1}h,\ y_i + q_{n-1,1}k_1h + q_{n-1,2}k_2h + \cdots + q_{n-1,n-1}k_{n-1}h)
\end{aligned}$$

สังเกตว่าค่า k_i มีความสัมพันธ์กันแบบ Recurrence Relation ซึ่งเป็นผลให้วิธีการของ RK มีประสิทธิภาพเมื่อนำมาเขียนเป็นโปรแกรม

$$\phi = a_1 k_1 + a_2 k_2 + \cdots + a_n k$$

ค่า a_i เป็นค่าคงที่ และค่า k_i กำหนดดังนี้

$$\begin{aligned}
k_1 &= f(x_i, y_i) \\
k_2 &= f(x_i + p_1 h,\ y_i + q_{11} k_1 h) \\
k_3 &= f(x_i + p_2 h,\ y_i + q_{21} k_1 h + q_{22} k_2 h) \\
&\ \vdots \\
k_n &= f(x_i + p_{n-1} h,\ y_i + q_{n-1,1} k_1 h + q_{n-1,2} k_2 h + \cdots + q_{n-1,n-1} k_{n-1} h)
\end{aligned}$$

ด้วยการเปลี่ยนวิธีการหาเทอมต่างๆตามสมการข้างบน จากค่า n เราสามารถเปลี่ยนแปลงกรรมวิธีของ Runge-Kutta ได้หลายแบบ สังเกตว่าในกรณีของ First-Order Runge-Kutta ด้วยค่า $n = 1$ ที่จริงแล้วก็คือ Euler's Method ในการนำ Runge-Kutta Method ไปใช้ เมื่อเราเลือกค่าของ n (หรือ Order ในกรณีที่มี Order ต่ำๆ) ค่าของ a_i, p_j และ $q_{k,l}$ สามารถหาได้โดยตั้ง $y_{i+1} = y_i + \phi(x_i, y_i, h)h$ เท่ากับเทอมใน Taylor Series Expansion

เราจะเริ่มจากการศึกษา Second-Order Runge-Kutta Method ก่อน ซึ่งในกรณีนี้เราจะได้คำตอบที่แท้จริงถ้าคำตอบของสมการอยู่ในรูปสมการ Quadratic และในวิธีนี้จะมีค่า Local Truncation Error อยู่ที่ $O(h^3)$ และ Global Truncation Error อยู่ที่ $O(h^2)$ ต่อจากนั้น เราจะศึกษา Third-Order และ Forth-Order ซึ่งมีค่า Global Truncation Error อยู่ที่ $O(h^3)$ และ $O(h^4)$ ตามลำดับ

Second Order Runge-Kutta Method

สมการของ Second-Order RK จะอยู่ในรูป

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h$$

โดยที่

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

$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h)$$

ค่า Constant ต่างๆ สามารถหาได้จากการตั้งสมการให้เท่ากับ Taylor Series Expansion

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h = y_i + f(x_i, y_i)h + f'(x_i, y_i)\frac{h^2}{2}$$

โดยที่ค่า $f'(x_i, y_i)$ สามารถหาได้จากกฎ Chain-Rule ของการหาค่า Derivative

Second Order Runge-Kutta Method

$$f'(x_i, y_i) = \frac{df}{dx} + \frac{df}{dy}\frac{dy}{dx}$$

และเราได้

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h = y_i + f(x_i, y_i)h + \left(\frac{df}{dx} + \frac{df}{dy}\frac{dy}{dx}\right)\frac{h^2}{2}$$

จากนั้น เมื่อเราใช้ Taylor Series Expansion สำหรับ Two-Variable Function ซึ่งมีรูปแบบดังนี้

$$g(x+r, y+s) = g(x, y) + r\frac{dg}{dx} + s\frac{dg}{dy} + \cdots$$

กับสมการของ k_2 เราจะได้

$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h) = f(x_i, y_i) + p_1 h\frac{df}{dx} + q_{11} k_1 h\frac{df}{dy} + O(h^2)$$

Second Order Runge-Kutta Method

และเมื่อนำไปแทนค่าในสมการ Second-Order RK และจัดเรียงเทอมที่เหมือนกัน เราได้

$$y_{i+1} = y_i + [a_1 f(x_i, y_i) + a_2 f(x_i, y_i)]h + \left[a_2 p_1 \frac{df}{dx} + a_2 q_{11} f(x_i, y_i) \frac{df}{dy} \right] h^2 + O(h^3)$$

เมื่อเปรียบเทียบกับสมการของ Taylor Series Expansion ของ y_{i+1} และทำการ Equate Term เราสรุปได้ว่า

$$a_1 + a_2 = 1$$

$$a_2 p_1 = 1/2$$

$$a_2 q_{11} = 1/2$$

Second Order Runge-Kutta Method

เมื่อเปรียบเทียบกับสมการของ Taylor Series Expansion ของ y_{i+1} และทำการ Equate Term เราสรุปได้ว่า

$$a_1 + a_2 = 1$$

$$a_2 p_1 = 1/2$$

$$a_2 q_{11} = 1/2$$

ชุดของสมการข้างบนประกอบด้วยสามสมการ แต่มี 4 Unknown และจะไม่มีคำตอบอันเดียว การหาคำตอบสามารถทำได้โดยเลือกค่า Constant อันหนึ่งก่อน จากนั้นใช้สามสมการข้างบนหาค่า Constant ที่เหลือ ดังนั้น Second-Order RK Method จะมีหลาย Variation และที่สำคัญมีดังนี้

Second Order Runge-Kutta Method

1. Heun's Method with a Single Corrector: ถ้าเราสมมุติค่า $a_2 = 1/2$ เราแก้สมการได้ $a_1 = 1/2, p_1 = q_{11} = 1$ เมื่อนำไปแทนค่าในสมการของ RK เราได้

$$y_{i+1} = y_i + \left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)h$$

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

$$k_2 = f(x_i + h, y_i + hk_1)$$

ซึ่ง k_1 ที่จริงแล้วคือค่า Slope ที่ส่วนต้นของ Interval และ k_2 คือค่า Slope ที่ส่วนท้ายของ Interval และสมการดังกล่าวก็คือ Heun's Method ที่มี 1 Iteration

Second Order Runge-Kutta Method

2. The Improved Polygon Method: ถ้าเราสมมุติค่า $a_2 = 1$ เราแก้สมการได้ $a_1 = 0, p_1 = q_{11} = 1/2$ เมื่อนำไปแทนค่าในสมการของ RK เราได้

$$y_{i+1} = y_i + k_2 h$$
$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}hk_1)$$

และสมการดังกล่าวก็คือ Improved Polygon Method

Second Order Runge-Kutta Method

3. Ralston's Method: ถ้าเราสมมุติค่า $a_2 = 2/3$ เราแก้สมการได้ $a_1 = 1/3, p_1 = q_{11} = 3/4$ เมื่อนำไปแทนค่าในสมการของ RK เราได้

$$y_{i+1} = y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)h$$

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

$$k_2 = f(x_i + \tfrac{3}{4}h, y_i + \tfrac{3}{4}hk_1)$$

สมการนี้รู้จักกันในนาม Ralston's Method ซึ่งจะให้ค่า Truncation Error ของ Second-Order RK ที่เป็น Minimum Bound

Second Order Runge-Kutta Method

Example 10.3 จงใช้ Second-Order RK Methods ทั้งสามวิธี เปรียบเทียบการหาค่า Integrate ของสมการ $y' = f(x,y) = -2x^3 + 12x^2 - 20x + 8.5$ จาก $x = 0$ ถึง 4 โดยใช้ขนาดของ Step Size = 0.5 ค่า Initial Condition คือ $f(0) = 1$

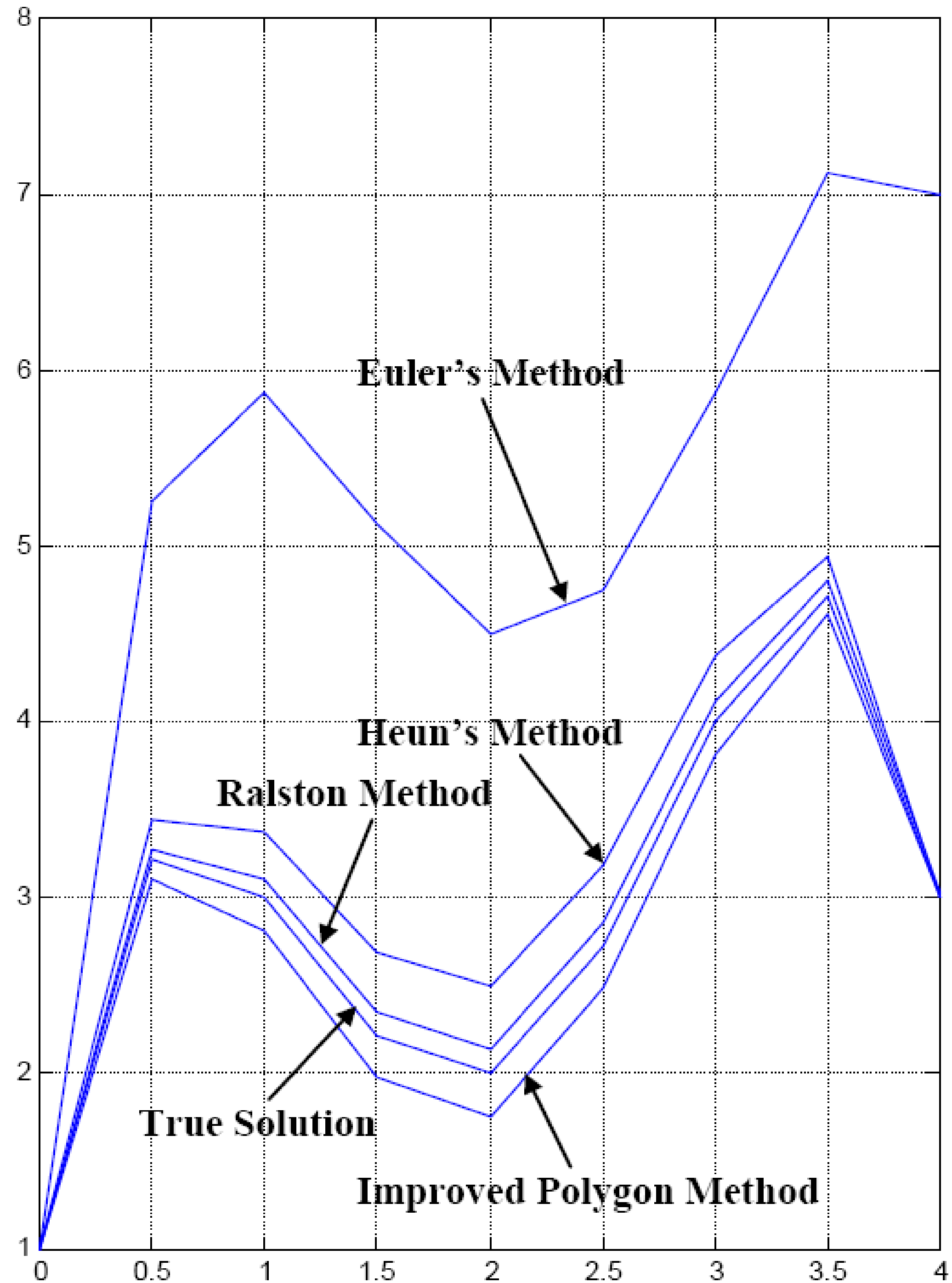
Answer:

คำตอบที่แท้จริงคือ(จากตัวอย่างที่ 1) $y = -0.5x^4 + 4x^3 - 10x^2 + 8.5x + 1$

ค่าคำตอบของทั้งสามวิธี และค่า Error สรุปได้ในตารางข้างล่าง สำหรับ Source Code ของ MATLAB ดูได้จากภาคผนวก

Second Order Runge-Kutta Method

x	y_{true}	Single Corrector Heun		Improved Polygon		Second-Order Ralston	
		y	$\lvert e_t \rvert$,%	y	$\lvert e_t \rvert$,%	y	$\lvert e_t \rvert$,%
0.0	1.00000000000000	1.00000000000000	0	1.00000000000000	0	1.00000000000000	0
0.5	3.21875000000000	3.21875000000000	6.8	3.10937500000000	3.4	3.27734375000000	1.8
1.0	3.00000000000000	3.00000000000000	12.5	2.81250000000000	6.3	3.10156250000000	3.4
1.5	2.21875000000000	2.21875000000000	21.1	1.98437500000000	10.6	2.34765625000000	5.8
2.0	2.00000000000000	2.00000000000000	25.0	1.75000000000000	12.5	2.14062500000000	7.0
2.5	2.71875000000000	2.71875000000000	17.2	2.48437500000000	8.6	2.85546875000000	5.0
3.0	4.00000000000000	4.00000000000000	9.4	3.81250000000000	4.7	4.11718750000000	2.9
3.5	4.71875000000000	4.71875000000000	4.6	4.60937500000000	2.3	4.80078125000000	1.7
4.0	3.00000000000000	3.00000000000000	0	3.00000000000000	0	3.03125000000000	1.0



Euler's Method
Heun's Method
Ralston Method
True Solution
Improved Polygon Method

Third Order Runge-Kutta Method

ในกรณีที่ $n = 3$ เราจะได้ Third-Order Runge-Kutta Methods และผลลัพธ์จะประกอบไปด้วยการหา 8 Unknown จาก 6 สมการ ดังนั้นเราจะต้องกำหนดค่าของสอง Unknown และหาค่าค่า Constant ที่เหลืออีก 6 ตัว และจะมีคำตอบได้มากมาย คำตอบหนึ่งที่นิยมใช้กันได้แก่ชุดสมการต่อไปนี้

$$y_{i+1} = y_i + \left[\tfrac{1}{6}(k_1 + 4k_2 + k_3)\right]h$$

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

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

$$k_3 = f(x_i + h, y_i - hk_1 + 2hk_2)$$

สังเกตว่า ถ้า Function ของสมการประกอบด้วยค่า x อย่างเดียว ชุดสมการข้างบนก็คือ Simson 1/3 Rule ค่า Local Truncation Error และ Global Truncation Error ของ Third-Order Runge-Kutta Method อยู่ใน $O(h^4)$ และ $O(h^3)$ ตามลำดับ ในกรณีที่สมการเป็น Cubic Polynomial เราได้ Solution เป็น Quartic ดังนั้นคำตอบของ Third-Order จะให้ค่าที่แท้จริง

Third Order Runge-Kutta Method

Example 10.4 จงใช้ Third-Order RK Method หา Solution ของสมการ

$$\frac{dy}{dx} = 4e^{0.8x} - 0.5y$$

จากค่า $x = 0$ จนถึง $x = 1$ โดยใช้ Step Size = 1 กำหนดให้ $y(0) = 2$

Answer: คำตอบได้จากการ Run MATLAB Program ท้ายบทนี้

```
» [x,y,e]=o3rk(0,1,1,2)
x =
     0
     1
y =
   2.00000000000000   2.00000000000000
   6.19463137720937   6.17567668094419
e =
                  0
   0.30598586277341
```

Forth Order Runge-Kutta Method

10.5.3 Forth-Order Runge-Kutta Methods

เช่นเดียวกับที่กล่าวในหัวข้อก่อน Forth-Order Runge-Kutta Method จะมี Variation ที่ไม่มีที่สิ้นสุด อย่างไรก็ตาม Forth-Order RK Method เป็นวิธีที่นิยมมากที่สุด ในที่นี้จะกล่าวกรรมวิธีที่เรียก Classical Forth-Order Runge-Kutta Method ซึ่งเป็นชุดของสมการต่อไปนี้

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

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

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

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

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

Forth Order Runge-Kutta Method

Example 10.5 จงใช้ Forth-Order RK Method หา Solution ของสมการ

$$y' = f(x, y) = -2x^3 + 12x^2 - 20x + 8.5$$

จากค่า $x = 0$ จนถึง $x = 1$ โดยใช้ Step Size = 0.5 กำหนดให้ $y(0) = 1$

Answer: คำตอบได้จากการ Run MATLAB Program ท้ายบทนี้

```
x =
                   0
   0.50000000000000
   1.00000000000000
y =
   1.00000000000000   1.00000000000000
   3.21875000000000   3.21875000000000
   3.00000000000000   3.00000000000000
e =
     0
     0
     0
```

สังเกตว่าในกรณีนี้เราจะได้คำตอบที่แท้จริง เนื่องจาก Solution เป็น Quartic และกรรมวิธีของ Forth-Order RK จะให้คำตอบที่แท้จริงของ Forth-Order Polynomial

Higher Order Runge-Kutta Method

10.5.4 Higher-Order Runge-Kutta Methods and System of ODEs

ถ้าต้องการคำตอบที่ถูกต้องกว่านี้ เราสามารถใช้ Order ที่สูงกว่าของ Runge-Kutta Method ยกตัวอย่างเช่นในกรณีของ Fifth-Order Method ที่รู้จักกันในนามของ Butcher's Method ประกอบด้วยชุดของสมการดังนี้

$$y_{i+1} = y_i + \left[\tfrac{1}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6)\right]h$$

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

$$k_2 = f(x_i + \tfrac{1}{4}h, y_i + \tfrac{1}{4}hk_1)$$

$$k_3 = f(x_i + \tfrac{1}{4}h, y_i + \tfrac{1}{8}hk_1 + \tfrac{1}{8}hk_2)$$

$$k_4 = f(x_i + \tfrac{1}{2}h, y_i - \tfrac{1}{2}hk_2 + hk_3)$$

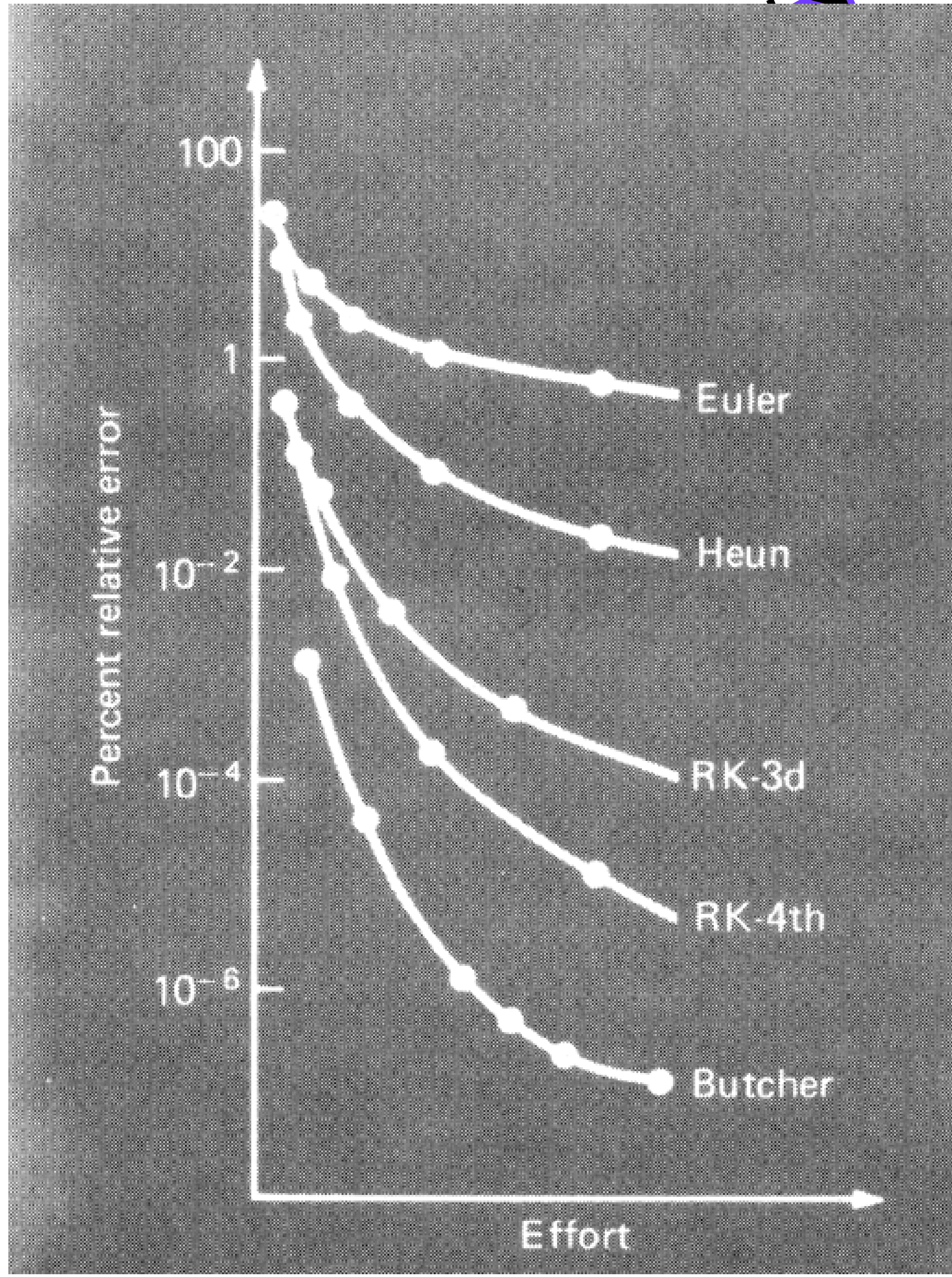
$$k_5 = f(x_i + \tfrac{3}{4}h, y_i + \tfrac{3}{16}hk_1 + \tfrac{9}{16}hk_4)$$

$$k_6 = f(x_i + h, y_i - \tfrac{3}{7}hk_1 + \tfrac{2}{7}hk_2 + \tfrac{12}{7}hk_3 - \tfrac{12}{7}hk_4 + \tfrac{8}{7}hk_5)$$

รายละเอียดจะไม่กล่าวถึง นักศึกษาที่สนใจสามารถเขียนโปรแกรม MATLAB สำหรับสมการข้างบนและทดลองนำไปใช้แก้สมการ ODE เพื่อเปรียบเทียบกับวิธีการที่กล่าวมาข้างต้น

Comparison

$$dy/dx = 4e^{0.8x} - 0.5y$$



Chapter 12 Homework (HW 11)

- **Download HW 11(ODE)และทำใน Sheet**
  - **Option ถ้าใครส่งครบ 10 ครั้งและได้เต็ม ไม่ต้องส่ง**
  - **จะนับ 10 HW ที่คะแนนมากที่สุด**
  - **ส่งพุธที่ 4 พ.ค. ที่ห้องภาค 5-310 ก่อนเที่ยง**
- **ไม่มีบทที่ 13 Curve Fitting**
- **Course Ends**

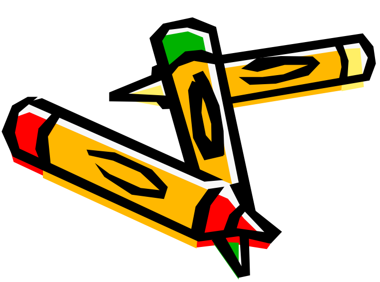
Final Exam Preparation

- สูตรจะให้มา
- ข้อสอบมี 7 ข้อ เลือกทำ 5 ข้อ 10x5 = 50 คะแนน เทียบเป็นคะแนนเก็บ 50 %
- 2 ข้อ เป็นเรื่องก่อน Midterm (Part 1 หนึ่งข้อ และ Part 2 หนึ่งข้อ)
- 5 ข้อเป็นเรื่องใหม่ ดังนี้

- **5 ข้อ เป็นเรื่องใหม่ หลัง MT ดังนี้**
  - 1. Taylor Series และการประมาณค่าของ Function รวมถึง Error
  - 2. Root of Function (Bisection หรือ Newton)
  - 3. Linear Equation 1 ข้อ
    - Gauss Elimination, Gauss Jordan, Gauss Seidel และ LU Decompositon
  - 4. Numerical Integration (Trapezoidal หรือ Simpson)
  - 5. ODE โดยใช้ 4th Order RK

Formulas

[illegible] MT+[illegible]



Numerical Methods:

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

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

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

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

(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:

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

Simpson's 3/8 Rule:

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

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:

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

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

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

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

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

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

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

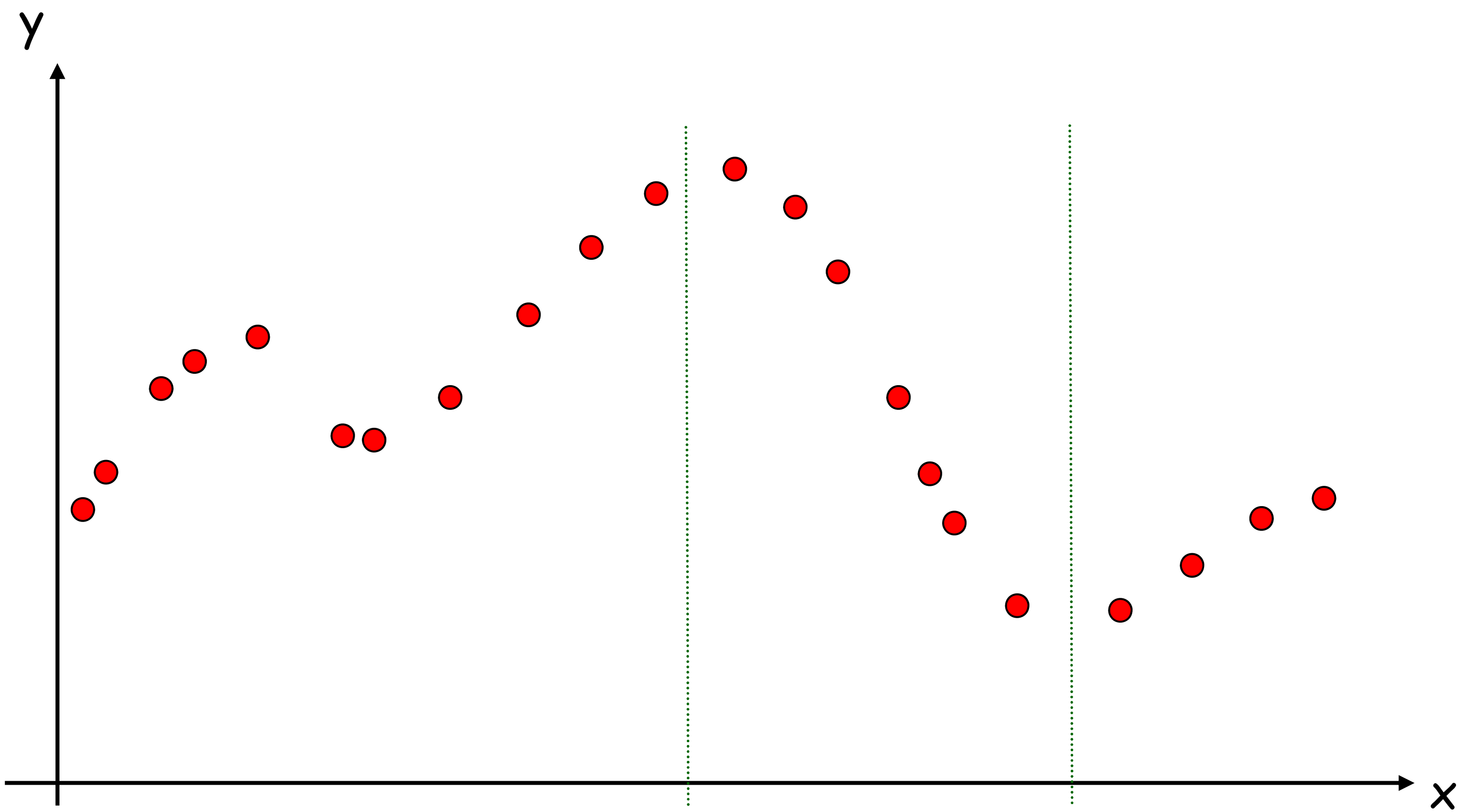
CPE 332
Computer Engineering Mathematics II

Chapter 12
Curve Fitting
(Last)

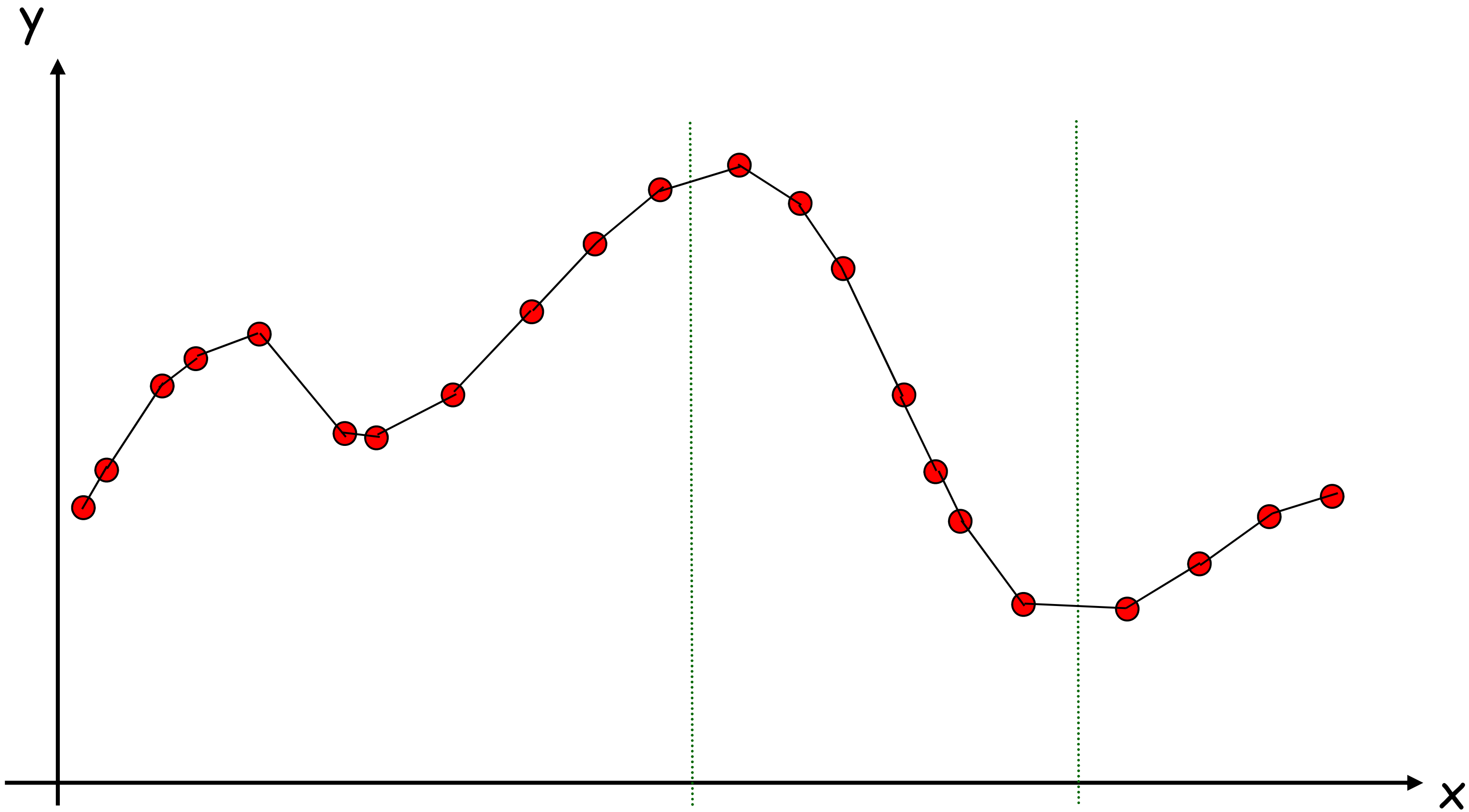
Curve Fitting

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

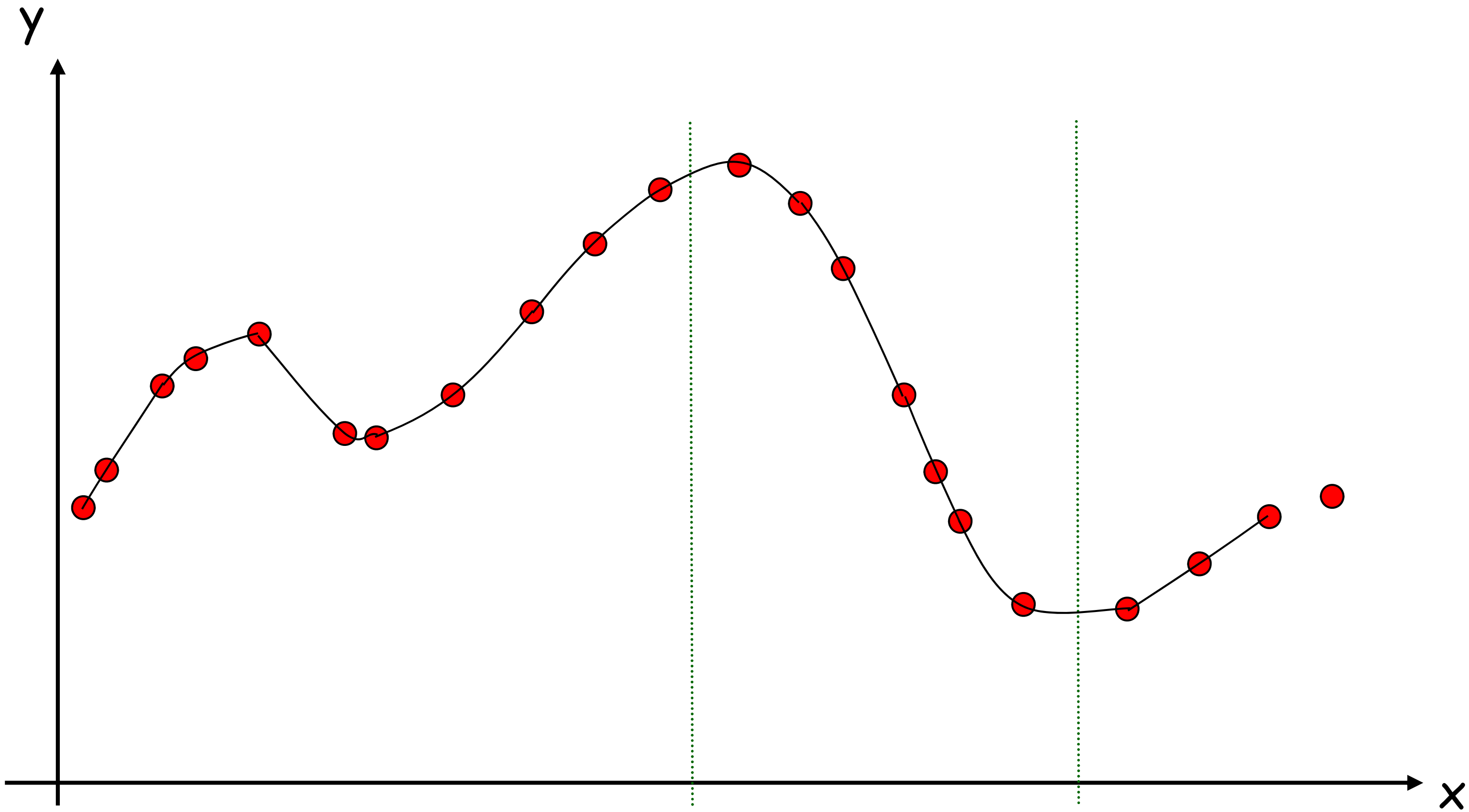
Interpolation



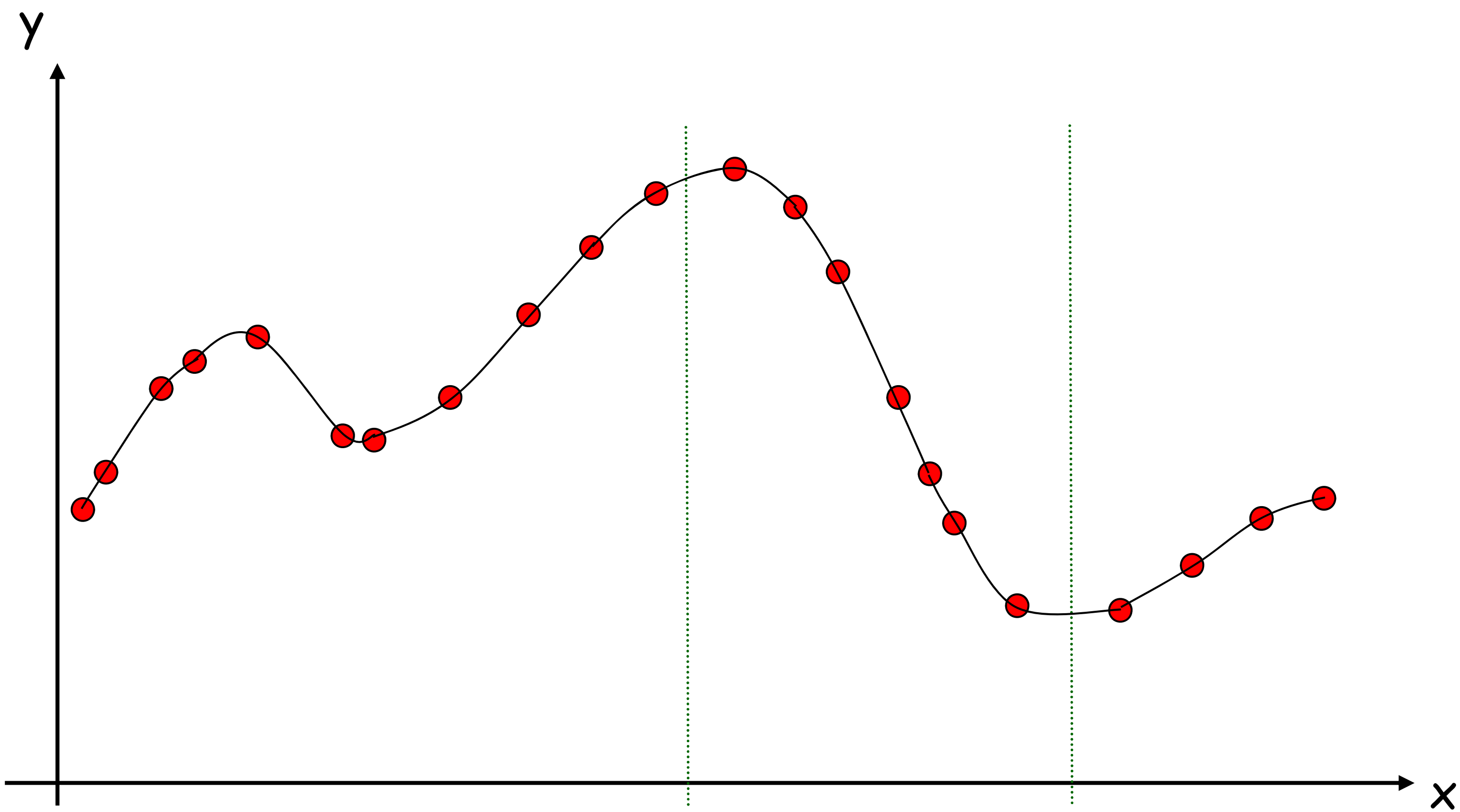
Linear Interpolation



Second Degree Piecewise Interpolation



Third Degree Piecewise Interpolation



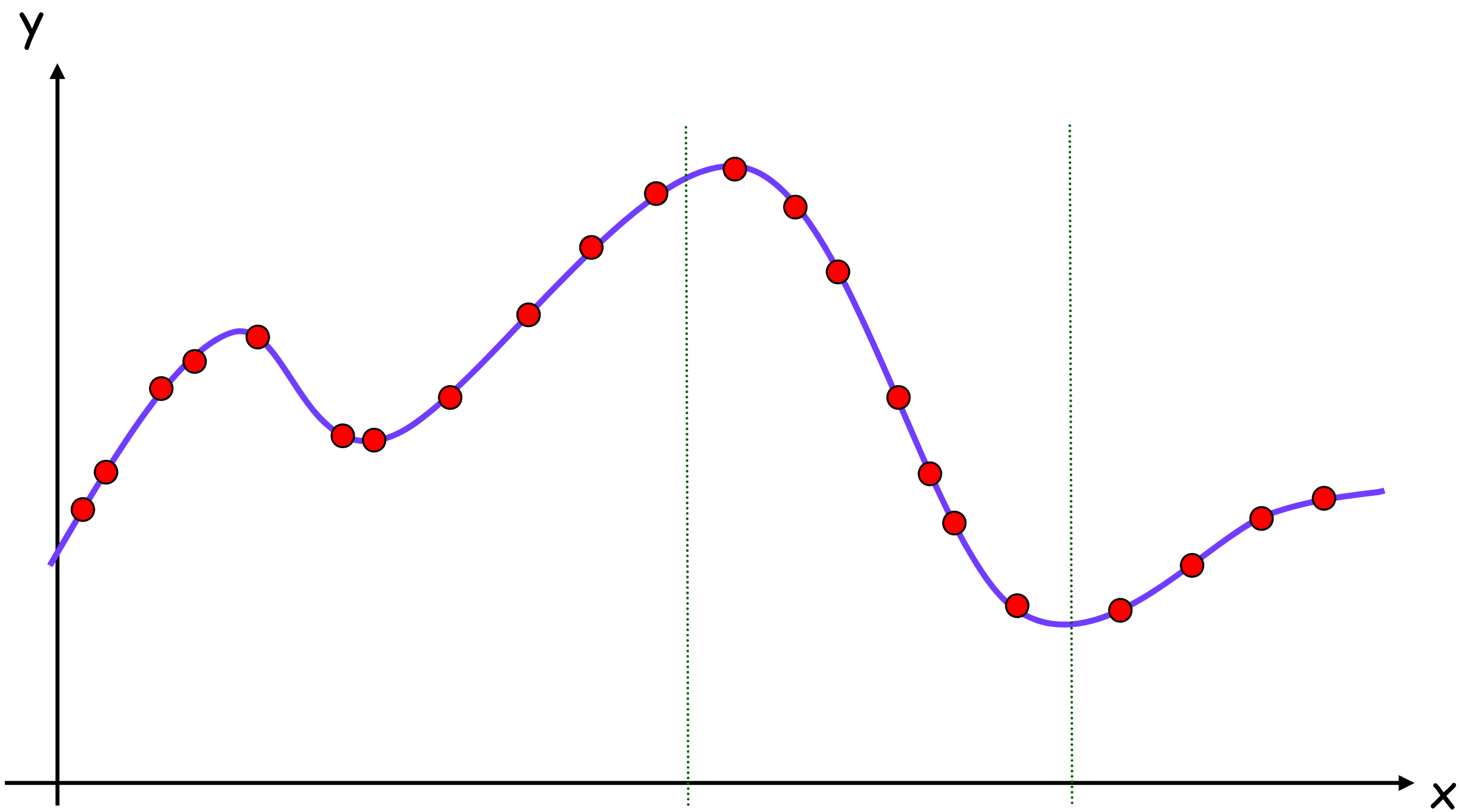
Higher Order Degree

- 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

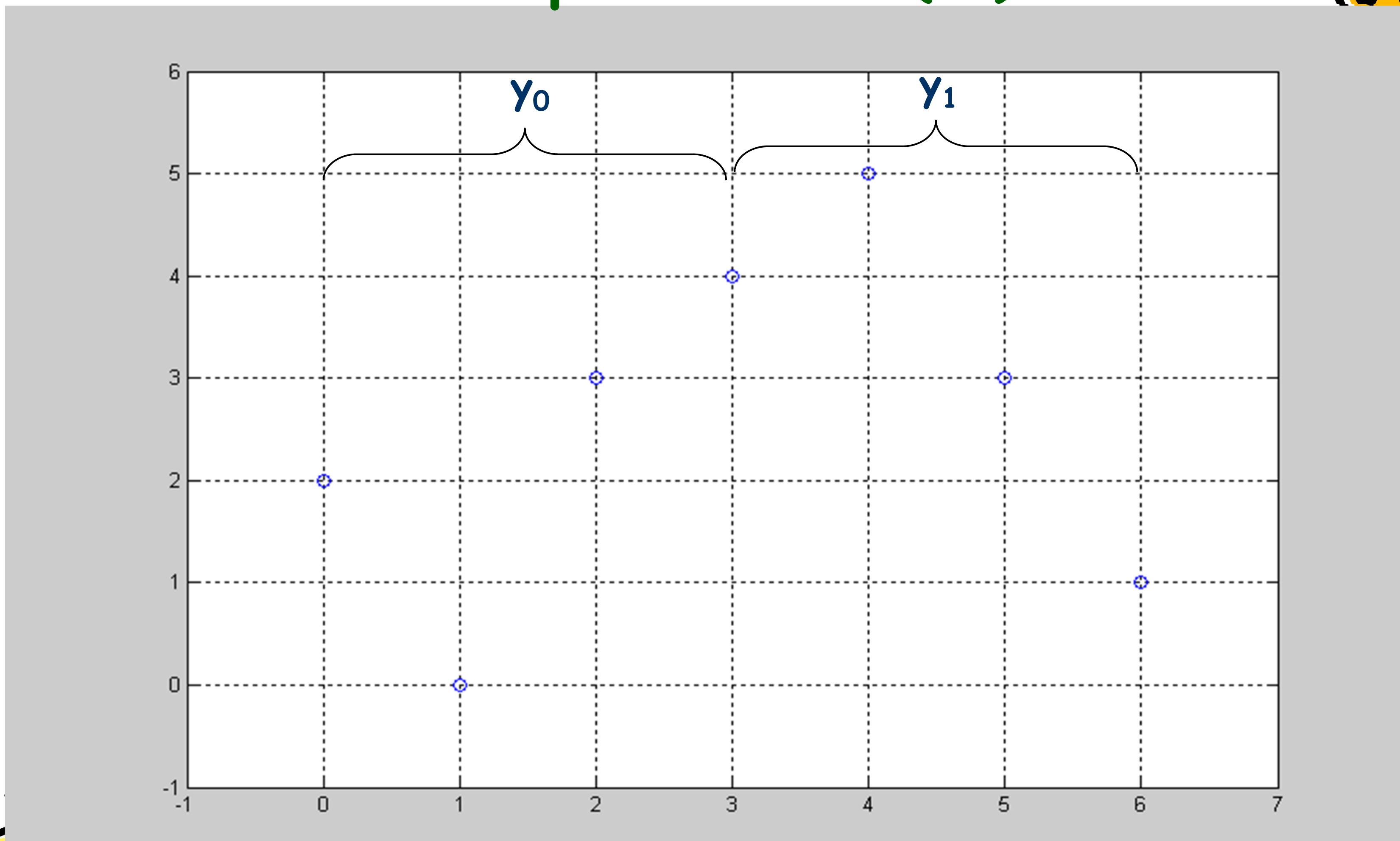
$$y = f_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

  - 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

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

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

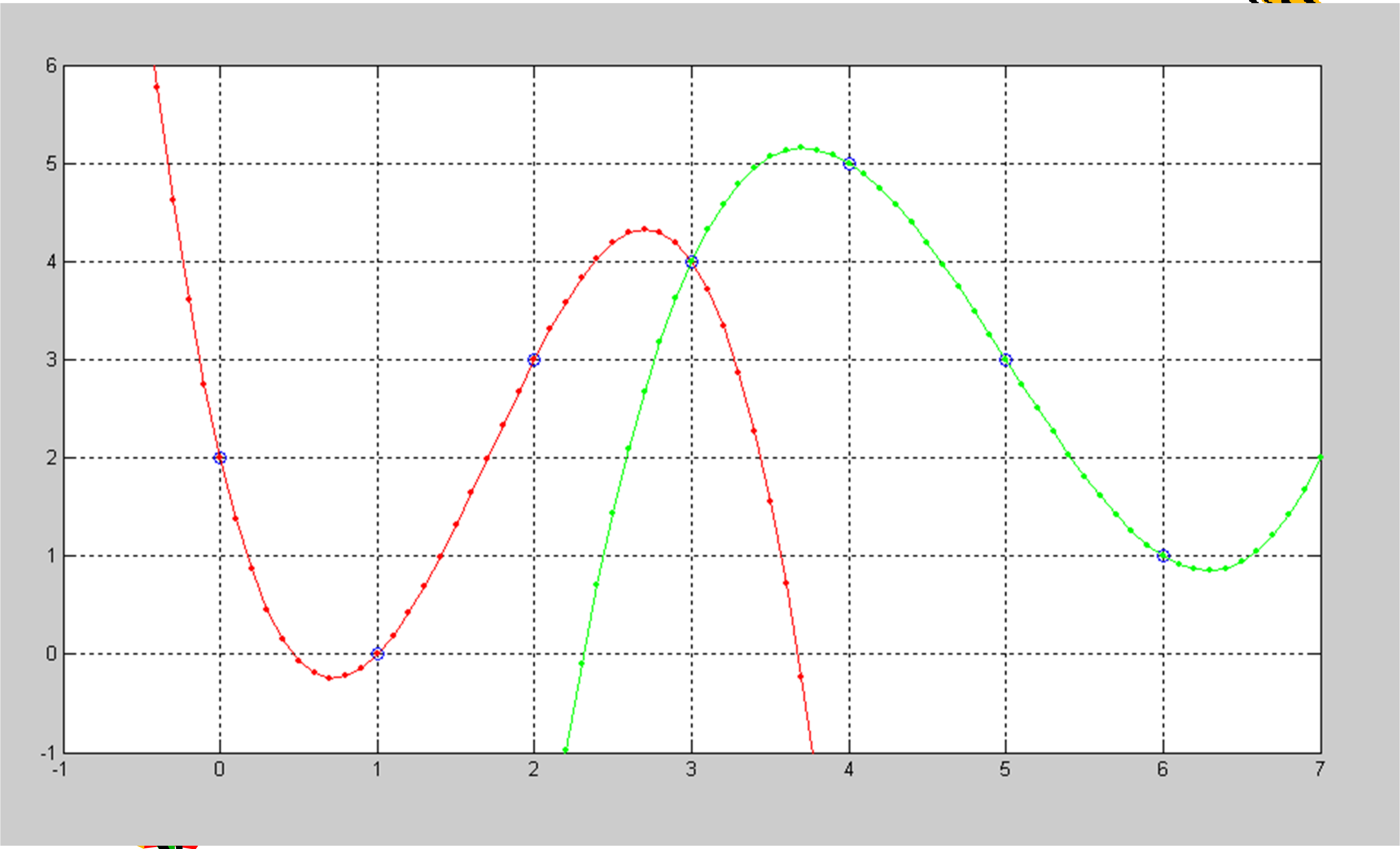
$$3 = a_0 + 5a_1 + 25a_2 + 125a_3$$

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

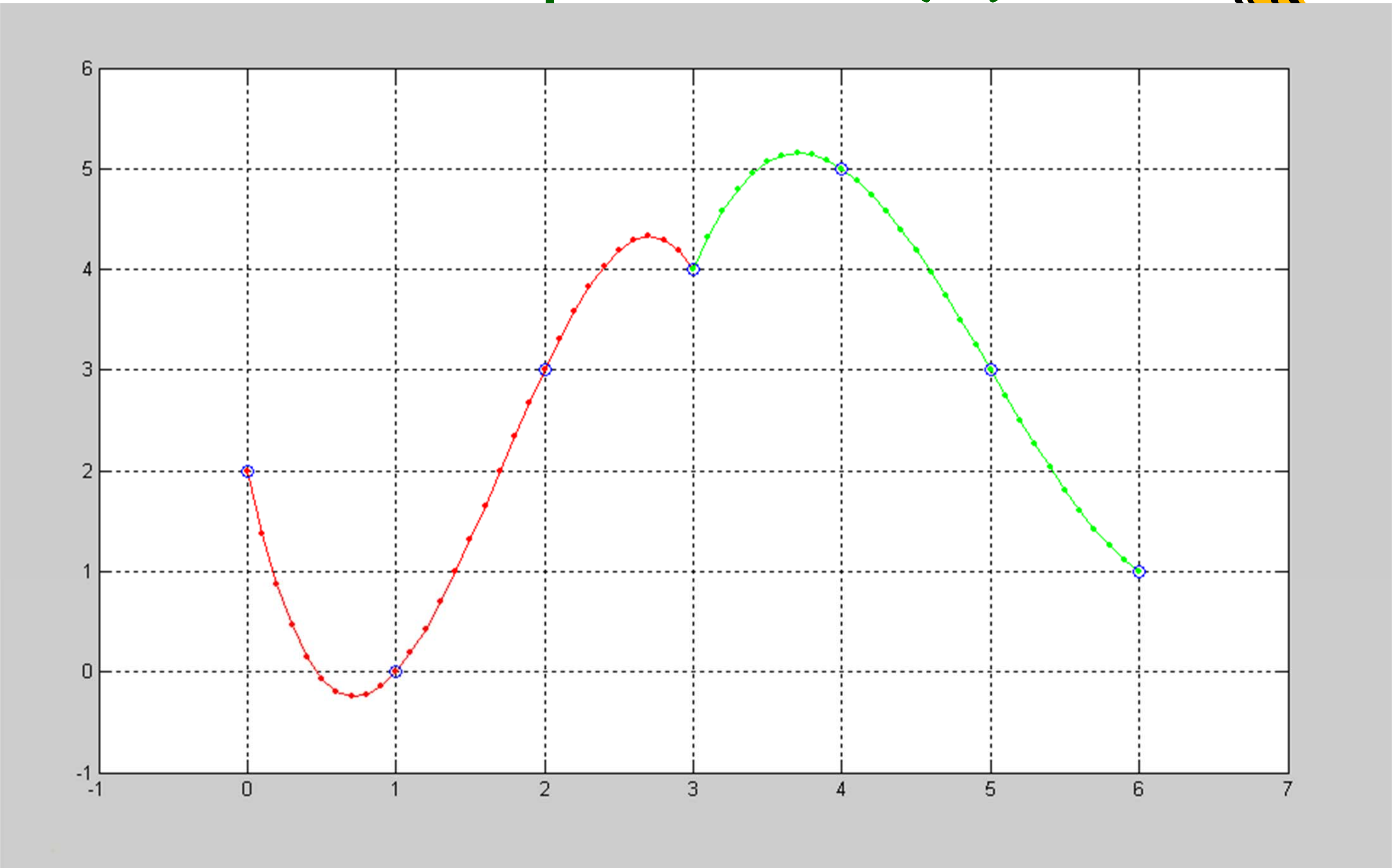
- แก้สมการได้

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

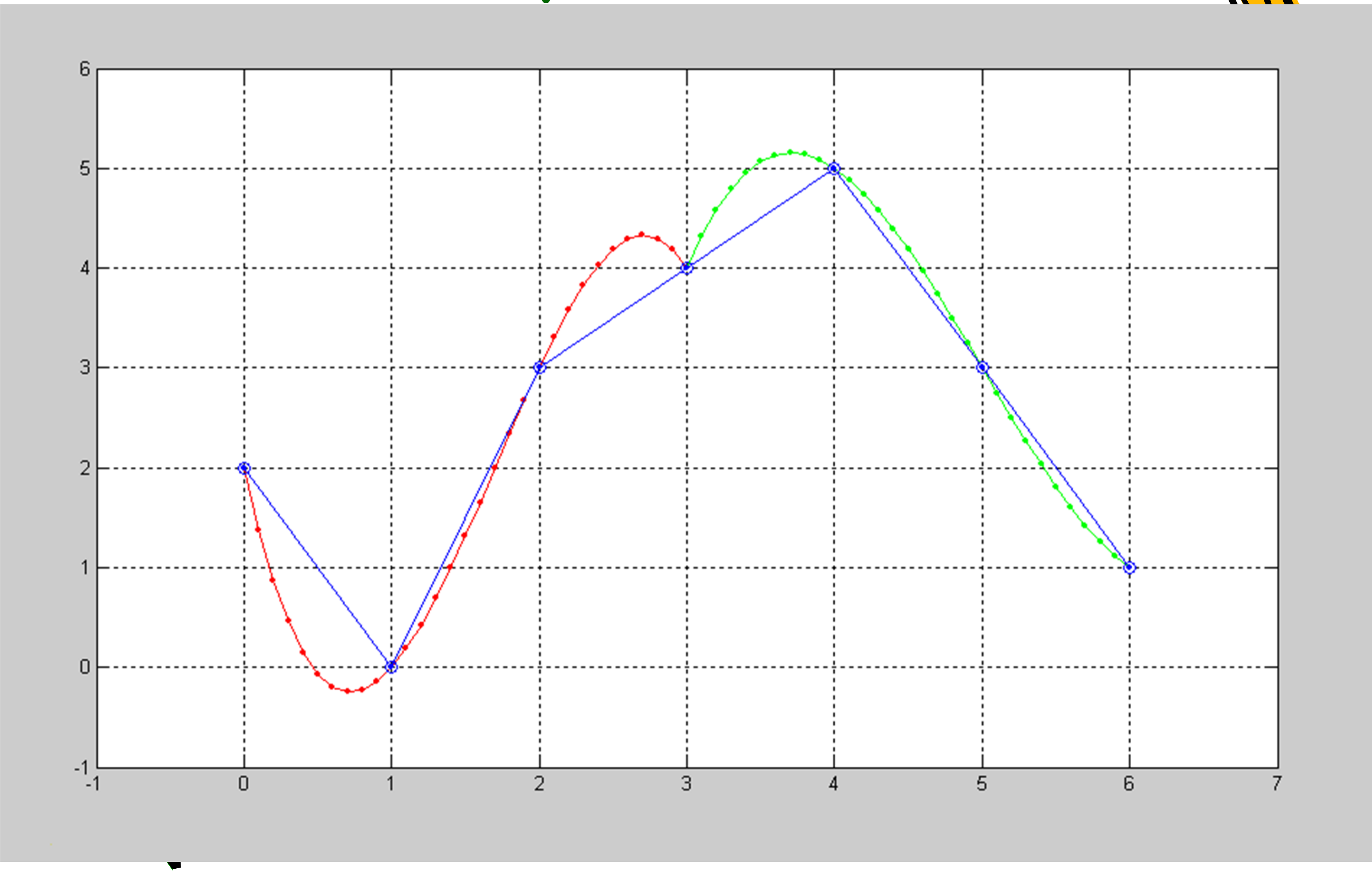
Cubic Spline Int.(6)



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_ii=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

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

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

- นอกจากนี้แล้ว ค่า Second Derivative จะต้องต่อเนื่องกันตลอดช่วง

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

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

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

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

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

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

- 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 นี้และ Levinson Algorithm อยู่นอกขอบเขตของวิชานี้ ผู้สนใจสามารถศึกษาเพิ่มเติมได้จากตำราด้าน 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

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

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

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

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

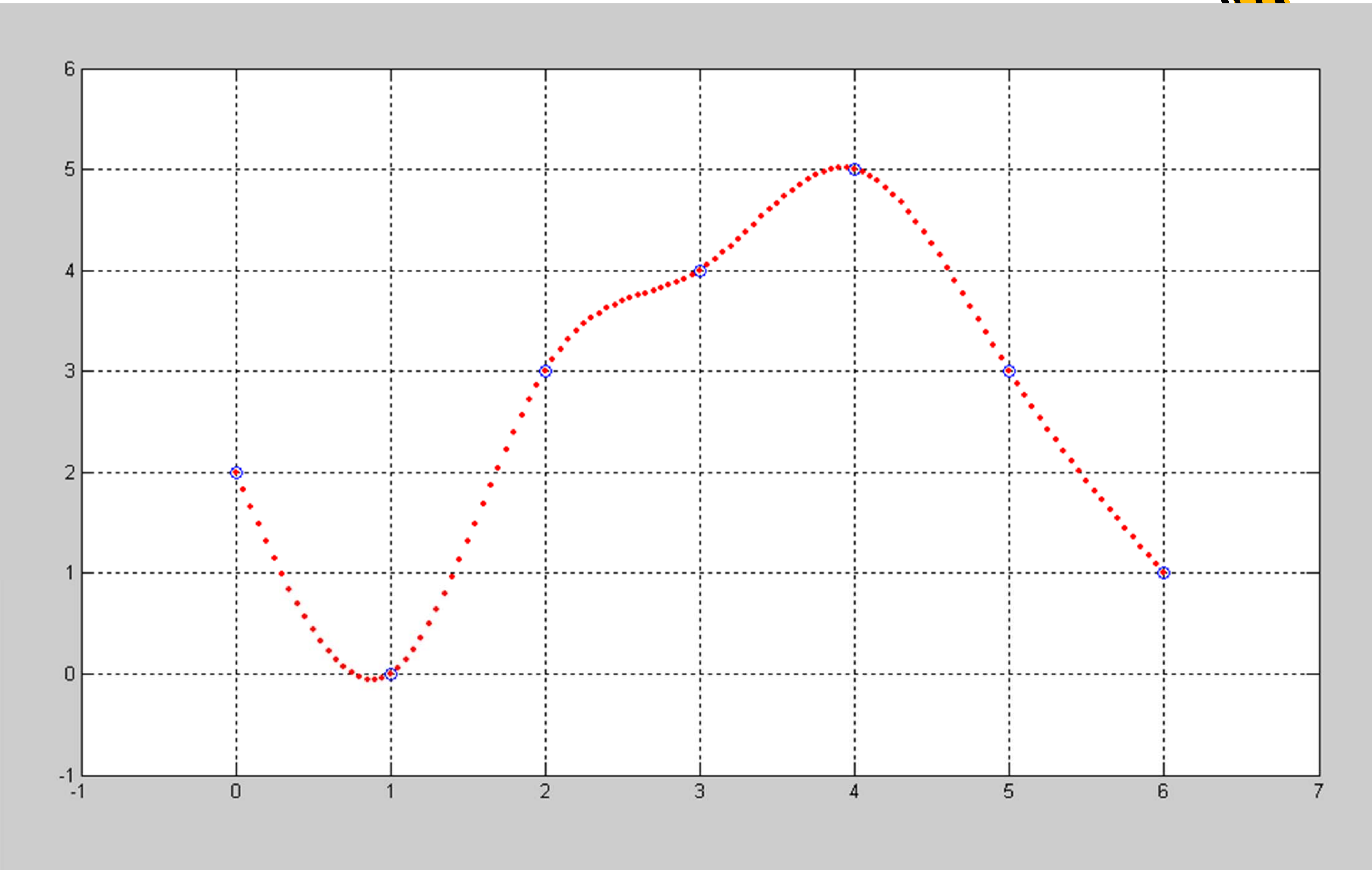
$$d_i = y_i = [2, 0, 3, 4, 5, 3]$$

Polynomial Coefficient Matrix

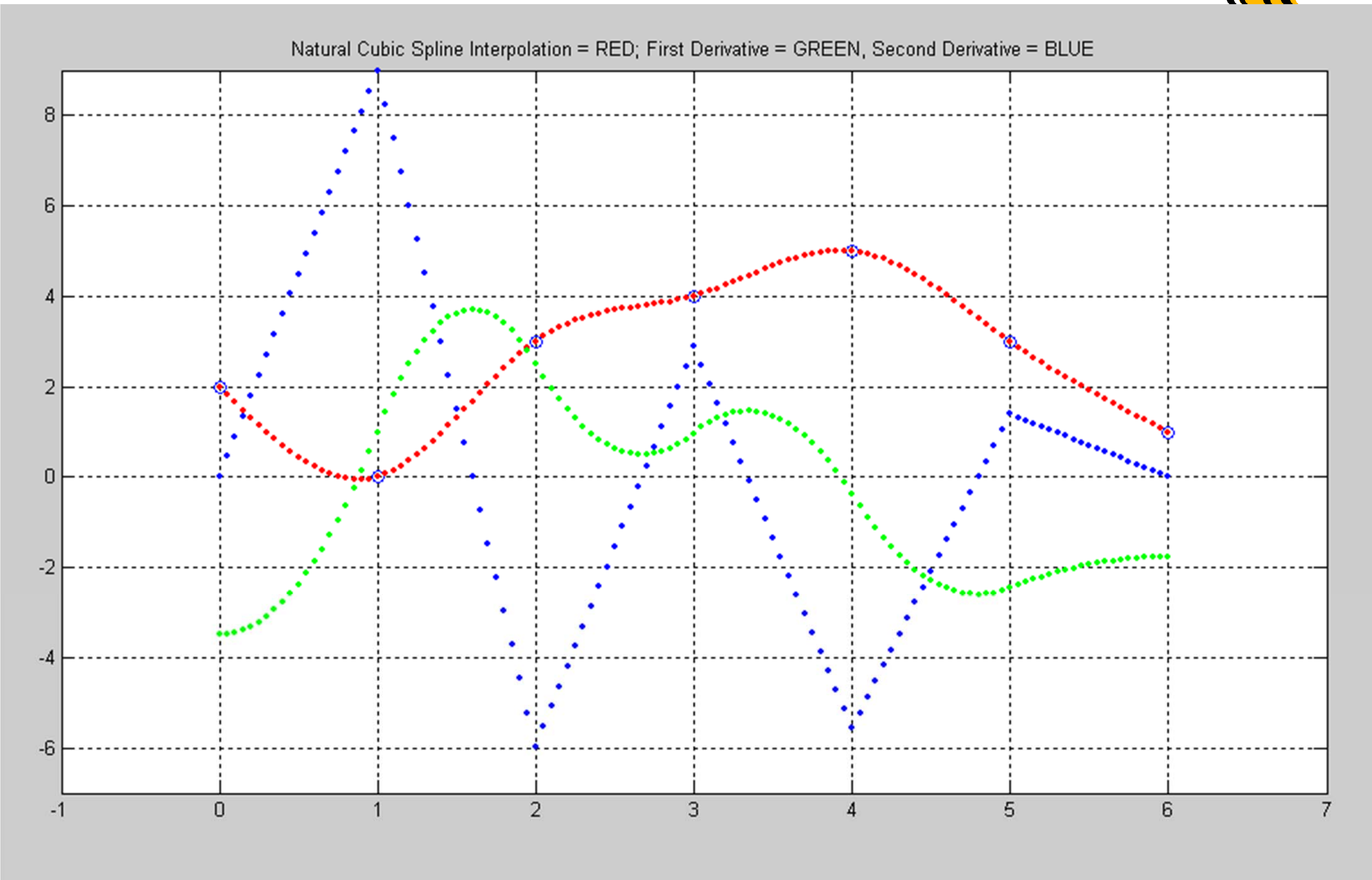
- `>> p=[a b c d]`
- `p =`
- `1.4987        0   -3.4987   2.0000`
- `-2.4936   4.4962    0.9974        0`
- `1.4756   -2.9846    2.5090   3.0000`
- `-1.4090   1.4423    0.9667   4.0000`
- `1.1603   -2.7846   -0.3756   5.0000`
- `-0.2321   0.6962   -2.4641   3.0000`

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

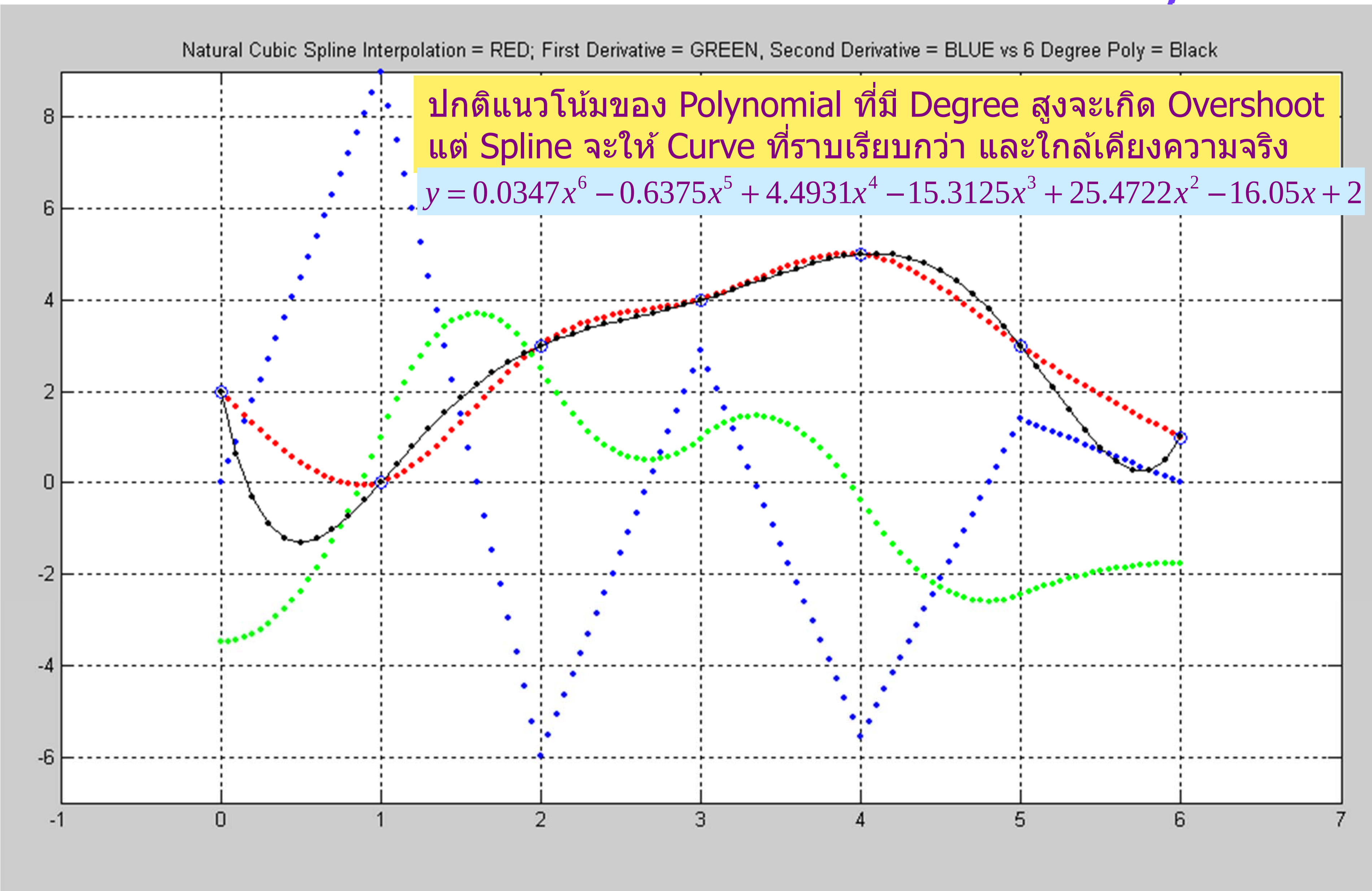
Natural Spline Interpolation



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



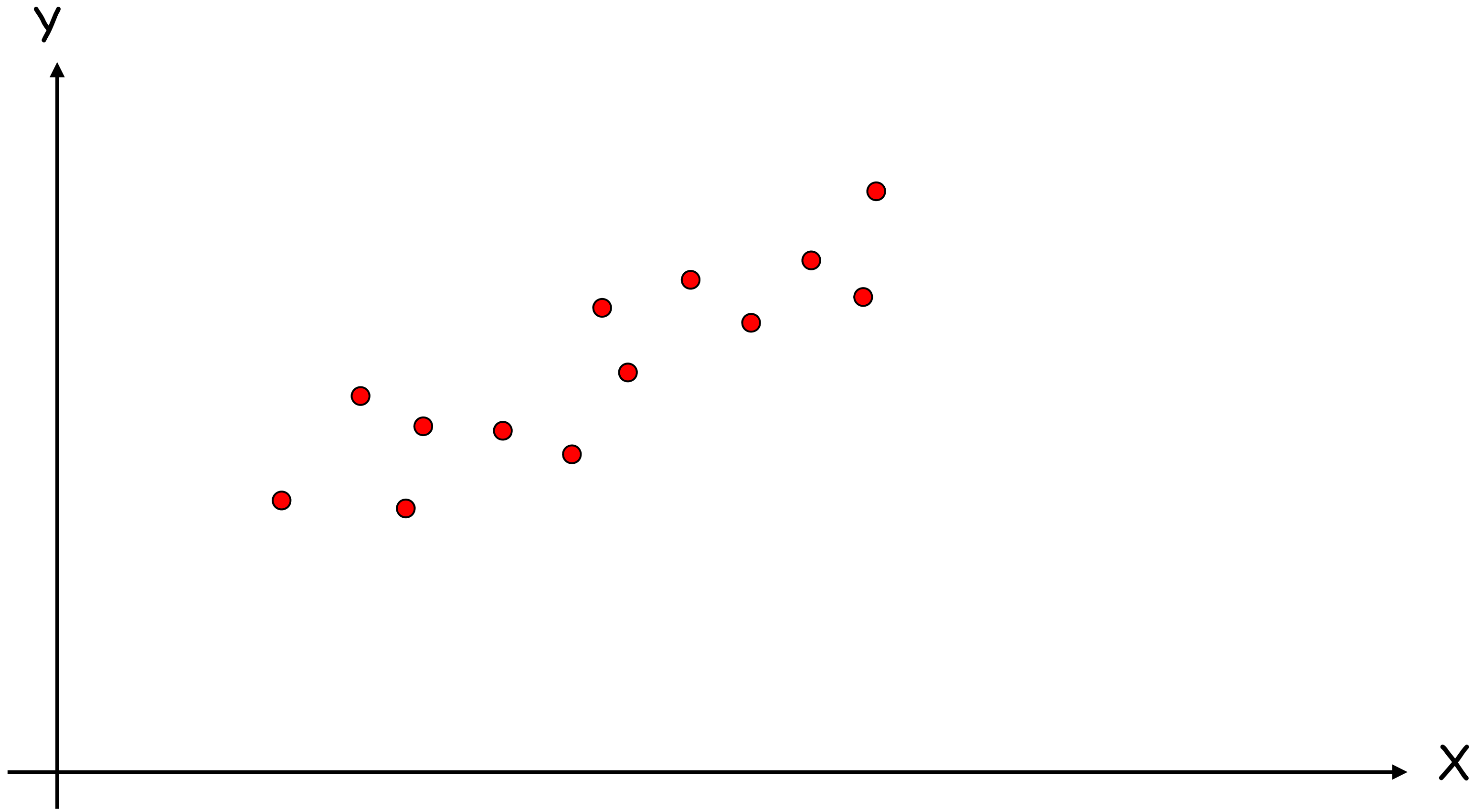
Natural Spline Interpolation = R;
First D = G; Second D = B vs 6th D Poly = K



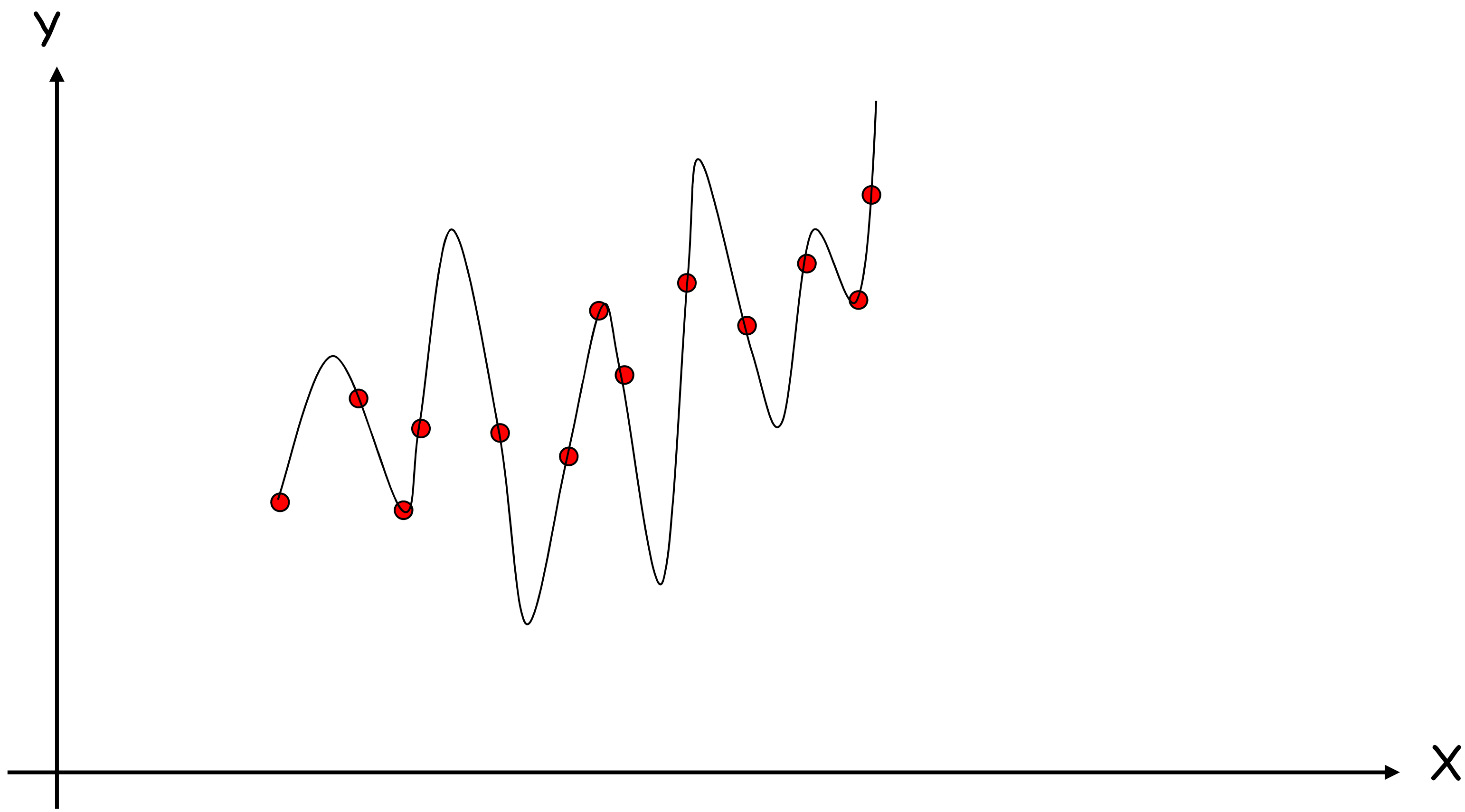
Regression

- 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



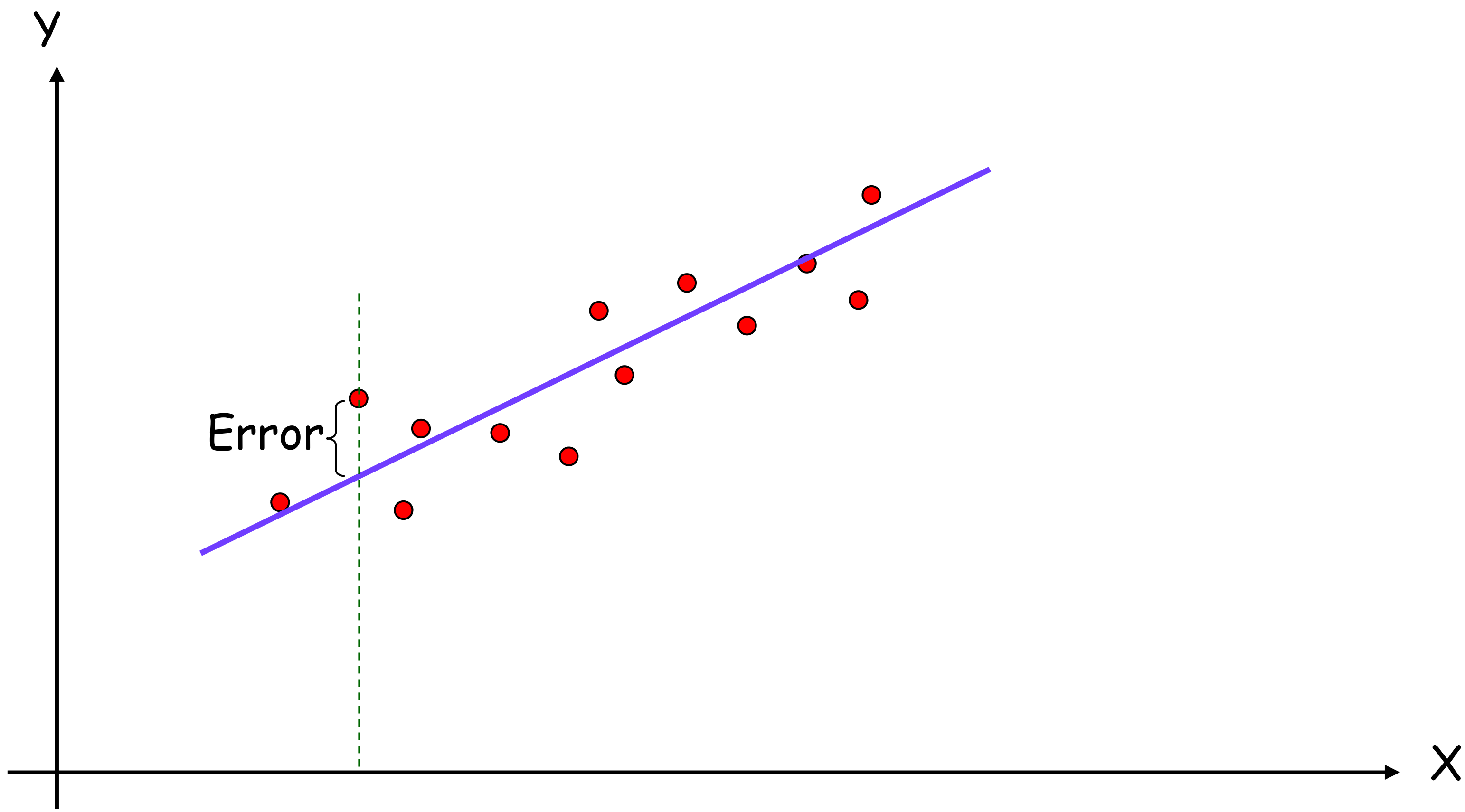
Ordinary Simple Linear Least-Squares Regression

- 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

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

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

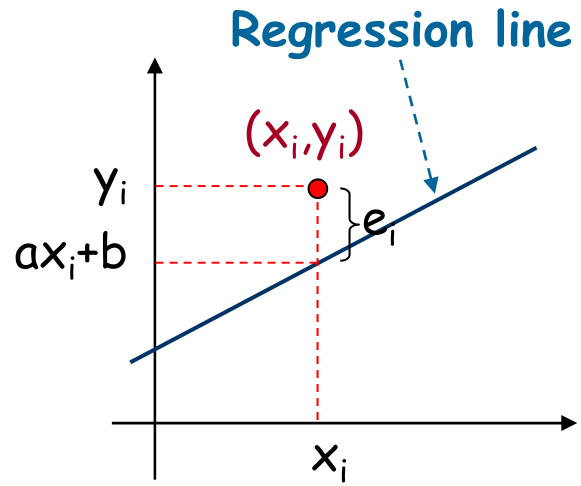
$$\frac{df}{da} = -2\sum_{i=1}^{n} [(y_i - ax_i - b)x_i] = 0$$

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

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

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

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



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

Variable X และ Y อาจจะเป็นได้ทั้ง Continuous และ Discrete

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

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

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

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

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

replace b in (1)

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

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

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

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

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

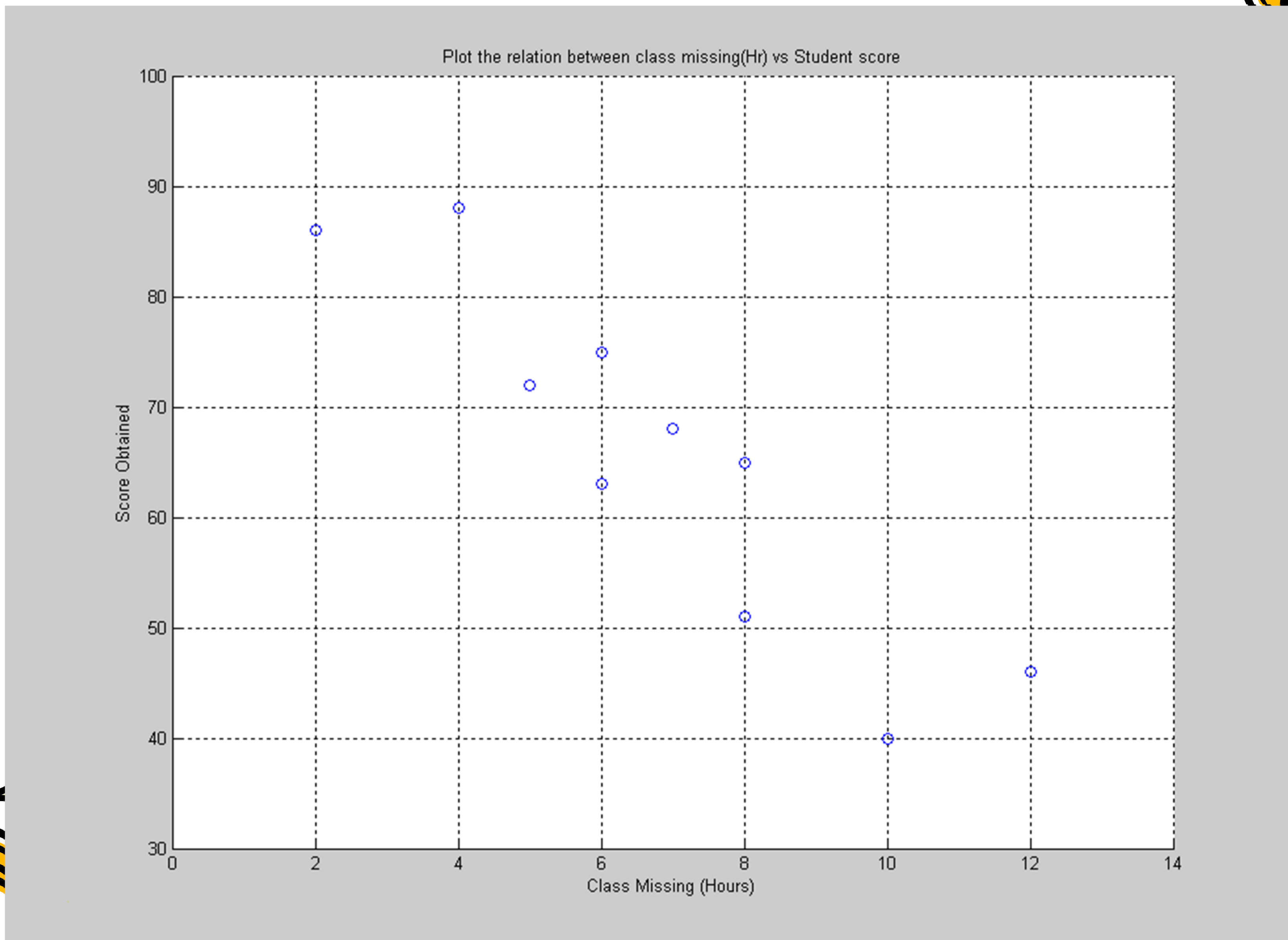
- 1. จากการเก็บข้อมูลตัวอย่าง วิชา CPE 332 ระหว่างคะแนน ที่นักศึกษาได้ กับจำนวนรวมเป็นชั่วโมงที่นักศึกษาขาดเรียน หรือมาสาย ได้ข้อมูลดังตารางข้างล่าง

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

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

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

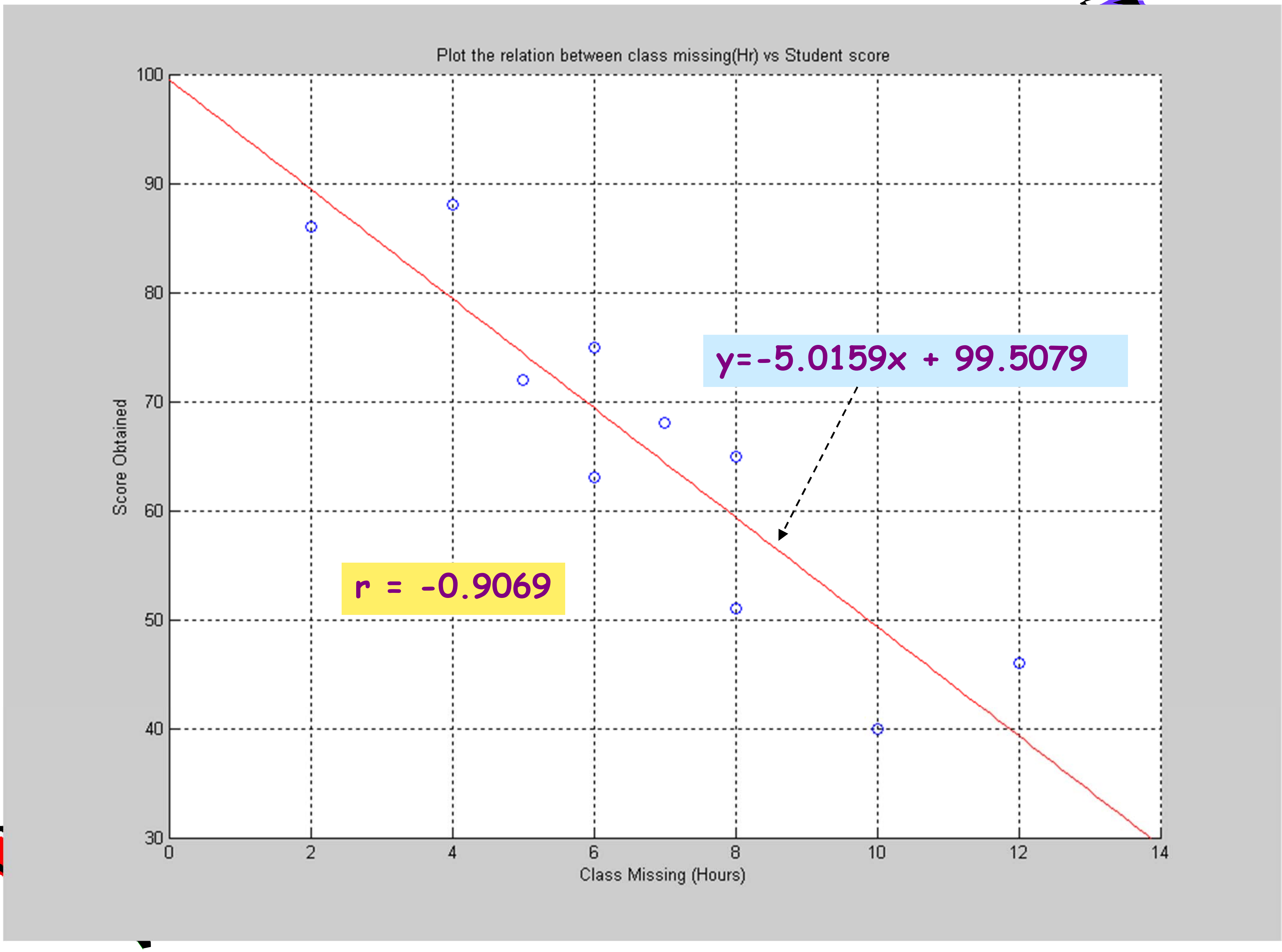
$$\sum_{i=1}^{10} x_i y_i = 4068; \quad \sum_{i=1}^{10} x_i^2 = 538; \quad \sum_{i=1}^{10} y_i^2 = 45084;$$

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

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

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

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


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:

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

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

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

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

(Secant Method Formula)

LU Decomposition:

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

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

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

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

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

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

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