พัฒนาการปฏิบัติการวิเคราะห์เชิงตัวเลขโดยใช้ Mathematica V.3 สำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์

โดย

นางสาวจิตติมา เจริญพิริยะเวศ 38054113

นายสิทธิศักดิ์ เติมนาค 38054160

ปัญหาพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาคณิตศาสตร์ประยุกต์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

NUMERICAL ANALYSIS WORKSHOP WITH MATHEMATICA V.3

FOR SCIENCE / ENGINEER

BY

MISS JITTTIMA	**CHAROENPIRIYAVES**	**38054113**
MR. SITTISAK	**TERMNAK**	**38054160**

A Speacial Project Submitted in Partial Fulfillment of the Requirement

of the Degree of Bachelor of Science

Department of Mathematics and Computer Science

Branch of Applied Mathematics

Faculty of Science

King Mongkut's Institute of Technology Chaokhuntaharn

Lardkrabang

1998

ปัญหาพิเศษเรื่อง	พัฒนาการปฏิบัติการวิเคราะห์เชิงตัวเลขโดยใช้ Mathematica V.3 สำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์
ชื่อนักศึกษา	1.นางสาวจิตติมา เจริญพิริยะเวศ 38054113
	2.นายสิทธิศักดิ์ เติมนาค 38054160
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา	คณิตศาสตร์ประยุกต์
อาจารย์ที่ปรึกษา	อาจารย์กาญจนา คำนึงกิจ

ปัญหาพิเศษฉบับนี้ กรรมการสอบปัญหาพิเศษได้ตรวจพิจารณาแล้วจึงอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต สาขาคณิตศาสตร์ประยุกต์ ประจำปีการศึกษา 2541

(รองศาสตราจารย์ภัคคินี ชิตสกุล)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะกรรมการโครงงานพิเศษ

(รองศาสตราจารย์ ดร.ไมตรี โพธิ์สุข)

ประธานกรรมการ

(อาจารย์กาญจนา คำนึงกิจ)

กรรมการและอาจารย์ที่ปรึกษา

ใจปอง วงษ์สวัสดิ์

(อาจารย์ใจปอง วงษ์สวัสดิ์)

กรรมการ

หัวข้อปัญหาพิเศษ	พัฒนาการปฏิบัติการปฏิบัติการวิเคราะห์เชิงตัวเลขโดยใช้ Mathematica V.3 สำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์
ชื่อนักศึกษา	นางสาวจิตติมา เจริญพิริยะเวศ 38054113
	นายสิทธิศักดิ์ เติมนาค 38054160
อาจารย์ที่ปรึกษา	อาจารย์กาญจนา คำนึงกิจ
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา	คณิตศาสตร์ประยุกต์
ปีการศึกษา	2541

บทคัดย่อ

ปัญหาพิเศษฉบับนี้ได้จัดทำขึ้นโดยมีวัตถุประสงค์คือ นำ software Mathematica V.3 มาพัฒนาการปฏิบัติการวิเคราะห์เชิงตัวเลข เพื่อเป็นประโยชน์และเพื่อความสะดวกรวดเร็วสำหรับนักศึกษาที่เรียนวิชาวิเคราะห์เชิงตัวเลขในการเขียนโปรแกรมสำหรับกรรมวิธีต่าง ๆ ในวิชานี้ และได้นำ software Visual Basic V.5.0 มาใช้ในการสร้าง interface เพื่อรับ input จากนั้น input จะถูกคำนวณโดย Mathematica ซึ่งได้ทำการ link ไว้กับ Visual Basic

ผลจากปัญหาพิเศษนี้จะได้แอพพลิเคชันที่สามารถเป็นการปฏิบัติการ (workshop) สำหรับวิชาวิเคราะห์เชิงตัวเลขที่ให้ผลลัพธ์ที่ถูกต้อง และสะดวกในการใช้

Special Topic Numerical Analysis Workshop with Mathematica V.3 for Science/Engineer

Student MISS JITTIMA CHAROENPIRIYAVES 38054113

MR. SITTISAK TERMNAK 38054160

Advisor MISS KANJANA KUMNUNGKIT

Department Mathematics and Computer Sciences

Branch Applied Mathematics

ABTRACT

The purpose of this special project is to using software Mathematica V.3 to develop Numerical Analysis Workshop. As usefulness and convenience for student who study Numerical Subject for programming in any method. And using software Visual basic V.5.0 to create interface for get input from users, After that computing by Mathematica V.3 that link with Visual Basic V.5.0

The result of this project is an application that can be workshop for Numercal Subject that gives correct result and convenient to use.

กิตติกรรมประกาศ

ปัญหาพิเศษฉบับนี้สำเร็จลงได้ด้วยดีก็เพราะหลายเหตุปัจจัย โดยเฉพาะอย่างยิ่ง

อาจารย์กาญจนา คำนึงกิจ อาจารย์ที่ปรึกษา

ที่ได้ให้แนวทางในการวิจัย ตลอดจนคำปรึกษาอันก่อให้เกิดแนวความคิดที่สามารถแก้ปัญหาต่าง ๆ ที่เกิดขึ้นในระหว่างการทำการวิจัย นอกจากนี้ยังช่วยแนะนำแนวทางในการดำเนินงานและตรวจทานแก้ไขด้วยความเอาใจใส่เป็นอย่างดียิ่ง

ขอบคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ทุกท่านที่สนับสนุนในการใช้ห้องปฏิบัติการคอมพิวเตอร์ และให้ความสะดวกในการเบิกอุปกรณ์ต่าง ๆ ที่ใช้ในการวิจัย

คณะผู้จัดทำขอกราบขอบพระคุณอาจารย์ทุกท่านที่ได้ประสาทวิชาความรู้ทั้งในภาคทฤษฎีและภาคปฏิบัติแก่ผู้จัดทำ จนกระทั่งงานวิจัยสัมฤทธิ์ผลได้ดีทุกประการ

ขอขอบคุณคุณพ่อ คุณแม่และเพื่อน ๆ ที่คอยให้กำลังใจจนงานลุล่วงไปได้ด้วยดี

ขอขอบพระคุณ

คณะผู้จัดทำ

สารบัญ

บทที่ 1
บทนำ

1. ความสำคัญ /ที่มาของปัญหาพิเศษ

การเรียนการสอนในวิชา การวิเคราะห์เชิงตัวเลข สำหรับวิทยาศาสตร์หรือวิศวกรรมศาสตร์นั้น จุดประสงค์คือ การให้นักศึกษาที่เรียนได้ทราบถึงหลักการ หรือกรรมวิธีการวิเคราะห์เชิงตัวเลข และสิ่งที่จำเป็นตามมาคือ การเขียนโปรแกรมเพื่อความสะดวกรวดเร็วในในการใช้หลักการหรือกรรมวิธีดังกล่าวข้างต้น ซึ่งในการเขียนโปรแกรมนั้นสามารถใช้ Basic, Fortran, Pascal หรือ C ได้ แต่ในปัจจุบันมี Software ทางคณิตศาสตร์ เช่น Maple, Mathematica หรือ MathLab ที่สามารถให้ ความสะดวกในการแก้ปัญหาทางคณิตศาสตร์ได้ดียิ่งขึ้น ดังนั้นปัญหาพิเศษนี้ จึงได้เห็นความสำคัญ ของ Mathematica V.3 มาเป็นประโยชน์ในเชิงปฏิบัติการวิเคราะห์เชิงตัวเลข สำหรับวิทยาศาสตร์หรือวิศวกรรมศาสตร์

2. วัตถุประสงค์ของปัญหาพิเศษ / โครงงานพิเศษ

1. ศึกษา Software Mathematica V.3 เพื่อใช้ในการทำปัญหาพิเศษ
2. ศึกษาขอบเขตของการวิเคราะห์เชิงตัวเลข สำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์
3. พัฒนาการวิเคราะห์เชิงตัวเลข สำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์

3. ขอบเขตของปัญหา

1. สามารถเข้าใจ Software Mathematica V.3 ที่ใช้ประโยชน์ในส่วนของการวิเคราะเชิงตัวเลขสำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์ได้
2. สามารถนำ Software Mathematic V.3 มาประยุกต์ใช้กับการวิเคราะห์เชิงตัวเลขสำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์
3. นำผลที่ได้มาเป็นแนวทางในเรื่องของการเรียนการสอนในวิชาการวิเคราะห์เชิงตัวเลขสำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์

4. ประโยชน์ที่คาดว่าจะได้รับ

1. ทราบถึงการใช้ประโยชน์ของ Software Mathematic V.3
2. สามารถเขียนโปรแกรมบน Software Mathematica V.3
3. นำประโยชน์ของ Software Mathermatica V.3 มาประยุกต์ใช้กับการวิเคราะห์เชิงตัวเลขสำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์
4. เพิ่มความสะดวกรวดเร็วให้กับผู้ศึกษาวิชาการวิเคราะห์เชิงตัวเลข สำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์

5. ขั้นตอนในการปฏิบัติงาน

1. ศึกษาขอบเขตของการวิเคราะห์เชิงตัวเลข สำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์
2. ศึกษา Software Mathematic V.3 ในเชิงทั่วไป
3. ศึกษาการเขียนโปรแกรมใน Software Mathematic V.3
4. ทำการพัฒนาการวิเคราะห์เชิงตัวเลข สำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์
5. ทดสอบและแก้ไขให้มีประสิทธิภาพ
6. จัดทำคู่มือ/เอกสารประกอบปัญหาพิเศษ

6. อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

1. HARDWARE
   - CPU ไม่ต่ำกว่า Pentium 166
   - RAM 64 MB
   - HARD DISK 1.6 GB
2. SOFTWARE
   - MATHEMATICA V.3
   - VISUAL BASIC V.5.0

ระยะเวลาในการทำโครงงานพิเศษ

วัน/เดือน/ปี	การทำงาน
1 มิย. –30 มิย. 41	ศึกษาปัญหาและที่มาของหัวข้อปัญหาพิเศษ
1 สค. – 16 กย. 41	ศึกษาขอบเขตของการวิเคราะห์เชิงตัวเลข สำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์
17 กย. – 30 กย. 41	ศึกษา Software Mathematica V.3 ในเชิงทั่วไป
9 ตค. – 5 ตค. 41	จัดทำเอกสารประกอบโครงงานพิเศษ (ส่วนแรก)
6 ตค. – 20 พย. 41	ศึกษาการเขียนโปรแกรมใน Software Mathematica V.3
21 พย. – 10 มค. 42	ทำการพัฒนาการวิเคราะห์เชิงตัวเลข สำหรับวิทยาศาสตร์และวิศวกรรมศาสตร์
11 มค. – 11 กพ. 42	ทำการทดสอบและแก้ไขให้มีประสิทธิภาพ
12 กพ. – 28 กพ. 42	จัดทำเอกสารประกอบโครงงาน

บทที่ 2

การวิเคราะห์ตัวเลข และMathematica V.3

การวิเคราะห์ตัวเลข

- Approximation and Error
- Root of Equation
- System of Linear Algebraic Equations
- Numerical Differentiation and Integration
- Ordinary Differential Equation
- Partial Differential Equation

1. Approximation and Errors (ค่าประมาณและค่าผิดพลาด)

1.1 นิยามของค่าผิดพลาด (Error)

ค่าผิดพลาดที่เกิดขึ้นนั้นมาจากการนำค่าประมาณมาใช้แทนค่าจริง โดยค่าผิดพลาดที่เกิดขึ้นนั้นจะประกอบไปด้วย ค่าผิดพลาดเพราะตัดปลาย(Truncation Error) และ ค่าผิดพลาดเพราะการตัดเศษ(Round-Off Error) โดยที่ทั้งสองประเภทนั้น เป็นความสัมพันธ์ระหว่างค่าจริงและค่าประมาณ โดยสามารถเขียนอยู่ในรูป

$$\text{ค่าจริง} = \text{ค่าประมาณ} + \text{ค่าผิดพลาด} \tag{1.1}$$

เราจะสามารถหาค่าผิดพลาดได้จาก

$$E_t = \text{ค่าจริง} - \text{ค่าประมาณ} \tag{1.2}$$

โดยที่ E_t แทนค่าจริงสำหรับค่าผิดพลาด

จากสมการที่ (1.2) จากค่าผิดพลาดสัมพัทธ์เราจะได้

$$\varepsilon_t = \frac{\text{ค่าจริง - ค่าใกล้เคียง}}{\text{ค่าจริง}} \times 100\% \tag{1.3}$$

เมื่อ ε_t แทน***ร้อยละของค่าผิดพลาดสัมพัทธ์ (True percent relative error)***

จากสมการที่ (1.2) และ(1.3) นั้น E_t และ ε_t เป็น ค่าผิดพลาดที่เทียบกับค่าจริง แต่ในทางระเบียบวิธีเชิงตัวเลขนั้น จะไม่สามารถทราบค่าจริงได้ ดังนั้นค่าผิดพลาดที่ได้ก็จะเป็นค่าผิดพลาดที่เทียบกับค่าประมาณ

$$\varepsilon_a = \frac{\text{ค่าผิดพลาดประมาณ}}{\text{ค่าประมาณ}} 100\% \tag{1.4}$$

หรือ

$$\varepsilon_a = \frac{\text{ค่าประมาณสุดท้าย - ค่าประมาณก่อนสุดท้าย}}{\text{ค่าประมาณสุดท้าย}} 100\% \tag{1.5}$$

เมื่อ ε_a แทน***ร้อยละของค่าผิดพลาดเทียบกับค่าประมาณ***

ในสมการที่ (1.2) และ (1.5) นั้น เครื่องหมายที่ได้อาจเป็น ค่าบวก หรือค่าลบ ก็ได้ ถ้าค่าประมาณนั้นใหญ่กว่า ค่าจริง (หรือค่าประมาณก่อนสุดท้ายมากกว่าค่าประมาณสุดท้าย) นั้นค่าผิดพลาดจะเป็นค่าลบ และถ้าค่าประมาณนั้นน้อยกว่าค่าจริง ค่าผิดพลาดก็จะเป็นบวก สำหรับสมการที่ (1.3) และ (1.5) ตัวหารอาจจะน้อยกว่าศูนย์ ซึ่งก็สามารถจะทำให้เป็นค่าผิดพลาดที่เป็นลบได้เหมือนกัน แต่บ่อยครั้งที่เราไม่ได้สนใจเกี่ยวกับเครื่องหมายหน้าค่าผิดพลาด แต่สนใจในค่าสมบูรณ์ที่น้อยกว่าค่าของ ε_s เพราะฉะนั้นเราจะทำการคำนวณจนกระทั่ง

$$|\varepsilon_a| < \varepsilon_s \tag{1.6}$$

โดยที่

$$\varepsilon_s = \left(0.5 \times 10^{2-n}\right)\% \tag{1.7}$$

เมื่อ $|\varepsilon_a| < \varepsilon_s$ เป็นผลลัพธ์ที่ได้จะมีความถูกต้องอย่างน้อยที่สุดถึงตำแหน่งที่ n

<u>ตัวอย่างที่ 1</u> จงหาค่าประมาณของ $e^{0.5}$ โดยคำนวณจากฟังก์ชันเอกซโพเนนเชียล

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!}$$

ให้มีความถูกต้องถึงทศนิยมตำแหน่งที่ 3 และคำนวณร้อยละของค่าผิดพลาดเทียบกับค่าประมาณ (ε_s) และร้อยละของค่าผิดพลาดสัมพัทธ์(ε_t) เมื่อค่าจริงของ $e^{0.5} = 1.648721271$

<u>วิธีทำ</u> จะเห็นว่า ถ้าต้องการประมาณค่า e^x ให้เข้าใกล้มากยิ่งขึ้น จะต้องเพิ่มจำนวนพจน์จนกระทั่ง $|\varepsilon_a| < \varepsilon_s$ โดยมีความถูกต้องถึงทศนิยมตำแหน่งที่ 3

คำนวณหาค่า ε_s จาก $\varepsilon_s = \left(0.5 \times 10^{2-3}\right)\% = 0.05\%$

ดังนั้นเราจะเพิ่มจนกระทั่ง $|\varepsilon_a| < 0.05\%$

ใช้หนึ่งพจน์

$$e^x \approx 1$$
$$e^{0.5} \approx 1$$

ใช้สองพจน์

$$e^x \approx 1 + x$$
$$e^{0.5} \approx 1 + 0.5 = 1.5$$

$$\varepsilon_t = \frac{\text{ค่าจริง - ค่าประมาณ}}{\text{ค่าจริง}} \times 100\%$$
$$= \frac{1.648721271 - 1.5}{1.648721271} \times 100\%$$
$$= 9.02\%$$

$$\varepsilon_a = \frac{\text{ค่าประมาณสุดท้าย - ค่าประมาณก่อนสุดท้าย}}{\text{ค่าประมาณสุดท้าย}} 100\%$$
$$= \frac{1.5 - 1}{1.5} \times 100\%$$
$$= 33.3\%$$

$|\varepsilon_a| = |33.3|$ ยังน้อยกว่า ε_s

ทำจนกระทั่ง $|\varepsilon_a| < 0.05$ จะได้ค่าดังตาราง

จำนวนพจน์	ผลลัพธ์($e^{0.5}$)	ε_t%	ε_a%
1	1	39.3	
2	1.5	9.02	33.3
3	1.625	1.44	76.9
4	1.645833333	0.1757	1.27
5	1.648437500	0.00172	0.158
6	1.648697917	0.00142	0.0158

1.2 ค่าที่ผิดพลาดเพราะปัดเศษ (Round-off Error)

คือค่าผิดพลาดที่เกิดจากการคำนวณทางคณิตศาสตร์ ซึ่งผลลัพธ์ที่ได้มีขนาดเกินกว่าที่จะเก็บบันทึกไว้ได้ ตัวอย่างเช่น คอมพิวเตอร์ ซึ่งแทนตัวเลขด้วย fix-point number ก็จะแทนตัวเลขที่จำกัดค่า $\pi, e, \sqrt{7}$ ไม่สามารถแทนด้วย fix number ค่าผิดพลาดนี้ เรียกว่า ค่าผิดพลาดเพราะการปัดเศษ

1.2.1 การแทนตัวเลขในคอมพิวเตอร์ (Computer Representation of Numbers)

ค่าผิดพลาดเพราะปัดเศษนั้นเกี่ยวข้องกับการเก็บค่าตัวเลขในคอมพิวเตอร์

1.2.1.1 การแทนเลขจำนวนเต็ม (Integer Representation)

เราจะแสดงการแปลงเป็นเลขฐาน 10 บนคอมพิวเตอร์ เรียกว่า ***Signed magnitude method*** โดยบิตแรกของเลขฐานสองที่เราแปลง จะเป็นบิตเครื่องหมาย ถ้าเป็น 0 แสดงว่าเป็นจำนวนเต็มบวก และ 1 เป็นจำนวนเต็มลบ ส่วนบิตอื่น ๆ ที่เหลือจะเป็นเลขฐานสิบ

ตัวอย่างเช่น -173 แปลงเป็นเลขฐานสอง 16 บิต บนคอมพิวเตอร์ จะแสดงได้ดังรูป

1	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1

บิตเครื่องหมาย เลขฐาน 2

Signed magnitude method จะไม่สามารถใช้ในการแสดงเลขจำนวนเต็มบนคอมพิวเตอร์ ดังนั้นจึงใช้การทำเลข 2 คอมพลีเมนต์แทนโดยรวมบิตเครื่องหมายกับบิตอื่น ๆ แต่คอมพิวเตอร์ก็มีข้อจำกัดของความสามารถในการแสดงเลขจำนวนเต็ม

1.2.1.2 การแทนเลขทศนิยม (Floating-Point Representation)

เลขทศนิยม จะแสดงได้โดย

$$mb^{e}\ e$$

m คือ แมนทิสซา

b คือ เลขฐาน

e คือ เลขชี้กำลัง

ตัวอย่างเช่น 156.78 จะแสดงด้วย $0.15678.10^{3}$ ในเลขทศนิยมฐานสิบ

เครื่องหมาย เลขยกกำลัง แมนทิสซา

จากรูป แสดงการเก็บเลขทศนิยม โดยที่บิตแรกเก็บเครื่องหมาย บิตต่อไปเก็บเลขชี้กำลัง และบิตสุดท้ายเก็บแมนทิสซา

แมนทิสซาจะถูก normalized ถ้าเป็น 0 เช่น 1/34 = 0.029411765… ให้เก็บเลขทศนิยมได้ 4 ตำแหน่ง ดังนั้น 1/34 จะถูกเก็บเป็น 0.2941×10^{-1} จะเห็นได้ว่าค่า m จะถูกจำกัดเนื้อที่เนื่องจาก $\frac{1}{b} \le m < 1$ ตัวอย่างเช่น สำหรับเลขฐานสิบ m จะมีค่าระหว่าง 0.1 กับ 1 , เลขฐานสอง m จะมีค่าระหว่าง 0.5 กับ 1

เลขทศนิยมแทนตัวเลขได้มาก ๆ ได้ แต่ก็จะใช้เนื้อที่ในการเก็บจำนวนมาก และมีการประมวลผลมากกว่าเลขจำนวน นอกจากนี้ค่าผิดพลาดยังเกิดขึ้นจากค่าแมนทิสซาที่จำกัด

1.2.2 การคำนวณทางคณิตศาสตร์

(Arithmetic Manipulations of Computer Number)

การคำนวณทางคณิตศาสตร์ เช่น บวก ลบ คูณ และ หาร จะทำให้เกิดค่าผิดพลาดได้ เนื่องจากการปัดเศษ

Common Arithmetic Operation : การบวก ลบ คูณ และ หาร เลขทศนิยมนั้น คอมพิวเตอร์จะเก็บค่าแมนทิสซา ได้ 4 ตัว และเลขชี้กำลังได้ 1 ตัว

การบวก

$0.1557 \times 10^{1} + 0.4381 \times 10^{-1}$

จะได้ $0.1557 \quad \times 10^{1}$

$\underline{0.4381 \quad \times 10^{1}}$

$\underline{0.160081 \quad \times 10^{1}}$

เพราะฉะนั้นจะได้ผลลัพธ์ คือ 0.1600×10^{1}

การลบ

จาก $36.41 - 26.86$

จะได้ 0.3641×10^{2}

$\underline{0.2686 \times 10^{2}}$

$\underline{0.0955 \times 10^{2}}$

เพราะฉะนั้นจะได้ผลลัพธ์ คือ 0.9550×10^{1}

การคูณและการหาร

ค่าผิดพลาดสำหรับการคูณการหารนั้น จะมากกว่าการบวกและการลบ

การคูณ

เลขชี้กำลัง บวกกัน

แมนทิสซา คูณกัน

0.1363×10^{3} คูณกับ 0.6423×10^{-1}

$= 0.08754549 \times 10^{2}$

$\approx 0.8754 \times 10^{1}$

การบวกตัวเลขน้อย ๆ บวกเข้ากับตัวเลขมาก ๆ

เราจะบวก 0.0010 กับ 4000 คอมพิวเตอร์จะเก็บค่า m ได้ 4 หลัก และเลขชี้กำลัง 1 หลัก จะได้

$$
\begin{array}{l}
0.0000001 \times 10^{4} \\
\underline{0.4000000 \times 10^{4}} \\
\underline{0.4000001 \times 10^{4}} \quad \approx \quad 0.4000 \times 10^{4}
\end{array}
$$

ซึ่งจะไม่มีผลต่อการบวก

1.3 ค่าผิดพลาดเพราะการตัดปลาย (Truncation Error)

1.3.1 อนุกรมเทย์เลอร์

ค่าฟังก์ชันของจุด ๆ หนึ่ง สามารถหาได้โดยใช้ค่าฟังก์ชัน และค่าอนุพันธ์ของอีกจุดหนึ่ง ตัวอย่าง เช่น ถ้าใช้เพียงพจน์แรกของอนุกรมเทย์เลอร์จะได้

$$f(x_{i+1}) \approx f(x_i) \qquad (1.6)$$

เรียกว่า ***การประมาณโดยใช้อนุกรมเทย์เลอร์อันดับศูนย์***

จากสมการ 1.6 ค่าฟังก์ชันที่จุด x_{i+1} มีค่าเท่ากับค่าฟังก์ชันที่จุด x_i ซึ่งจะเป็นค่าประมาณที่ถูกต้อง โดยการเพิ่มจำนวนพจน์ ก็สามารถได้ค่าประมาณที่ถูกต้อง คือถ้าฟังก์ชันเปลี่ยนไป โดยการเพิ่มจำนวนพจน์ ก็สามารถที่จะประมาณค่าได้ใกล้เคียง เช่น

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)(x_{i+1} - x_i) \qquad (1.7)$$

เรียกว่า ***การประมาณค่าโดยใช้อนุกรมเทย์เลอร์อันดับ 1***

Complete Taylor Series Expansion เขียนได้ดังนี้

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(x_i)(x_{i+1} - x_i)^2}{2!} + \frac{f'''(x_i)(x_{i+1} - x_i)^3}{3!} + \cdots + \frac{f^{n}(x_i)(x_{i+1} - x_i)^n}{n!} + R_n$$

$$R_n = \frac{f^{(n+1)}(\xi)(x_{i+1} - x_i)^{(n+1)}}{(n+1)!}$$

โดยที่ R_n คือ เศษเหลือ สำหรับการประมาณโดยใช้อนุกรมเทย์เลอร์อันดับที่ n

ξ คือ ค่าของ x ซึ่งอยู่ระหว่าง x_i และ x_{i+1}

ให้ h คือ step size $= x_{i+1} - x_i$

เมื่อแทนโดย h จะได้

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)h^2}{2!} + \frac{f'''(x_i)h^3}{3!} + \cdots + \frac{f^n(x_i)h^n}{n!} + R_n$$

$$R_n = \frac{f^{(n+1)}(\xi)h^{(n+1)}}{(n+1)!}$$

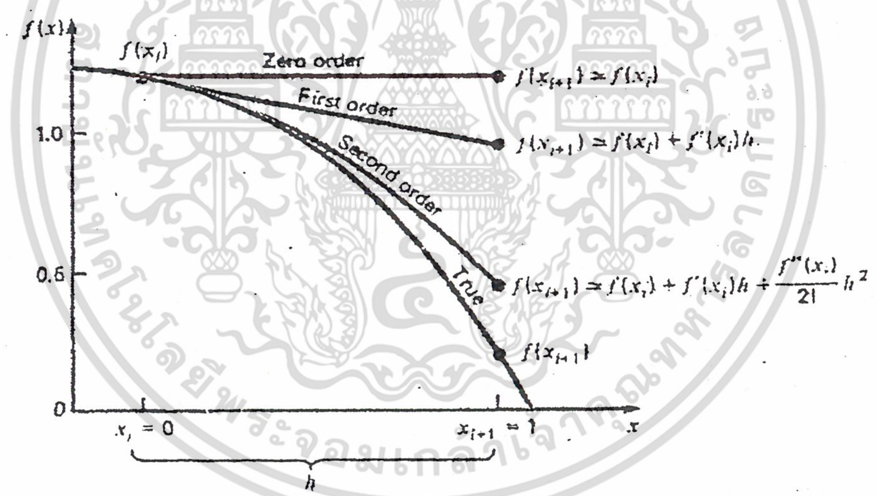
<u>ตัวอย่าง 1.3</u> จงหาค่าประมาณของฟังก์ชันที่ $x_{i+1} = 1$

$$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$$

โดยใช้ อนุกรมเทย์เลอร์อันดับ 0 ถึงอันดับ 4

เมื่อกำหนดค่าเริ่มต้น $x_i = 0$ และ h = 1

(ค่าจริงของ f(1) = 0.2 และ f(0) = 1.2)



รูป 1.1

<u>วิธีทำ</u>

E_t = ค่าจริง - ค่าประมาณ

n = 0 : ใช้อนุกรมเทย์เลอร์อันดับ 0

$$f(x_{i+1}) \approx f(x_i)$$

$$f(1) \approx f(0)$$

$$\approx 1.2$$

$$E_t = 0.2 - 1.2$$

$$= -1.0$$

n = 1 : ใช้อนุกรมเทย์เลอร์อันดับ 1

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)(x_{i+1} - x)$$

หาค่า $f'(x)$ จะได้

$$f'(x) = -4.0x^3 - 0.45x^2 - 1.0x - 0.25$$
$$f'(0) = -0.4(0) - 0.45(0) - 1.0(0) - 0.25$$
$$= -0.25$$
$$f(x_{i+1}) \approx f(0) + f'(0)(1-0)$$
$$f(1) \approx 1.2 - 0.25$$
$$\approx 0.95$$
$$E_t = 0.2 - 0.95 = -0.75$$

n = 2 : ใช้อนุกรมเทย์เลอร์อันดับ 2

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(x_i)(x_{i+1} - x_i)^2}{2!}$$

หาค่า $f''(x)$

$$f''(x) = -1.2x^2 - 0.9x - 1.0$$
$$f''(0) = -1.2(0) - 0.9(0) - 1.0$$
$$= -1.0$$
$$f(x_{i+1}) \approx f(0) + f'(0)(1-0) + \frac{f''(0)(1-0)^2}{2!}$$
$$f(1) \approx 1.2 - 0.25(1) - \frac{1}{2}$$
$$\approx 0.45$$
$$E_t = 0.2 - 0.45$$
$$= -0.25$$

n = 3 : ใช้อนุกรมเทย์เลอร์อันดับ 3

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)h^2}{2!} + \frac{f'''(x_i)h^3}{3!}$$

หาค่า $f'''(x)$

$$f'''(x) = -2.4x - 0.9$$
$$f'''(0) = -2.4(0) - 0.9$$
$$= -0.9$$

$$f(1) \approx f(0) + f'(0)(1) + \frac{f''(0)(1)^2}{2!} + \frac{f'''(0)(1)^3}{3!}$$
$$\approx 1.2 - 0.25 - 0.5 - 0.15$$
$$\approx 0.3$$
$$E_t = 0.2 - 0.3$$
$$= -0.1$$

n = 4 : ใช้อนุกรมเทย์เลอร์อันดับ 4

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)h^2}{2!} + \frac{f'''(x_i)h^3}{3!} + \ldots + \frac{f^{(4)}(x_i)h^4}{4!} + R_4$$
$$R_4 = \frac{f^{(5)}(\xi)h^{(5)}}{5!}$$
$$f'''(x) = -2.4x - 0.9$$
$$f^{(4)}(x) = -2.4$$
$$f(1) \approx 1.2 - 0.25(1) - 0.5(1)^2 - 0.15(1)^3 - 0.10(1)^4$$
$$\approx 0.2$$

จะเห็นว่า $R_4 = 0$ เพราะว่า $f^{(5)}(\xi) = 0$

โดยทั่วไปถ้าใช้อนุกรมเทย์เลอร์อันดับที่ n จะได้ค่าที่ถูกต้อง สำหรับฟังก์ชันพหุนามอันดับที่ n จากสมการ (1.9) เขียนได้เป็น

$$R_n = o(h^{n+1})$$

นั่นหมายความว่า ค่าผิดพลาดเพราะปัดเศษ ขึ้นอยู่กับเลขชี้กำลังของ h^{n+1} หรือค่าผิดพลาดเป็นสัดส่วน กับ h (step size)

ถ้าค่าผิดพลาด = o(h) ก็คือ ถ้าลด h ลงครึ่งหนึ่ง ค่าผิดพลาดก็จะลดไปครึ่งหนึ่ง

ถ้าค่าผิดพลาด = $o(h)^2$ ก็คือ ถ้าลด h (step size) ลงครึ่งหนึ่ง ก็จะลดค่าผิดพลาดได้ ¾

ดังนั้น ค่าผิดพลาดเพราะตัดปลายลดลง เมื่อเพิ่มจำนวนพจน์ของ อนุกรมเทย์เลอร์ แต่ถ้าขนาดของ h เล็กพอ ใช้อนุกรมเทย์เลอร์อันดับ 1 ก็สามารถจะลดค่าผิดพลาดได้มาก

ตัวอย่าง 1.4 จงใช้อนุกรมเทย์เลอร์ตั้งแต่อันดับที่ 0 ถึง 6 เพื่อหาค่าประมาณ $f(x) = Cos(x)$ ที่ $x_{i+1} = \pi/3, x_i = \pi/4$ ค่าจริงของ $Cos(\pi/3) = 0.5$

วิธีทำ

$$\varepsilon_t = \frac{\text{ค่าจริง - ค่าใกล้เคียง}}{\text{ค่าจริง}} \times 100\%$$

$$h = \pi/3 - \pi/4 = \pi/12$$

ใช้อนุกรมเทย์เลอร์อันดับ 0

$$f(x_{i+1}) \approx f(x_i)$$

$$f(\pi/3) \approx Cos(\pi/4)$$

$$\approx 0.70710781$$

$$\varepsilon_t = \frac{0.5 - 0.707106781}{0.5} \times 100\%$$

$$= -41.4\%$$

ใช้อนุกรมเทย์เลอร์อันดับ 1

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)(x_{i+1} - x_i)$$

หาค่า $f'(x)$

$$f'(x) = -Sinx$$

$$f(\pi/3) \approx Cos(\pi/4) - Sin(\pi/4)(\pi/12)$$

$$\approx 0.521986659$$

$$\varepsilon_t = \frac{\text{ค่าจริง - ค่าใกล้เคียง}}{\text{ค่าจริง}} \times 100\%$$

$$= \frac{0.5 - 0.521986659}{0.5} \times 100\%$$

$$= -4.40\%$$

ใช้อนุกรมเทย์เลอร์อันดับ 2

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(x_i)(x_{i+1} - x_i)^2}{2!}$$

หาค่า $f''(x)$

$$f''(x) = -Cos(x)$$

$$f(\pi/3) \approx Cos(\pi/4) - Sin(\pi/4)\frac{\pi}{12} - \frac{Cos(\pi/4)(\pi/12)^2}{2!}$$

$$\approx 0.497754491$$

$$\varepsilon_t = \frac{\text{ค่าจริง - ค่าใกล้เคียง}}{\text{ค่าจริง}} \times 100\%$$

$$= 0.449$$

1.3.2 เศษเหลือของอนุกรมเทย์เลอร์ (Remainder of Taylor Series Expansion)

สมมุติว่าเราตัดปลายอนุกรมเทย์เลอร์

$$f(x_{i+1}) \approx f(x_i)$$

$$R_0 = f'(x_i)h + \frac{f''(x_i)h^2}{2!} + \cdots$$

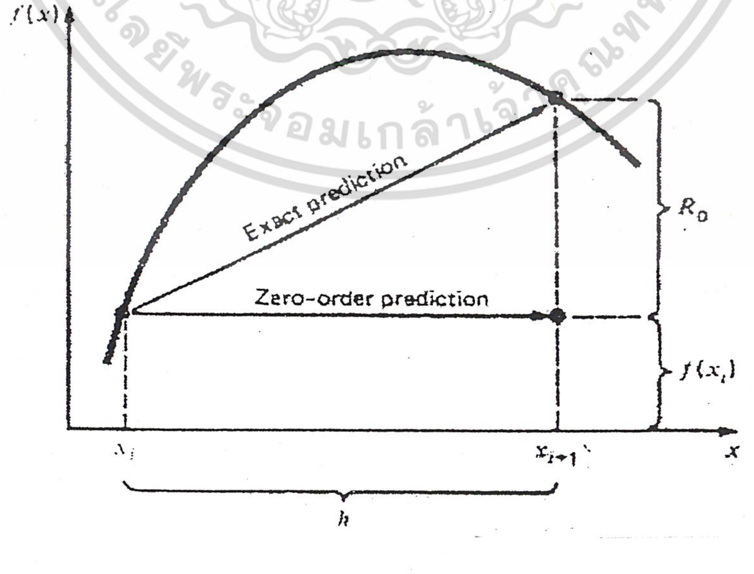
ตัดปลาย R_0 จะได้

$$R_0 = f'(x_i)h$$

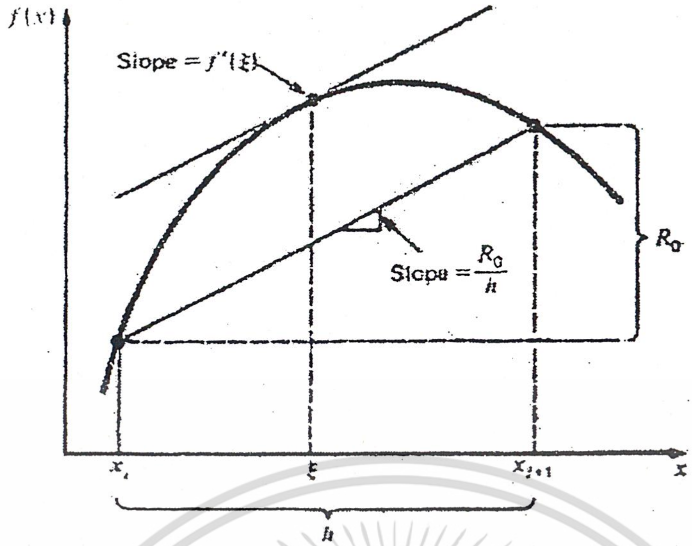
Derivative Mean Value Theorem กล่าวว่า ถ้าฟังก์ชัน f(x) และอนุพันธ์อันดับหนึ่งต่อเนื่องในช่วง x_i ถึง x_{i+1} แล้ว จะมีค่า x อย่างน้อยหนึ่งค่า ที่มีค่าเท่ากับความชันเท่ากับ เส้นที่ลากระหว่าง $f(x_i)$ กับ $f(x_{i+1})$

$$f'(\xi) = \frac{R_0}{h}$$

$$R_0 = f'(\xi)h$$



รูป 1.2

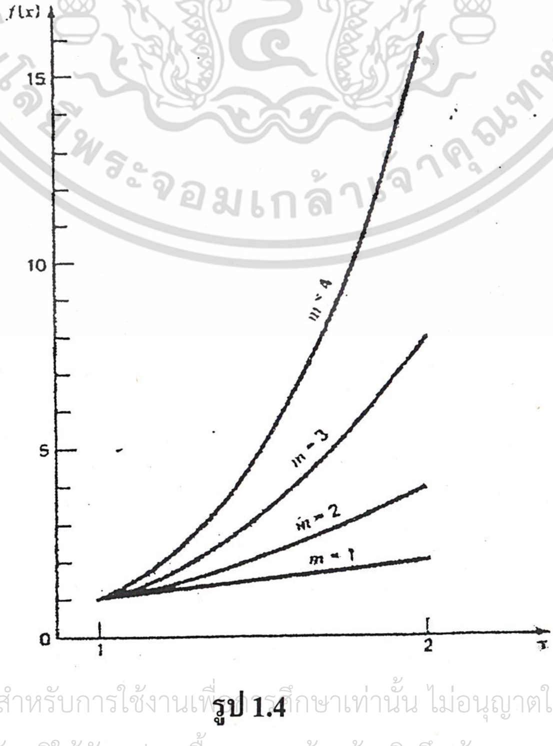


รูป 1.3

ตัวอย่าง 1.5 จงหาค่าประมาณที่จุด x_{I+1} และ R_1 โดยใช้อนุกรมเทย์เลอร์อันดับ 1 จากรูป 1.4 เป็นฟังก์ชัน

$$f(x) = x^m$$

ให้ m มีค่าตั้งแต่ 1 – 4 และ $x_i = 1$, $x_{i+1} = 2$



รูป 1.4

<u>วิธีทำ</u>

ใช้อนุกรมเทย์เลอร์อันดับที่ 1

$$f(x_{i+1}) \approx f(x_i) + mx_i^{m-1}h$$

$$R_1 = \frac{f''(x_i)}{2}h^2 + \frac{f^{(3)}(x_i)}{3!}h^{(3)} + \frac{f^{(4)}(x_i)}{4!}h^{(4)} + \cdots$$

กรณี m = 1

$$f(2) = f(1) + 1(1) \approx 1 + 1 = 2$$

$$R_1 = 0$$

กรณี m = 2

$$f(2) = f(1) + 2(1)(1) = 1 + 2 = 3$$

$$R_1 = \frac{2}{2}(1)^2 + 0 + 0 + \cdots = 1$$

กรณี m = 3

$$f(2) = f(1) + 3(1)^2(1) = 1 + 3 = 4$$

$$R_1 = \frac{6}{2}(1)^2 + \frac{6}{6}(1)^3 + 0 + \cdots = 4$$

กรณี m = 4

$$f(2) = f(1) + 4(1)^3(1) = 1 + 4 = 5$$

$$R_1 = \frac{12}{2}(1)^2 + \frac{24}{6}(1)^3 + \frac{24}{24}(1)^4 + 0 + 0 + \cdots = 11$$

จากทั้ง 4 กรณีจะเห็นได้ว่า R_1 จะเพิ่มขึ้นถ้าฟังก์ชันไม่เป็นเชิงเส้น ถ้าเรามาพิจารณา กรณี m = 4 ถ้าลดค่า h (Step Size)

$$f(x_i + h) = f(x_i) + 4x_i^3 h$$

ถ้า $x_i = 1$, f(1) = 1 จะได้สมการ

$$f(1+h) = 1 + 4h$$

โดยที่

$$R_1 = 6h^2 + 4h^3 + h^4$$

จากสมการนี้ ถ้าเราลดค่า h ลง ค่า R_1 ก็จะลดลงด้วย สรุปได้ว่า ถ้าใช้อนุกรมเทย์เลอร์อันดับ 1 ก็จะถูกต้องมากขึ้น ถ้าลดค่า h หรือ ลดค่า m แต่การลดค่า m นั้นทำไม่ได้เพราะรูปแบบของฟังก์ชัน ขึ้นอยู่กับปัญหาที่เราสนใจ

ตาราง 1.2 แสดงการเปรียบเทียบค่าจริงและค่าประมาณโดยใช้อนุกรมเทย์เลอร์อันดับ 1 เมื่อ h ต่างกัน

h	ค่าจริง	ประมาณโดยใช้อนุกรมเทย์เลอร์อันดับ 1	R_1
1	16	5	11
0.5	5.0625	3	2.0625
0.25	2.441406	2	0.441406
0.125	1.601807	1.5	0.101807
0.0625	1.274429	1.25	0.024429
0.03125	1.130982	1.125	0.005982
0.015625	1.063980	1.0625	0.001480

1.3.3 การหาค่าอนุพันธ์เชิงตัวเลข(Numerical Differentiation)

1.3.3.1 การหาอนุพันธ์อันดับหนึ่งโดยใช้ผลต่างสืบเนื่องข้างหน้า

(Forward Difference Approximation of the First Derivative)

จากสมการ

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + R_1$$

จะได้

$$f'(x) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} + \frac{R_1}{x_{i+1} - x_i} \qquad (1.12)$$

โดยที่ $$R_1 = \frac{f''(\xi)(x_{i+1} - x_i)^2}{2!}$$

แทนค่า R_1 ใน (1.12) จะได้

$$f'(x_i) = \frac{\Delta f_i}{h} - \frac{f''(\xi)h^2}{2!}$$

หรือ $$f'(x_i) = \frac{\Delta f_i}{h} - O(h)$$

เมื่อ Δf_i **คือ ผลต่างข้างหน้าอันดับหนึ่ง (First forward difference)**

h คือ Step size หรือ $x_{i+1} - x_i$

ในสมการนี้จะเป็น ผลต่างข้างหน้า เพราะ การหาค่าจะอยู่ที่ i และ i + 1 ในการประมาณค่าผลต่าง และในพจน์ของ $\Delta f_i / h$ จะเรียกว่าเป็น สัดส่วนของผลต่างจำกัด อันดับที่ 1(first finite divided difference)

1.3.3.2 การหาอนุพันธ์อันดับหนึ่งโดยใช้ผลต่างสืบเนื่องย้อนหลัง

(Backward Difference Approximation of the First Derivative)

อนุกรมเทย์เลอร์ สามารถที่จะคำนวณหาค่าก่อนหน้าได้ จากสมการ

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)h^2}{2!} - \cdots$$

จะได้

$$f'(x) \approx \frac{f(x_i) - f(x_{i-1})}{h} = \frac{\nabla f_i}{h}$$

เมื่อ O(h) คือ ค่าผิดพลาด

∇f_i **คือ ผลต่างย้อนหลังอันดับหนึ่ง (First backward difference)**

1.3.3.3 การหาอนุพันธ์อันดับหนึ่งโดยใช้ผลต่างสืบเนื่องตรงกลาง

(Centered Difference Approximation of the First Derivative)

จากสมการ

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)h^2}{2!} + \cdots$$

และ

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)h^2}{2!} - \cdots$$

เราจะได้

$$f(x_{i+1}) = f(x_{i-1}) + 2f'(x_i)h + \frac{f'''(x_i)h^3}{3} + \cdots$$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - \frac{f'''(x_i)h^2}{6} + \cdots$$

หรือ

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - O(h^2)$$

ค่าผิดพลาดที่เกิดจากการตัดปลายคือ อันดับ h^2 ซึ่งจะแตกต่างจาก 2 แบบแรกที่มี ค่าผิดพลาด อันดับ h ก็คือ ถ้าลด h ไปครึ่งหนึ่ง ก็จะมีค่าผิดพลาด1/4 แต่ถ้าเป็นแบบไปข้างหน้า และย้อนหลัง จะมีค่าผิดพลาด 1/2

1.3.3.4 การหาอนุพันธ์อันดับสองโดยใช้ผลต่างสืบเนื่อง

(Finite Difference Approximation of Second Derivatives)

จากอนุพันอันดับหนึ่งนั้น อนุกรมเทย์เลอร์สามารถใช้ค่าตัวเลขที่ได้ มาใช้การประมาณค่าสำหรับอนุพันธ์อันดับสูง เราสามารถเขียนอนุกรมเทย์เลอร์แบบไปข้างหน้าสำหรับ $f(x_{i+2})$ ในพจน์ของ $f(x_i)$ จะได้

$$f(x_{i+2}) = f(x_i) + f'(x_i)(2h) + \frac{f''(x_i)(2h)^2}{2!} + \cdots$$

จากสมการ

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)h^2}{2!} + \cdots$$

เราทำการคูณด้วย 2 แล้วนำไปลบกับสมการ $f(x_{i+2})$ จะได้

$$f(x_{i+2}) - 2f(x_{i+1}) = -f(x_i) + f''(x_i)h^2 + \cdots$$

จัดสมการใหม่จะได้

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} + O(h)$$

เรียกว่า ผลต่างสืบเนื่องข้างหน้า อันดับ 2 **(Second forward finite divided difference)**

$$f''(x_i) = \frac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2} + O(h)$$

เรียกว่า ผลต่างสืบเนื่องย้อนหลัง อันดับ 2 **(Second backward finite divided difference)**

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2} + O(h^2)$$

<u>ตัวอย่าง 1.6</u> จงหาค่าอนุพันธ์อันดับที่ 1 และร้อยละของค่าผิดพลาดสัมพัทธ์ของฟังก์ชันที่ x = 0.5 เมื่อ h = 0.5 และ h = 0.25

$$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$$

1) โดยวิธีผลต่างสืบเนื่องข้างหน้า
2) โดยวิธีผลต่างสืบเนื่องย้อนหลัง
3) โดยวิธีผลต่างสืบเนื่องตรงกลาง

<u>วิธีทำ</u>

ค่าจริงหาได้จากสมการ

$$f'(x) = -0.4x^3 - 0.45x^2 - 10x - 0.25$$

จะได้ค่าจริง $f'(0.5) = -0.9125$

เมื่อ h = 0.5

$$x_{i-1} = 0 \;;\quad f(x_{i-1}) = 1.2$$
$$x_i = 0.5 \;;\quad f(x_i) = 0.925$$
$$x_{i+1} = 1.0 \;;\quad f(x_{i+1}) = 0.2$$

1) โดยวิธีผลต่างสืบเนื่องข้างหน้า

$$f'(0.5) \approx \frac{0.2 - 0.925}{0.5} = -1.45$$
$$\varepsilon_t = -58.9\%$$

2) โดยวิธีผลต่างสืบเนื่องย้อนหลัง

$$f'(0.5) \approx \frac{0.925 - 1.2}{0.5} = -1.55$$
$$\varepsilon_t = 39.7\%$$

3) โดยวิธีผลต่างสืบเนื่องตรงกลาง

$$f'(0.5) \approx \frac{0.2 - 1.2}{1.0} = -1.0$$
$$\varepsilon_t = -9.6\%$$

เมื่อ h = 0.25

$$x_{i-1} = 0.25 \;;\quad f(x_{i-1}) = 1.10351563$$
$$x_i = 0.5 \;;\quad f(x_i) = 0.925$$
$$x_{i+1} = 0.75 \;;\quad f(x_{i+1}) = 0.63632813$$

1) โดยวิธีผลต่างสืบเนื่องข้างหน้า

$$f'(0.5) \approx \frac{0.63632813 - 0.925}{0.25} = -1.155$$
$$\varepsilon_t = -26.5\%$$

2) โดยวิธีผลต่างสืบเนื่องย้อนหลัง

$$f'(0.5) \approx \frac{0.925 - 1.10351563}{0.25} = -1.0714$$
$$\varepsilon_t = 21.7\%$$

3) โดยวิธีผลต่างสืบเนื่องตรงกลาง

$$f'(0.5) \approx \frac{0.63632813 - 1.10351563}{0.5} = -0.934$$

$$\varepsilon_t = -2.4\%$$

1.4 Error Propagation

เป็นค่าที่เกิดขึ้นเนื่องจากค่าที่นำมาคำนวณนั้นมีค่าผิดพลาดอยู่ด้วย เช่น การคูณตัวเลขจำนวน 2 ตัวเลข ซึ่งแต่ละจำนวนเป็นค่าประมาณ(มีค่าผิดพลาดรวมอยู่ด้วย) ทำให้ผลลัพธ์ที่ได้มีค่าผิดพลาดเกิดขึ้น

1.4.1 ฟังก์ชันที่ขึ้นกับตัวแปรเดียว (Functions of a Single Variable)

ถ้า $f(x)$ ขึ้นอยู่กับตัวแปลเดียวคือค่า x จะมี $\tilde{x}$ เป็นค่าประมาณกับ x โดยที่ความแตกต่างระหว่างค่าจริงและค่าประมาณของฟังก์ชันเขียนได้ดังนี้

$$\Delta f(\tilde{x}) = |f(x) - f(\tilde{x})|$$

ปัญหาที่เกิดขึ้นจะอยู่ที่การหาค่าของ $f(\tilde{x})$ โดยที่เราไม่รู้ค่าของ $f(x)$ เพราะ x นั้นเราไม่รู้ค่า จากที่ $\tilde{x}$ เข้าใกล้ค่า x และ $f(\tilde{x})$ เป็นฟังก์ชันต่อเนื่อง และหาอนุพันธ์ได้ เราก็หาค่าได้โดยใช้อนุกรมเทย์เลอร์

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

โดยเราตัดพจน์ของพจน์อนุพันธ์อันดับ 2 และ พจน์ที่สูงกว่า และจัดสมการใหม่จะได้

$$f(x) - f(\tilde{x}) = f'(x)(x - \tilde{x})$$

หรือ

$$\Delta f(\tilde{x}) = |f'(\tilde{x})|\Delta\tilde{x}$$

โดยที่ $\Delta f(\tilde{x}) = |f(x) - f(\tilde{x})|$ เป็นค่าประมาณของค่าผิดพลาดของฟังก์ชัน

<u>**ตัวอย่างที่ 1.7**</u> กำหนด $\tilde{x} = 2.5, \Delta\tilde{x} = 0.01$ ประมาณค่าผิดพลาดของฟังก์ชัน

$$f(x) = x^3$$

<u>**วิธีทำ**</u>

จาก $\Delta f(\tilde{x}) = |f'(\tilde{x})|\Delta\tilde{x}$

$$\Delta f(\tilde{x}) = 3(2.5)^2(0.01) = 0.1875$$

โดยที่ $f(2.5) = 15.625$

จะได้ค่าผิดพลาดคือ 15.625 ± 01875

หรือจะได้ค่าจริงอยู่ระหว่าง 15.4875 กับ 15.8125

1.4.2 ฟังก์ชันที่ขึ้นอยู่กับตัวแปรมากกว่าหนึ่งตัว(Function of More than One Variable)

ฟังก์ชันที่ขึ้นกับตัวแปลอิสระมากกว่าหนึ่งตัว โดยใช้ สมการหลายตัวแปรสำหรับอนุกรมเทย์เลอร์ ตัวอย่างเช่น ฟังก์ชันที่มีตัวแปรอิสระสองตัว คือ u และ v จากอนุกรมเทย์เลอร์ จะเขียนได้ว่า

$$f(u_{i+1}, v_{i+1}) = f(u_i, v_i) + \frac{\partial f}{\partial u}(u_{i+1} - u_i) + \frac{\partial f}{\partial v}(v_{i+1} - v_i) +$$

$$\frac{1}{2!}\left[\frac{\partial^2 f}{\partial u^2}(u_{i+1} - u_i)^2 + 2\frac{\partial^2 f}{\partial u \partial v}(u_{i+1} - u_i)(v_{i+1} - v_i) + \frac{\partial^2 f}{\partial v^2}(v_{i+1} - v_i)^2\right] + \cdots$$

เราจะตัดพจน์อนุกรมอันดับ 2 และพจน์ที่สูงกว่า เราจะได้

$$\Delta f(\tilde{u}, \tilde{v}) \approx \left|\frac{\partial f}{\partial u}\right|\Delta\tilde{u} + \left|\frac{\partial f}{\partial v}\right|\Delta\tilde{v}$$

เมื่อ $\Delta\tilde{u}, \Delta\tilde{v}$ เป็นค่าประมาณของค่าผิดพลาดใน u, v ตามลำดับ

สำหรับตัวแปรอิสระ x จำนวน n ตัว คือ $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n$ จะมีค่าผิดพลาด $\Delta\tilde{x}_1, \Delta\tilde{x}_2, \ldots, \Delta\tilde{x}_n$

จะได้

$$\Delta f(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n) \cong \left|\frac{\partial f}{\partial x_1}\right|\Delta\tilde{x}_1 + \left|\frac{\partial f}{\partial x_2}\right|\Delta\tilde{x}_2 + \cdots + \left|\frac{\partial f}{\partial x_n}\right|\Delta\tilde{x}_n$$

1.4.3 Stability and Condition

เราจะเห็นได้ว่าในการคำนวณนั้นจะทำให้เกิดค่าผิดพลาด ในการตัดเศษได้มาก ซึ่งเราจะเรียกว่าเป็นNumerically unstable จาก อนุกรมเทย์เลอร์อันดับที่ 1

$$f(x) \approx f(\tilde{x}) + f'(\tilde{x})(x - \tilde{x})$$

ในความสัมพันธ์นี้สามารถใช้ค่าประมาณ

ความสัมพันธ์ของค่าผิดพลาดของ $f(x)$ (relative error of $f(x)$) ใน

$$\varepsilon[f(x)] = \frac{f(x) - f(\tilde{x})}{f(\tilde{x})} \approx \frac{f'(\tilde{x})(x - \tilde{x})}{f(\vec{x})}$$

ความสัมพันธ์ของค่าผิดพลาดของ x (relative error of x) ใน

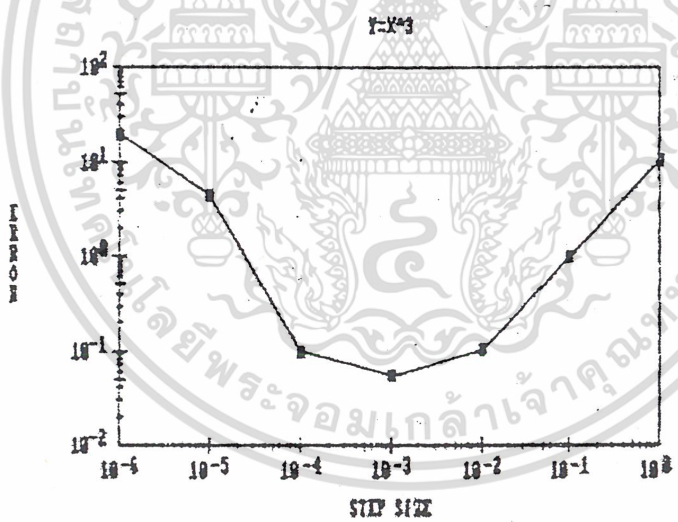
$$\varepsilon(x)=\frac{x-\tilde{x}}{\tilde{x}}$$

Condition number จะถูกกำหนดโดยสัดส่วนความสัมพันธ์ของค่าผิดพลาด

$$\frac{\varepsilon[f(x)]}{\varepsilon(x)}=\frac{f'(\tilde{x})(x-\tilde{x})}{f(\tilde{x})}\cdot\frac{\tilde{x}}{x-\tilde{x}}=\frac{\tilde{x}f'(\tilde{x})}{f(\tilde{x})}$$

1.5 ค่าผิดพลาดรวม(Total Numerical Error)

ค่าผิดพลาดรวมคือ ผลรวมของ ค่าผิดพลาดเพราะปัดเศษ และค่าผิดพลาดเพราะตัดปลาย โดยทั่วไปการลดค่าผิดพลาดเพราะปัดเศษทำได้โดยการเพิ่มเนื้อที่การเก็บในคอมพิวเตอร์ และค่าผิดพลาดเพราะตัดปลายจะลดลง เมื่อเราทำการลดค่า h โดยที่การลด h จะทำให้นำไปสู่การคำนวณที่ละเอียดขึ้น ทำให้ค่าผิดพลาดเพราะปัดเศษมากขึ้น ดังนั้นจะมีจุดหนึ่งที่การลดค่า h ไม่ทำให้ค่าผิดพลาดรวมลดลง ดังรูปที่ 1.5



รูป 1.5

2. Root of Equations (การหารากของสมการ)

2.1 Bracketing Methods

เป็นวิธีที่ต้องกำหนดช่วงหรือค่าเริ่มต้น 2 ค่า และลดความกว้างของช่วงลง โดยอาศัยความจริงที่ว่า ค่าของฟังก์ชันเครื่องหมายจะเปลี่ยนเมื่อเข้าใกล้ค่ารากสมการ

2.1.1 วิธีเชิงกราฟ

เป็นวิธีการง่าย ๆ ในการประมาณค่ารากสมการ โดยพล็อตกราฟระหว่างแกน x และ f(x) แล้วประมาณค่า x เมื่อ f(x) = 0

ตัวอย่างที่ 2.1 จงใช้วิธีเชิงกราฟเพื่อหาค่า drag coefficient c เมื่อนักกระโดดร่มมีมวล m = 68.1 กก. และจะมีความเร็ว v = 40 m/s หลังจากกระโดดลงมาเป็นเวลา t = 10 s (ค่าความโน้มถ่วง $g = 9.8\ m/s^2$)

วิธีทำ

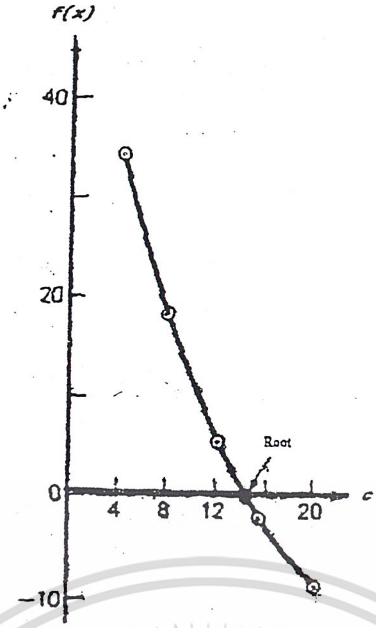
$$f(c) = \frac{gm}{c}\left(1 - e^{-\left(\frac{c}{m}\right)t}\right) - v$$

$$= \frac{9.8(68.1)}{c}\left(1 - e^{-\left(\frac{c}{68.1}\right)10}\right) - 40$$

หาค่า f(c) โดยกำหนดค่า c ดังนี้

c	f(c)
4	34.115
8	17.653
12	6.067
16	-2.269
20	-8.401

นำมาเขียนกราฟ ดังรูป 2.1



รูป 2.1

ประมาณค่ารากสมการอย่างคร่าว ๆ ได้เท่ากับ 14.75

$$f(14.75) = 0.059$$

$$v = \frac{9.8(68.1)}{14.75}\left(1 - e^{-\left(\frac{14.75}{68.1}\right)10}\right) = 40.059 m/s$$

ซึ่งใกล้เคียงกับ 40 m/s แต่วิธีเชิงกราฟเป็นวิธีประมาณค่ารากสมการ ไม่ละเอียดนัก

2.1.2 วิธี Bisection

หรือ Binary Chopping หรือ Interval Having หรือ Bolzano's Method

จากตัวอย่าง 2.1 ค่าฟังก์ชันจะเปลี่ยนเป็นเครื่องหมายตรงกันข้าม ถ้า f(x) เป็นจำนวนจริงและต่อเนื่องในช่วง x_l ถึง x_u และ $f(x_l)$ และ $f(x_u)$ มีเครื่องหมายตรงกันข้าม หรือ $f(x_l)f(x_u) < 0$ แล้วจะมีค่ารากสมการอย่างน้อยที่สุด 1 ค่า ระหว่าง x_l ถึง x_u

ซึ่งเป็นวิธีการหาค่ารากของสมการ โดยแบ่งเป็นช่วงย่อย ๆ ที่ละครึ่ง ปละเลือกช่วงที่ค่าฟังก์ชันมีเครื่องหมายตรงกันข้าม ซึ่งสามารถแบ่งเป็นขั้นตอนได้ดังนี้

ขั้นตอน 1 เลือก x_l และ x_u ซึ่งค่าฟังก์ชันมีเครื่องหมายแตกต่างกัน

$$f(x_l)f(x_u) < 0$$

ขั้นตอน 2 ประมาณค่ารากของสมการ x_r โดย

$$x_r = \frac{x_l + x_u}{2} \tag{2.1}$$

ขั้นตอน 3 ตรวจสอบเงื่อนไขต่อไปนี้

a) ถ้า $f(x_l)f(x_u)<0, \quad x_u = x_r \Rightarrow$ ขั้นตอน 2

b) ถ้า $f(x_l)f(x_u)>0, \quad x_l = x_r \Rightarrow$ ขั้นตอน 2

c) ถ้า $f(x_l)f(x_u)=0,$ ค่ารากสมการ $= x_r \Rightarrow$ ออกจากโปรแกรม

<u>**ตัวอย่าง 2.2**</u> จงใช้วิธี Bisection จากโจทย์ตัวอย่าง 2.1 (ค่ารากสมการคือ 14.7802) และหาค่า $|\varepsilon_t|, |\varepsilon_a|$ เมื่อกำหนด $|\varepsilon_s| = 0.5\%$

<u>**วิธีทำ**</u>

จากรูป 2.1 จะเป็นว่า ค่าฟังก์ชันเครื่องหมายเปลี่ยน ในช่วง 12 และ 16 ดังนั้น ให้

$$x_t = 12; \quad x_u = 16$$

$$x_r = \frac{12+16}{2} = 14$$

$$\varepsilon_t = \frac{14.7802-14}{14.7802} \times 100\% = 5.3\%$$

$$f(12)f(14) = 6.067(1.569) = 9.517 > 0$$

เพราะฉะนั้น $x_l = x_r$

$$x_r = \frac{14+16}{2} = 15$$

จะได้ $\varepsilon_t = -1.5\%$

$$f(14)f(15) = 1.569(-0.425) = -0.666$$

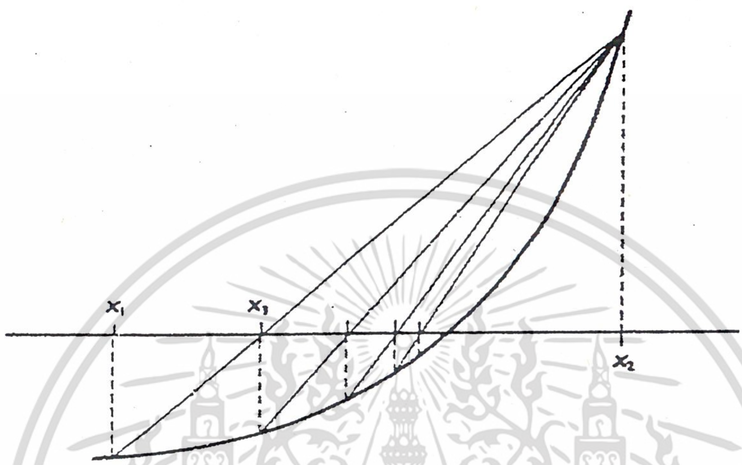
เพราะฉะนั้น $x_u = x_r$

ทำซ้ำจนกระทั่ง $|\varepsilon_a| < 0.5\%$ จะได้

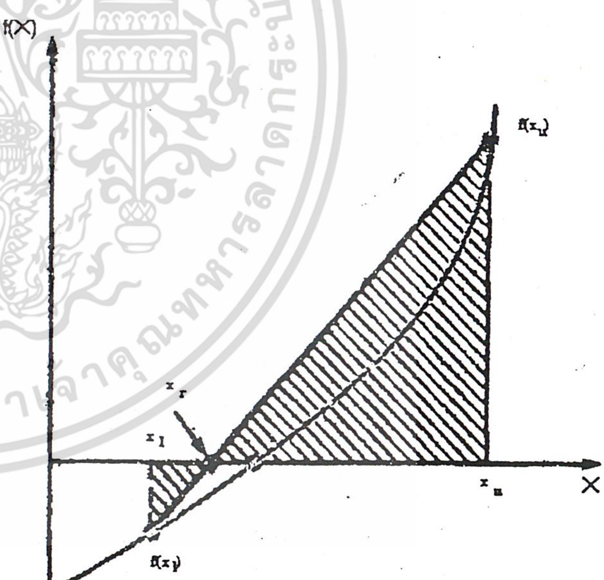
จำนวนครั้งที่ทำซ้ำ	x_l	x_u	x_r	$\lvert\varepsilon_a\rvert\%$	$\lvert\varepsilon_t\rvert\%$
1	12	16	14		5.279
2	14	16	15	6.667	1.487
3	14	15	14.5	3.448	1.896
4	14.5	15	14.75	1.695	0.204
5	14.75	15	14.875	0.840	0.641
6	14.75	14.875	14.8125	0.442	0.219

2.1.3 วิธี False – Position

วิธีนี้ก็จะพิจารณาค่า $f(x_l)$ และ $f(x_u)$ เช่น จากรูป 2.5 $f(x_l)$ มีค่าเข้าใกล้ศูนย์มากกว่า $f(x_u)$ ค่ารากสมการ ก็ควรจะอยู่ใกล้ค่า x_l มากกว่า x_u ถ้าลากเส้นเชื่อมต่อระหว่าง $f(x_l)$ กับ $f(x_u)$ ก็จะได้จุดที่ตัดแกน x ซึ่งเป็นที่มาของคำว่า false position หรือ บางทีเรียก **linear interpolation method**



รูป 2.2a



รูป 2.2b

จากรูป 2.2 b ตามกฎของสามเหลี่ยม 2 รูปคล้าย จะได้

$$\frac{f(x_l)}{x_r - x_l} = \frac{f(x_u)}{x_r - x_u}$$

$$\boxed{x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}} \qquad (2.2)$$

ขั้นตอนจะคล้ายกับ Bisection แต่ขั้นตอน 2 สูตรการหาค่า x_r จะเปลี่ยนไป

ตัวอย่าง 2.3 จงใช้วิธี false-position เพื่อหาค่ารากสมการ จากตัวอย่าง 2.1

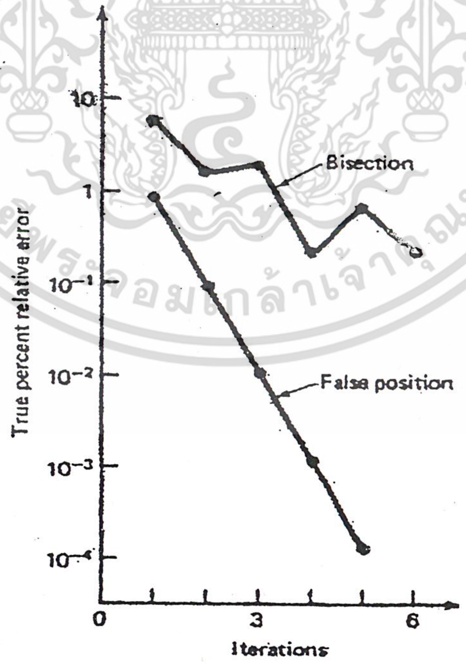
วิธีทำ

ครั้งแรก

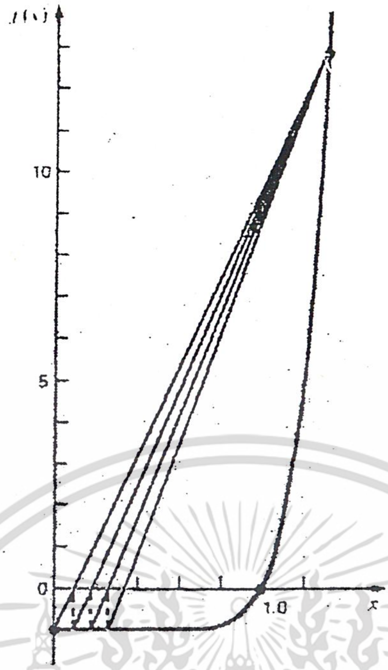
$$x_l = 12; \quad x_u = 16$$
$$f(x_l) = 6.0669$$
$$f(x_u) = -2.2688$$
$$x_r = 16 - \frac{-2.2688(12-16)}{6.0669-(-2.2688)} = 14.9113$$
$$\varepsilon_t = 0.89\%$$

ครั้งที่สอง

$$f(x_l)f(x_r) = -1.5426$$
$$x_l = 12; \quad x_u = 14.9113$$
$$f(x_l) = 6.0668; \quad f(x_u) = -0.2543$$
$$x_r = 14.9113 - \frac{-2.2543(12-14.9113)}{6.0669-(-0.2543)} = 14.7942$$
$$\varepsilon_t = 0.09\%$$



รูป 2.3 แสดงการเปรียบเทียบร้อยละของค่าผิดพลาดสัมพันธ์ เมื่อใช้วิธี Bisection และ false-position ในการทำซ้ำมากขึ้น



รูป 2.4 แสดง Pitfall ของวิธี false-position ค่า $f(x_l)$ เข้าใกล้ศูนย์มากกว่า $f(x_u)$ ค่ารากสมการควรจะอยู่ใกล้ x_l มากกว่า x_u แต่จากรูปค่ารากสมการจะอยู่ใกล้ x_u มากกว่าใกล้ x_l ถ้าใช้วิธี false-position คำนวณหาค่ารากสมการจะต้องทำซ้ำซากหลายครั้งมาก

2.2 Open Methods

เป็นวิธีที่ต้องการทราบ x เพียงค่าเดียว บางทีอาจจะลู่ออก (divergence) หรือค่าไม่เข้าใกล้ค่าจริงเมื่อมีการทำซ้ำมากขึ้น แต่ถ้าลู่เข้าแล้ว (Convergence) จะเข้าใกล้ค่าจริงเร็วกว่าวิธี Bracketing

2.2.1 วิธี Simple One-Point Iteration

ในการหาค่ารากสมการ จะให้ f(x)=0 ซึ่งสามารถเขียนในรูป x = g(x) ได้

เช่น

$$x^2 - 2x + 3 = 0$$

$$x = \frac{x^2 + 3}{2}$$

ซึ่งสามารถหาค่าประมาณ เมื่อกำหนดค่าเริ่มต้น x_i คือ $x_{i+1} = g(x_i)$

$$|\varepsilon_a| = \left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| \times 100\% \qquad (2.3)$$

ตัวอย่าง 2.4 จงใช้วิธี Simple One-Point Iteration เพื่อหาค่ารากของ สมการของ $f(x) = e^{-x} - x$ (ค่าจริงคือ 0.56714329)

วิธีทำ

$$f(x) = 0$$

$$e^{-x} - x = 0$$

$$x_{i+1} = e^{-xi}$$

กำหนดค่าเริ่มต้น $x_0 = 0$

$$x_1 = e^{-x_0} = e^0 = 1$$

$$x_2 = e^{-x_1} = \frac{1}{e} = 0.367879$$

$$|\varepsilon_a| = \frac{0.367879 - 1}{0.367879} = 171.5$$

$$x_3 = e^{-x_2} = e^{-0.367879}$$

การลู่เข้า (Convergence)

Two-Curve Graphical Method เป็นวิธีการเขียนกราฟในที่นี้จะนำมาใช้อธิบายลักษณะการลู่เข้า และลู่ออก เช่น

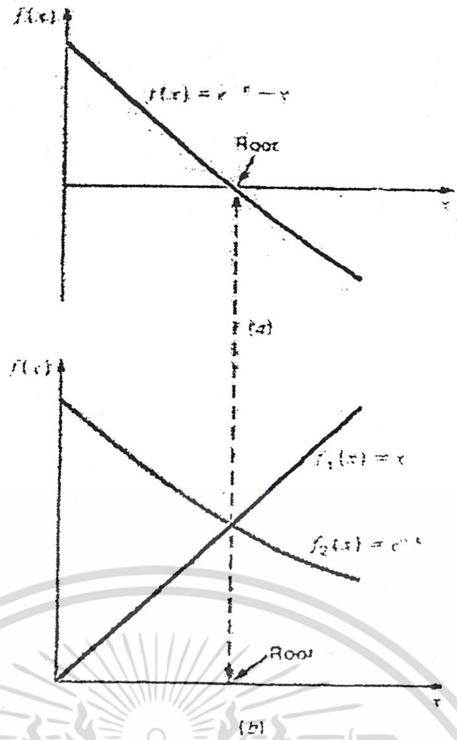
จากสมการ $e^{-x} - x = 0$

$$x = e^{-x}$$

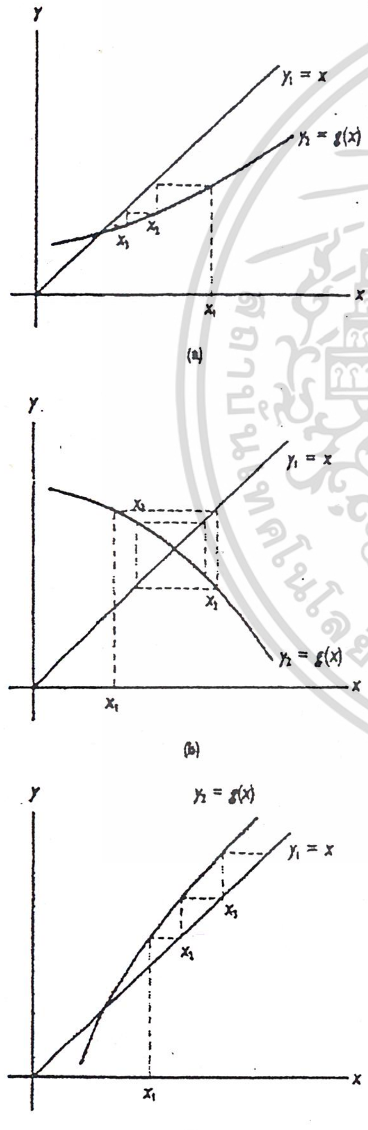
ให้ $f_1(x) = x$

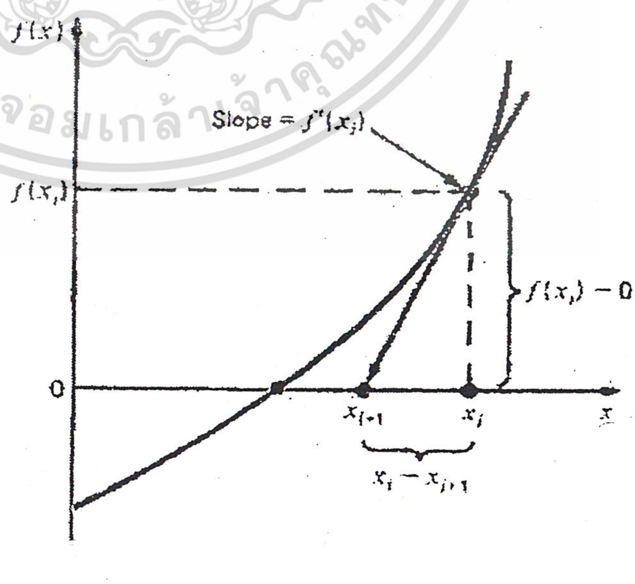
$$f_2(x) = e^{-x}$$

เมื่อนำ 2 สมการมาเขียนกราฟ จะได้ดังรูป 2.8 ค่าสมการก็คือ ค่า x ที่ทำให้ $f_1(x) = f_2(x)$ หรือค่า x ที่ตัดจุดตัดกันของกราฟ 2 เส้น



รูป 2.5





รูป 2.6

รูป 2.7

รูป 2.6 a และ b เป็นลักษณะการลู่เข้า ส่วนรูป 2.9c เป็นลักษณะการลู่ออก

2.2.2 วิธี Newton-Raphson

เป็นวิธีที่ใช้กันอย่างกว้างขวาง จากรูป 2.10 จะได้

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

จะได้

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \qquad (2.4)$$

สมการที่ 2.4 เรียกว่าสูตร Newton-Rephson

ตัวอย่าง 2.5 จงใช้วิธี Newton-Rephson เพื่อหารากสมการโดยประมาณของ $e^{-x} - x$ โดยกำหนดค่าเริ่มต้น $x_0 = 0$

วิธีทำ

จาก $f'(x) = -e^{-x} - 1$

โดยใช้ $x_{i+1} = x_i - \left[\frac{e^{-x} - x}{-e^{-x} - 1}\right]$

เมื่อ $x_0 = 0$

จะได้

$$x_1 = x_0 - \left[\frac{e^{-x_0} - x_0}{-e^{-x_0} - 1}\right]$$

$$= x_0 - \left[\frac{e^0 - 1}{-e^0 - 1}\right]$$

$$= 0 - \frac{1}{(-2)} = \frac{1}{2}$$

2.2.3 วิธีเซแคนต์(Secant Method)

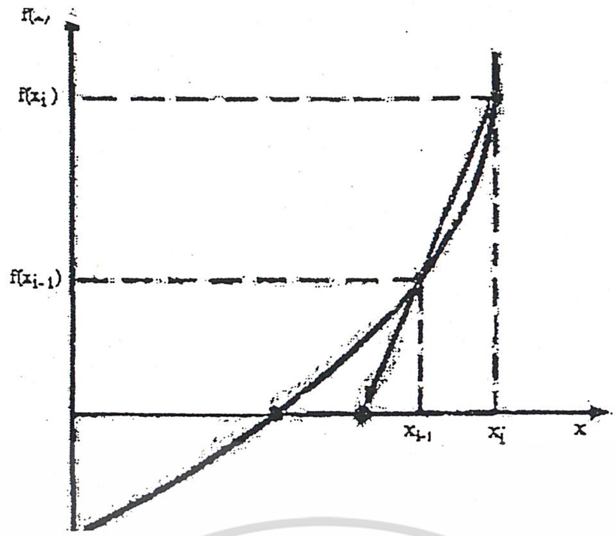
จาก finite divided difference

$$f'(x) = \frac{f(x_{i-1}) - f(x_i)}{x_{i-1} - x_i}$$

แทนค่าในสูตร Newton-Raphson

$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)} \qquad (2.5)$$

สมการ 2.5 เรียกว่าสูตร เซแคนต์



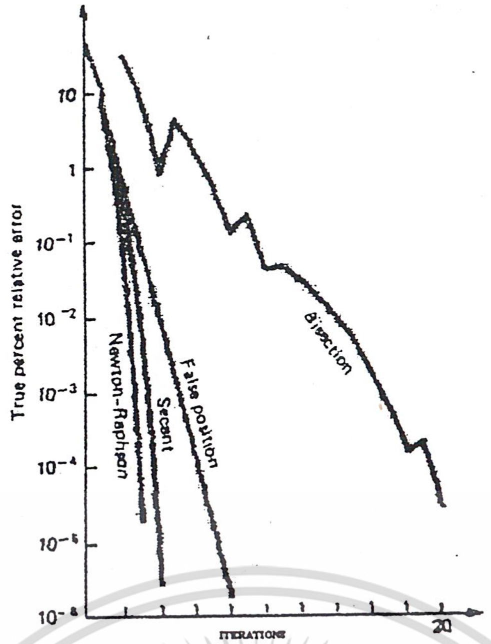
รูป 2.8

ตัวอย่าง 2.6 จงใช้วิธีเซแคนต์ เพื่อหาค่ารากสมการ โดยประมาณของ $f(x)=e^{-x}-x$ โดยกำหนดค่าเริ่มต้น $x_{-1}=0$ และ $x_0=1.0$ (ค่าจริงคือ 0.56714329)

วิธีทำ

ครั้งแรก

$$x_{-1}=0; \quad f(x_{-1})=e^{-0}-0=1$$
$$x_0=1; \quad f(x_0)=e^{-1}-1=-0.63212$$
$$x_1=x_0-\frac{f(x_0)(x_{-1}-x_0)}{f(x_{-1})-f(x_0)}$$
$$x_1=1-\frac{-0.56714329-0.61270}{0.56714329}$$
$$=0.61270$$
$$|\varepsilon_1|=\frac{0.56714329-0.61270}{0.56714329}\times 100\% =0.8\%$$



รูป 2.9

2.2.4 Multiple Roots

คือจุดที่ฟังก์ชันสัมผัสกับแกน x ตัวอย่าง เช่น

$$f(x) = (x-3)(x-1)(x-1)$$

หรือถ้านำแต่ละเทอมคูณกันจะได้

$$f(x) = x^3 - 5x^2 + 7x - 3$$

มีค่ารากสมการ 2 ค่า คือ 3 และ 1 โดยที่ 1 นั้นจะมีค่า 2 ค่า (Double Root) หรือ

$$f(x) = (x-3)(x-1)(x-1)(x-1)$$

จะมีค่า 1 จำนวน 3 ค่า (Triple Root)

โดยทั่วไปแล้ว ถ้าจำนวนรากที่ซ้ำกันของฟังก์ชันเป็นเลขคี่ ฟังก์ชัน จะตัดกับแกน x ถ้าเป็นเลขคู่ จะไม่ตัดกับแกน x

ให้ $$u(x) = \frac{f(x)}{f'(x)}$$

จากวิธี Newton-Raphson

$$x_{i+1} = x_i - \frac{u(x_i)}{u'(x_i)}$$

$$u'(x) = \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2}$$

$$x_{i+1} = x_i - \frac{f(x_i)f'(x_i)}{(f(x_i))^2 - f(x_i)f''(x_i)} \qquad (2.6)$$

สมการ 2.6 เรียกว่า Modified Newton-Raphson Method

ตัวอย่าง 2.7 จงใช้วิธี Newton-Raphson และวิธี Modified Newton-Raphson เพื่อหาค่า Multiple Root ของสมการ

$$f(x) = x^3 - 5x^2 + 7x - 3$$

1) เมื่อกำหนดค่าเริ่มต้น $x_0 = 0$

2) เมื่อกำหนดค่าเริ่มต้น $x_0 = 4$

วิธีทำ

1) ค่าเริ่มต้น $x_0 = 0$, วิธี Newton-Raphson

$$f'(x) = 3x^2 - 10x + 7$$

$$x_{i+1} = x_i - \frac{x_i^3 - 5x_i^2 + 7x_i - 3}{3x_i^2 - 10x_i + 7}$$

จะได้ผลลัพธ์ดังนี้

I	x_1	$\lvert\varepsilon_t\%\rvert$
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

วิธี Modified Newton-Raphson

$$x_{i+1} = x_i - \frac{(x_i^3 - 5x_i^2 + 7x_i - 3)(3x_i^2 - 10x_i + 7)}{(3x_i^2 - 10x_i + 7)^2 - (x_i^3 - 5x_i^2 + 7x_i - 3)(6x_i - 10)}$$

จะได้ผลลัพธ์ดังนี้

I	x_1	$\|\varepsilon_t\%\|$
0	0	100
1	1.105263158	11
2	1.003081664	0.31
3	1.000002382	0.00024

จาก 2 วิธี จะเห็นว่าวิธี Modified Newton-Raphson ค่าผิดพลาดจะลดลงได้เร็วกว่า เนื่องจาก x = 1 เป็น Double root

2) ค่าเริ่มต้น $x_0 = 4$ โดยวิธี Newton-Raphson และ Modified Newton-Raphson จะได้ผลลัพธ์ดังนี้

I	Newton-Raphson ($\|\varepsilon_t\%\|$)	Modified Newton-Raphson ($\|\varepsilon_t\%\|$)
0	4(33)%	4(33%)
1	3.4(13%)	2.636363637(12%)
2	3.1(3.3%)	2.820224720(6%)
3	3.008695652(0.29%)	2.961728211(1.3%)
4	3.000074641(2.5x10^{-3}%)	2.998478719(0.051)
5	3.000000006(2x10^{-7}%)	2.999997682(7.7x10^{-5}%)

ทั้ง 2 วิธี Newton-Raphson ค่าผิดพลาดจะลดลงเร็วกว่า แต่ก็มีอัตราไม่แตกต่างกันมากนัก แต่ในกรณีนี้เป็น single root คือค่า 3 เพียงค่าเดียว ถ้าเป็น single root การใช้วิธี Newton-Raphson จะมีประสิทธิภาพมากกว่า

2.3 การหาค่ารากของระบบสมการ

วิธีที่กล่าวในตอนต้นเป็นการหาค่ารากสมการของสมการเดียว แต่ถ้าต้องการหาค่ารากของสมการ ซึ่งอยู่ในรูป

$$f_1(x_1, x_2, x_3, \ldots, x_n) = 0$$
$$f_2(x_1, x_2, x_3, \ldots, x_n) = 0$$
$$f_3(x_1, x_2, x_3, \ldots, x_n) = 0$$
$$\vdots \quad \vdots \quad \vdots \quad \vdots$$
$$f_n(x_1, x_2, x_3, \ldots, x_n) = 0$$

เช่น

$$u(x, y) = x^2 + xy - 10 = 0$$
$$v(u, v) = y + 3xy^2 - 57 = 0$$

จะต้องหาค่า x,y ที่ทำให้

$$u(x, y) = 0 \text{ และ } v(x, y) = 0$$

2.3.1 วิธี One – Point Iteration

จะใช้แก้ปัญหา เมื่อ 2 สมการ ไม่เชิงเส้น

ตัวอย่าง 2.8 จงใช้วิธี One-point iteration

$$x^2 + xy - 10 = 0$$
$$y + 3xy^2 - 57 = 0$$

(ค่าจริงคือ x =2 และ y = 3)

กำหนดค่าเริ่มต้น x = 1.5 และ y = 3.5

วิธีทำ

$$x_{i+1} = \frac{10 - x_i^2}{y_i}$$
$$y_{i+1} = 57 - 3x_i y_i^2$$

ครั้งแรก

$$x_1 = \frac{10 - (1.5)^2}{3.5} = 2.21429$$
$$y_1 = 57 - 3(2.21429)(3.5)^2 = -24.37516$$

ครั้งสอง

$$x_2 = \frac{10 - (2.21429)^2}{-24.37516} = -0.20910$$
$$y_2 = 57 - 3(-0.20910)(-24.37516)^2 = 429.709$$

จากทั้งสองครั้งเราจะเห็นได้ว่า ค่าที่หาได้จะออกห่างจากค่าจริง

ถ้าเขียนสมการใหม่เป็น

$$x_{i+1} = \sqrt{10 - x_i y_i}$$

$$y_{i+1} = \sqrt{\frac{57 - y_i}{3x_i}}$$

ครั้งแรก

$$x_1 = \sqrt{10 - 1.5(3.5)} = 2.17945$$

$$y_1 = \sqrt{\frac{57 - 3.5}{3(2.17945)}} = 2.86051$$

ครั้งสอง

$$x_2 = \sqrt{10 - 2.17945(2.86051)} = 1.94053$$

$$y_2 = \sqrt{\frac{57 - 2.86051}{3(1.94053)}} = 3.04955$$

จากทั้งสองครั้ง ถ้าเราแทนด้วยสมการใหม่นี้ ค่าจะเข้าใกล้เคียงค่าจริง คือ x = 2 และ y = 3.5

2.3.2 วิธี NEWTON-RAPHSON

จากอนุกรมเทย์เลอร์อันดับ 1

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i)$$

จัดรูปใหม่ได้

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{2.7}$$

สมการที่ 2.7 เรียกว่า **Single-Equation form of the Newton-Raphson Method**

สำหรับ 2 ตัวแปล อนุกรมเทย์เลอร์อันดับหนึ่ง สามารถเขียนได้เป็น

$$u_{i+1} = u_i + (x_{i+1} - x_i)\frac{\partial u_i}{\partial x} + (y_{i+1} - y_i)\frac{\partial u_i}{\partial y}$$

$$v_{i+1} = v_i + (x_{i+1} - x_i)\frac{\partial v_i}{\partial x} + (y_{i+1} - y_i)\frac{\partial v_i}{\partial y}$$

เมื่อ u_{i+1} และ v_{i+1} จะมีค่าเท่ากับศูนย์ ดังนั้นเราจะจัดสมการใหม่ได้เป็น

$$\frac{\partial u_i}{\partial x} x_{i+1} + \frac{\partial u_i}{\partial y} y_{i+1} = -u_i + x_i \frac{\partial u_i}{\partial x} + y_i \frac{\partial u_i}{\partial y}$$

$$\frac{\partial v_i}{\partial x} x_{i+1} + \frac{\partial v_i}{\partial y} y_{i+1} = -v_i + x_i \frac{\partial v_i}{\partial x} + y_i \frac{\partial v_i}{\partial y}$$

จะได้

$$x_{i+1} = x_i - \frac{u_i \dfrac{\partial v_i}{\partial y_i} - v_i \dfrac{\partial u_i}{\partial y_i}}{\dfrac{\partial u_i}{\partial x}\dfrac{\partial v_i}{\partial y} - \dfrac{\partial u_i}{\partial y}\dfrac{\partial v_i}{\partial x}} \tag{2.8}$$

และ

$$y_{i+1} = y_i - \frac{u_i \dfrac{\partial v_i}{\partial x_i} - v_i \dfrac{\partial u_i}{\partial x_i}}{\dfrac{\partial u_i}{\partial x}\dfrac{\partial v_i}{\partial y} - \dfrac{\partial u_i}{\partial y}\dfrac{\partial v_i}{\partial x}} \tag{2.9}$$

โดยที่ $\dfrac{\partial u_i}{\partial x}\dfrac{\partial v_i}{\partial y} - \dfrac{\partial u_i}{\partial y}\dfrac{\partial v_i}{\partial x}$ จะเรียกว่า **Determinant of Jacobian**

จากสมการ 2.8,2.9 เรียกว่า **Two-Equation form of the Newton-Raphson Method**

ตัวอย่าง 2.9 จงใช้วิธี Two-Equation Newton-Raphson เพื่อหารากสมการ

$x^2 + xy - 10 = 0$

$y + 3xy^2 - 57 = 0$

กำหนดค่าเริ่มต้น x = 1.5 และ y = 3.5

วิธีทำ

$$\frac{\partial u}{\partial x} = 2x + y = 2(1.5) + 3.5 = 6.5$$

$$\frac{\partial u}{\partial y} = x = 15$$

$$\frac{\partial v}{\partial x} = 3y^2 = 3(3.5)^2 = 36.75$$

$$\frac{\partial v}{\partial y} = 1 + 6xy = 1 + 6(1.5)(3.5) = 32.5$$

ค่าของ Determinant of Jacobian คือ

$$6.5(32.5) - 1.5(36.75) = 156.15$$

โดยที่แทนค่าเริ่มต้นจะได้

$$u_0 = (1.5)^2 + 1.5(3.5) - 10 = -2.5$$

$$v_0 = 3.5 + 3(1.5)(3.5)^2 - 57 = 1.625$$

แทนค่าในสมการ 2.8,2.9 จะได้

$$x_1 = 1.5 - \frac{-25(32.5) - 1.625(1.5)}{156.125} = 2.03603$$

$$y_1 = 3.5 + \frac{-2.5(36.75) - 1.625(6.5)}{156.125} = 2.84388$$

จะเข้าใกล้ค่าจริงคือ x = 2 และ y = 3

3. System of Linear Algebraic Equation
(การหาค่าอินทิเกรตเชิงตัวเลข)

3.1 NEWTON-COTES INTEGRATION

Newton-Cotes Integration เป็นวิธีพื้นฐานของการอินทิเกรตเชิงตัวเลข โดยมีหลักการพื้นฐานว่าจะแทนฟังก์ชันที่ซับซ้อนด้วยการประมาณฟังก์ชันที่ง่ายต่อการอินทิเกรต เขียนสมการแทนได้ดังนี้

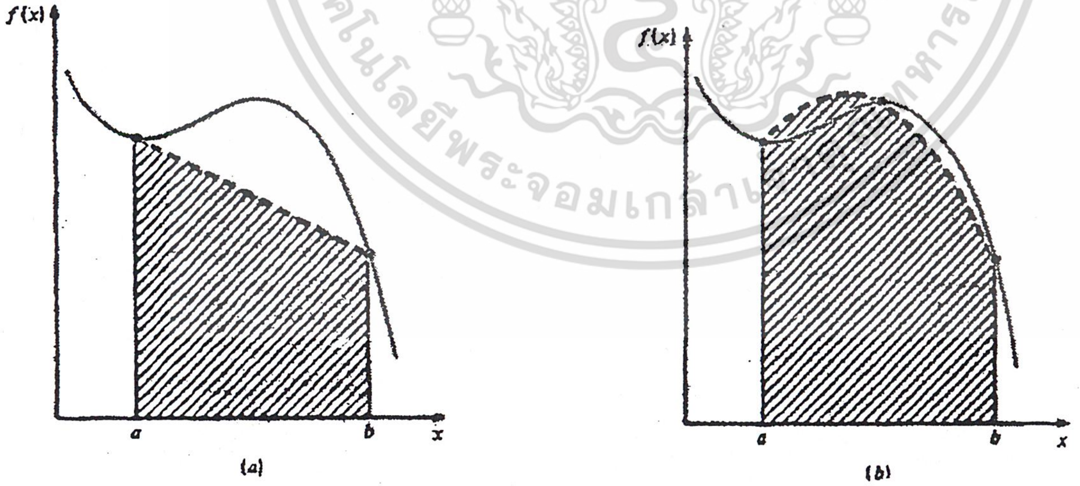
$$I = \int_a^b f(x)dx \approx \int_a^b f_n(x)dx \qquad (3.1)$$

ซึ่ง $f_n(x)$ **คือ ฟังก์ชันพหุนาม (Polynomial Function)** ที่อยู่ในรูปของ

$$f_n(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1} + a_n x^n$$

เมื่อ n คือ **อันดับของฟังก์ชันพหุนาม**

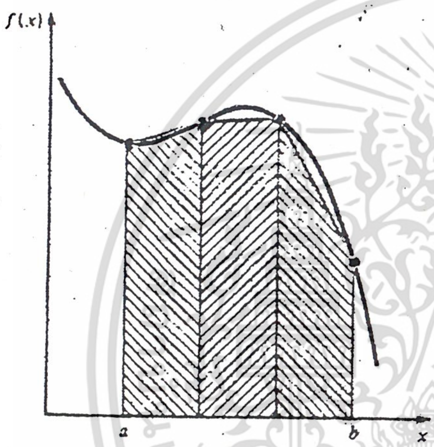
ตัวอย่าง เช่น รูป 3.1a ใช้ฟังก์ชันพหุนามอันดับหนึ่ง (สมการเส้นตรง) ในการประมาณ
รูป 3.1b ใช้ฟังก์ชันพหุนามอันดับสอง (สมการพาราโบลา) ในการประมาณ



รูปที่ 3.1

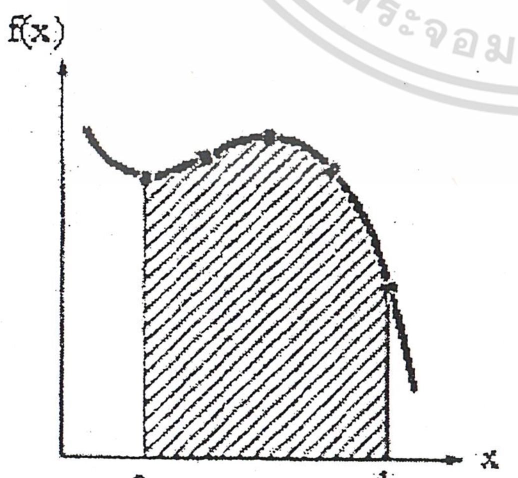
ค่าอินทิเกรตสามารถประมาณโดยการแบ่งเป็นเซกเมนต์ๆ ด้วยช่วงที่เท่าๆ กัน ตัวอย่างเช่น รูป 3.2a แบ่งเป็น 3 เซกเมนต์ แต่ละเซกเมนต์ประมาณด้วยฟังก์ชันที่อยู่ในรูปเส้นตรง จากพื้นฐานนี้เอง ถ้าใช้ฟังก์ชันพหุนามอันดับศูนย์ (ฟังก์ชันเป็นค่าคงที่) และแบ่งเป็นหลายๆ เซกเมนต์ ก็จะเป็นดังรูป 3.2b

Newton-Cotes Integration สามารถหาอินทิเกรตได้ทั้งช่วงเปิดและช่วงปิด (Close and Open Forms) ช่วงปิด ก็คือรู้จุดเริ่มต้นและจุดสิ้นสุดของการอินทิเกรต ดังรูป 3.3a ช่วงเปิด จะเป็นการหาอินทิเกรตที่ไม่มีจุดสิ้นสุด ดังรูป 3.3b ในบทนี้จะกล่าวถึงช่วงปิดเท่านั้น

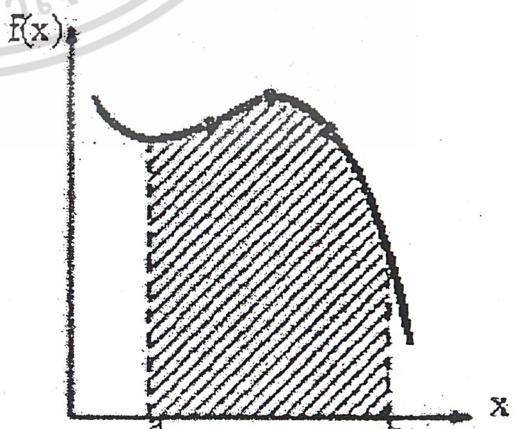


รูป 3.2a

รูป 3.2b



รูป 3.3a



รูป 3.3b

3.1.1 วิธี TRAPEZOIDAL RULE

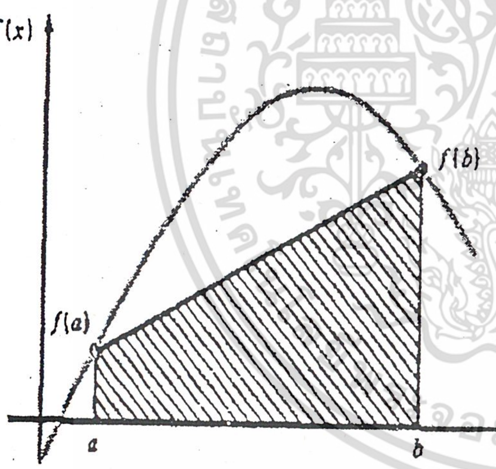
วิธี Trapezoidal Rule เป็น First Newton-cotes Closed Integration โดยการประมาณฟังก์ชันด้วย ฟังก์ชันพหุนามอันดับหนึ่ง จากสมการ 3.1 จะได้

$$I = \int_a^b f(x)dx \approx \int_a^b f_1(x)dx \qquad (3.2)$$

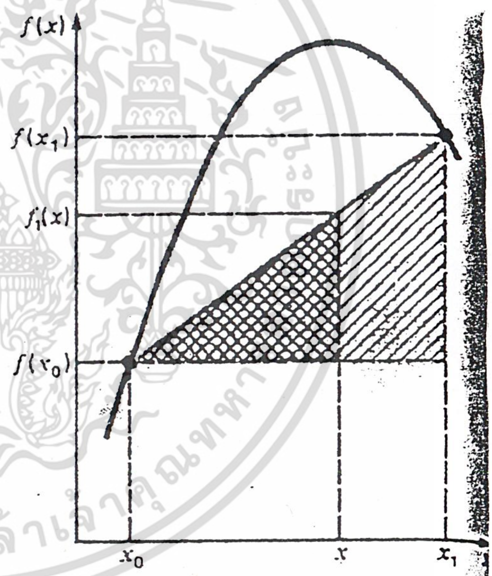
จากรูป 3.4b คุณสมบัติของสามเหลี่ยมคล้ายกัน จะได้

$$\frac{f_1(x) - f(a)}{x - a} = \frac{f(b) - f(a)}{b - a}$$

$$f_1(x) = f(a) + \frac{f(a) - f(b)}{b - a}(x - a) \qquad (3.3)$$



รูป 3.4a



รูป 3.4b

พื้นที่ใต้เส้นตรงก็จะเป็นการประมาณค่าอินทิเกรตของฟังก์ชัน f(x) ในช่วง a และ b

$$I = \int_a^b f(x)dx \approx \int_a^b f_1(x)dx \approx \int_a^b \left[f(a) + \frac{f(b) - f(a)}{b - a}(x - a) \right] dx$$

จาก

$$f_1(x) = f(a) + \frac{f(a) - f(b)}{b - a}(x - a)$$

$$f_1(x) = \frac{f(b) - f(a)}{b - a}x + f(a) - \frac{af(b) - af(a)}{b - a}$$

$$f_1(x) = \frac{f(b) - f(a)}{b - a}x + \frac{bf(a) - af(a) - af(b) + af(a)}{b - a}$$

$$f_1(x) = \frac{f(b) - f(a)}{b - a}x + \frac{bf(a) - af(b)}{b - a}$$

อินทิเกรตในช่วง x=a และ x=b จะได้

$$I \approx \frac{f(b) - f(a)}{b - a}\frac{x^2}{2} + \frac{bf(a) - af(b)}{b - a}x \Big|_a^b$$

$$I \approx \frac{f(b) - f(a)}{b - a}\frac{(b^2 - a^2)}{2} + \frac{bf(a) - af(b)}{b - a}(b - a)$$

เนื่องจาก $b^2 - a^2 = (b - a)(b + a)$

$$I \approx [f(b) - f(a)]\frac{(b + a)}{2} + bf(a) - af(b)$$

$$\approx \frac{(b + a)}{2}f(b) - \frac{(b + a)}{2}f(a) + bf(a) - af(b)$$

$$\approx f(a)\left(b - \frac{b + a}{2}\right) + f(b)\left(\frac{b + a}{2} - a\right)$$

$$\approx f(a)\left(\frac{b - a}{2}\right) + f(b)\left(\frac{b - a}{2}\right)$$

$$I \approx (b - a)\frac{f(a) + f(b)}{2} \qquad (3.4)$$

สมการ 3.4 เรียกว่า **Trapezoidal Rule**

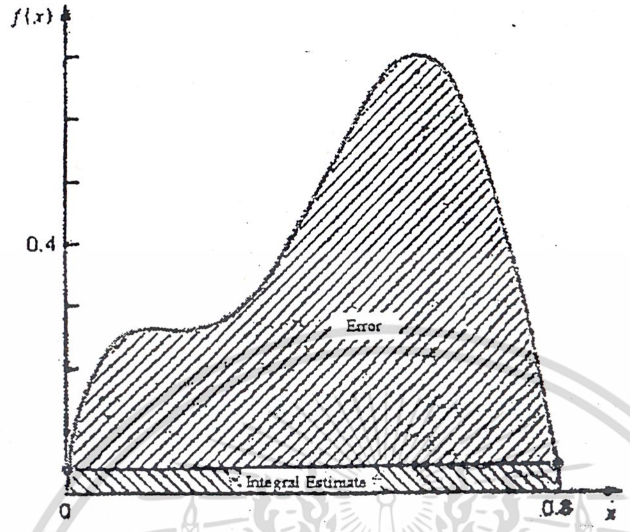
จะเห็นว่า Trapezoidal Rule ก็คือการประมาณพื้นที่ใต้เส้นตรงที่เชื่อมระหว่าง f(a) และ f(b) ดังรูป 3.4 จากสมการ 3.4 จะเห็นว่าเป็นสูตรของการหาพื้นที่ คือ

$I \approx$ ความกว้าง $\times$ ความสูงเฉลี่ย

หรือ $I \approx$ (b-a) $\times$ ความสูงเฉลี่ย (3.5)

3.1.1.1 ค่าผิดพลาดของวิธี Trapezoidal Rule

การประมาณค่าอินทิเกรตโดยการหาพื้นที่ใต้เส้นตรง ทำให้เกิดค่าผิดพลาด แสดงดังรูป 3.5



รูป 3.5

ซึ่งค่าผิดพลาดเพราะตัดปลาย ที่ได้จากการใช้วิธี Trapezoidal Rule คือ

$$E_t = \frac{-1}{12} f''(\xi)(b-a)^3 \qquad (3.6)$$

ซึ่ง ξ อยู่ระหว่าง a และ b จากสมการ 3.6 นี้ จะเห็นว่าถ้าฟังก์ชันเป็นเส้นตรง โดยวิธี Trapezoidal Rule จะได้ค่าที่ถูกต้อง (เพราะว่า $f''(\xi) = 0$)

ในการหาค่า $f''(\xi)$ จะใช้วิธีการหาค่าเฉลี่ย เขียนแทนด้วย $\overline{f}''$

ดังนั้นจะได้

$$\boxed{\overline{f}'' = \frac{\int_a^b f'' dx}{b-a}}$$

จะได้

$$\boxed{E_a = \frac{-1}{12} \overline{f}'' (b-a)^3}$$

<u>**ตัวอย่าง3.1**</u> จงใช้วิธี Trapezoidal Rule เพื่อประมาณค่าอินทิเกรตของ

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

ช่วง a =0 ถึง b = 0.8 และประมาณค่าผิดพลาด เมื่อค่าอินทิเกรตจริง คือ 1.64053334

<u>**วิธีทำ**</u> จากโจทย์จะได้

$$f(0) = 0.2 + 25(0) - 200(0)^2 + 675(0)^3 - 900(0)^4 + 400(0)^5$$
$$= 0.2$$

$$f(0.8) = 0.2 + 25(0.8) - 200(0.8)^2 + 675(0.8)^3 - 900(0.8)^4 + 400(0.8)^5$$
$$= 0.232$$

จากสมการ 3.4 จะได้

$$I \approx (0.8-0)\frac{f(0)+f(0.8)}{2}$$
$$\approx 0.8\frac{0.2+0.232}{2} = 0.1728$$

$$E_t = 1.64053334 - 0.1728 = 1.46773334$$
$$\varepsilon_t = \frac{1.64053334 - 0.1728}{1.64053334} \times 100 = 89.5\ \%$$
$$f''(x) = -400 + 4050x - 10800x^2 + 8000x^3$$

ค่าเฉลี่ยของอนุพันธ์อันดับที่ 2 สามารถคำนวณได้ดังนี้

$$\bar{f}'' = \frac{\int_0^{0.8}(-400 + 4050x - 10800x^2 + 8000x^3)dx}{0.8-0}$$

$$= \frac{-400x + \frac{4050x^2}{2} - \frac{10800x^3}{3} + \frac{8000x^4}{4}\Big|_0^{0.8}}{0.8} = -60$$

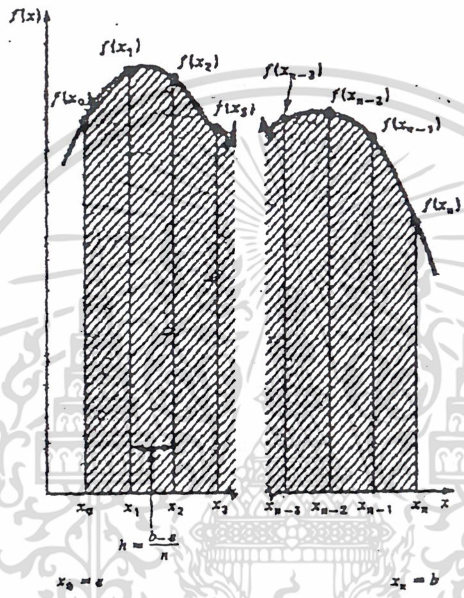
$$E_a = -\frac{1}{12}\bar{f}''(b-a)^3$$
$$E_a = -\frac{1}{12}(-60)(0.8-0)^3 = 2.56$$

3.1.2 วิธี MUTIPLE – APPLICATION TRAPEZOIDAL RULE

วิธีหนึ่งที่จะเพิ่มความถูกต้องเมื่อใช้วิธี Trapezoidal Rule คือ แบ่งช่วงการอินทิเกรตเป็นหลายๆ เซกเมนต์ ดังรูป 3.6 ค่าอินทิเกรตก็คือผลบวกของการอินทิเกรตแต่ละเซกเมนต์ ซึ่งเราเรียกสมการนี้ว่า **mutiple-application หรือ composite integration formulas**

รูป 3.6 แสดงการแบ่งเป็น n ช่วง โดยแต่ละช่วงมีความกว้างเท่ากัน ซึ่งจะมี n+1 จุด คือ $x_0, x_1, x_2, \ldots, x_n$ ดังนั้นแต่ละช่วงจะมีความกว้าง

$$h = \frac{b-a}{n} \tag{3.7}$$



รูป 3.6

จะได้ว่า

$$I = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \int_{x_2}^{x_3} f(x)dx + \ldots + \int_{x_{n-1}}^{x_n} f(x)dx$$

แทนค่าด้วย trapezoidal rule

$$I \approx h\frac{f(x_0)+f(x_1)}{2} + h\frac{f(x_1)+f(x_2)}{2} + \ldots + h\frac{f(x_{n-1})+f(x_n)}{2} \tag{3.8}$$

หรือ

$$I \approx \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n) \right] \tag{3.9}$$

หรือใช้สมการ3.7 แทนค่าใน 3.9 เขียนให้อยู่ในรูปของสมการ 3.5 จะได้

$$I \approx (b-a)\frac{f(x_0)+2\sum_{i=1}^{n-1} f(x_i)+f(x_n)}{2n} \tag{3.10}$$

ค่าผิดพลาด สำหรับ multiple-application trapezoidal rule หาได้โดยการหาผลรวมของ ค่าผิดพลาด ของแต่ละเซกเมนต์ ซึ่งจะได้

$$E_t = -\frac{(b-a)^3}{12n^3}\sum_{i=1}^{n} f''(\xi_i) \tag{3.11}$$

ซึ่ง $f''(\xi_i)$ คือ ค่าอนุพันธ์อันดับ 2 ที่จุด ξ_i ในแต่ละเซกเมนต์ที่ i ซึ่งใช้เป็นค่าประมาณของค่า mean หรือค่าเฉลี่ยอนุพันธ์อันดับ 2 ของทุกๆ เซกเมนต์

$$\bar{f}'' \approx \frac{\sum_{i=1}^{n} f''(\xi_i)}{n}$$

$$n\bar{f}'' \approx \sum_{i=1}^{n} f''(\xi_i) \tag{3.12}$$

แทนในสมการ 3.11

ดังนั้นสมการ 3.11 สามารถเขียนใหม่ได้เป็น

$$E_a = -\frac{(b-a)^3}{12n^2}\bar{f}'' \tag{3.13}$$

<u>**ตัวอย่างที่ 3.2**</u> จงใช้ Trapezoidal Rule เพื่อประมาณค่าอินทิเกรต

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

โดยแบ่งเป็น 2 เซกเมนต์ ในช่วง a = 0 ถึง b = 0.8 และประมาณค่า ค่าผิดพลาด โดยค่าที่ถูกต้องของการอินทิเกรต คือ 1.64053334

<u>**วิธีทำ**</u> จากโจทย์จะได้

$$n = 2$$

$$h = \frac{0.8-0}{2} = 0.4$$

$$f(0) = 0.2 + 25(0) - 200(0)^2 + 675(0)^3 - 900(0)^4 + 400(0)^5$$
$$= 0.2$$
$$f(0.4) = 0.2 + 25(0.4) - 200(0.4)^2 + 675(0.4)^3 - 900(0.4)^4 + 400(0.4)^5$$
$$= 2.456$$
$$f(0.8) = 0.2 + 25(0.8) - 200(0.8)^2 + 675(0.8)^4 - 900(0.8)^5 + 400(0.8)^5$$
$$= 0.232$$

จากสมการ 3.10 จะได้

$$I \approx (0.8-0)\frac{f(0)+2f(0.4)+f(0.8)}{4}$$
$$\approx 0.8\frac{0.2+2(2.456)+0.232}{4} = 1.0688$$
$$E_t = 1.64053334 - 1.0688 = 0.5717$$
$$\varepsilon_t = \frac{1.64053334-1.0688}{1.64053334}100\% = 34.9\%$$
$$f'(x) = 25 - 400x + 2025x^2 - 3600x^3 + 2000x^4$$
$$f''(x) = -400 + 4050x - 10800x^2 + 8000x^3$$

ค่าเฉลี่ยของอนุพันธ์อันดับที่ 2 สามารถคำนวณได้ดังนี้

$$\bar{f}'' = \frac{\int_0^{0.8}\left(-400+4050x-10800x^2+8000x^3\right)dx}{0.8-0}$$
$$= -400x + \frac{4050}{2}x^2 - \frac{10800}{3}x^3 + \frac{8000}{4}x^4\Big|_0^{0.8}$$
$$= -60$$
$$E_a = \frac{(b-a)^3}{12n^2}\bar{f}''$$
$$E_a = -\frac{(0.8-0)^3}{12(2)^2}(-60) = 0.64$$

ถ้าเพิ่มจำนวนเซกเมนต์ (แบ่งช่วงมากขึ้น) จะสามารถลดค่าผิดพลาดได้ จากตาราง 3.1 จะเห็นว่าถ้าเพิ่ม เซกเมนต์ เป็น 2 เท่า จะลดลง ค้าผิดพลาดได้ประมาณ ¾

N	h	I	$\varepsilon_t\%$
2	0.4	1.0688	3409
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

ตาราง 3.1 แสดงค่าอินทิเกรตโดยใช้วิธี Mutiple-Application Trapezoidal Rule

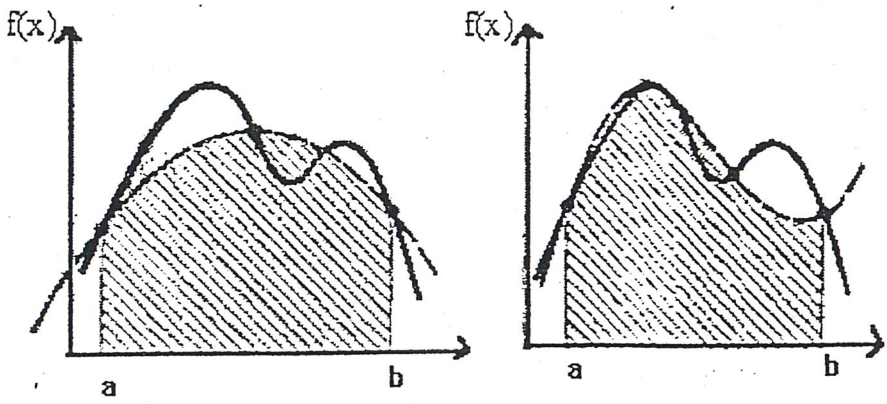
3.13 วิธี SIMPSON'S RULE

การประยุกต์วิธี trapezoidal rule โดยการแบ่งเป็นเซกเมนต์ๆ (segment) อีกวิธีการหนึ่งที่จะได้ค่าประมาณของการอินทิเกรตที่ถูกต้องมากขึ้น คือการใช้ฟังก์ชันพหุนามที่มีอันดับที่สูงกว่าในการเชื่อมระหว่างจุด ตัวอย่างเช่น ถ้ามีจุดกึ่งกลางระหว่าง f(a) และ f(b) สามจุดนี้สามารถเชื่อมต่อกันโดยใช้พาราโบลา (รูป 3.7a) ถ้ามี 2 จุด ซึ่งมีความกว้างช่วงเท่ากันระหว่าง f(a) และ f(b) สี่จุดนี้ สามารถเชื่อมต่อกันด้วย ฟังก์ชันพหุนามอันดับสาม (รูป 3.7b) สูตรในการหาค่าอินทิเกรตโดยการใช้ ฟังก์ชันพหุนามเหล่านี้เรียกว่า Simpson's Rule

3.1.3.1 วิธี SIMPSON'S 1/3 RULE

เป็นการประมาณค่าอินทิเกรตของฟังก์ชัน โดยใช้ฟังก์ชันพหุนามอันดับสอง ดังนั้นสมการ 3.1 จะเป็น

$$I = \int_a^b f(x)dx \approx \int_a^b f_2(x)dx$$



รูป 3.7

ถ้า a และ b แทนด้วย x_0 และ x_2

$f_2(x)$ แทนด้วยฟังก์ชันพหุนามลากรองจ์ อันดับ 2 (Lagrang Polynomial order =2) จะได้

$$I \approx \int \left[\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x-x_0)(x_2-x_1)} f(x_2) \right] dx$$

จากการอินทิเกรตจะได้

$$I \approx \frac{h}{3}\left[f(x_0)+4f(x_1)+f(x_2)\right] \tag{3.14}$$

โดย h = (b-a) / 2 สมการ 3.14 เรียกว่า simpson's 1/3 rule ซึ่งเป็น second Newton-Cotes Closed Integration Formula

Simpson's 1/3 Rule สามารถเขียนอยู่ในรูป

$$I \approx (b-a)\frac{f(x_0)+4f(x_1)+f(x_2)}{6} \tag{3.15}$$

ซึ่ง $a = x_0, b = x_2$ และ x_1 คือจุดกึ่งกลางระหว่าง a และ b หรือเท่ากับ (b+a)/2 สังเกตว่าจากสมการ 3.15 จุดกึ่งกลางจะถ่วงน้ำหนัก ด้วย 2/3 และที่จุดเริ่มต้นกับจุดสิ้นสุดจะถ่วงน้ำหนัก ด้วย 1/6

Simpson's 1/3 Rule มีค่าผิดพลาดเพราะตัดปลาย คือ

$$E_t = -\frac{1}{90}h^5 f^{(4)}(\xi)$$

แต่ h = (b-a) / 2

$$E_t = -\frac{(b-a)^5}{2880} f^{(4)}(\xi) \tag{3.16}$$

ซึ่ง ξ อยู่ในช่วงระหว่าง a และ b

<u>**ตัวอย่าง 3.3**</u> จงใช้สมการ 3.15 เพื่อหาค่าอินทิเกรตของ

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

ในช่วง a = 0 ถึง b = 0.8 และประมาณค่า ค่าผิดพลาด โดยค่าที่ถูกต้องของการอินทิเกรต คือ 1.64053334

<u>**วิธีทำ**</u> จากโจทย์จะได้

$$f(0) = 0.2 + 25(0) - 200(0)^2 + 675(0)^3 - 900(0)^4 + 400(0)^5$$
$$= 0.2$$
$$f(0.4) = 0.2 + 25(0.4) - 200(0.4)^2 + 675(0.4)^3 - 900(0.4)^4 + 400(0.4)^5$$
$$= 2.456$$
$$f(0.8) = 0.2 + 25(0.8) - 200(0.8)^2 + 675(0.8)^4 - 900(0.8)^5 + 400(0.8)^5$$
$$= 0.232$$

จากสมการ 3.15 จะได้

$$I \approx 0.8\frac{(0.2 + 4(2.456) + 0.232)}{6} = 1.36746667$$
$$E_t = 1.64053334 - 1.36746667 = 0.27306666$$
$$\varepsilon_t = \frac{1.64053334 - 1.36746667}{1.64053334} \times 100\% = 16.6\%$$
$$f'(x) = 25 - 400x + 2025x^2 - 3600x^3 + 2000x^4$$
$$f''(x) = -400 + 4050x - 10800x^2 + 8000x^3$$
$$f'''(x) = 4050 - 21600x + 24000x^2$$
$$f^{(4)} = -21600 + 4800x$$

ค่าเฉลี่ยของอนุพันธ์อันดับที่ 4 สามารถคำนวณได้ดังนี้

$$\bar{f}^{(4)} = \frac{\int_0^{0.8}(-21600 + 48000x)dx}{0.8 - 0}$$

$$= \frac{-21600x + \frac{48000x^2}{2} \Big|_0^{0.8}}{0.8}$$
$$= -2400$$

$$E_a = -\frac{(b-a)^5}{2880}\bar{f}^{(4)}$$

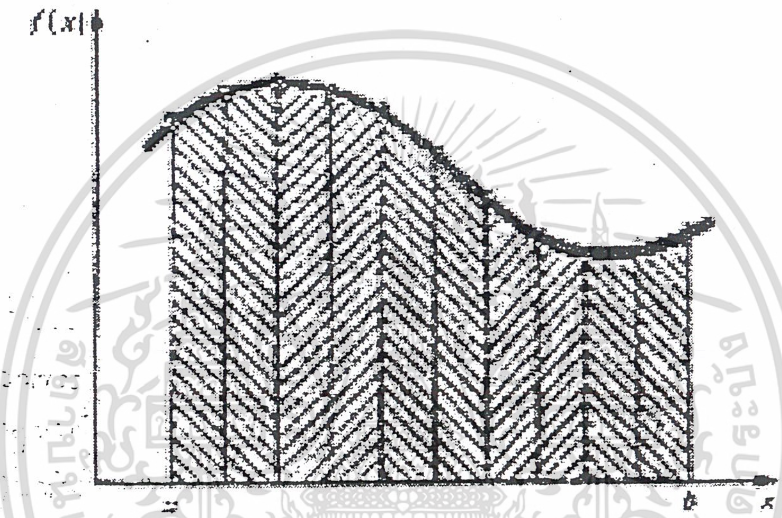
$$E_a = -\frac{(0.8-0)^5}{2880}(-2400) = 0.27306667$$

3.1.3.2 วิธี MULTIPLE-APPLICATION SIMPSON'S 1/3 RULE

เช่นเดียวกับ trapezoidal rule แต่ simpson's rule สามารถเพิ่มความถูกต้องโดยแบ่งช่วงของการอินทิเกรตเป็นหลายๆ เซกเมนต์ ด้วยช่วงที่เท่ากัน ดังรูป 3.8 ค่าของการอินทิเกรตก็คือ ผลบวกของการอินทิเกรตแต่ละเซกเมนต์

$$h = \frac{(b-a)}{n}$$

$$I = \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx + \int_{x_4}^{x_6} f(x)dx + \ldots + \int_{x_{n-2}}^{x_n} f(x)dx \qquad (3.17)$$



รูป 3.8

แทนค่าอินทิเกรตแต่ละช่วง

$$I \approx \frac{h}{3}\left[f(x_0)+4f(x_1)+f(x_2)\right]+\frac{h}{3}\left[f(x_0)+4f(x_1)+f(x_2)\right]+\ldots+\frac{h}{3}\left[f(x_{n-2})+4f(x_{n-1})+f(x_n)\right]$$

หรือ

$$I \approx (b-a)\frac{f(x_0)+4\sum_{i=1,3,5}^{n-1} f(x_i)+2\sum_{i=2,4,6}^{n-2} f(x_i)+f(x_n)}{3n} \qquad (3.18)$$

ค่าผิดพลาด สำหรับ Multiple-Application Simpson's 1/3 Rule หาได้จากรวมผลบวกของค่าผิดพลาด ในแต่ละเซกเมนต์ และหาค่าเฉลี่ยของอนุพันธ์

$$E_a = -\frac{(b-a)^5}{180n^4}\bar{f}^{(4)} \qquad (3.19)$$

ซึ่ง $\bar{f}^{(4)}$ คือ ค่าเฉลี่ยของอนุพันธ์อันดับที่ 4

ตัวอย่าง3.4 จงใช้ Multiple-Application Simpson's 1/3 Rule เพื่อประมาณค่าอินทิเกรตของ

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

เมื่อ n = 4 ในช่วง a = 0 ถึง b = 0.8 และประมาณค่า ค่าผิดพลาด โดยค่าที่ถูกต้องของการอินทิเกรต คือ 1.64053334

วิธีทำ จากโจทย์จะได้

$$f(0) = 0.2 + 25(0) - 200(0)^2 + 675(0)^3 - 900(0)^4 + 400(0)^5$$
$$= 0.2$$
$$f(0.2) = 0.2 + 25(0.2) - 200(0.2)^2 + 675(0.2)^3 - 900(0.2)^4 + 400(0.2)^5$$
$$= 1.288$$
$$f(0.4) = 0.2 + 25(0.4) - 200(0.4)^2 + 675(0.4)^3 - 900(0.4)^4 + 400(0.4)^5$$
$$= 2.456$$
$$f(0.6) = 0.2 + 25(0.6) - 200(0.6)^2 + 675(0.6)^3 - 900(0.6)^4 + 400(0.6)^5$$
$$= 3.464$$
$$f(0.8) = 0.2 + 25(0.8) - 200(0.8)^2 + 675(0.8)^4 - 900(0.8)^5 + 400(0.8)^5$$
$$= 0.232$$

จากสมการ 3.18 จะได้

$$I \approx 0.8\frac{0.2 + 4(1.288 + 3.464) + 2(2.456) + 0.232}{12}$$
$$\approx 1.62346667$$
$$E_t = 1.64053334 - 1.62346667 = 0.01706667$$
$$\varepsilon_t = \frac{1.64053334 - 1.62346667}{1.64053334} \times 100\% = 1.04\%$$
$$f'(x) = 25 - 400x + 2025x^2 - 3600x^3 + 2000x^4$$
$$f''(x) = -400 + 4050x - 10800x^2 + 8000x^3$$
$$f'''(x) = 4050 - 21600x + 24000x^2$$
$$f^{(4)}(x) = -21600 + 48000x$$

ค่าเฉลี่ยของอนุพันธ์อันดับที่ 4 สามารถคำนวณได้ดังนี้

$$\bar{f}^{(4)} = \frac{\int_0^{0.8} (-21600 + 48000x)dx}{0.8 - 0}$$

$$= \frac{-21600x + \frac{48000x^2}{2} \Big|_0^{0.8}}{0.8}$$

$$= -2400$$

$$E_a = -\frac{(b-a)^5}{180(n)^4} \bar{f}^{(4)}$$

$$E_a = -\frac{(0.8-0)^5}{180(4)^4}(-2400) = 0.01706667$$

3.1.3.3 วิธี Simpson's 3/8 Rule

การหาอนุพันธ์คล้ายกับ Trapezoidal และ Simpson's 1/3 Rule แต่จะใช้ฟังก์ชันพหุนามลากรองจ์อันดับ 3 เนื่องจากจะมีจุด 2 จุด ซึ่งความกว้างของช่วงเท่ากันในระหว่าง f(a) และ f(b)

จากการอินทิเกรตจะได้

$$I = \int_a^b f(x)dx \approx \int_a^b f_3(x)dx$$

$$I \approx \frac{3h}{8}\left[f(x_0) + 3f(x_1) + f(x_3)\right]$$

ซึ่ง h = (b-a) / 3 สมการนี้เรียกว่า **Simpson's 3/8 Rule**

วิธีนี้สามารถเขียนในรูปสมการ 3.5 ได้ดังนี้

$$I = (b-a)\frac{f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)}{8} \tag{3.20}$$

ค่าผิดพลาดของ Simpson's 3/8 Rule คือ

$$E_t = -\frac{3}{80}h^5 f^{(4)}(\xi)$$

เนื่องจาก h = (b-a)/3 ดังนั้นจะได้

$$E_t = -\frac{(b-a)^5}{6480} f^{(4)}(\xi) \tag{3.21}$$

การอินทิเกรตสำหรับเซกเมนต์ที่ไม่เท่ากัน (Integration with Unequal Segments)

จากวิธีต่างๆ ที่ผ่านมา เซกเมนต์จะถูกแบ่งเป็นช่วงที่เท่าๆ กัน แต่ในความเป็นจริงแล้วจะมีกรณีที่ เซกเมนต์ถูกแบ่งเป็นช่วงที่ไม่เท่ากัน ดังนั้นกรณีนี้จะนำ Trapezoidal Rule มาใช้หาค่าประมาณในแต่ละเซกเมนต์ และทำการบวกผลลัพธ์ของแต่ละเซกเมนต์ จะได้

$$I = h_1 \frac{f(x_1) + f(x_0)}{2} + h_2 \frac{f(x_2) + f(x_1)}{2} + \ldots + h_n \frac{f(x_n) + f(x_{n-1})}{2} \quad (3.22)$$

h คือ ความกว้างของเซกเมนต์ I

สังเกตเห็นว่าจะคล้ายคลึงกับ Multiple-Application Trapezoidal Rule จะต่างกันก็เพียงค่า h ในวิธี Multiple-Application Trapezoidal Rule เป็นค่าคงที่ แต่สำหรับวิธีนี้ค่า h ของงแต่ละเซกเมนต์ ไม่เท่ากัน

3.2 ROMBERG INTEGRATION

3.2.1 RECHARDSON'S EXTRAPOLATION

เป็นเทคนิคในการประมาณค่าอินทิเกรต โดยใช้ค่าประมาณ 2 ค่า เพื่อให้ได้ค่าประมาณที่ถูกต้องมากขึ้น

เช่น โดยวิธี Multiple-Application Trapezoidal Rule

$$I = I(h) + E(h)$$

เมื่อ I = ค่าอินทิเกรต

$I(h)$ = ค่าอินทิเกรตโดยประมาณ โดยใช้วิธี Multiple-Application Trapezoidal Rule จำนวน n เซกเมนต์

h = step size

= (b-a) / n

E(h) = ค่าผิดพลาดเพราะตัดปลาย

ถ้ากำหนด step size ต่างกันจะได้

$$I = I(h_1) + E(h_1) = I(h_2) + E(h_2) \quad (3.23)$$

จากสมการ 3.13 โดย n = (b-a) / h จะได้

$$E \approx E_a = -\frac{(b-a)}{12} h^2 \bar{f}'' \tag{3.24}$$

$$\frac{E(h_1)}{E(h_2)} \approx \frac{h_1^2}{h_2^2} \tag{3.25}$$

$$E(h_1) \approx E(h_2) \frac{h_1^2}{h_2^2}$$

แทน $E(h_1)$ ในสมการ (3.23)

$$I(h_1) + E(h_2)\left(\frac{h_1}{h_2}\right)^2 \approx I(h_2) + E(h_2)$$

$$I(h_1) - I(h_2) \approx E(h_2) - E(h_2)\left(\frac{h_1}{h_2}\right)^2$$

$$E(h_2) \approx \frac{I(h_1) - I(h_2)}{1 - (h_1 / h_2)^2}$$

จาก

$$I = I(h_2) + E(h_2)$$

$$I \approx I(h_2) + \left[\frac{1}{(h_1 / h_2)^2 - 1}\right]\left[I(h_2) - I(h_1)\right] \tag{3.26}$$

จากสมการ 3.26

ถ้า $h_2 = \frac{h_1}{2}$

$$I \approx I(h_2) + \frac{1}{2^2 - 1}\left[I(h_2) - I(h_1)\right]$$

$$I \approx \frac{4}{3} I(h_2) - \frac{1}{3} I(h_1) \tag{3.27}$$

ตัวอย่าง 3.5 จากตารางที่กำหนดให้ จงหาค่าอินทิเกรตโดยประมาณด้วยวิธี Richardson's Extrapolation

เซกเมนต์	h	ค่าอินทิเกรต	ε_t(%)
1	0.8	0.1728	89.5
2	0.4	1.0688	34.9
4	0.2	1.484	9.5

ค่าจริงคือ 1.64053334

วิธีทำ

ใช้ค่าประมาณของอินทิเกรต 1 เซกเมนต์ และ 2 เซกเมนต์

จากสมการ 3.27

$$I \approx \frac{4}{3}I(h_2) - \frac{1}{3}I(h_1)$$

$$\approx \frac{4}{3}(1.0688) - \frac{1}{3}(0.1728)$$

$$\approx 1.3674$$

$$E_t = 1.64053334 - 1.36746667 = 0.27306667$$

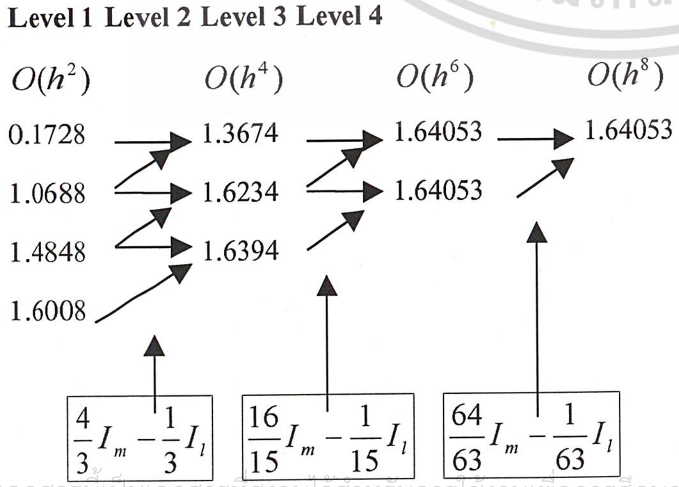
$$\varepsilon_t = 16.6\%$$

ทำนองเดียวกัน ถ้าใช้ 2 และ 4 เซกเมนต์

$$I \approx \frac{4}{3}(1.4848) - \frac{1}{3}(1.0688) \approx 1.62346667$$

$$E_t = 1.64053334 - 1.62346667 = 0.01706667$$

$$\varepsilon_t = 1.0\%$$



Richardson's Extrapolation

เมื่อ I_m และ I_l คือค่าอินทิเกรตโดยประมาณที่มีจำนวนเซกเมนต์มากกว่าและน้อยกว่าตามลำดับ (m = more , l = less)

เขียนรูปทั่วไป

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

เรียกว่า **Romberg Integration**

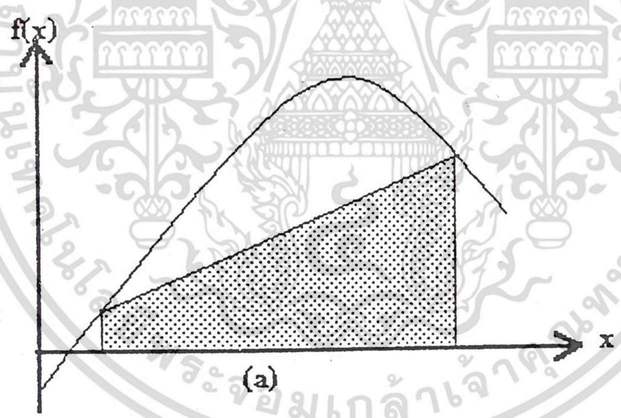
$I_{j+1,k-1}$ และ $I_{j,k-1}$ คือค่าอินทิเกรตโดยประมาณ

k คือ level

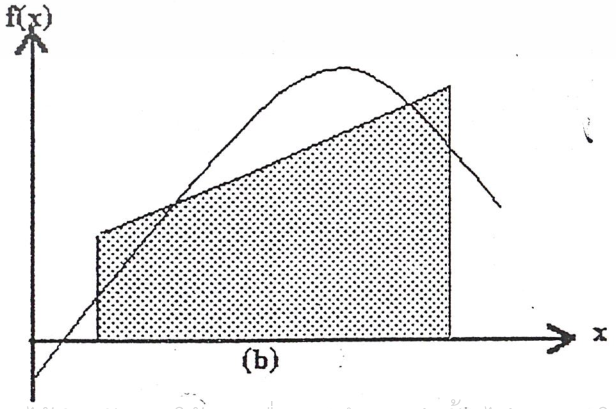
เช่น $I_{1,2} \approx \dfrac{4I_{2,1} - I_{1,1}}{4^1 - 1} \approx \dfrac{4I_{2,1} - I_{1,1}}{3}$

3.3 GAUSS QUADRATURE

สูตรของ เกาส์เลอจองด์ ซึ่งเป็นเทคนิคหนึ่งของ gauss quadrature



(a)



รูป 3.9

3.3.1 ระเบียบวิธีเทียบสัมประสิทธิ์

(METHOD OF UNDETERMINED COEFFICIENTS)

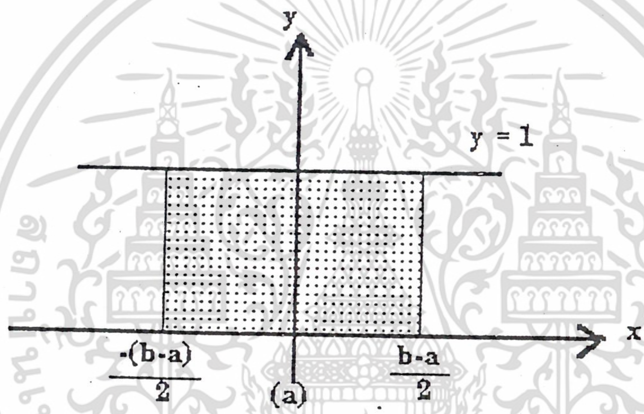
จากวิธี trapezoidal rule

$$I \approx (b-a)\frac{f(a)+f(b)}{2}$$

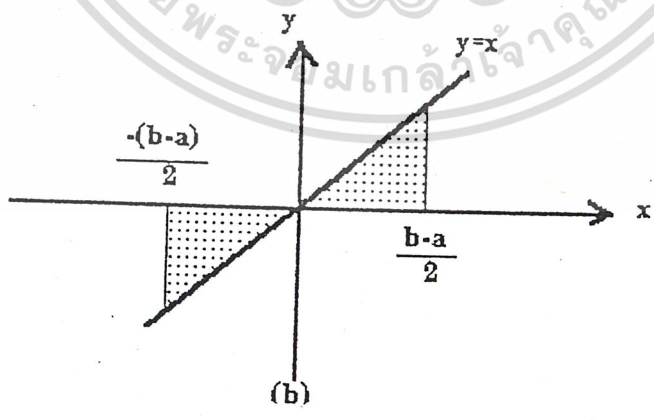
หรือเขียนในรูป

$$I \approx c_0 f(a) + c_1 f(b) \qquad (3.28)$$

จากวิธี trapezoidal rule ซึ่งจะใช้ฟังก์ชันที่อยู่ในรูปของเส้นตรงหรือค่าคงที่ นั่นคือ y = x หรือ y = 1



(a)



(b)

รูป 3.10

จะได้

$$c_0 f(a) + c_1 f(b) = \int_{-(b-a)/2}^{(b-a)/2} 1dx$$

หรือ

$$c_0 f(a) + c_1 f(b) = \int_{-(b-a)/2}^{(b-a)/2} xdx$$

เมื่อหาค่าอินทิเกรตจะได้

$$c_0 f(a) + c_1 f(b) = \frac{(b-a)}{2} + \frac{(b-a)}{2}$$

$$= b - a$$

$$c_0 f(a) + c_1 f(b) = \frac{x^2}{2} \Big|_{-(b-a)/2}^{(b-a)/2}$$

$$= \frac{(b-a)^2}{8} - \frac{(b-a)^2}{8}$$

$$= 0$$

เพราะฉะนั้น

$$c_0 f(a) + c_1 f(b) = b - a$$

$$c_0 f(a) + c_1 f(b) = 0$$

จะได้

$$c_0 + c_1 = b - a$$

$$\frac{(b-a)}{2c_0} - \frac{(b-a)}{2c_1} = 0$$

เมื่อแก้สมการหาค่า c_0 และ c_1

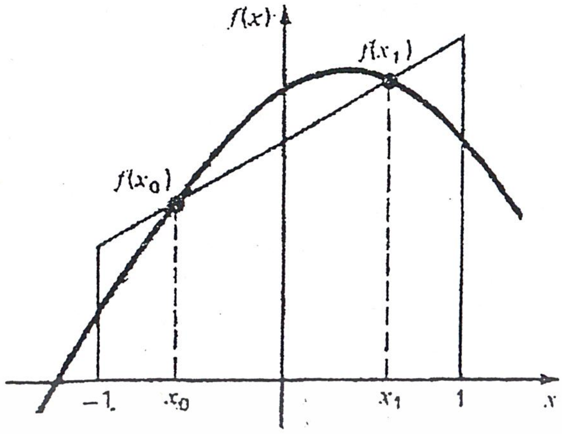
จะได้ $c_0 = c_1 = \dfrac{b-a}{2}$

แทนค่าในสมการ 3.28

$$I \approx \frac{b-a}{2} f(a) + \frac{b-a}{2} f(b)$$

ซึ่งก็คือวิธี trapezoidal rule

3.3.2 DERIVATION OF THE TWO-POINT GAUSS-LEGENDRE FORMULA



จากวิธี trapezoidal rule

$$I \approx c_0 f(a) + c_1 f(b) \quad (3.26)$$

ซึ่งจะ fix ว่าเป็นการอินทิเกรตในช่วง a ถึง b แต่โดยวิธีของ เกาส์เลอจองด์ จะไม่ fix จุด และไม่รู้ค่า

จากสมการ 3.29เขียนได้เป็น

$$I \approx c_0 f(x_0) + c_1 f(x_1)$$

จากวิธี trapezoidal rule จะใช้ฟังก์ชันเป็นเส้นตรงและค่าคงที่

โดยวิธีของ เกาส์เลอจองด์ จะเพิ่ม parabolic ($y = x^2$) และ cubic ($y = x^3$) จะได้ 4 สมการ คือ

$$c_0 f(a) + c_1 f(b) = \int_{-1}^{1} 1dx = 2 \quad (3.30)$$

$$c_0 f(a) + c_1 f(b) = \int_{-1}^{1} xdx = 0 \quad (3.31)$$

$$c_0 f(a) + c_1 f(b) = \int_{-1}^{1} x^2 dx = \frac{2}{3} \quad (3.32)$$

$$c_0 f(a) + c_1 f(b) = \int_{-1}^{1} x^3 dx = 0 \quad (3.33)$$

$$c_0 + c_1 = 2$$

$$x_0 c_0 + x_1 c_1 = 0$$

$$x_0^2 c_0 + x_1^2 c_1 = \frac{2}{3}$$

$$x_0^3 c_0 + x_1^3 c_1 = 0$$

ผลจากการแก้สมการ 4 สมการ จะได้

$$c_0 = c_1 = 1$$

$$x_0 = -\frac{1}{\sqrt{3}}$$

$$x_1 = \frac{1}{\sqrt{3}} \tag{3.34}$$

$$I = c_0 f(x_0) + c_1 f(x_1) = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)$$

จากสมการ 3.30 – 3.33 จะเห็นว่าเป็นการอินทิเกรตในช่วง –1 ถึง 1

เพราะฉะนั้น เมื่อกำหนด a,b มาให้ จะต้องเปลี่ยนช่วงใหม่

$$x = a_0 + a_1 x_d \tag{3.35}$$

ขีดจำกัดล่าง : x = a

$$x_d = -1$$

$$a = a_0 + a_1(-1) \tag{3.36}$$

ขีดจำกัดบน : x = b

$$x_d = 1$$

$$b = a_0 + a_1(1) \tag{3.37}$$

จากสมการ 3.36 และ 3.37

$$a_0 - a_1 = a$$

$$a_0 - a_1 = b$$

$$2a_0 = a + b$$

$$a_0 = \frac{a+b}{2}$$

$$a_1 = a_0 - a$$

$$= \frac{b-a}{2}$$

แทนค่าใน 3.35

$$x = \frac{(b+a)+(b-a)x_d}{2}$$

$$= \frac{b+a+bx_d - ax_d}{2} \tag{3.38}$$

$$dx = \frac{bdx_d - adx_d}{2}$$

$$dx = \frac{b-a}{2} dx_d$$

<u>ตัวอย่าง3.6</u> จงใช้วิธีเกาส์เลอจองด์ เพื่อหาค่าอินทิเกรตของ

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

ช่วง x = 0 ถึง x = 0.8 ค่าจริงคือ 1.64053334

<u>วิธีทำ</u> ก่อนอินทิเกรต จะต้องเปลี่ยนช่วงของการอินทิเกรต

$$a = 0$$

$$b = 0.8$$

$$a_0 = \frac{0.8+0}{2} = 0.4$$

$$a_1 = \frac{0.8-0}{2} = 0.4$$

$$x = 0.4 + 0.4x_d$$

$$dx = 0.4dx_d$$

$$\int_0^{0.8} \left(0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5\right)dx$$

$$= \int_{-1}^{1} \left[0.2 + 25(0.4+0.4x_d) - 200(0.4+0.4x_d)^2 + 675(0.4+0.4x_d)^3 \right.$$

$$\left. - 900(0.4+0.4x_d)^4 + 400(0.4+0.4x_d)^5 \right]0.4dx_d$$

$$I \approx f\left(\frac{-1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)$$

$$I \approx 0.51674055 + 1.30583723 \approx 1.82257778$$

SYSTEM OF LINEAR ALGEBRAIC EQUATIONS

- CRAMER'S RULE
- GUASS ELIMINATION AND LU DECOMPOSITION
- MATRIX INVERSION AND GUASS-SEIDEL
- LU DECOMPOSITION METHODS

4. SYSTEM OF LINEAR ALGEBRAIC EQUATIONS
(การแก้ระบบสมการเชิงเส้น)

การแก้ระบบสมการเชิงเส้น สามารถแบ่งเป็น 2 วิธี

1.ตัววิธีกำจัดไม่รู้ค่า (Elimination methods) ได้แก่วิธี Guass Elimination , Guass-Jordan,LU decomposition

2.วิธีการทำซ้ำ (Iteration methods) ได้แก่ Guass-Seidel

4.1 วิธีเชิงกราฟ (Graphical Method)

ถ้าให้สมการทั่วไป คือ

$$a_{11}x_1 + a_{12}x_2 = c_1$$
$$a_{21}x_1 + a_{22}x_2 = c_2$$

เขียนใหม่ได้ว่า

$$x_2 = -\left(\frac{a_{11}}{a_{12}}\right)x_1 + \frac{c_1}{a_{12}}$$

$$x_2 = -\left(\frac{a_{21}}{a_{22}}\right)x_1 + \frac{c_2}{a_{22}}$$

สมการทั้งสองเป็นสมการเส้นตรง

$$x_2 = (slope)x_1 + \text{int}\,ercept$$

เมื่อนำทั้งสองสมการ มาเขียนกราฟ จุดที่เส้นทั้งสองตัดกัน ก็คือ คำตอบของสมการ

4.2 วิธี CRAMER'S RULE

การใช้กฎของ CRAMER เป็นระเบียบวิธีการหนึ่งที่สามารถใช้แก้ระบบสมการขนาดเล็ก ๆ ได้โดยง่าย หัวใจของกฎCRAMER คือการหาค่าตัวกำหนด (Determinant) โดยกฎของ CRAMER กล่าวคือ ตัวไม่รู้ค่า x_i ของระบบสมการนั้นหาได้จาก

$$x_i = \frac{\det[A]_i}{\det[A]} \tag{1}$$

โดย det[A] แทนค่าตัวกำหนดของเมตริกซ์ [A] และ $\det[A]_i$ แทนค่าตัวกำหนดของเมตริกซ์ [A] หลังจากที่เมตริกซ์ [A] นี้ ได้เปลี่ยนค่าไปในแนวแถวตั้ง i ด้วยค่าในเวกเตอร์ {B} แล้ว การหาผลลัพธ์ด้วยวิธี CRAMER โดยใช้สมการ (1)

4.3 ระเบียบวิธีการกำจัดแบบเกาส์ (Guass Elimination Method)

ระเบียบวิธีการกำจัดแบบเกาส์ จัดได้ว่าเป็นระเบียบวิธีแก้ระบบสมการที่ได้รับความนิยมมากวิธีหนึ่ง เป็นวิธีที่ปรกติจะใช้ในโปรแกรมคอมพิวเตอร์ขนาดใหญ่ที่ใช้แก้ปัญหาทางวิศวกรรมศาสตร์และวิทยาศาสตร์โดยทั่วไป ระเบียบวิธีการกำจัดแบบเกาส์ในภาพรวม สามารถแบ่งออกได้เป็น 2 ขั้นตอน ดังนี้

1. การกำจัดไปข้างหน้า (Forward elimination) หากเรามีระบบสมการที่ประกอบด้วย 3 สมการย่อย ดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix} \tag{2}$$

การกำจัดไปข้างหน้าจะเปลี่ยนระบบสมการ (2) ไปให้อยู่ในรูปแบบซึ่งเมตริกซ์จัตุรัสทางด้านซ้ายของสมการ จะเป็นเมตริกซ์ที่ประกอบด้วยค่าศูนย์ตลอดแถบล่างซ้ายของเมตริกซ์นั้น ในรูปแบบดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b'_2 \\ b''_3 \end{Bmatrix} \tag{3}$$

โดยเครื่องหมายที่เป็นดรรชนีบนของสัมประสทธิ์แสดงถึงว่าสัมประสิทธิ์นั้นเป็นค่าใหม่ซึ่งเปลี่ยนแปลงไปจากสัมประสิทธิ์เดิมในสมการ (2)

2. การแทนค่าย้อนกลับ (Back substitution) เมื่อจัดระบบสมการให้อยู่ในรูปแบบของสมการ (3) ได้แล้ว ก็เป็นการง่ายทีเดียวที่จะคำนวณหาค่า x_i โดยเริ่มจากสมการท้ายสุดก่อน แล้วทำไล่ย้อนกลับขึ้นไปเพื่อหาค่า x_i ทีละสมการ ดังนี้

$$\begin{aligned} x_3 &= b_3'' / a_{33}'' \\ x_2 &= \left(b_2' - a_{23}' x_3\right) / a_{22}' \\ x_1 &= \left(b_1 - a_{12} x_2 - a_{13} x_3\right) / a_{11} \end{aligned} \tag{4}$$

ปัญหาที่อาจเกิดขึ้นจากระเบียบวิธีการกำจัดแบบเกาส์

1. ***ปัญหาที่เกิดจากการหารด้วยศูนย์ (Division by Zero)*** ปัญหานี้เกิดขึ้นเมื่อสัมประสิทธิ์เข้าใกล้ศูนย์หรือเป็นศูนย์ เช่น

$$\begin{aligned} 2x_2 + 3x_3 &= 8 \\ 4x_1 + 6x_2 + 7x_3 &= -3 \\ 2x_1 + x_2 + 6x_3 &= 5 \end{aligned}$$

2. ***ปัญหาความผิดพลาดจากการปัดเศษ (Round-off Errors)*** ค่าผิดพลาดเพราะการปัดเศษจะมีผลอย่างมากเมื่อระบบสมการมีหลายสมการ เพราะว่าคำตอบของตัวแปรขึ้นอยู่กับตัวแปรก่อนหน้า

3. ***ปัญหาระบบสมการในภาวะไม่เหมาะสม (ill-Conditioned System)*** เมื่อเปลี่ยนแปลงสัมประสิทธิ์ไปไม่มาก แต่ทำให้คำตอบที่ได้เปลี่ยนแปลงไปมาก

4.4 การปรับปรุงระเบียบวิธีการกำจัดแบบเกาส์

เราสามารถสรุปได้ว่าปัญหาในภาพรวมนั้นเกิดจากความผิดพลาดจากการปัดเศษและการหารด้วยศูนย์ ความผิดพลาดจากการปัดเศษสามารถแก้ไขให้ลดน้อยลงได้โดยการใช้เครื่องคอมพิวเตอร์ที่สามารถเก็บตัวเลขนัยสำคัญได้มากขึ้น ส่วนการหารด้วยศูนย์นั้นสามารถแก้ไขได้ดังนี้

1.การเลือกตัวหลัก (pivoting)

2.การจัดสเกล (scaling)

4.5 ระเบียบวิธีการของเกาส์-ชอร์ดอง(Guass-Jordan Method)

ระเบียบวิธีการของเกาส์-ชอร์ดอง เป็นระเบียบวิธีที่ขยายเพิ่มเติมต่อไปจากระเบียบวิธีการกำจัดแบบเกาส์ โดยมีหลักการในภาพรวมที่เริ่มจากระบบสมการทั่วไปในรูปแบบดัง เช่น

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix} \qquad (5)$$

จากนั้น ทำการกำจัดไปข้างหน้าในทำนองเดียวกันกับที่ใช้ระเบียบวิธีการกำจัดแบบเกาส์ โดยเปลี่ยนระบบสมการ (5) ให้ไปอยู่ในรูปแบบที่เมตริกซ์จัตุรัสทางด้านซ้ายของระบบสมการประกอบด้วยสัมประสิทธิ์ที่มีค่าเป็นศูนย์ตลอดแถบล่างซ้ายและแถบบนขวาของเมตริกซ์ และมีค่าเป็นหนึ่งตลอดแนวแกนเฉียงของเมตริกซ์นั้น ดังนี้

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1^* \\ b_2^* \\ b_3^* \end{Bmatrix} \qquad (6)$$

ซึ่งก่อให้เกิดผลลัพธ์โดยทันที คือ

$$\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1^* \\ b_2^* \\ b_3^* \end{Bmatrix}$$

4.6 ระเบียบวิธีการทำเมตริกซ์ผกผัน(Matrix Inversion)

การหาเมตริกซ์ผกผัน จากเมตริกซ์ที่กำหนดให้ สามารถทำได้ง่ายโดยใช้ระเบียบวิธีของเกาส์-ชอร์ดอง สมมติว่าเรามีเมตริกซ์ขนาด 3×3 ดังนี้

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad (7)$$

และต้องการเมตริกซ์ผกผัน $[A]^{-1}$ เราสามารถเริ่มจากความสัมพันธ์ที่ว่า

$$[A]_{(3\times3)}[A]^{-1}_{(3\times3)} = [I]_{(3\times3)} \tag{8}$$

โดย [I] คือ เมตริกซ์เอกลักษณ์ (identity matrix) และหากเรากำหนดให้เมตริกซ์ผกผันนี้ ประกอบนี้

$$[A]^{-1}_{(3\times3)} = \left[\underset{(3\times1)}{\{x\}_1} \quad \underset{(3\times1)}{\{x\}_2} \quad \underset{(3\times1)}{\{x\}_3} \right] \tag{9}$$

แล้วแทนกลับลงในสมการ (8) จะได้

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \left[\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}_1 \quad \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}_2 \quad \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}_3 \right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

ซึ่งก็หมายความในทำนองที่ว่า เราจะแก้ระบบสมการนี้ถึง 3 ครั้ง นั่นคือ เปรียบเสมือนมีเวกเตอร์ทางด้านขวาถึง 3 ชุด ดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix}, \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}, \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \tag{11}$$

เนื่องจากระบบสมการ (11) ประกอบด้วยเวกเตอร์ทางด้านขวาของสมการที่ต่างกันแต่มีเมตริกซ์จัตุรัสเพียงเมตริกซ์เดียวทางด้านซ้ายของสมการ ดังนั้นเราสามารถใช้ระเบียบวิธีเกาส์-ชอร์-ดอง เพื่อทำการกำจัดไปข้างหน้าไปโดยพร้อมเพรียงกันทั้ง 3 เวกเตอร์

4.7 ระเบียบวิธีการแยกแบบแอลยู (LU decomposition method)

ระเบียบวิธีการแยกแบบแอลยู เป็นอีกระเบียบวิธีการหนึ่งที่นิยมใช้ในการแก้ระบบสมการเชิงเส้นสำหรับปัญหาทางด้านวิศวกรรมศาสตร์และวิทยาศาสตร์หลักการและระเบียบวิธีการนี้เกิดจากแนวความคิดที่ว่าหากระบบสมการสามารถถูกกำจัดให้อยู่ในรูปดังเช่น สมการ (10) กล่าวคือ สัมประสิทธิ์ต่าง ๆ ที่อยู่ในแถบล่างซ้ายของเมตริกซ์ต่างมีค่าเป็นศูนย์ แล้วเราก็สามารถคำนวณหาผลลัพธ์โดยการแทนค่าย้อนกลับได้โดยง่าย

เพื่อให้ง่ายในการทำความเข้าใจ เราจะเริ่มจากระบบสมการ

$$[A]\{X\} = \{B\} \tag{12}$$

ซึ่งประกอบด้วย 3 สมการ ขั้นตอนแรกคือ การแยกเมตริกซ์[A] ออกเป็น 2 เมตริกซ์

$$[A] = [L][U] \tag{13}$$

ซึ่งมีรูปแบบโดยละเอียด ดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

กล่าวคือ เมตริกซ์ [L] เป็นเมตริกซ์ที่มีสัมประสิทธิ์ตลอดแบบแถบบนขวาเป็นศูนย์หมด ในขณะที่เมตริกซ์ [U] เป็นเมตริกซ์ที่มีค่าสัมประสิทธิ์ตลอดแถบล่างซ้ายเป็นศูนย์หมดเช่นกัน หากเราแทนสมการ (13) ลงในสมการ (12) จะได้

$$[L]\,[U]\{X\} = \{B\} \tag{15}$$

ดังนั้น เราสามารถแก้ระบบสมการ

$$[L]\{Y\} = \{B\} \tag{16}$$

ก่อน ด้วยการแทนค่าแบบไปข้างหน้า (forward substitution) เพื่อหา {Y} และจากนั้นจึงแก้ระบบสมการ

$$[U]\{X\} = \{Y\}$$

ด้วยการแทนค่าแบบย้อนกลับ (backward substitution) เพื่อหาผลลัพธ์ {X} ที่ต้องการ

จากคำอธิบายนี้ เราจะเห็นได้ว่าการแก้ระบบสมการ [A]{X} = {B} ด้วยระเบียบวิธีแยกแบบแอลยูนั้นประกอบด้วย 3 ขั้นตอนคือ

(ก) ทำการแยกเมตริกซ์ [A] ให้เป็นผลคูณระหว่างเมตริกซ์ [L][U]

(ข) แก้ระบบสมการ [L] {Y} ={B} ด้วยการแทนค่าแบบไปข้างหน้าเพื่อหาเวกเตอร์ {Y}

(ค) แก้ระบบสมการ [U] {X} ={Y} ด้วยการแทนค่าแบบย้อนกลับเพื่อหาผลลัพธ์เวกเตอร์ {X}

จากกระบวนการดังกล่าว เราพอจะเห็นได้ว่าเวลาส่วนใหญ่ที่จำเป็นต้องใช้ไปในการคำนวณนั้น จะเป็นการแยกเมตริกซ์ [A] ออกเป็นเมตริกซ์ [L][U] สำหรับปัญหาทางวิศวกรรมศาสตร์โดยทั่วไป เช่น ปัญหาการคำนวณความเค้นตามตำแหน่งต่าง ๆ ของสะพานที่เป็นโครงสร้างเหล็ก เป็นต้น เมตริกซ์ [A] จะมีคุณสมบัติทางกายภาพ (physical meaning) ที่เกี่ยวข้องกับการแสดงลักษณะรูปแบบของโครงสร้างรวมทั้งความแข็งของโลหะที่ใช้ ในขณะที่เวกเตอร์ {B} ทางด้านขวาของระบบสมการอาจแทนน้ำหนักของรถที่วิ่งผ่าน ดังนั้นเวกเตอร์ {B} จึงอาจเปลี่ยนแปลงไปโดยขึ้นอยู่ปริมาณรถและตำแหน่งของรถเหล่านั้นบนสะพาน นั่นหมายถึง หากเราคำนวณสิ่งที่เกิดขึ้นกับโครงสร้างสะพานนี้ ซึ่งในที่นี้ก็คือ เวกเตอร์ {X} เราสามารถทำการแยกเมตริกซ์ [A] เพียงครั้งเดียว และเปลี่ยนแปลงเวกเตอร์ {B} เพื่อหาผลลัพธ์ที่เกิดขึ้นสำหรับกรณีต่าง ๆ ซึ่งจะช่วยทุ่นเวลาในการคำนวณไปได้มากโดยเฉพาะหากปัญหานั้นมีขนาดใหญ่ เหตุผลดังกล่าวนี้เองทำให้ระเบียบวิธีการนี้ได้รับความนิยมค่อนข้างมากในการแก้ปัญหาทางวิศวกรรมศาสตร์ในทางปฏิบัติ

อนึ่ง หากเราพิจารณาการแยกของเมตริกซ์ [A] ออกเป็นเมตริกซ์ [L] และ [U] ดังแสดงในสมการ(14) เราจะเห็นได้ว่าสัมประสิทธิ์ตลอดแนวแกนเฉียงของเมตริกซ์ [L] นั้นเป็นค่าใด ๆ ในขณะที่สัมประสิทธิ์ตลอดแนวแกนเฉียงของเมตริกซ์ [U] นั้นต่างมีค่าเท่ากับหนึ่ง การจัดรูปแบบเมตริกซ์ [L] และ [U] ในลักษณะเช่นนี้ เรียกว่าเป็นการจัดของเคราท์ (crout) หากสัมประสิทธิ์ตลอดแนวแกนเฉียงของเมตริกซ์ [L] มีค่าเท่ากับหนึ่งโดยสัมประสิทธิ์ตลอดแนวแกนเฉียงของเมตริกซ์ [U] นั้นเป็นค่าใด ๆ การจัดรูปแบบเมตริกซ์ในลักษณะ หลังนี้เรียกว่าเป็นการจัดแบบของดูลิตเติล (Doolittle)

4.8 ระเบียบวิธีการการแยกแบบโชเลซกี(Cholesky Decomposition)

ระเบียบวิธีการต่าง ๆ ที่ได้อธิบายมา เป็นระเบียบวิธีที่ใช้แก้ระบบสมการ [A]{X} = {B} โดยเมตริกซ์ [A] เป็นเมตริกซ์ทั่วไป มีปัญหาทางวิศวกรรมศาสตร์จำนวนมากซึ่งใช้ระบียบวิธีผลต่างสืบเนื่องและวิธีไฟไนต์เอลิเมนท์ ที่จะก่อให้เกิดระบบสมการในลักษณะดังกล่าวโดยเมตริกซ์ [A] เป็นเมตริกซ์ที่มีความสมมาตร ในกรณีเช่นนี้ระเบียบวิธีแยกแบบแอลยูสามารถนำมา

ใช้ได้ และเมื่อใช้คุณสมบัติของเมตริกซ์ [A] ที่มีความสมมาตร จำนวนครั้งของการคูณหารจะลดลงไปอย่างมากอีกทั้งระเบียบขั้นตอนนั้นก็ง่ายแก่การทำความเข้าใจขึ้น ผลประโยชน์ดังกล่าวทำให้ระเบียบวิธีการแยกแบบโชเลซกีนี้นิยมใช้ในโปรแกรมคอมพิวเตอร์ทางวิศวกรรมศาสตร์โดยทั่วไป อาทิเช่น โปรแกรมที่ใช้แก้ปัญหาทางด้านโครงสร้าง เป็นต้น หลักการของวิธีการนี้เริ่มจากระบบสมการ

$$[A]\{X\}=\{B\} \tag{17}$$

ซึ่งในกรณีที่ [A] เป็นเมตริกซ์สมมาตร เราสามารถแยกเมตริกซ์ [A] ออกเป็น 2 เมตริกซ์ ได้ดังนี้

$$[A] = [L][L]^T \tag{18}$$

โดย [L] เป็นเมตริกซ์ ที่ค่าสัมประสิทธิ์ทางด้านขวาบนของเมตริกซ์เป็นศูนย์หมด เช่น หากระบบสมการ (17) ประกอบด้วย 3 สมการย่อย สมการ (18) นี้คือ

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix}\begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix} \tag{19}$$

สัมประสิทธิ์ในเมตริกซ์ [L] เหล่านี้ สามารถคำนวณหาได้ดังนี้

$$\begin{aligned}
l_{11}^2 &= a_{11} & &\rightarrow & l_{11} &= \sqrt{a_{11}} \\
l_{11}l_{21} &= a_{12} & &\rightarrow & l_{21} &= \frac{a_{12}}{l_{11}} \\
l_{11}l_{31} &= a_{13} & &\rightarrow & l_{31} &= \frac{a_{13}}{l_{11}} \\
l_{21}^2 + l_{22}^2 &= a_{22} & &\rightarrow & l_{22} &= \sqrt{a_{22} - l_{21}^2} \\
l_{21}l_{31} + l_{22}l_{32} &= a_{23} & &\rightarrow & l_{32} &= \frac{a_{23} - l_{21}l_{31}}{l_{22}} \\
l_{31}^2 + l_{32}^2 + l_{33}^2 &= a_{33} & &\rightarrow & l_{33} &= \sqrt{a_{33} - l_{31}^2 - l_{32}^2}
\end{aligned}$$

เมื่อคำนวณสัมประสิทธิ์เหล่านี้ได้แล้ว ระบบสมการ (17) สำหรับกรณีนี้ คือ

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix}\begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix} \tag{20}$$

นั่นคือ ทำการหาเวกเตอร์ {Y} ด้วยการแทนค่าแบบไปข้างหน้า

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix} \tag{21}$$

แล้วจึงคำนวณหาผลลัพธ์ในเวกเตอร์ {X} ด้วยการแทนค่าแบบย้อนกลับ

$$\begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \end{Bmatrix} \tag{22}$$

ลำดับขั้นตอนในการหาค่าสัมประสิทธิ์ต่าง ๆ ในเมตริกซ์ [L] ดังแสดงด้วยระบบสมการที่ประกอบด้วย 3 สมการย่อยนี้ สามารถประยุกต์ไปใช้ได้กับระบบสมการโดยทั่วไปที่ประกอบด้วย n สมการย่อย ได้ดังนี้

สำหรับแถวนอนแรกในเมตริกซ์ [L]

$$l_{11} = \sqrt{a_{11}} \tag{23a}$$

จากนั้นสำหรับแถวนอน k ถัดไป k = 2,3,…,n

$$l_{ik} = \frac{a_{ki} - \sum_{j=1}^{i-1} l_{ij} l_{kj}}{l_{ii}} \qquad i = 1,2,\ldots,k-1 \tag{23b}$$

และ

$$l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2} \tag{23c}$$

เมื่อได้เมตริกซ์ [L] นี้แล้ว เวกเตอร์ {Y} และ {X} จึงสามารถคำนวณได้ตามลำดับ ดังนี้

$$y_1 = \frac{b_1}{l_{11}}$$

$$y_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} y_j}{l_{ii}} \qquad i = 2,3,\ldots,n \tag{24}$$

และ

$$x_n = \frac{y_n}{l_{nn}}$$

$$x_i = \frac{y_i - \sum_{j=i+1}^{n} l_{ji} x_j}{l_{ii}} \qquad i = n-1, n-2, \ldots, 1 \tag{25}$$

CURVE FITTING

LEAST-SQUARES REGRESSION

- LINEAR REGRESSION
- POLYNOMAIL REGRESSION
- MULTIPLE LINEAR REGRESSION
- GENERAL LINEAR LEAST SQUARES

5. CURVE FITTING

ส่วนใหญ่ข้อมูลที่เราได้นั้นมีค่าไม่ต่อเนื่อง ซึ่งบางครั้งเราต้องการค่าประมาณที่อยู่ระหว่างช่วงที่ไม่ต่อเนื่องนั้น ในเรื่อง curve fitting จะอธิบายถึงวิธีการหาเส้นที่มีค่าต่อเนื่องที่เหมาะสมกับข้อมูลที่มีค่าไม่ต่อเนื่อง โดยทั่วไปจะมี 2 วิธี จะเลือกใช้วิธีไหนนั้นขึ้นอยู่กับว่าข้อมูลที่ไม่ต่อเนื่องนั้นมีค่าผิดพลาด (error) มากหรือน้อย

ถ้าคาดว่าข้อมูลมีค่าผิดพลาดมาก เส้นที่ค่าต่อเนื่องที่เหมาะสมก็ควรจะเป็นแนวโน้มของข้อมูล เพราะข้อมูลที่ได้มาแต่ละค่าอาจจะไม่ถูกต้อง เส้นที่เหมาะสมกับข้อมูลไม่จำเป็นต้องผ่านจุดข้อมูล ที่ได้มานั้นก็ได้ แต่จะแทนข้อมูลโดยรวม ซึ่งวิธีการหาเส้นที่เหมาะสมนี้เรียกว่า least-square regression

ถ้าคาดว่าข้อมูลถูกต้อง เส้นที่มีค่าต่อเนื่องที่เหมาะสม ก็ควรจะผ่านทุก ๆ จุดข้อมูล วิธีการหาเส้นที่เหมาะสมนี้ เรียกว่า Interpolation

5.1 LEAST-SQUARE REGRESSION

ข้อมูลที่คาดว่ามีค่าผิดพลาดมาก เช่น ข้อมูลที่ได้จากผลการทดลองในห้องปฏิบัติการ ตัวอย่างเช่น รูป 1a ซึ่งมีข้อมูลอยู่ 7 จุด จะเห็นว่าแนวโน้มของข้อมูลคือ ค่า x และ ค่า y มีค่าไปในทิศทางเดียวกัน นั่นคือ ถ้า x มีค่ามากขึ้น y ก็จะมีค่ามากขึ้น

รูป 1c แสดงเส้นตรงที่มีค่าต่อเนื่องที่สามารถใช้แทนข้อมูลได้

วิธีการหาเส้นที่เหมาะสมกับข้อมูล โดยเส้นที่เหมาะสมนั้นไม่จำเป็นต้องผ่านจุดข้อมูลก็อาจจะเป็นการplot ข้อมูล และ sketch เส้นที่เห็นว่าดีที่สุด แต่วิธีการนี้ไม่ใช่เป็นวิธีที่ดี แต่ละคนก็จะได้เส้นตรงที่ต่างกัน วิธีหนึ่งที่จะได้เส้นที่เหมาะสม ก็คือให้ความแตกต่างระหว่างจุดข้อมูล และเส้นเหมาะสมมีค่าน้อยที่สุด ซึ่งวิธีนี้เรียกว่า **Least-Squares Regression**

5.1.1 LINEAR REGRESSION

รูปแบบที่ง่ายที่สุดของ Least-Square Approximation ก็คือ การใช้เส้นตรงประมาณกับเซ็ตของจุด $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

สมการเส้นตรง คือ

$$y_a = a_0 + a_1 x$$

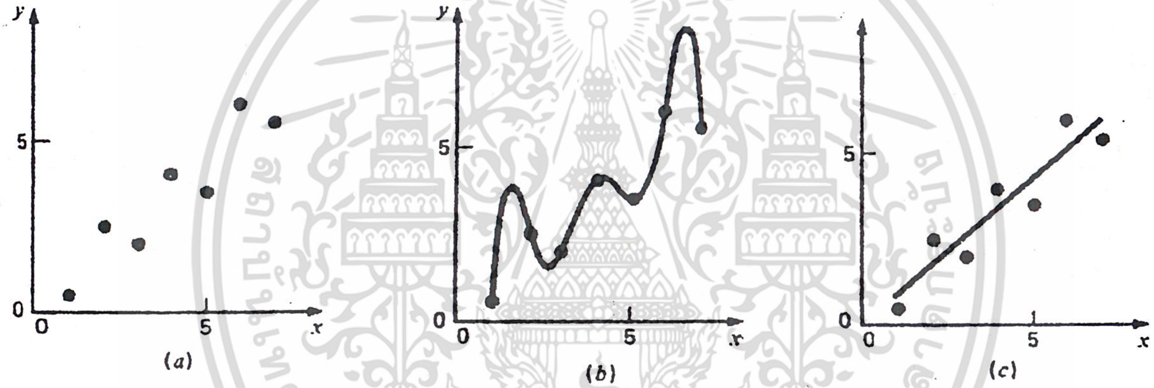
$$y = y_a + e$$

$$y = a_0 + a_1 x + e \qquad (1)$$

a_0, a_1 คือ สัมประสิทธิ์

e คือ ค่าผิดพลาด (error หรือ residual) หรือค่าความแตกต่างระหว่างค่าจริงและค่าประมาณหรือเราสามารถเขียนสมการ (1) ในรูป

$$e = y - a_0 - a_1 x$$



รูป 5.1

5.1.1.1 เกณฑ์การหาเส้นที่เหมาะสมที่สุด

วิธีการหนึ่งในการหาเส้นที่เหมาะสมที่สุด คือ ทำให้ผลรวมของค่าผิดพลาดมีค่าน้อยที่สุด

$$S_r = \sum_{i=1}^{n} e_i^2$$

$$= \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$

เมื่อ n คือจำนวนจุดข้อมูล

เพื่อหาค่า minimize เพราะฉะนั้น

$$\frac{\partial S_r}{\partial a_0} = 0$$

และ

$$\frac{\partial S_r}{\partial a_1} = 0$$

$$\frac{\partial S_r}{\partial a_0} = -2\sum_{i=1}^{n}(y_i - a_0 - a_1 x_i)$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum_{i=1}^{n}(y_i - a_0 - a_1 x_i)x_i$$

$$\sum_{i=1}^{n}(y_i - a_0 - a_1 x_i) = 0$$

$$\sum_{i=1}^{n}(y_i - a_0 - a_1 x_i)x_i = 0$$

หรือ

$$a_0\sum_{i=1}^{n}1 + a_1\sum_{i=1}^{n}x_i = \sum_{i=1}^{n}y_i$$

$$a_0\sum_{i=1}^{n}x_i + a_1\sum_{i=1}^{n}x_i^{\;2} = \sum_{i=1}^{n}y_i x_i$$

เรียกว่า normal equations

เพราะว่า $\sum_{i=1}^{n}1 = n$

ค่า mean ของ x_i และ y_i คือ

$$\bar{x} = \frac{\sum_{i=1}^{n}x_i}{n} \quad ; \quad \bar{y} = \frac{\sum_{i=1}^{n}y_i}{n}$$

หรือ

$$n\bar{x} = \sum_{i=1}^{n}x_i \quad , \quad n\bar{y} = \sum_{i=1}^{n}y_i$$

normal equations เขียนใหม่ได้เป็น

$$na_0 + n\bar{x}a_1 = n\bar{y}$$

$$n\bar{x}a_0 + \sum_{i=1}^{n}x_i^2 a_1 = \sum_{i=1}^{n}x_i y_i$$

หรือเขียนในรูปเมตริกซ์

$$\begin{bmatrix} n & n\bar{x} \\ n\bar{x} & \sum_{i=1}^{n} x_i^2 \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} n\bar{y} \\ \sum_{i=1}^{n} x_i y_i \end{bmatrix}$$

$$a_1 = \frac{\begin{bmatrix} n & n\bar{y} \\ n\bar{x} & \sum_{i=1}^{n} x_i y_i \end{bmatrix}}{\begin{bmatrix} n & n\bar{x} \\ n\bar{x} & \sum_{i=1}^{n} x_i^2 \end{bmatrix}}$$

$$= \frac{n\sum x_i y_i - n^2\bar{x}\bar{y}}{n\sum x_i^2 - n^2\bar{x}^2}$$

$$= \frac{n^2(\frac{1}{n}\sum x_i y_i - \bar{x}\bar{y})}{n^2(\frac{1}{n}\sum x_i^2 - \bar{x}^2)}$$

$$= \frac{\overline{xy} - \bar{x}\,\bar{y}}{x^2 - \bar{x}^2}$$

$$a_0 = \bar{y} - a_1\bar{x}$$

5.1.1.2 คุณภาพของเส้นเหมาะสม

$$S_r = \sum (y_i - a_0 - a_1 x_i)^2$$

$= sum\ of\ squares\ of\ residuals\ about\ regression\ line\ y = a_0 + a_1 x$

$$S_{y/x} = \sqrt{\frac{S_r}{n-2}}$$

$= standard\ error\ of\ the\ estimate$

$$S_t = \sum (y_i - \bar{y})^2$$

$= sum\ of\ squares\ of\ residuals\ about\ the\ mean\ \bar{y}$

$$r^2 = \frac{S_t - S_r}{S_t}$$

$= coefficient\ of\ determination$

$$r = \sqrt{\frac{S_t - S_r}{S_t}}$$

$= correlation\ coefficient$

สำหรับเส้นที่เหมาะสมที่สุด จะได้ $S_r = 0; r^2 = 1; r = 1$ แสดงว่าเส้นที่มีค่าต่อเนื่องที่ได้นี้สามารถแทนข้อมูลได้ 100%

5.1.1.3 การประยุกต์ของ Linear Regression

Linearization of Nonlinear Relationships ความสัมพันธ์ระหว่างตัวแปรไม่เป็นเชิงเส้นเสมอไป เช่น

Exponential Equation

$$y = ae^{bx}$$

a และ b เป็นค่าคงที่

ทำการ take logarithm ฐาน e

$$\ln y = \ln a + \ln(e^{bx})$$

$$\ln y = \ln a + bx$$

ให้ z = ln y และ c = ln a

จะได้

$$z = c + bx$$

จะเห็นว่า z และ x มีความสัมพันธ์เชิงเส้น

นั่นคือ สามารถใช้ linear regression กับเซ็ตของจุด (x_i, z_i) หรือ $(x_i, \ln y_i)$

Power Equation

$$y = ax^b$$

a และ b เป็นค่าคงที่

ทำการ take logarithm

$$\log y = \log a + b \log x$$

ให้

$$z = \log y$$

$$w = \log x$$

$$c = \log a$$

จะได้

$$z = c + bw$$

จะเห็นว่า z และ w มีความสัมพันธ์เชิงเส้น

นั่นคือ สามารถใช้ linear regression กับเซ็ตของจุด (w_i, z_i) หรือ $(\log x_i, \log y_i)$

Saturation-Growth Rate Equation

$$y = a\frac{x}{b+x}$$

a และ b เป็นค่าคงที่

เราสามารถแปลงให้อยู่ในรูป linear form กลับเศษเป็นส่วน

$$\frac{1}{y} = \frac{b+x}{ax} = \frac{b}{a}\cdot\frac{1}{x} + \frac{1}{a}$$

ให้

$$z = \frac{1}{y} \qquad w = \frac{1}{x}$$

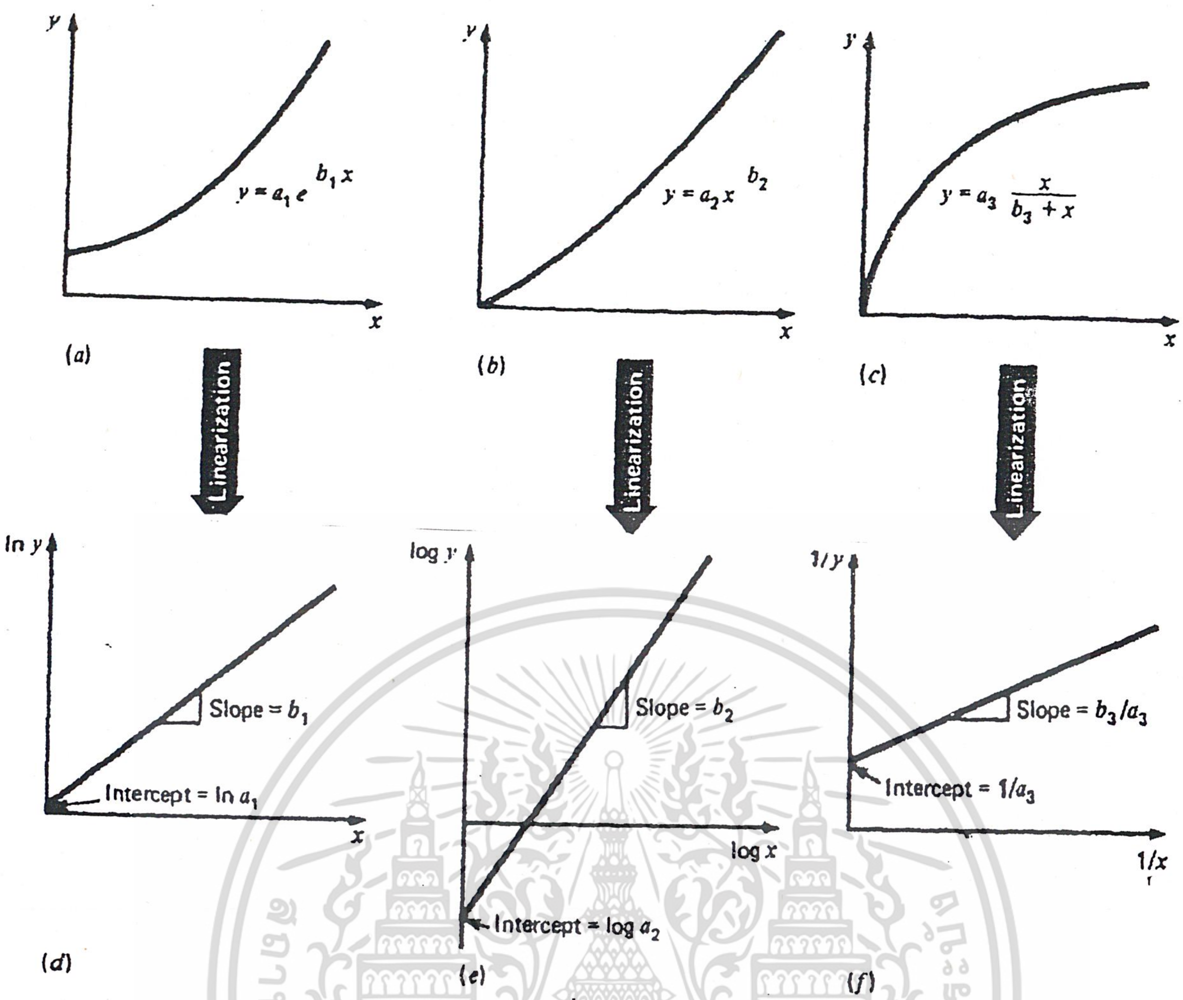
$$c = \frac{1}{a} \qquad d = \frac{b}{a}$$

จะได้

z = c + dw

จะเห็นว่า z และ w มีความสัมพันธ์เชิงเส้น

นั่นคือ สามารถใช้ linear regression กับเซ็ตของจุด (w_i, z_i) หรือ $\left(\frac{1}{x_i}, \frac{1}{y_i}\right)$



รูป 5.2

5.2 POLYNOMIAL REGRESSION

ในหัวข้อก่อนหน้านี้ เป็นการใช้สมการเส้นตรง โดยใช้ least-squares criterion บางครั้งข้อมูลที่ได้มาอาจจะเป็นดังรูป 5.3 ซึ่งไม่เหมาะสมที่ใช้เส้นตรง การใช้เส้นโค้งจะดีกว่า ในหัวข้อนี้จะกล่าวถึง polynomial regression

สมการพหุนามอันดับที่ m

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m + e$$

ผลรวมกำลังสองของ residual

$$S_r = \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_m x_i^m \right)^2$$

เพื่อหาค่า minimize เพราะฉะนั้น

$$\frac{\partial S_r}{\partial a_0} = -2\sum\left(y_i - a_0 - a_1x_i - a_2x_i^2 - \ldots - a_m x_i^m\right) = 0$$
$$\frac{\partial S_r}{\partial a_1} = -2\sum x_i\left(y_i - a_0 - a_1x_i - a_2x_i^2 - \ldots - a_m x_i^m\right) = 0$$
$$\frac{\partial S_r}{\partial a_2} = -2\sum x_i^2\left(y_i - a_0 - a_1x_i - a_2x_i^2 - \ldots - a_m x_i^m\right) = 0$$

.

.

.

$$\frac{\partial S_r}{\partial a_m} = -2\sum x_i^m\left(y_i - a_0 - a_1x_i - a_2x_i^2 - \ldots - a_m x_i^m\right) = 0$$

จะได้ normal equations คือ

$$a_0 n + a_1\sum x_i + a_2\sum x_i^2 + \ldots + a_m\sum x_i^m = \sum y_i$$
$$a_0\sum x_i + a_1\sum x_i^2 + a_2\sum x_i^3 \ldots + a_m\sum x_i^{m+1} = \sum x_i y_i$$
$$a_0\sum x_i^2 + a_1\sum x_i^3 + a_2\sum x_i^4 + \ldots + a_m\sum x_i^{m+2} = \sum x_i^2 y_i$$

.

.

.

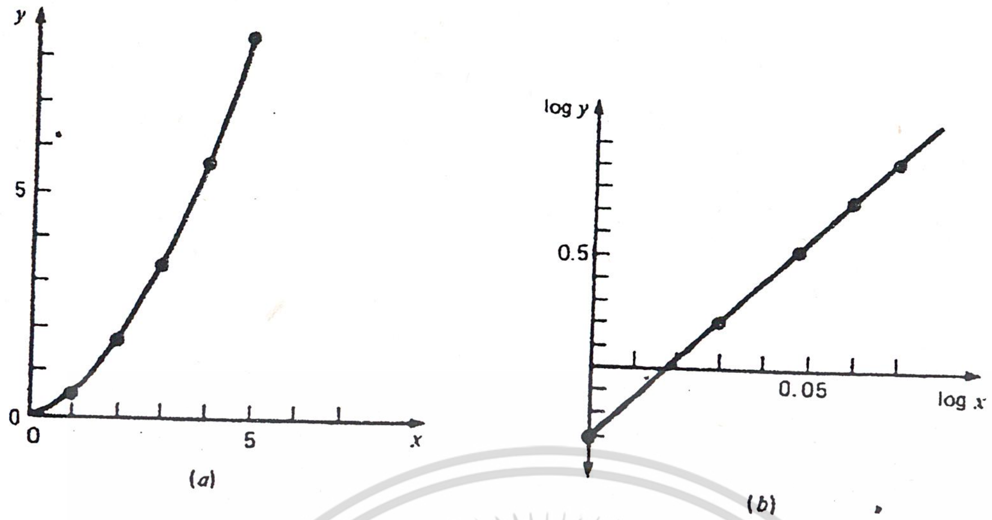
$$a_0\sum x_i^m + a_1\sum x_i^{m+1} + a_2\sum x_i^{m+2} + \ldots + a_m\sum x_i^{2m} = \sum x_i^m y_i$$

สังเกตว่ามี m+1 สมการ และมีตัวไม่รู้ค่า m+1 ตัว คือ $a_0, a_1, a_2, \ldots, a_m$

ค่าผิดพลาดมาตราฐานของการประมาณของ polynomial regression

$$S_{y/x} = \sqrt{\frac{S_r}{n-(m+1)}}$$

m คือ อันดับของสมการพหุนาม



รูป 5.3

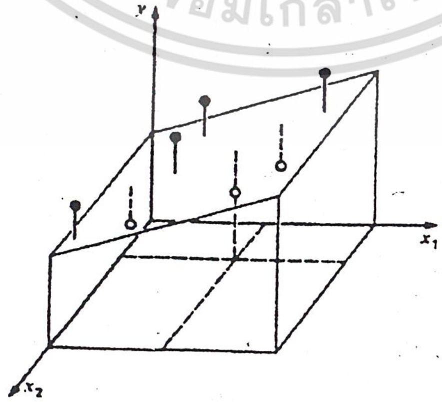
$$r^2 = \frac{S_t - S_r}{S_t}$$

5.3 MULTIPLE LINEAR REGRESSION

ในกรณีที่ y เป็นฟังก์ชันเชิงเส้นที่ขึ้นอยู่กับตัวแปรมากกว่า 2 ตัวแปร ตัวอย่างเช่น y ขึ้นอยู่กับตัวแปร x_1 และ x_2

$$y = a_0 + a_1 x_1 + a_2 x_2 + e$$

สำหรับกรณีนี้จะเป็นระนาบ ดังรูป 5.4



รูป 5.4

เพื่อหาค่า minimize เพราะฉะนั้น

$$\frac{\partial S_r}{\partial a_0} - 2\sum\left(y_i - a_0 - a_1x_{1i} - a_2x_{2i}\right) = 0$$

$$\frac{\partial S_r}{\partial a_1} - 2\sum x_{1i}\left(y_i - a_0 - a_1x_{1i} - a_2x_{2i}\right) = 0$$

$$\frac{\partial S_r}{\partial a_2} - 2\sum x_{2i}\left(y_i - a_0 - a_1x_{1i} - a_2x_{2i}\right) = 0$$

จะได้ normal equation ดังนี้

$$na_0 + \sum x_{1i}a_1 + \sum x_{2i}a_2 = \sum y_i$$

$$\sum x_{1i}a_0 + \sum x_{1i}{}^2 a_1 + \sum x_{1i}x_{2i}a_2 = \sum x_{1i}y_i$$

$$\sum x_{2i}a_0 + \sum x_{1i}x_{2i}a_1 + \sum x_{2i}^2 a_2 = \sum x_{2i}y_i$$

หรือเขียนในรูปเมตริกซ์ จะได้

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}x_{2i} \\ \sum x_{2i} & \sum x_{1i}x_{2i} & \sum x_{2i}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_{1i}y_i \\ \sum x_{2i}y_i \end{bmatrix}$$

5.4 GENERAL LINEAR LEAST SQUARES

ซึ่งอยู่ในรูป

$$y = a_0z_0 + a_1z_1 + a_2z_2 + \ldots + a_mz_m + e \qquad (2)$$

เมื่อ $z_0, z_1, \ldots, z_m$ คือ ฟังก์ชัน

เช่น ถ้าเป็น multiple linear regression

$$z_0 = 1, z_1 = x_1, z_2 = x_2, \ldots, z_m = x_m$$

ถ้าเป็น polynomial regression

$$z_0 = x^0 = 1, z_1 = x, z_2 = x^2, \ldots, z_m = x^m$$

จากสมการ (2) เขียนอยู่ในรูป

$$y(x) = a_0z_0(x) + a_1z_1(x) + a_2z_2(x) + \ldots + a_mz_m(x) + e$$

$$= \sum_{j=0}^{m} a_jz_j(x) + e$$

5.4.1 เกณฑ์เพื่อหาเส้นที่เหมาะสมที่สุด

ค่าต่ำสุดของผลรวมกำลังสองของ residual (S_r)

$$S_r = \sum_{i=1}^{n} e_i^2$$

$$= \sum_{i=1}^{n} \left[y_i - \sum_{j=0}^{m} a_j z_j(x_i) \right]^2$$

ให้ $z_{ji} = z_j(x_i)$

$$S_r = \sum_{i=1}^{n} \left[y_i - \sum_{j=0}^{m} a_j z_{ji} \right]^2$$

หาค่า minimize

$$\frac{\partial S_r}{\partial a_j} = 0 \qquad (j = 0,1,2,\ldots,m)$$

$$\frac{\partial S_r}{\partial a_j} = \frac{\partial \sum_{i-1}^{n} \left[y_i = \sum_{k=o}^{m} a_k z_{ki} \right]^2}{\partial a_j}$$

$$= \frac{\partial \sum_{i-1}^{n} \left[y_i - \sum_{k=o}^{m} a_k z_{ki} \right]^2}{\partial a_j}$$

$$= \sum_{i=1}^{n} \left\{ 2 \left[y_i - \sum_{i=1}^{n} a_k z_{ki} \right] \frac{\partial \left[y_i - \sum_{i=1}^{n} a_k z_{ki} \right]}{\partial a_j} \right\}$$

$$= -2 \sum_{i=1}^{n} \left\{ \left[y_i - \sum_{i=1}^{n} a_k z_{ki} \right] z_{ji} \right\}$$

$$\sum_{i=1}^{n} \left\{ \left[y_i - \sum_{k=0}^{m} a_k z_{ki} \right] z_{ji} \right\} = 0$$

$$\sum_{i=1}^{n} \sum_{k=o}^{m} a_k z_{ki} z_{ji} = \sum_{i=1}^{n} y_i z_{ji} \qquad (j = 0,\ldots,m)$$

เขียน normal equation ในรูปเมตริกซ์

$$y = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix} \qquad A = \begin{bmatrix} a_0 \\ a_1 \\ . \\ . \\ . \\ a_m \end{bmatrix}$$

$$z = \begin{bmatrix} z_{01} & \dots & z_{j1} & \dots & z_{m1} \\ z_{02} & & z_{j2} & & z_{m2} \\ . & & & & \\ . & & & & \\ . & & & & \\ z_{0n} & & z_{jn} & & z_{nm} \end{bmatrix}$$

normal equation คือ

$$(Z^T Z)A = Z^T Y$$

INTERPOLATION

- NEWTON'S DIVIDED-DIFFERENCE INTERPOLATING POLYNOMAILS
- LAGRANGE INTERPOLATING POLYNOMAILS
- SPINE INTERPOLATION

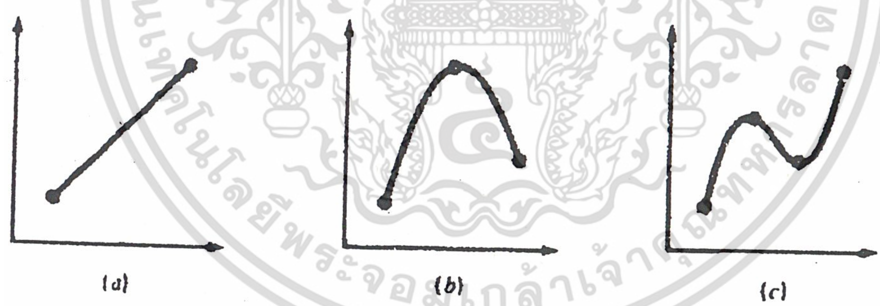
6. INTERPOLATION

วิธีการที่จะกล่าวถึงนี้ คือ polynomial interpolation

มีรูปแบบทั่วไปคือ

$$f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$$

เรียกว่า ฟังก์ชันพหุนามอันดับ n มีข้อมูล n+1 จุด ตัวอย่างเช่น สมการเส้นตรง ซึ่งเป็นสมการอันดับ 1 ที่ผ่านจุด 2 จุด ดังรูป 6.1a และสมการพาราโบลา ที่ผ่านจุด 3 จุด ดังรูป 6.1b และสมการพหุนามอันดับ 3 (cubic) ที่ผ่านจุด 4 จุด ดังรูป 6.1c



รูป 6.1

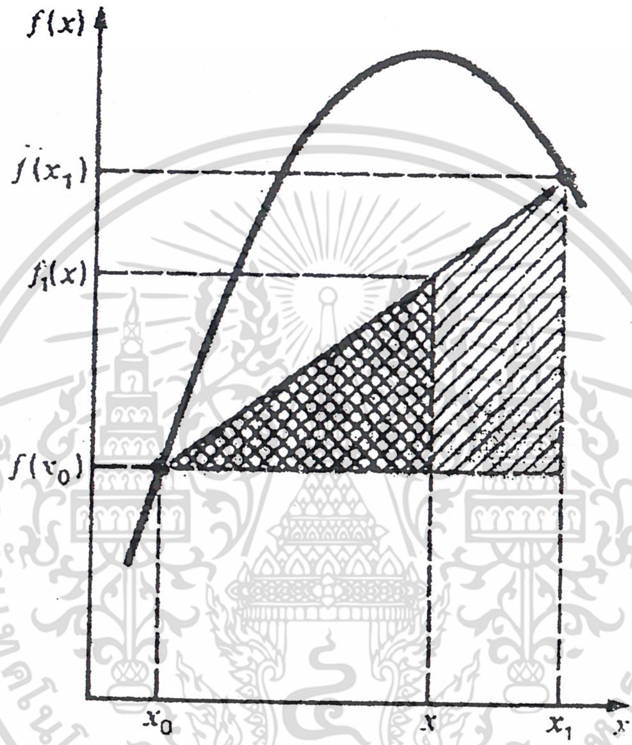
ถ้ามีข้อมูล n + 1 จุด จะได้สมการพหุนามที่มีอันดับที่ n เพียงสมการเดียวที่ผ่านจุด n+1 จุด นั้น

6.1 NEWTON'S DIVIDED-DIFFERENCE INTERPOLATING POLYNOMIALS

ก่อนที่จะกล่าวถึงรูปแบบทั่วไป จะกล่าวถึงในรูปแบบ linear และ quadratic ก่อนเป็นพื้นฐาน

6.1.1 Linear Interpolation

รูปแบบเส้นตรง (straight line) เป็นรูปแบบที่ง่ายที่สุด แสดงได้ดังรูป 2 โดยใช้คุณสมบัติสามเหลี่ยมคล้าย จะได้



รูป 6.2

$$\frac{f_1(x)-f(x_0)}{x-x_0}=\frac{f(x_i)-f(x_0)}{x_1-x_0}$$

$$f_1(x)=f(x_0)+\frac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_0)$$

LINEAR-INTERPOLATION FORMULA

$f_1(x)$ คือ ฟังก์ชันพหุนามอันดับ 1 ที่ ในการประมาณค่า

โดยทั่วไป ถ้าช่วงระหว่างจุดข้อมูลน้อยลง จะประมาณค่าได้ใกล้เคียงมากขึ้น

6.1.2 Quadratic Interpolation

ถ้ามีข้อมูล 3 จุดสามารถใช้ฟังก์ชันพหุนามอันดับ 2 เพื่อหาสมการที่ผ่านจุด 3 จุด

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) \quad (1)$$

$$= b_0 + b_1 x - b_1 x_0 + b_2 x^2 + b_2 x_0 x_1 - b_2 x x_0 - b_2 x x_1$$

หรือ

$$f_2(x) = a_0 + a_1 x + a_2 x^2$$

เมื่อ

$$a_0 = b_0 - b_1 x_0 - b_2 x_0 x_1$$
$$a_1 = b_1 - b_2 x_0 - b_2 x_1$$
$$a_2 = b_2$$

ในการหาค่าสัมประสิทธิ์ของสมการ (1)

ให้ $x = x_0$ จะได้ $b_0 = f(x_0)$

แทนค่า b_0 ในสมการ (1)

ให้ $x = x_1$ จะได้ $b_1 = \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}$

แทนค่า b_0, b_1 ในสมการ (1)

ให้ $x = x_2$

$$b_2 = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

6.1.3 GENERAL NEWTON'S INTERPOLATING POLYNOMIAL

รูปแบบทั่วไปฟังก์ชันพหุนามอันดับ n คือ

$$f_n(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + \ldots + b_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

$$b_0 = f(x_0)$$
$$b_1 = f[x_1, x_0]$$
$$b_2 = f[x_2, x_1, x_0]$$

.

.

.

$$b_n = f[x_n, x_{n-1}, \ldots, x_2, x_1, x_0]$$

เมื่อ

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j}$$

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k}$$

$$f[x_n, x_{n-1}, \ldots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \ldots, x_1] - f[x_{n-1}, x_{n-2}, \ldots, x_0]}{x_n - x_0}$$

TABLE OF DIVIDED DIFFERENCES

i	x_i	$f(x_i)$	*first*	*second*	*third*
0	x_0	$f(x_0)$	$f[x_1, x_0]$	$f[x_2, x_1, x_0]$	$f[x_3, x_2, x_1, x_0]$
1	x_1	$f(x_1)$	$f[x_2, x_1]$	$f[x_3, x_2, x_1]$	
2	x_2	$f(x_2)$	$f[x_3, x_2]$		
3	x_3	$f(x_3)$			

6.2 LAGRANG INTERPOLATING POLYNOMIALS

Lagrang Interpolating Polynomial เป็นรูปแบบที่นำมาจาก Newton Polynomial เช่น Linear Interpolation

จาก newton จะได้

$$f_1(x) = f(x_0) + (x - x_0) f[x_1, x_0] \tag{2}$$

$$f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x_1)}{x_1 - x_0} + \frac{f(x_0)}{x_0 - x_1} \tag{3}$$

แทนสมการ (3) ลงใน (2)

$$f_1(x) = f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1) + \frac{(x - x_0)}{(x_0 - x_1)} f(x_0)$$

$$f_1(x) = \frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1)$$

ซึ่งได้รูปทั่วไป คือ

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

เมื่อ

$$L_i(x) = \prod_{j=0, j \neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

$\prod = "\, product \quad of \,"$

ถ้าเป็น linear (n=1)

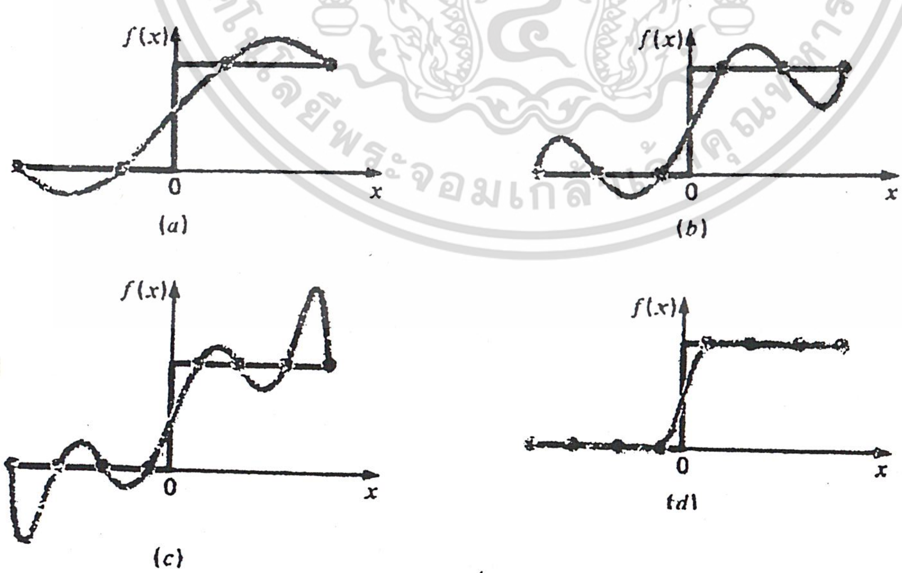
$$f_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

ถ้าเป็น second-order (n=2)

$$f_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2)$$

6.3 SPINE INTERPOLATION

เป็นวิธีการหนึ่งที่ใช้ฟังก์ชันพหุนามที่มีอันดับต่ำ ๆ กับสับเซ็ตของจุดข้อมูล และนำฟังก์ชันพหุนามทั้งหมดนั้นมาเชื่อมต่อกัน เรียกว่า SPINE รูป 7d เป็นฟังก์ชันที่ส่วนใหญ่จะเรียบ รูป 7a ถึง 7c ใช้ฟังก์ชันพหุนามที่มีอันดับสูง ๆ ซึ่งมีแนวโน้มที่แกว่งไปมา



รูป 6.3

6.3.1 LINEAR SPINE

ซึ่งเป็นเส้นตรง เชื่อมระหว่างจุด 2 จุด จะได้เซ็ตของสมการเส้นตรง ดังนี้

$f(x) = f(x_0) + m_0(x - x_0); x_0 \le x \le x_1$

$f(x) = f(x_1) + m_1(x - x_1); x_1 \le x \le x_2$

.

.

.

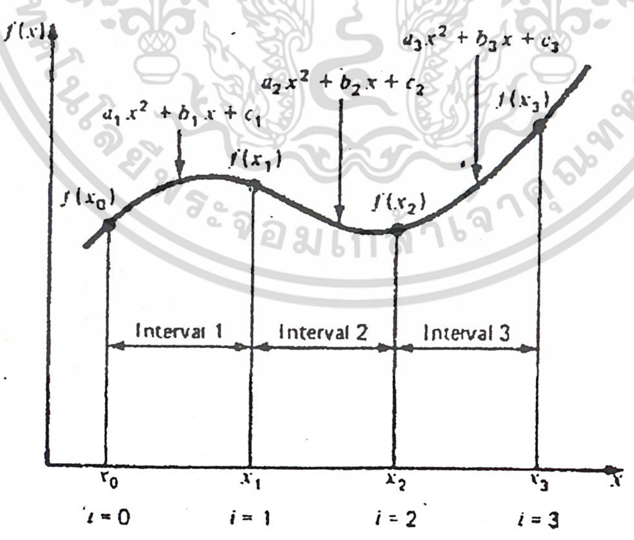
$f(x) = f(x_{n-1}) + m_{n-1}(x - x_{n-1}); x_{n-1} \le x \le x_n$

m_i = ความชันของเส้นตรง

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

6.3.2 QUADRATIC SPINE

จะใช้ฟังก์ชันพหุนามอันดับ 2 ในแต่ละช่วงระหว่างจุดข้อมูล อนุพันธ์อันดับหนึ่งจะต่อเนื่องที่จุด knot รูป 8 แสดง ถ้าข้อมูลมี n+1 จุด จะมี n ช่วง แต่ละช่วงใช้ฟังก์ชันพหุนามอันดับ 2



รูป 6.4

$f_i(x) = a_i x^2 + b_i x + c_i$

ถ้าข้อมูลมีทั้งหมด n+1 จุด $i = 0,1,2,\ldots,n$ จะมีทั้งหมด n ช่วง

มีตัวไม่รู้ค่า 3 ตัว ในแต่ละช่วง แต่มีทั้งหมด n ช่วง เพราะฉะนั้น จะมีตัวไม่รู้ค่า 3n ตัว ดังนั้นต้องมีสมการทั้งหมด 3n สมการ มีดังนี้

1. ค่าฟังก์ชันจะต้องเท่ากันที่จุด interior knots

$$a_{i-1}x_{i-1}^2 + b_{i-1}x_{i-1} + c_{i-1} = f(x_{i-1})$$
$$a_i x_{i-1}^2 + b_i x_{i-1} + c_i = f(x_{i-1})$$

$i = 2$ ถึง n

n-1 interior knots และ 2(n-1) สมการ

2. ฟังก์ชันแรกและสุดท้าย จะต้องผ่าน end points

$$a_1 x_0^2 + b_1 x_0 + c_1 = f(x_0)$$
$$a_n x_n^2 + b_n x_n + c_n = f(x_n)$$

2 สมการ

3. อนุพันธ์อันดับ 1 ที่จุด interior knots จะต้องเท่ากัน

$$f'(x) = 2ax + b$$
$$2a_{i-1}x_{i-1} + b_{i-1} = 2a_i x_{i-1} + b_i$$

$i = 2$ ถึง n

n-1 interior knots และ n-1 สมการ

4. สมมติให้อนุพันธ์อันดับ 2 เท่ากับศูนย์ ที่จุดแรก (first point)

$$f''(x) = 2a$$
$$2a_1 = 0$$
$$a_1 = 0$$

1 สมการ

หรือ 2 จุดแรก (first two points) เชื่อมกันด้วยเส้นตรง นั่นเอง

6.3.3 CUBIC SPINE

จะใช้ฟังก์ชันพหุนามอันดับ 3 $f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$

ถ้าข้อมูลมีทั้งหมด n+1 จุด $i = 0,1,2,\ldots,n$ จะมีทั้งหมด n ช่วง เพราะฉะนั้น มีตัวไม่รู้ค่า 4n ตัว ดังนั้นต้องมี

สมการทั้งหมด 4n สมการ มีดังนี้

1. ค่าฟังก์ชันจะต้องเท่ากัน ที่จุด interior knots

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

$$a_i x_{i-1}^3 + b_i x_{i-1}{}^2 + c_i x_{i-1} + d_i = f(x_{i-1})$$

2(n-1) สมการ

2. ฟังก์ชันแรกและสุดท้าย จะต้องผ่าน end points

$$a_0 x_0^3 + b_0 x_0{}^2 + c_0 x_0 + d_0 = f(x_0)$$

$$a_n x_n^3 + b_n x_n{}^2 + c_n x_n + d_n = f(x_n)$$

2 สมการ

3. อนุพันธ์อันดับ 1 ที่จุด interior knots จะต้องเท่ากัน

$$f'(x) = 3ax^2 + 2bx + c$$

$$3a_{i-1}x_{i-1}^2 + 2b_{i-1}x_{i-1} + c_{i-1} = 3a_i x_{i-1}^2 + 2b_i x_{i-1} + c_i$$

n-1 สมการ

4. อนุพันธ์อันดับ 2 ที่ interior knots จะต้องเท่ากัน

$$f''(x) = 6ax + 2b$$

$$6a_{i-1}x_{i-1} + 2b_{i-1} = 6a_i x_{i-1} + 2b_i$$

n-1 สมการ

5. อนุพันธ์อันดับ 2 เท่ากับศูนย์ที่จุดแรก และจุดสุดท้าย

$$6a_0 x_0 + 2b_0 = 0$$

$$6a_n x_n + 2b_n = 0$$

2 สมการ

ORDINARY DIFFERENTIAL EQUATIONS

- EULER'S METHOD
- MODIFICATIONS AND IMPROVEMENTS OF EULER'S METHOD
- RUNGE-KUTTA METHODS
- SYSTEM OF EQUATIONS

7. ORDINARY DIFFERENTIAL EQUATIONS

(การแก้ระบบสมการอนุพันธ์)

สมการอนุพันธ์ ซึ่งอยู่ในรูป

$$\frac{dy}{dx} = f(x, y) = slope$$

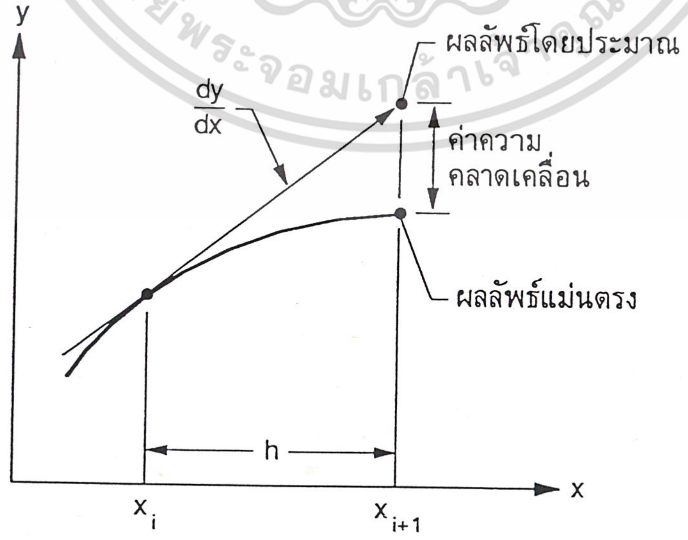
มีวิธีการแก้สมการหลายวิธีด้วยกัน

7.1 EULER'S METHOD

ระเบียบวิธีของออยเลอร์ (Euler's method) จัดได้ว่าเป็นระเบียบวิธีที่ง่ายแก่การทำความเข้าใจมากที่สุดสำหรับการแก้สมการเชิงอนุพันธ์สามัญที่อยู่ในรูปแบบโดยทั่วไปดังนี้

$$\frac{dy}{dx} = f(x, y) \qquad (1)$$

หลักการที่ใช้ในการแก้สมการเชิงอนุพันธ์สามัญนี้ สามารถอธิบายได้โดยใช้รูป 7.1



รูป 7.1

จากรูป 7.1 นี้ เราจะหาค่าผลลัพธ์โดยประมาณ y_{i+1} ที่ x_{i+1} จากผลลัพธ์ y_i ซึ่งรู้ค่าที่ x_i โดยใช้ค่าความชันที่ x_i ดังนี้

$$\frac{dy}{dx} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{h} \tag{2}$$

โดย $h = x_{i+1} - x_i$ คือความกว้างช่วง (step size) ที่ใช้ในการคำนวณ ทำการแทนค่าของความชันที่ x_i จาก
สมการ(2) นี้ ลงในสมการ (1) จะได้

$$\frac{y_{i+1} - y_i}{h} = f(x_i, y_i)$$

นั่นคือ

$$y_{i+1} = y_i + f(x_i, y_i)h \tag{3}$$

ซึ่งหมายความว่า เราสามารถทำการคำนวณโดยเริ่มจากเงื่อนไขเริ่มต้นของ y_i ที่ x_i และสามารถคำนวณค่า y_{i+1} ใหม่จากความกว้างช่วง h ที่กำหนดให้ และจากรูป 7.1 เราจะเห็นได้ว่า ความเที่ยงตรงของผลลัพธ์โดยประมาณนั้นจะขึ้นอยู่กับค่า h ที่ใช้ในการคำนวณนี้ กล่าวคือ ยิ่งใช้ h ที่มีค่าน้อยเท่าใด ก็จะได้ผลลัพธ์ที่มีความเที่ยงตรงมากยิ่งขึ้นเท่านั้น

7.1.1 ค่าผิดพลาดของวิธี EULER

คำตอบของการแก้สมการอนุพันธ์ สามารถแบ่งค่าผิดพลาดเป็น 2 แบบ

1.ค่าผิดพลาดเพราะตัดปลาย (Truncation error)

ค่าผิดพลาดเพราะตัดปลาย ยังแบ่งออกเป็น 2 ส่วน คือ local truncation error เป็นค่าผิดพลาดที่เกิดขึ้นในขั้นตอนนั้น และ propagated truncation error ซึ่งเป็นค่าผิดพลาดที่มีผลมาจากการประมาณในขั้นตอนแรก ทำให้ค่าประมาณในขั้นตอนต่อไปเกิดค่าผิดพลาดขึ้นด้วย เช่น ค่า y ที่ได้จากการประมาณในขั้นตอนแรก เมื่อนำค่า y นี้ไปใช้ในขั้นตอนต่อไป ก็จะทำให้ค่า y ที่ได้ครั้งใหม่เกิดค่าผิดพลาดขึ้นด้วย ผลบวกของค่าผิดพลาดทั้ง 2 ส่วน เรียกว่า global truncation error

7.1.1.2 ค่าผิดพลาดเพราะปัดเศษ (Round-off Error)

ดังที่กล่าวไว้ในหัวข้อที่ 1

7.2 MODIFICATION AND IMPROVEMENTS OF EULER'S METHODS

7.2.1 HEUN'S METHOD

ระเบียบวิธีของฮวน เป็นระเบียบวิธีที่ดัดแปลงมาจากระเบียบวิธีของออยเลอร์เพื่อก่อให้เกิดผลลัพธ์จากการแก้สมการเชิงอนุพันธ์สามัญที่มีความเที่ยงตรงมากยิ่งขึ้น ในระเบียบวิธีของออยเลอร์ดังแสดงในรูป 7.1 จะเห็นได้ว่าผลลัพธ์โดยประมาณที่คำนวณได้จะมีความเที่ยงตรงมากขึ้นหากเราลดขนาดช่วงความกว้าง h ลง หรือหากเราสามารถคำนวณค่าความชัน $(y' = dy/dx)$ ให้มีความเที่ยงตรงได้มากยิ่งขึ้น ค่าความชันในระเบียบวิธีของออยเลอร์นั้นคำนวณที่จุดต้นของความกว้างช่วงที่ตำแหน่ง x_i นั่นคือ

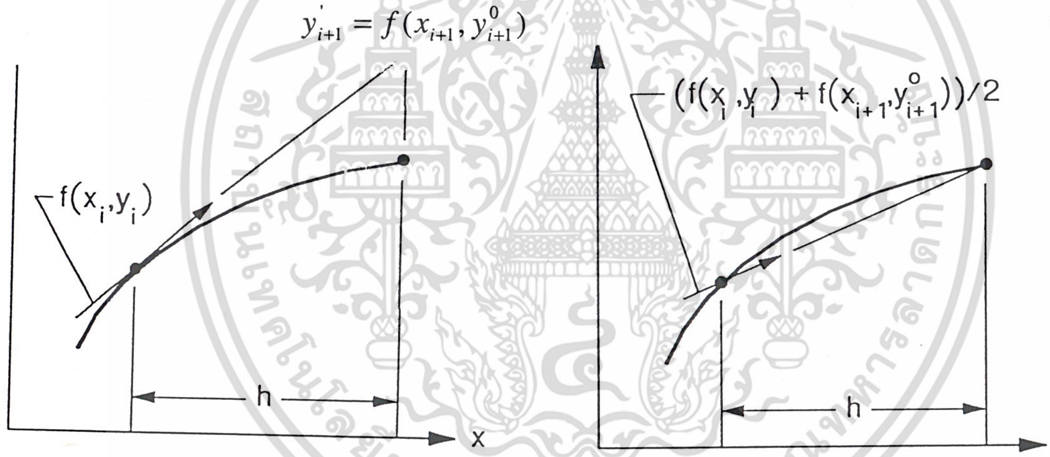
$$y_i' = f(x_i, y_i) \tag{4}$$

แล้วจึงนำไปใช้ในการคำนวณค่าผลลัพธ์โดยประมาณที่ x_{i+1} ดังนี้

$$y_{i+1}^0 = y_i + f(x_i, y_i)h \tag{5}$$

หากเรานำค่าที่ได้นี้ไปคำนวณหาค่าความชันที่จุดปลายของความกว้างช่วงที่ x_{i+1} ดังแสดงในรูป 7.2a เราจะได้

$$y_{i+1}' = f(x_{i+1}, y_{i+1}^0) \tag{6}$$



(a) ความชันที่ x_i และ x_{i+1} (b) ค่าความชันเฉลี่ย

รูป 7.2

หลักการในระเบียบวิธีของฮวน ก็คือ เราจะนำค่าความชันที่จุดต้นและจุดปลายของความกว้างช่วงนี้มาทำการเฉลี่ยกัน ซึ่งจะก่อให้เกิดค่าความชันเฉลี่ยที่ใกล้เคียงความเป็นจริงมากขึ้น ดังแสดงในรูป 7.2b และหากเราใช้ค่าความชันเฉลี่ยนี้ในการคำนวณ เราควรจะได้ผลลัพธ์ที่มีความเที่ยงตรงมากขึ้นตามไปด้วย

จากสมการ (4) และ (5) ค่าความชันเฉลี่ยคือ

$$\bar{y}' = \frac{y_i' + y_{i+1}'}{2} = \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)}{2} \tag{7}$$

ดังนั้นคำตอบของผลลัพธ์โดยประมาณที่ตำแหน่ง x_{i+1} คือ

$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)}{2} h \tag{8}$$

โดยสรุป ระเบียบวิธีของฮวนประกอบด้วย 2 ขั้นตอน ขั้นตอนแรกเป็นการทำนายค่าโดยประมาณที่ x_{i+1} ดังแสดงในสมการ (5) ก่อให้เกิดผลลัพธ์ y_{i+1}^0 ที่เรียกว่าตัวทำนาย (predictor) ผลลัพธ์ที่ได้นี้จะนำไปใช้ในการคำนวณหาค่าความชันที่ x_{i+1} ค่าความชันที่ได้นี้จะนำไปใช้หาค่าความชันเฉลี่ยในขั้นตอนที่สองเพื่อใช้ในการคำนวณผลลัพธ์ y_{i+1} ที่เรียกว่าตัวแก้ (corrector) ซึ่งจะมีความเที่ยงตรงสูงขึ้น

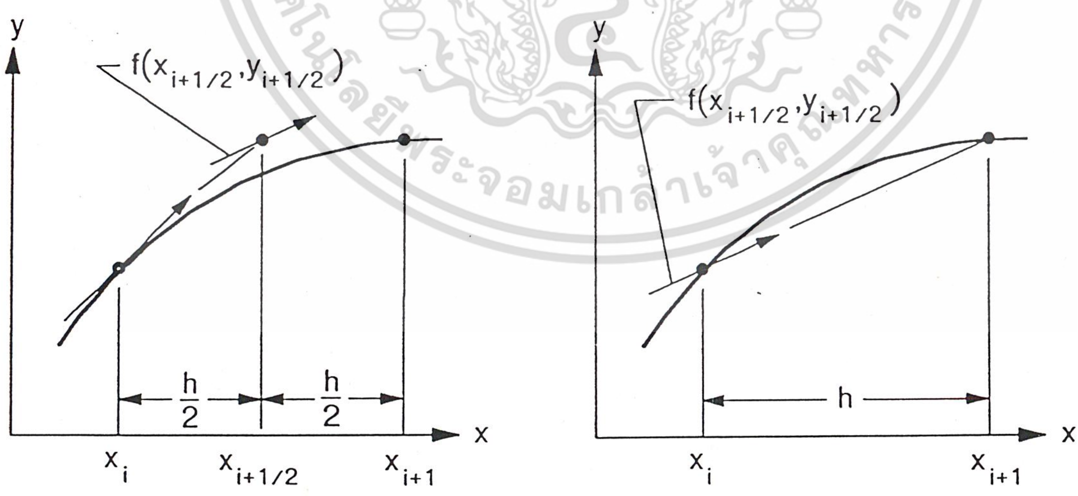
ดังนั้น เราอาจกล่าวได้ว่า ระเบียบวิธีการของฮวน เป็นระเบียบวิธีการที่ประกอบด้วยการคำนวณตัวทำนายและตัวแก้ (predictor-corrector method) ที่มีขั้นตอนดังนี้

ตัวทำนาย : $y_{i+1}^0 = y_i + f(x_i, y_i)h$ (9a)

ตัวแก้ : $y_{i+1} = y_i + \dfrac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)}{2}h$ (9b)

7.2.2 MODIFICATIONS AND IMPROVEMENTS OF EULER'S METHODS

ระเบียบวิธีของออยเลอร์ที่ปรับปรุงแล้ว (modification Euler's method) เป็นระเบียบวิธีที่เกิดจากความเข้าใจในทำนองเดียวกันกับความเข้าใจในระเบียบวิธีของฮวน กล่าวคือ ผลลัพธ์ที่คำนวณได้จะมีความเที่ยงตรงมากขึ้นหากเราสามารถหาค่าความชันที่จะนำมาใช้ในการคำนวณได้แม่นยำมากขึ้น ดังนั้นในระเบียบวิธีของออยเลอร์ที่ปรับปรุงแล้วนี้ เราจะหาค่าความชันที่จุดกึ่งกลางของขนาดช่วงความกว้าง h ดังแสดงในรูป 7.3(b)



(a) การหาความชันเฉลี่ย (b) การใช้ความชันเฉลี่ยทั้งช่วงกว้าง

รูป 7.3

เราจะเริ่มจากการใช้ระเบียบวิธีออยเลอร์เดิมในการคำนวณผลลัพธ์ที่จุดกึ่งกลางของความกว้างช่วง ดังนี้

$$y_{i+1/2} = y_i + f(x_i, y_i)\frac{h}{2} \tag{10}$$

จากนั้น จึงคำนวณค่าความชันที่จุดกึ่งกลางของความกว้างช่วงนี้

$$y'_{i+1/2} = f(x_{i+1/2}, y_{i+1/2}) \tag{11}$$

แล้วจึงใช้ความชันที่จุดกึ่งกลางของความกว้างช่วงที่คำนวณได้นี้ในการคำนวณผลลัพธ์ที่จุดปลายของช่วง

$$y_{i+1} = y_i + f(x_{i+1/2}, y_{i+1/2})h \tag{12}$$

7.3 RUNGE-KUTTA METHODS

ระเบียบวิธีของรุงเง-คุตตา จัดได้ว่าเป็นระเบียบวิธีที่ได้รับความนิยมและใช้กันอย่างกว้างขวางโดยเฉพาะในการคำนวณที่ต้องการผลลัพธ์ที่มีความเที่ยงตรงสูง แนวความคิดที่ใช้ในการประดิษฐ์ระเบียบวิธีของรุงเง-คุตตานี้คือ การหาค่าความชันที่มีความเที่ยงตรงสูงเพื่อก่อให้เกิดผลลัพธ์ที่มีความเที่ยงตรงสูงตามมา สมการหลักที่ใช้ในการคำนวณผลลัพธ์ในระเบียบวิธีของรุงเง-คุตตานั้นอยู่ในรูปแบบทำนองเดียวกันกับระเบียบวิธีต่าง ๆ ที่เราได้ศึกษามาก่อนหน้านี้ นั่นคือ

$$y_{i+1} = y_i + \phi\,(x_i, y_i, h)h \tag{13}$$

โดย $\phi\,(x_i, y_i, h)$ เรียกว่าฟังก์ชันส่วนเพิ่ม (increment function) ซึ่งมีความหมายของความชันเฉลี่ยตลอดขนาดช่วงความกว้าง h ที่จะนำไปใช้ในการคำนวณหาส่วนที่เพิ่มขึ้นจากผลลัพธ์เดิม ฟังก์ชันส่วนเพิ่มนี้สามารถเขียนให้อยู่ในรูปแบบโดยทั่วไปได้ดังนี้

$$\phi = a_1k_1 + a_2k_2 + \ldots + a_nk_n \tag{14}$$

โดย $a_i, i = 1,2,\ldots,n$ เป็นค่าคงที่ และ

$$k_1 = f(x_i, y_i) \quad (15a)$$
$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h) \quad (15b)$$
$$k_3 = f(x_i + p_2 h, y_i + q_{21} k_1 h + q_{22} k_2 h) \quad (15c)$$

.

.

.

$$k_n = f(x_i + p_{n-1} h, y_i + q_{n-1} k_1 h + q_{n-1,2} k_2 h + \ldots + q_{n-1,n-1} k_{n-1} h) \quad (15n)$$

โดยตัวห้อย n บ่งถึงอันดับที่ของระเบียบวิธีรุงเง-คุตตาที่เลือกใช้ เช่น เมื่อ n=1 เราจะเรียกว่า เป็นระเบียบวิธีรุงเง-คุตตาอันดับหนึ่ง ซึ่งหากเราพิจารณาสมการดังกล่าวที่เกิดขึ้นสำหรับกรณีนี้แล้ว จะพบว่ามีความทัดเทียมกับสมการจากระเบียบววิธีของออยเลอร์ ในทำนองเดียวกัน เมื่อเราเลือกใช้ n=2 เราจะเรียกว่าเป็นระเบียบวิธีรุงเง-คุตตาอันดับสอง เป็นต้น ค่า $k_i, i = 1,2,\ldots,n$ ในสมการ (15) ขึ้นอยู่กับฟังก์ชันของสมการเชิงอนุพันธ์สามัญที่กำหนดให้มา ส่วนค่า p และ q ต่าง ๆ นั้นเป็นค่าคงที่ซึ่งเราจะศึกษาขั้นตอนในการหาค่าเหล่านี้ต่อไป อนึ่ง หากเราพิจารณาสมการ (15) นี้ เราจะพบว่า เราจำเป็นต้องรู้ค่า k_1 ก่อนทำการคำนวณค่า k_2 และรู้ค่า k_2 ก่อนทำการคำนวณหาค่า k_3 เช่นนี้เรื่อยไป เพื่อให้เกิดความเข้าใจในวิธีการนี้ได้โดยง่าย เราจะมาศึกษาระเบียบวิธีรุงเง-คุตตาอันดับสอง (n=2) โดยละเอียดในหัวข้อย่อยต่อไปนี้

7.3.1 SECOND-ORDER RUNGE-KUTTA (รุงเง-คุตตาอันดับสอง)

จากสมการ (13)-(15) ที่อยู่ในรูปแบบทั่วไปหากเราใช้ n=2 จะก่อให้เกิดระเบียบวิธีของรุงเง-คุตตาอันดับสอง ดังนี้

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2) h \quad (16)$$

โดย $k_1 = f(x_i, y_i)$ (17)

และ $k_2 = (x_i + p_1 h, y_i + q_{11} k_1 h)$ (18)

จะเห็นได้ว่า เรามีค่าคงตัวที่ไม่รู้ค่าทั้งหมด 4 ตัวคือ a_1, a_2, p_1 และ q_{11} ซึ่งจะทำการหาโดยใช้หลักการที่ว่า ระเบียบวิธีของรุงเง-คุตตาอันดับสองดังแสดงในการแสดง (16) นี้จะให้ผลลัพธ์ที่มีความเที่ยงตรงเท่ากับผลลัพธ์ที่เกิดจากการใช้อนุกรมเทย์เลอร์ที่อยู่ในรูปแบบดังนี้

$$y_{i+1} = y_i + f(x_i, y_i)h + f'(x_i, y_i)\frac{h^2}{2} + \dots \tag{19}$$

จากกฎลูกโซ่ (chain-rule) ค่าอนุพันธ์อันดับหนึ่งของฟังก์ชันในสมการ (19) สามารถเขียนได้ดังนี้

$$f'(x_i, y_i) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx} = \frac{\partial f}{\partial x} + f(x_i, y_i)\frac{\partial f}{\partial y} \tag{20}$$

แทนสมการ (18) ลงในสมการ (19) จะได้

$$y_{i+1} = y_i + f(x_i, y_i)h + \left(\frac{\partial f}{\partial x} + f(x_i, y_i)\frac{\partial f}{\partial y}\right)\frac{h^2}{2} + \dots.. \tag{21}$$

ดังนั้น เพื่อหาค่าคงตัวที่ไม่รู้ค่าทั้งหมด 4 ตัว ที่จะก่อให้เกิดผลลัพธ์ที่มีความเที่ยงตรงเท่ากับอนุกรมเทย์เลอร์ดังกล่าว เราจำเป็นต้องเขียนสมการของรุงเง-คุตตา (16) ให้อยู่ในรูปแบบของสมการอนุกรมเทย์เลอร์ (21) นี้ แล้วทำการเปรียบเทียบสัมประสิทธิ์ แต่ก่อนอื่น เราจะประยุกต์อนุกรมเทย์เลอร์ที่ประกอบด้วย 2 ตัวแปร คือ

$$g(x + r, y + s) = g(x, y) + r\frac{\partial g}{\partial x} + s\frac{\partial g}{\partial y} + \dots.. \tag{22}$$

เข้ากับสมการ (18) ซึ่งจะก่อให้เกิด

$$k_2 = f(x_i, y_i) + p_1 h\frac{\partial f}{\partial x} + q_{11}k_{11}h\frac{\partial f}{\partial y} + \dots... \tag{23}$$

จากนั้นแทนค่า k_1 จากสมการ (17) ลงในสมการนี้ แล้วจึงแทนสมการนี้ลงในสมการรุงเง-คุตตา (16) และทำการจัดพจน์จะได้

$$\begin{aligned} y_{i+1} &= y_i + [a_1 f(x_i, y_i) + a_2 f(x_i, y_i)]h \\ &\quad + \left[a_2 p_1\frac{\partial f}{\partial x} + a_2 q_{11} f(x_i, y_i)\frac{\partial f}{\partial y}\right]h^2 + \dots\dots \end{aligned} \tag{24}$$

หากเราเปรียบเทียบสัมประสิทธิ์ของสมการรุงเง-คุตตา (18) ที่เกิดขึ้นนี้กับสัมประสิทธิ์ของสมการอนุกรมเทย์เลอร์ (21) เราจะพบว่า

$$a_1 + a_2 = 1 \tag{25a}$$

$$a_2 p_1 = \frac{1}{2} \tag{25b}$$

$$a_2 q_{11} = \frac{1}{2} \tag{25c}$$

กล่าวคือ มีเงื่อนไขที่เกิดขึ้นเพียง 3 เงื่อนไขในขณะที่มีตัวไม่รู้ค่าถึง 4 ตัว ดังนั้นเราสามารถสมมติค่าใดค่าหนึ่งขึ้นมาก่อน แล้วจึงหาค่าที่เหลืออีก 3 ค่าจากเงื่อนไขทั้ง 3 นี้

สมมติว่าเราเลือก $a_2 = 1/2$ ดังนั้นจากสมการ (25a-c) เราจะได้

$$a_1 = 1/2 \quad ; \quad p_1 = 1 \quad ; \quad q_{11} = 1 \tag{26}$$

ดังนั้น สูตรของรุงเง-คุตตาในสมการ (16) คือ

$$y_{i+1} = y_i + (\frac{1}{2}k_1 + \frac{1}{2}k_2)h \tag{27}$$

และสมการ (17)-(18) คือ

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

$k_2 = f(x_i + h, y_i + hk_1) =$ ความชันที่จุดปลายของช่วงกว้าง h

ดังนั้น สมการ(50) ที่เกิดขึ้นนี้ก็คือ สมการ(9b) ซึ่งเป็นสูตรในระเบียบวิธีของฮวนที่เราได้ศึกษามาแล้ว

หากเราเลือก $a_2 = 1$ ดังนั้นจากสมการ(25a-c) เราจะได้

$$a_1 = 0 \quad ; \quad p_1 = 1/2 \quad ; \quad q_{11} = 1/2 \tag{28}$$

ดังนั้น สูตรของรุงเง-คุตตาในสมการ (16) คือ

$$y_{i+1} = y_i + (0 + k_2)h \tag{29}$$

และสมการ(17)-(18) คือ

$$k_1 = f(x_i, y_i) = \text{ ความชันที่จุดต้นของช่วงกว้าง h}$$

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

ดังนั้น สมการ(27) ที่เกิดขึ้นนี้ก็คือ สมการ(12) ซึ่งเป็นสูตรอันเกิดจากระเบียบวิธีของออยเลอร์ที่ปรับปรุงแล้วซึ่งเราได้ทำการศึกษาผ่านมาแล้ว

ผลลัพธ์ต่าง ๆ ที่เกิดขึ้นดังเช่นแสดงในสมการ(27) และ (28) จากระเบียบวิธีรุงเง-คุตตาอันดับสอง (n=2) นี้ ล้วนมีค่าความผิดพลาดในรูปแบบของความกว้างช่วงอันดับสอง $o(h^2)$ ซึ่งมีความหมายโดยทั่วไปว่า หากเราลดขนาดของความกว้างช่วงที่ใช้ในการคำนวณลงไปครึ่งหนึ่ง ค่าความผิดพลาดที่เกิดขึ้นจะลดลงไปเป็นเศษหนึ่งส่วนสี่จากค่าความผิดพลาดเดิม เป็นต้น

7.3.2 THIRD-ORDER RUNGE-KUTTA METHODS (รุงเง-คุตตาอันดับสาม)

สมการ(13)-(15) สำหรับระเบียบวิธีของรุงเง-คุตตาที่อยู่ในรูปแบบทั่วไป สามารถนำมาดัดแปลง เมื่อ n=3 เพื่อก่อให้เกิดระเบียบวิธีของรุงเง-คุตตาอันดับสาม รูปแบบของสมการรุงเง-คุตตาอันดับสาม ที่ใช้กันโดยทั่วไปซึ่งสามารถนำไปใช้ในการประดิษฐ์โปรแกรมคอมพิวเตอร์ได้โดยตรงนั้นมีลักษณะดังนี้

$$y_{i+1} = y_i + \left[\frac{1}{6}(k_1 + 4k_2 + k_3)\right]h \qquad (30)$$

โดย

$$k_1 = f(x_i, y_i) \qquad (31a)$$

$$k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1) \qquad (31b)$$

$$k_3 = f(x_i + h, y_i - hk_1 + 2hk_2) \qquad (31c)$$

ผลลัพธ์ที่คำนวณได้จากสมการ(30) โดยระเบียบวิธีของรุงเง-คุตตาอันดับสามนี้จะมีค่าความผิดพลาดในรูปแบบของความกว้างช่วงอันดับสาม $o(h^3)$ ขั้นตอนการคำนวณในระเบียบวิธีรุงเง-คุตตาอันดับสามนี้สามารถทำได้โดยง่าย

7.3.3 FOURT-ORDER KUNGE-KUTTA METHODS (รุงเง-คุตตาอันดับสี่)

ระเบียบวิธีรุงเง-คุตตาอันดับสี่ ถูกจัดว่าเป็นระเบียบวิธีที่ได้รับความนิยมใช้กันโดยแพร่หลาย การดัดแปลงสมการ (13)-(15) ที่อยู่ในรูปแบบทั่วไปโดยใช้ n = 4 ก่อให้เกิดสมการรุงเง-คุต

ตาอันดับสี่ ซึ่งให้ค่าความผิดพลาดในรูปแบบของความกว้างช่วงอันดับสี่ $o(h^4)$ลักษณะของผลลัพธ์ที่มีความเที่ยงตรงสูงนี้ทำให้ระเบียบวิธีการนี้ถูกนำไปประยุกต์กับงานการคำนวณในหลาย ๆ ด้านโดยเฉพาะในงานวิจัยค้นคว้าที่ต้องการความเที่ยงตรงสูง รูปแบบของสมการรุงเง-คุตตาอันดับสี่ที่ใช้กันโดยทั่วไปซึ่งสามารถนำไปประดิษฐ์เป็นโปรแกรมคอมพิวเตอร์ได้โดยตรงนั้นมีลักษณะดังนี้

$$y_{i+1} = y_i + \left[\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\right]h \tag{32}$$

โดย

$$k_1 = f(x_i, y_i) \tag{33a}$$

$$k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1) \tag{33b}$$

$$k_3 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_2) \tag{33c}$$

$$k_4 = f(x_i + h, y_i + hk_3) \tag{33d}$$

7.2 SYSTEM OF EQUATIONS

ปัญหาทางวิศวกรรมศาสตร์และวิทยาศาสตร์ส่วนใหญ่ไม่เพียงประกอบด้วยสมการเชิงอนุพันธ์อันดับหนึ่งเพียงสมการเดียวดังเช่นที่เราได้ศึกษาวิธีการแก้มาโดยตลอด แต่จะประกอบด้วยสมการเชิงอนุพันธ์อันดับหนึ่งหลายสมการที่เกี่ยวข้องสัมพันธ์กัน ซึ่งก่อให้เกิดระบบสมการที่ประกอบด้วย n สมการย่อยในรูปแบบดังนี้

$$\frac{dy_1}{dx} = f(x, y_1, y_2, ..., y_n) \tag{34a}$$

$$\frac{dy_2}{dx} = f(x, y_1, y_2, ..., y_n) \tag{34b}$$

.

.

.

$$\frac{dy_n}{dx} = f(x, y_1, y_2, ..., y_n) \tag{34n}$$

ระเบียบวิธีการต่าง ๆ ที่ใช้แก้สมการเชิงอนุพันธ์อันดับหนึ่งสมการเดียวดังที่เราได้ศึกษามาแล้ว อาทิเช่น ระเบียบวิธีของออยเลอร์ หรือ ระเบียบวิธีของรุงเง-คุตตาอันดับต่าง ๆ สามารถนำมาใช้แก้ระบบสมการเชิงอนุพันธ์ไปโดยพร้อมกันทั้งระบบสมการ

Mathematatica V.3

1. Running Mathematica

1.1 Notebook Interfaces

การรัน *MATHEMATICA* ด้วย notebook interface

ดับเบิ้ลคลิกที่ไอคอน *Mathematica*	การแสดงในรูปแบบกราฟฟิกสำหรับการเริ่มต้น *Mathematica*
mathematica	คำสั่งระบบปฏิบัติการ เพื่อสั่งเริ่มต้น *Mathematica*
การจบข้อความด้วย SHIFT-**R**ETURN	การป้อน input สำหรับ *Mathematica*
เลือก Quit จากเมนู	การออกจาก *Mathematica*

ถ้าเราใช้คอมพิวเตอร์โดยผ่านทาง graphical interface เราต้องดับเบิ้ลคลิกที่ไอคอน *Mathematica* เพื่อการเริ่มต้นของ *Mathematica* แต่ถ้าเราใช้คอมพิวเตอร์โดยผ่านทางระบบปฏิบัติการโดยพื้นฐานเป็นข้อความ (textually based O.S.) เราต้องพิมพ์คำสั่ง mathematica เพื่อการเริ่มต้นของ *Mathematica*

เมื่อ *Mathematica* เริ่มทำงาน มันจะแสดงหน้าจอว่างเพื่อให้เราทำการป้อน input ลงไป แล้วกดปุ่ม shift และ return (หมายถึง ปุ่ม enter นั่นเอง) โดยจะกดปุ่ม shift ค้างเอาไว้แล้วกดปุ่ม return จากนั้น *Mathematica* จะทำการประมวลผล input และแสดงผลลัพธ์มาให้

```
2+2
```

เราเพียงแค่พิมพ์ 2+2 แล้วกดปุ่ม shift - return จากนั้น *Mathematica* ทำการประมวลผล input แล้วเพิ่มข้อความ In[1] := ให้ และแสดงผลลัพธ์ หลังข้อความ Out[1] =

```
In[1] := 2+2

Out[1] = 4
```

ถ้าต้องการออกจาก *Mathematica* ทำได้โดยการเลือก Quit จากเมนูใน notebook interface

1.2 Text-Based Interfaces

การรัน *Mathematica* ด้วย text-based interface

math	คำสั่งระบบปฏิบัติการเพื่อสั่งเริ่มต้น *Mathematica*
การจบข้อความด้วย SHIFT-RETURN	การป้อน input สำหรับ *Mathematica* ที่ใช้ในทุกระบบ
การจบข้อความด้วย RETURN	รูปแบบอย่างง่ายของการป้อน input ที่ใช้ได้ในบางระบบ
CONTROL-D หรือ Quit[ ]	การออกจาก *Mathematica*

เพื่อการเริ่มต้น *Mathematica* เราจะต้องพิมพ์คำสั่ง math ที่ O.S. prompt ในบางระบบอาจจะเริ่มต้น *Mathematica* ด้วยการดับเบิ้ลคลิกที่ไอคอน *Mathematica* kernel

เมื่อ *Mathematica* เริ่มต้นการทำงาน มันจะพิมพ์ In[1] := ขึ้นมา เพื่อเป็นการบอกว่าพร้อมสำหรับการป้อน input แล้ว และเมื่อทำการป้อน input เสร็จในบางระบบจะกดปุ่ม shift-return บางระบบ จะกดปุ่ม return (หรือ enter) เพียงปุ่มเดียว หลังจากนั้น *Mathematica* จะทำการประมวลผล input และแสดงผลลัพธ์หลังข้อความ Out[1] =

สำหรับการออกจาก *Mathematica* นั้น ทำได้โดยการกด Control-D หรือ พิมพ์ Quit[ ] ที่ input prompt

2. Numerical calculations

2.1 เลขคณิต (Arithmetic)

เราสามารถจัดการเกี่ยวกับเลขคณิตได้ด้วย *Mathematica* ดังเช่นเดียวกับการใช้เครื่องคิดเลข

ตัวอย่าง เช่น

```
In[1] := 2.3+5.63
```

```
Out[1] = 7.93
```

- แสดงผลรวมของจำนวน 2 จำนวน

```
In[2] := 2.4 / 8.9 ^ 2
Out[2] = 0.0302992
```

- เครื่องหมาย (/) แทนการหาร และเครื่องหมาย (^) แทนการยกกำลัง

```
In[3] := 2 3 4
Out[3] = 24
```

- ใน *Mathematica* การเว้นช่องว่าง หมายถึง การคูณ แต่สามารถใช้เครื่องหมาย (*) ได้ถ้าต้องการ

Operation เกี่ยวกับเลขคณิตใน Mathematica

1.	*x ^ y*	การยกกำลัง
2.	*–x*	จำนวนลบ
3.	*x / y*	การหาร
4.	*x y z* หรือ *x*y*z*	การคูณ
5.	*x+y+z*	การบวก

operations เลขคณิต ใน *Mathematica* จะถูกจับกลุ่มตามแบบแผนมาตราฐานทางคณิตศาสตร์ เช่น 2^3+4 จะหมายถึง (2^3)+4 ไม่ใช่ 2^(3+4) เราสามารถควบคุมการจับกลุ่มการจับกลุ่มได้โดยใช้เครื่องหมาย ( ) เพื่อให้แน่ใจ

```
In[1] := 2 ^ 3 + 4
Out[1] = 12


In[2] := (2 ^ 3) + 4
Out[2] = 12
```

- จะเห็นว่าการป้อน input ทั้งสองแบบนี้จะได้ output เท่ากัน

2.2 ค่าที่แท้จริงและค่าประมาณ (Exact and Approximate Results)

ในการใช้เครื่องคิดเลขนั้น ผลลัพธ์ที่ได้นั้นมีความถูกต้องเพียงบางส่วนเท่านั้น นั่นคือ เครื่องคิดเลขจะแสดงผลลัพธ์ถึงตำแหน่งที่ 10 (แล้วแต่ขนาดขนาดของหน้าปัดเครื่องคิดเลข) ซึ่งผลลัพธ์จริงๆ อาจจะมีถึงตำแหน่งที่ 20 ก็ได้

สำหรับใน *Mathematica* จะให้ค่าที่แท้จริงของผลลัพธ์ เช่น ผลลัพธ์ของ 2^{100} ค่าที่แท้จริงมีถึง 31 ตำแหน่ง

In[1] := 2 ^ 100

Out[1] = 1267650600228229401496703205376

แต่เราสามารถบอกให้ *Mathematica* แสดงผลลัพธ์เป็นค่าโดยประมาณเชิงตัวเลข ได้โดยพิมพ์ // N หลังการป้อน input (N ต้องเป็นอักษรตัวพิมพ์ใหญ่) เช่น

In[2] := 2 ^100 // N

Out[2] = 1.26765×10^{30}

ในการป้อน input ถ้าเราพิมพ์เลขจำนวนเต็ม เช่น 7,9,10,... *Mathematica* จะสันนิษฐานว่าจำนวนนั้นเป็นค่าที่แท้จริง (exact value) แต่ถ้าพิมพ์จำนวนที่มีจุด ทศนิยม เช่น 2.5,1.23,... *Mathematica* จะสันนิษฐานว่าจำนวนนั้นมีความถูกต้องถึงทศนิยมตำแหน่งที่ถูกพิมพ์

In[1] := 452 / 62

Out[1] = $\frac{226}{31}$

- เราป้อน input เป็นอัตราส่วนของจำนวนเต็ม (หรือค่าที่แท้จริง) output จะแสดงผลลัพธ์ที่ได้จากการลดทอนเศษส่วน

In[2] := 452.3 / 62

Out[2] = 7.29516

- เมื่อไรก็ตามที่ป้อน input เป็นจำนวนที่มีจุดทศนิยม *Mathematica* จะแสดงผลลัพธ์เป็นค่าประมาณ

2.3 ฟังก์ชันทางคณิตศาสตร์ (Some Mathematical Functions)

ใน *Mathematica* ได้รวบรวมฟังก์ชันทางคณิตศาสตร์ไว้มากมาย ที่นำมานี้เป็นเพียงบางส่วนของฟังก์ชันทั้งหมด ที่มีการใช้กันทั่วไป

Squrt[x]	รากของ x ($\sqrt{x}$)
Exp[x]	เอ็กซ์โปเนนเชียล (e^x)
Log[x]	ลอการิทึมธรรมชาติ ($\log_e x$)
Log[b,x]	ลอการิทึมฐาน b ($\log_b x$)
Sin[x],Cos[x],Tan[x]	ฟังก์ชันตรีโกณมิติ
ArcSin[x],ArcCos[x],ArcTan[x]	ส่วนกลับของฟังก์ชันตรีโกณมิติ
n!	แฟคทอเรียล (n เป็นจำนวนเต็ม)
Abs[x]	ค่าสัมบูรณ์ของ x
Round[x]	จำนวนเต็มที่ใกล้ x ที่สุด
Mod[x]	n มอดูโล m
Max[x,y,..],Min[x,y,..]	ค่าสูงสุด,ค่าต่ำสุดของ x,y,..

สิ่งสำคัญ 2 สิ่งที่เกี่ยวกับการใช้ฟังก์ชันทางคณิตศาสตร์ใน *Mathematica* ก็คือ

1.Arguments ของฟังก์ชันจะต้องอยู่ภายในเครื่องหมาย [ ] (square braket)

2.ชื่อของฟังก์ชันจะต้องขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่

```
In[1] := Log[8.4]
Out[1] = 2.1282
```

- นี่เป็นการหาค่าลอการิทึมของ 8.4 จะสังเกตเห็นว่าชื่อฟังก์ชันขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่ (Log) และ argument (นั่นคือ 8.4) อยู่ภายในเครื่องหมาย [ ]

สำหรับการคำนวณของฟังก์ชันทางคณิตศาสตร์ *Mathematica* พยายามที่จะแสดงผลลัพธ์เป็นค่าที่แท้จริง (exact value) เมื่อเราป้อน input เป็นค่าที่แท้จริง เช่น

```
In[2] := Sqrt[16]
Out[2] = 4
```

- ค่าของรากที่สองของ 16 เท่ากับ 4 เป็นค่าที่แท้จริงซึ่งเป็นจำนวนเต็มพอดี

```
In[3] := Sqrt[2 .]
Out[3] = 1.41421
```

- การที่เราป้อน argument เป็นเลขที่มีจุดทศนิยม (2 .) เป็นการบอกให้ *Mathematica* แสดงผลลัพธ์เป็นค่าประมาณเชิงตัวเลข หรืออาจจะพิมพ์ // N หลังจากป้อน input ผลลัพธ์ก็จะออกมาเหมือนกัน ดังนี้

```
In[4] := Sqrt[2] // N
Out[4] = 1.41421
```

แต่ถ้าป้อน argument เป็นจำนวนเต็ม และไม่ได้จบด้วย // N แล้ว *Mathematica* จะแสดงผลลัพธ์เป็นค่าที่แท้จริงในรูปแบบของสัญลักษณ์ ดังนี้

In[5] := Sqrt[2]

Out[5] = $\sqrt{2}$

พิจารณาตัวอย่างอื่น

```
In[6] := 30!
Out[6] = 265252859811121910586363084800000000
```

แต่ถ้าจบด้วย // N จะได้

In[7] := 30!

Out[7] = 2.65253×10^{32}

ค่าคงที่ทางคณิตศาสตร์ที่ใช้กันทั่วไป

Pi	$\pi = 3.14159$
E	$e \approx 2.71828$
Degree	$\frac{\pi}{180}$: การเปลี่ยนจากองศาเป็นเรเดียน
I	$i = \sqrt{-1}$
Infinity	∞

สังเกตว่าชื่อของค่าคงที่ทั้งหมดจะต้องขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่

```
In[8] := Pi ^ 2  // N
Out[8] = 9.8696
```

- เป็นการหาค่าของ π^2

```
In[9] := Sin[Pi / 2]
Out[9] = 1
```

- เป็นการหาค่าของ sin $(\frac{\pi}{2})$ สังเกตว่า argument ของฟังก์ชันตรีโกณมิติจะต้องเป็นเรเดียนเสมอ

```
In[10] := Sin[20  Degree]  // N
Out[[10] = 0.34202
```

- Degree คือการเปลี่ยนองศาเป็นเรเดียน ในที่นี้คือการเปลี่ยน 20° เป็นเรเดียน $(20\times\frac{\pi}{180})$ แล้วหาค่า Sin

2.4 การคำนวณแบบกำหนดเอง (Arbitrary-Precision Calculations)

ใน *Mathematica* เราสามารถกำหนดจำนวนของตำแหน่งทศนิยม ที่ผลลัพธ์จะแสดงได้เอง โดยใช้ฟังก์ชันดังนี้

expr // N หรือ N[expr]	ค่าประมาณเชิงตัวเลขของ expr
N[expr,n]	ค่าประมาณเชิงตัวเลขของ expr ที่แสดงทศนิยมถึงตำแหน่งที่ n

เช่น

```
In[1] := N[Pi]
Out[1] = 3.14159
```

- จะแสดงค่าของ π ถึงทศนิยมที่มีนัยสำคัญ แต่ถ้าเรากำหนดตำแหน่งทศนิยมเอง จะได้ดังนี้

```
In[2] := N[Pi,40]
Out[2] = 3.1415926535897932384626433832795028841972
```

หรือ การหาค่าของ $\sqrt{7}$ ถึงทศนิยมตำแหน่งที่ 30

```
In[3] := N[Sqrt[7],30]
Out[3] = 2.645751311064590590501615753639
```

2.5 จำนวนเชิงซ้อน (Complex Numbers)

operation ของ จำนวนเชิงซ้อน

$x + Iy$	จำนวนเชิงซ้อน x + Iy
Re[z]	ส่วนจริง
Im[z]	ส่วนจินตภาพ
Conjugate[z]	conjugate เชิงซ้อน z* หรือ $\bar{z}$
Abs[z]	ค่าสัมบูรณ์ $\|z\|$
Arg[z]	argument $\varnothing$ ใน $\|z\|e^{i\varnothing}$

การคำนวณจำนวนเชิงซ้อนใน *Mathematica* จะทำโดยการรวมค่าคงที่ I เข้าไปด้วย ซึ่งค่าคงที่ I มีค่าเท่ากับ $\sqrt{-1}$ และต้องแน่ใจว่าพิมพ์ I เป็นอักษรตัวพิมพ์ใหญ่

<u>ตัวอย่าง</u>

```
In[1] := Sqrt[-4]
Out[1] = 2I
```

- ผลลัพธ์ที่ได้เป็นจำนวนจินตภาพ 2I

```
In[2] := (4 + 3I ) / (2 – I)
Out[2] = 1+2I
```

- เป็นการหาอัตราส่วนของจำนวนเชิงซ้อน 2 จำนวน

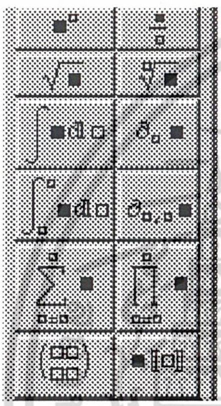
2.6 เครื่องหมายทางคณิตศาสตร์ใน Notebook

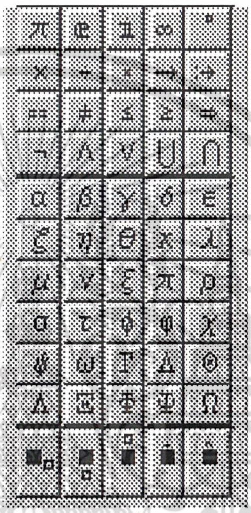
(Mathematical Notation in Notebooks)

ถ้าเราใช้ text-based ติดต่อ (interface) กับ *Mathematica* แล้ว input ที่เราป้อนเข้าไปจะต้องประกอบด้วยตัวอักษร (character) เพียงเท่านั้น แต่ถ้าเราใช้ notebook
ติดต่อกับ *Mathematica* แล้วการป้อน input สามารถทำในรูปแบบอื่นๆ ได้

โดยปรกติจะมี palettes จัดเตรียมไว้ให้ ซึ่งเราสามารถ click ที่ปุ่มของสัญลักษณ์ที่เราต้องการใช้ได้เลย และเราสามารถเลือก standard palettes ได้โดยใช้เมนูไฟล์ ซึ่งจะมี submenu palettes อยู่ในเมนูไฟล์

__ตัวอย่าง__





นอกจากนี้เรายังสามารถป้อน input โดยใช้คีย์พิเศษ (special keys) ได้ดังนี้

ปุ่มESC+p+ปุ่มESC	สัญลักษณ์ π
ปุ่มESC+inf+ปุ่มESC	สัญลักษณ์ ∞
ปุ่มCRT+^ หรือ ปุ่มCRT+6	ใส่เลขยกกำลัง
ปุ่มCRT+/	ใส่ตัวส่วนของเศษส่วน
ปุ่มCRT+@ หรือ ปุ่มCRT+2	ใส่เลขใน $\sqrt{\ }$
ปุ่มCRT+SPACE	แสดงผลจากการยกกำลัง,ใส่ตัวส่วน,ใส่ square root

<u>ตัวอย่าง</u> การหาค่าของ $\frac{\pi^2}{6}$ สามารถป้อน input ได้ในรูปแบบต่างๆ ดังนี้

In[1] := N[Pi ^ 2 / 6]

Out[1] = 1.64493

In[2] := N[$\frac{\pi^2}{6}$]

Out[2] = 1.64493

In[3] := N[ESC+p+ESC CRT+^ 2 CRT+SPACE CRT+/ 6 CRT+SPACE]

Out[3] = 1.64493

คำสั่งของ Mathematica ที่ใช้ใน Visual Basic

MLInitialize()	เป็นการเริ่มต้นการใช้ฟังก์ชัน Library ใน MathLink
MLOpenArgv()	เป็นการทำการติดต่อกับMathLink โดยผ่านทางตัวแปร argv
MLOpenString()	เป็นการทำการติดต่อกับMathLink โดยผ่านทางตัวแปรที่ประกาศเป็น character
MLClose()	เป็นการทำการยกเลิกการติดต่อกับ MathLink
MLEndPacket()	เป็นการแสดง packet สุดท้ายในการเชื่อมต่อกับ MLink
MLNextPacket()	เป็นการหาจุดเริ่มต้นของ packet ต่อไปในการเชื่อมต่อกับ MLink
MLNewPacket()	เป็นการข้ามไปที่จุดสุดท้ายของ packet ปัจจุบันในการเชื่อมต่อกับ MLink
MLReady()	เป็นการทดลองการรออ่านของข้อมูลในการเชื่อมต่อกับ MLink
MLFlush()	เป็นการช่วยในการบรรจุข้อมูลนอกที่อยู่นอก Buffers เพื่อรอที่จะส่งบนข้อมูลในการเชื่อมต่อกับ Mlink

คำสั่ง MathLink ที่ใช้ใน Visual Basic

Packets

ILLEGALPKT As Integer = 0

CALLPKT As Integer = 7

EVALUATEPKT As Integer = 13

RETURNPKT As Integer = 3

INPUTNAMEPKT As Integer = 8

ENTERTEXTPKT As Integer = 14

ENTEREXPRPKT As Integer = 15

OUTPUTNAMEPKT As Integer = 9

RETURNTEXTPKT As Integer = 4

RETURNEXPRPKT As Integer = 16

DISPLAYPKT As Integer = 11

DISPLAYENDPKT As Integer = 12

MESSAGEPKT As Integer = 5

TEXTPKT As Integer = 2

INPUTPKT As Integer = 1

INPUTSTRPKT As Integer = 21

MENUPKT As Integer = 6

SYNTAXPKT As Integer = 10

SUSPENDPKT As Integer = 17

RESUMEPKT As Integer = 18

BEGINDLGPKT As Integer = 19

ENDDLGPKT As Integer = 20

FIRSTUSERPKT As Integer = 128

LASTUSERPKT As Integer = 255

Messages

MLTerminateMessage As Long = 1

MLInterruptMessage As Long = 2

MLAbortMessage As Long = 3

Error Codes

MLEUNKNOWN As Integer = -1

MLEOK As Integer = 0

MLEDEAD As Integer = 1

MLEGBAD As Integer = 2

MLEGSEQ = 3

MLEPBTK = 4

MLEPSEQ = 5

MLEPBIG = 6

MLEOVFL = 7

MLEMEM = 8

MLEACCEPT = 9

MLECONNECT = 10

MLECLOSED = 11

MLENOACK = 15

MLENODATA = 16

MLENOTDELIVERED = 17

MLENOMSG = 18

MLEFAILED = 19

MLEPUTENDPACKET = 21

MLENEXTPACKET = 22

MLEUNKNOWNPACKET = 23

MLEGETENDPACKET = 24

MLEABORT = 25

MLEINIT = 32

MLEARGV = 33

MLEPROTOCOL = 34

MLEMODE = 35

MLELAUNCH = 36

MLELAUNCHAGAIN = 37

MLELAUNCHSPACE = 38

MLENOPARENT = 39

MLENAMETAKEN = 40

MLENOLISTEN = 41

MLEBADNAME = 42

MLEBADHOST = 43

MLERESOURCE = 44

Tokens

MLTKSTR = 34

MLTKSYM = 35

MLTKINT = 43

MLTKREAL = 42

MLTKFUNC = 70

MLTKERROR = 0

Function

MLInitialize As Long

MLDeinitialize As Long

MLOpenString (ByVal ep As Long, ByVal command_line As String, ByRef errp As Long) As Long

MLLoopbackOpen (ByVal ep As Long, ByRef errp As Long) As Long

MLConnect (ByVal mlp As Long) As Integer

MLClose (ByVal mlp As Long)

MLError (ByVal mlp As Long) As Integer

MLClearError (ByVal mlp As Long) As Integer

MLPutMessage (ByVal mlp As Long, ByVal msg As Long) As Integer

MLGetNext (ByVal mlp As Long) As Integer

MLGetType (ByVal mlp As Long) As Integer

MLGetArgCount (ByVal mlp As Long, ByRef countp As Long) As Integer

MLBytesToGet (ByVal mlp As Long, ByRef leftp As Long) As Integer

MLNewPacket (ByVal mlp As Long) As Integer

MLReady (ByVal mlp As Long) As Integer

```
MLPutNext (ByVal mlp As Long, ByVal tok As Integer) As Integer
MLPutArgCount (ByVal mlp As Long, ByVal argc As Long) As Integer
MLBytesToPut (ByVal mlp As Long, ByRef leftp As Long) As Integer
MLEndPacket (ByVal mlp As Long) As Integer
MLFlush (ByVal mlp As Long) As Integer
MLGetShortInteger (ByVal mlp As Long, ByRef hp As Integer) As Integer
MLGetInteger Alias "MLGetShortInteger" (ByVal mlp As Long, ByRef ip As Integer) As
        Integer
MLGetLongInteger (ByVal mlp As Long, ByRef lp As Long) As Integer
MLGetFloat (ByVal mlp As Long, ByRef fp As Single) As Integer
MLGetDouble (ByVal mlp As Long, ByRef dp As Double) As Integer
MLGet16BitCharacters  (ByVal mlp As Long, ByRef chars_left As Long, ByRef buf As Any,
        ByVal cardof_buf As Long, ByRef got As Long) As Integer
MLCheckFunction (ByVal mlp As Long, ByVal s As String, ByRef countp As Long) As Integer
MLPutShortInteger (ByVal mlp As Long, ByVal h As Integer) As Integer
MLPutInteger Alias "MLPutShortInteger" (ByVal mlp As Long, ByVal i As Integer) As Integer
MLPutLongInteger (ByVal mlp As Long, ByVal l As Long) As Integer
MLPutFloat (ByVal mlp As Long, ByVal f As Double) As Integer
MLPutDouble (ByVal mlp As Long, ByVal d As Double) As Integer
MLPutByteString (ByVal mlp As Long, ByVal s As String, ByVal len1 As Long) As Integer
MLPutByteSymbol (ByVal mlp As Long, ByVal s As String, ByVal len1 As Long) As Integer
MLPutFunction (ByVal mlp As Long, ByVal s As String, ByVal argc As Long) As Integer
MLPutSize (ByVal mlp As Long, ByVal size As Long) As Integer
MLPutData (ByVal mlp As Long, ByVal buff As String, ByVal len1 As Long) As Integer
MLCreateMark (ByVal mlp As Long) As Long
MLSeekMark (ByVal mlp As Long, ByVal mark As Long, ByVal index As Long) As Long
Declare Sub MLDestroyMark (ByVal mlp As Long, ByVal mark As Long)
MLTransferExpression (ByVal dmlp As Long, ByVal smlp As Long) As Integer
MLNextPacket (ByVal mlp As Long) As Integer
MLActivate Alias "MLConnect" (ByVal mlp As Long) As Integer
MLGetReal Alias "MLGetDouble" (ByVal mlp As Long, ByRef dp As Double) As Integer
MLPutReal Alias "MLPutDouble" (ByVal mlp As Long, ByVal d As Double) As Integer
```

VISUAL BASIC

ประวัติของภาษาเบสิกโดยย่อ

ภาษาเบสิกถูกคิดค้นขึ้นมาในปี ค.ศ.1963 โดย John Kemeny และ Thomas Kurtrz แห่งสถาบัน Dartmouth College โดยมีจุดมุ่งหมายที่จะออกแบบภาษาเบสิกขึ้นมาเพื่อใช้สอนหลักการเขียนโปรแกรม โดยเน้นที่ความชัดเจนรวดเร็วและประสิทธิภาพในการเรียนรู้ พวกเขาสามารถสร้างภาษาเบสิกได้เป็นผลสำเร็จ โดยยกเลิกการใช้ภาษาควบคุมงาน (Job Control Language) และหันมาใช้ภาษาที่ใช้สำหรับสร้างโปรแกรมอื่นๆ โดยการใช้ขั้นตอนการคอมไพล์และเชื่อมโยง เช่น ภาษาฟอร์แทรน และภาษาแอสเซมบลี จึงทำให้ภาษาเบสิกเป็นภาษาที่ง่ายต่อการต่อการใช้งานภาษาแรก ที่เน้นให้ผู้ใช้สนใจขั้นตอนการทำงานในส่วนของฮาร์ดแวร์ ภาษาเบสิกในเวอร์ชันแรกๆ มีคุณสมบัติต่างๆ หลายประการ อันเป็นที่รู้จักกันทั่วไป อาทิเช่น ในแต่ละบันทัดของโปรแกรมจะถูกขึ้นต้นด้วยหมายเลขบรรทัด ไม่มีการย่อหน้าในแต่ละกลุ่มคำสั่ง

คุณลักษณะของภาษาเบสิกดังที่กล่าวข้างต้นนั้น ก่อให้เกิดความยุ่งยากในการทำความเข้าใจขั้นตอนการทำงานของโปรแกรม ซึ่งถูกเรียกว่าสปาเก็ตตี้โค้ด (spaghetti code) ที่เรียกเช่นนี้ก็เพราะว่า ความต่อเนื่องของโปรแกรมจะถูกเชื่อมโยงกันนกลับไปกลับมาคล้ายกับเส้นสปาเก็ตตี้บนจาน ถึงแม้จะเป็นโปรแกรมที่สั้นๆ แต่ทำให้คุณลำบากในการทำความเข้าใจได้โดยง่าย จึงนับว่าเป็นความบังเอิญอย่างยิ่งที่ภาษาเบสิกยังคงเป็นที่ยอมรับจนถึงทุกวันนี้

ในปัจจุบันภาษาเบสิกถูกมองว่าเป็นภาษาเด็กเล่น ซึ่งไม่เหมาะสมกับงานโปรแกรมในปัจจุบัน แต่อย่างไรก็ดีภาษาเบสิกก็ยังคงถูกพัฒนาขึ้นมาอย่างต่อเนื่อง จากภาษาที่ช้าไม่มีโครงสร้างและแปลโปรแกรมทีละคำสั่ง กลายไปเป็นภาษาที่รวดเร็วมีโครงสร้างแน่นอน และแปลโปรแกรมแบบภาษาชั้นสูง ทำให้มีความเหมาะสมในการสร้างแอปพลิเคชันต่างๆ ได้หลากหลายขึ้น เป็นผลให้ Hewlett Packard Company, Microsoft Corporation และอีกหลายบริษัทได้ผลิตเวอร์ชันต่างๆ ของภาษาเบสิกออกมา โดยมีการพัฒนารายละเอียดการใช้งานให้มีความก้าวหน้ายิ่งขึ้น

การพัฒนาของภาษาเบสิก

ความก้าวหน้าของภาษาเบสิกนั้นถูกพัมนาควบคู่ไปกับการปฏิวัติทางคอมพิวเตอร์ส่วนบุคคล โดยในระหว่างคริสต์ทศวรรษที่1970ไมโครซอฟต์ได้ริเริ่มที่จะพัฒนาให้พัฒนาภาษาเบสิกเป็นตัวแปลภาษาพื้นฐานในไมโครโปรเซสเซอร์ของคอมพิวเตอร์ ต่อมาบริษัทRadio Shack TRS-8Oได้แนะนำภาษาเบสิก(และหลักการจัดการส่วนบุคคล)ให้แก่สาธารณชนได้รู้จักไมโครซอฟต์เบสิกเวอร์ชั่นแรกยังคงถูกใช้มาจนถึงทุกวันนี้ โดยไม่ได้มีการดัดแปลงอะไรมากมาย โดยอยู่ในรูปของGW-BASIC ซึ่งเป็นตัวแปลภาษาเบสิกที่อยู่ในระบบปฏิบัติการ MS-DOS เวอร์ชั่นล่าสุด

ถึงแม้ว่า GW-BASIC จะเป็นภาษาที่สามารถทำการคำนวณ และทำงานพื้นฐานง่ายๆ ได้อย่างรวดเร็ว แต่อย่างไรก็ดี มันยังมีลักษณะรายละเอียดบางอย่างที่ดูคล้ายกับภาษาเด็กเล่นอยู่ดี มันยังมีลักษณะรายละเอียดบางอย่างที่ดูคล้ายกับภาษาเด็กเล่นอยู่ดี ซึ่งนักพัฒนาซอฟต์แวร์ที่ไม่สนใจจุดนี้จะนิยมเลือกใช้ภาษา GW-BASIC ในการผลิตซอฟต์แวร์ออกสู่ตลาดจึงทำให้ยูทิลิตี้บน MS-DOS ไม่ถูกเขียนขึ้นมาขายในรูปของแบตช์ไฟล์ เนื่องจากโปรแกรมจะทำงานช้าเกินไป ต้องมีโปรแกรมต้นแบบให้กับผู้ใช้ และยังเป็นการไม่จำเป็นเนื่องจากยังมีวิธีอื่นๆ อีกหลายวิธีที่จะสร้างโปรแกรมเช่นนั้นขึ้นมา

ในปี ค.ศ.1982 การเกิดขึ้นของ Microsoft QuickBasic ทำให้เกิดการปฏิวัติภาษาเบสิกขึ้นมา และมีการจดลิขสิทธิ์ให้เป็นภาษาที่ใช้ในการพัฒนาบน MS-DOS ภาษา Quick Basic เป็นภาษาที่มีประสิทธิภาพในการตอบโต้กับผู้ใช้ และมีความรวดเร็วในการแปลชุดคำสั่งของโปรแกรม ซึ่งเป็นลักษณะเดิมของ GW-BASIC และมีการยกเลิกการใช้หมายเลขบรรทัดและเพิ่มลักษณะของโปรแกรมที่ทันสมัยเข้าไป อาทิเช่น มีโปรแกรมย่อย มีการกำหนดผู้ใช้ และชนิดโครงสร้างของข้อมูล มีกราฟิกที่ทันสมัยและจัดการเรื่องเสียงได้จนทำให้โปรแกรมเมอร์ภาษา QuickBasic มีความรู้สึกว่ากำลังใช้ภาษาที่ทันสมัยกว่า ภาษาซี ปาสคาล และฟอร์แทรน ภาษา QuickBasic ยังมีข้อดีอื่น ๆ อีก เช่น สามารถเลือกที่จะรัดได้ทั้งแบบทีละคำสั่ง หรือแบบแปลที่ละชุดคำสั่งได้โดยตัวมันเอง ซึ่งเป็นโปรแกรมที่เหมาะสมกันสภาพการตลาดในปัจจุบัน (ในปัจจุบันไมโครซอฟต์ก็ได้นำเอา Qbasic ซึ่งเป็นเวอร์ชันที่รันในแบบทีละคำสั่งของ QuickBasic มาเป็นส่วนหนึ่งของ MS-DOS)

Visual Basic

ในปัจจุบัน การปฏิวัติของไมโครซอฟต์วินโดวส์ ส่งผลดีทำให้เกิดความเป็นมาตราฐานในการาจัดการสภาวะแวดล้อมของระบบในการดึงเอาความสามารถที่มีอยู่ในตัวไมโครโปรเซสเซอร์แบบล่าสุดของบริษัท Intel แต่สำหรับผู้ใช้งานแล้วละก็ วินโดวส์ทำให้เครื่องคอมพิวเตอร์ส่วนบุคคลมีความเป็นส่วนตัวและใช้งานง่ายมากขึ้น ในขณะเดียวกันนักเขียนโปรแกรมก็ต้องเตรียมตัวเรียนรู้หลักการเขียนโปรแกรมใหม่เข้ามา เพื่อจะสามารถพัฒนาโปรแกรมที่สามารถใช้งานบนวินโดวส์ได้ และ Visual Basic ซึ่งเกิดจากการพัฒนาครั้งใหญ่ของภาษาเบสิกก็เป็นภาษาที่จะทำให้การเรียนรู้ที่จะสร้างแอปพลิเคชันบนวินโดวส์กลายเป็นเรื่องง่าย

ภาษาเบสิกมีการเปลี่ยนแปลงไปมากในช่วง 2 ทศวรรษที่ผ่านมา ในขณะที่ภาษา Visual Basic สำหรับ Windows เวอร์ชัน 5 ได้ถูกสร้างขึ้นมาโดยมีโปรแกรมประยุกต์บน Windows เป็นพันๆ โปรแกรมที่ถูกสร้างขึ้นมาโดยใช้ภาษา Visual Basic นี้ และยังมีอีกหลายโปรแกรมที่กำลังจะถูกพัฒนาโดยภาษา Visual Basic 5 ซึ่งเป็นเวอร์ชั่นที่ทันสมัยมาก

คุณสมบัติและข้อดีของ Visual Basic for Windows

Visual Basic for Windows มีคุณสมบัติที่ดีหลายประการ เป็นภาษาในฝันสำหรับการพัฒนาโปรแกรมต่างๆ บนวินโดวส์ คุณสมบัติซึ่งช่วยให้การเขียนโปรแกรมง่ายขึ้นประกอบไปด้วยเครื่องมือ(Tools) หลากหลายอันจะช่วยในการเขียนโปรแกรมแบบซับซ้อนบนวินโดวส์ให้สำเร็จได้โดยง่ายและรวดเร็ว

Visual Basic for Windows ยังคงไว้ด้วยข้อดีต่างๆ ของ Microsoft QuickBasic และยังเพิ่มเติมคุณสมบัติอันหลากหลายที่จะสนับสนุนให้ตัวมันเองเป็นโปรแกรมพื้นฐานสำหรับการพัฒนาโปรแกรมบน Microsoft Windows อีกด้วย ตัวอย่างเช่น กราฟิกเอาต์พุตที่สามารถถูกส่งออกไปยังส่วนต่างๆ ของวินโดวส์ หรือแม้กระทั่งไปยังเครื่องพิมพ์ คุณสามารถถูกส่งออกไปยังส่วนต่างๆ ของวินโดวส์ หรือแม้กระทั่งไปยังเครื่องพิมพ์ คุณสามารถเลือกสีสำหรับทำงานกราฟิกได้มากกว่า 16 ล้านเฉดสี (โดยวินโดวส์จะจัดการแสดงผลกราฟิกนั้นตามที่คุณต้องการ หรือมันจะลดลงมาได้เท่าที่ฮาร์ดแวร์ของเครื่องนั้นๆ จะสนับสนุนในการแสดงผลได้) โดยที่คุณไม่ต้องกังวลในส่วนนี้ว่ามีกระบวนการในการจัดการอย่างไร ไม่ว่าในตอนนี้หรือต่อไปในอนาคต

ข้อดีเหนือ QuickBasic อีกอย่างก็คือ การจัดการตัวแปลใน Visual Basic มีกฎเกณฑ์ ซึ่งง่ายในการเข้าใจและจดจำ เพราะว่ามันถูกพัฒนาให้ง่าย และมีประสิทธิภาพ โดยที่โปรแกรมของ Visual Basic จะประกอบไปด้วยไฟล์ 2 แบบ คือ Form และ Module ยกเว้นเมื่อ declare globally ที่อื่นตัวแปร และค่าคงที่ในโปรแกรมย่อย และฟังก์ชันนั้นจะเป็น Local สำหรับกระบวนการที่เกิดขึ้น (กฎต่างๆ ตรงนี้ คุณจะเข้าใจเอง เมื่อคุณมีประสบการณ์ในการใช้งาน Visual Basic สักพัก)

ข้อดีอีกอย่างหนึ่งของVisual Basic สำหรับผู้ที่จะมาเป็นนักโปรแกรมผู้ยิ่งใหญ่ก็คือ ถ้าคุณมีความคุ้นเคยอยู่กับQuickBasicอยู่แล้วก็ยิ่งจะง่ายต่อการใช้งาน Visual Basic และจะดียิ่งขึ้นถ้าคุณมีความรู้เล็กๆ น้อยๆ สำหรับการเขียนโปรแกรม Visual Basic ได้อย่างแท้จริงและ Visual Basic จะกลายเป็นเครื่องมือที่ดีที่สุดที่จะช่วยให้คุณสามารถเขียนโปรแกรมบนวินโดวส์ได้อย่างรวดเร็วและมีประสิทธิภาพ และถึงแม้คุณจะมีความเชี่ยวชาญในการใช้ภาษา C พัฒนาโปรแกรมบนวินโดวส์ก็ตาม คุณก็จะชอบการเรียกใช้งาน กระบวนการ Interface- Development ที่ Visual Basic มีให้

ภาษา Basic ได้ถูกพัฒนามาให้เป็นภาษาสำหรับการพัฒนาโปรแกรมแบบโต้ตอบ (Interactive) ดังตัวอย่างเช่น เป็นการง่ายที่จะใช้ GW-Basic รันคำสั่งเพียงสองสามคำสั่งเพื่อดูว่ามันทำงานอย่างไร มากกว่าที่จะเปิดตกราดูเอกสารมากมาย หลักการพัฒนางานแบบโต้ตอบนี้ ได้ถูกนำมาใช้ในการสร้างระบบงาน User interface สำหรับโปรแกรมของคุณบน Windows โดยใช้ Visual Basic

คงจะไม่สามารถเปรียบเทียบได้ว่าการออกแบบโปรแกรมด้วย Visual Basic นั้นจะดีกว่าอย่างไร จนกว่าคุณจะลืมว่า Interface design progress ของโปรแกรมเก่าๆ ที่คุณเขียนมานานแล้ว

นั้นมันมีกระบวนการทำงานแต่ละส่วนอย่างไร เมื่อนั้นคุณจะพบว่าการใช้โปรแกรมแบบโต้ตอบของ Visual Basic นั้นดีกว่า ง่ายกว่า และเร็วกว่าอย่างไร

Dynamic Link Libraries ของ Windows

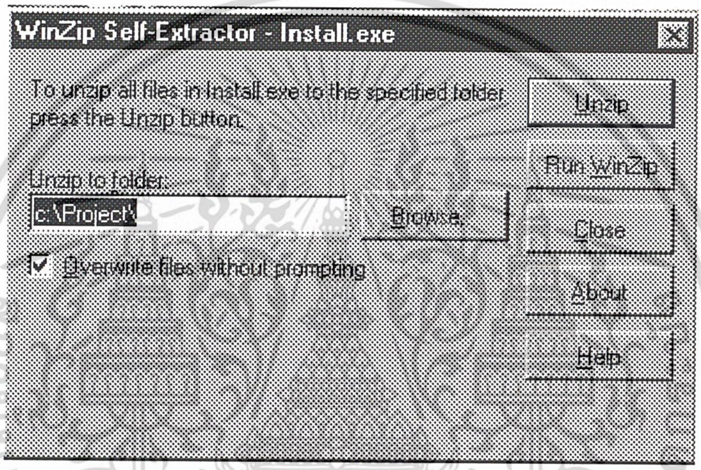
ข้อดีอีกข้อของ Visual Basic ก็คือ ความสามารถในการเพิ่มขยายของตัวมันเองถึงแม้ Visual Basic for Windows จะทำงานได้อย่างรวดเร็วเพียงใดก็ตาม แต่ในบางทีนั้น Optimized Code ของภาษา C นั้น จะสามารถทำงานได้เร็วกว่า ถ้าคุณมี C Compiler คุณก็จะสามารถสร้าง Dynamic Link Libraries ใช้เองได้

ประโยชน์ของ DLL เหล่านี้จะเกิดขึ้นกับคุณ เพราะมันเป็นส่วนเพิ่มประสิทธิภาพที่ระบบปฏิบัติการ Windows มีให้กับคุณอยู่แล้ว Microsoft Windows Software Development Kit (SDK) ได้อธิบายฟังก์ชันต่างๆ เหล่านี้ไว้ถึงกระบวนการทำงาน และยังให้รูปแบบ ในการเรียกใช้ฟังก์ชันเหล่านี้มาใช้งานจากภาษา C อีกด้วย นอกจากนี้ Visual Basic ในรุ่น Professional ยังได้รวม Special Help ซึ่งได้บรรจุการเรียกใช้ Standard DLLs ของ Windows ไว้ให้ด้วยแล้ว และเมื่อคุณลองใช้ Visual Basic เรียกเอาฟังก์ชันเหล่านี้มาใช้งาน คุณก็จะเข้าใจว่า Visual Basic นั้น ช่วยให้คุณประหยัดเวลาในการพัฒนาโปรแกรมเพียงใด

บทที่ 3
คู่มือการใช้โปรแกรม

ขั้นตอนการลงโปรแกรม

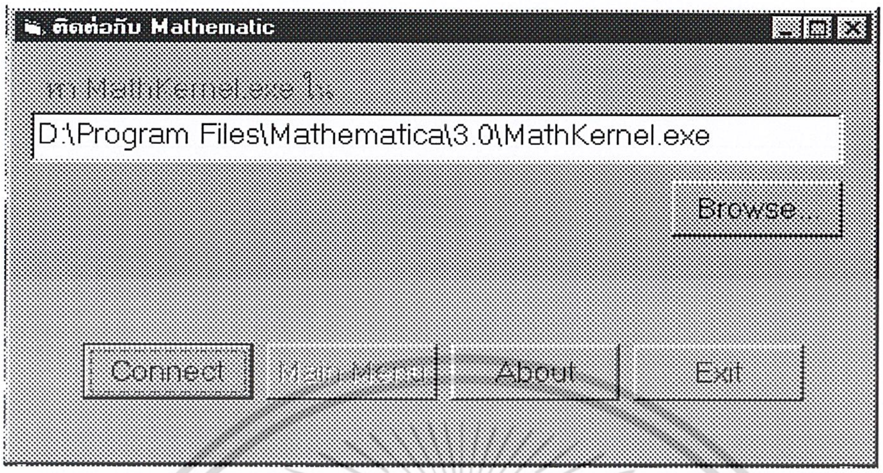
1. ให้นำแผ่นที่มีตัว Install ของโปรแกรมใส่ลงไปใน Drive |A: หรือถ้าเป็น CD-Rom ก็ใส่ไว้ที่เครื่องเล่น CD
2. เข้าไปใน Drive A: หรือใน Drive CD-Rom ทำการหา file ชื่อว่า Install.exe แล้วคลิกซ้าย 2 ครั้ง แล้วจะมีรูปแสดงให้เห็นดังนี้



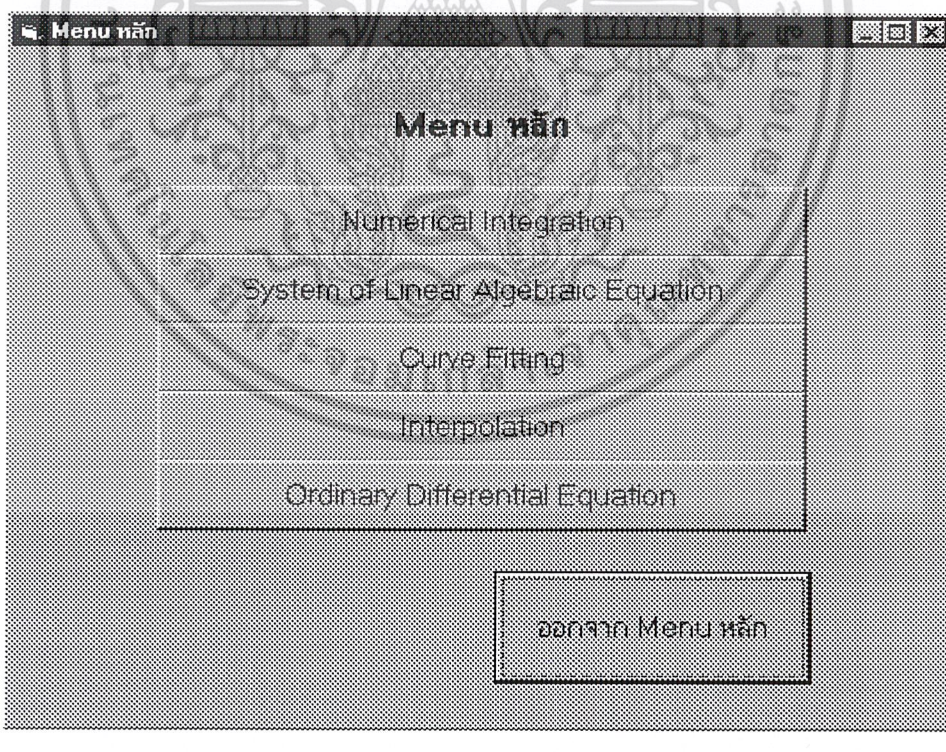
3. กดที่ปุ่ม Unzip ก็จะทำการเสร็จสิ้นการ Install

<u>ขั้นตอนการใช้โปรแกรม</u>

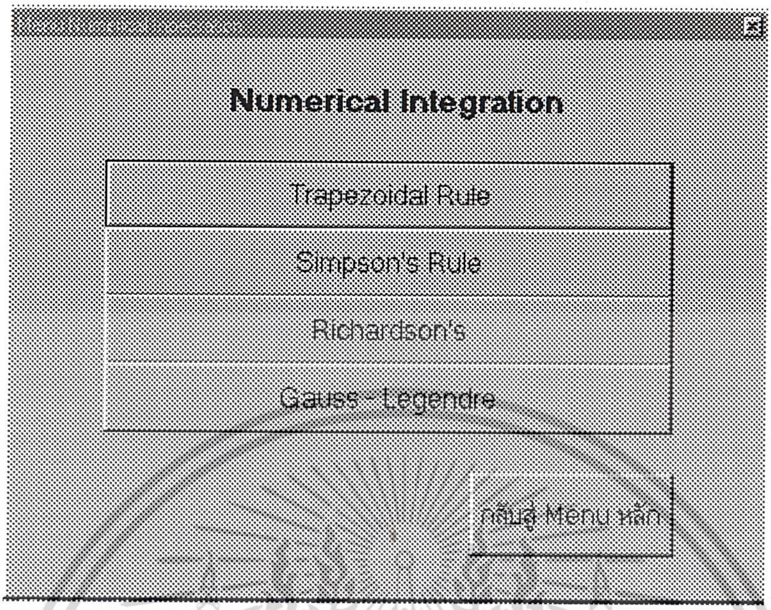
1. ไปเรียกใช้โปรแกรมใน Directory c:\Project\Project.exe แล้วจะพบ Form ของการ Link Mathematica ดังรูปด้านล่าง



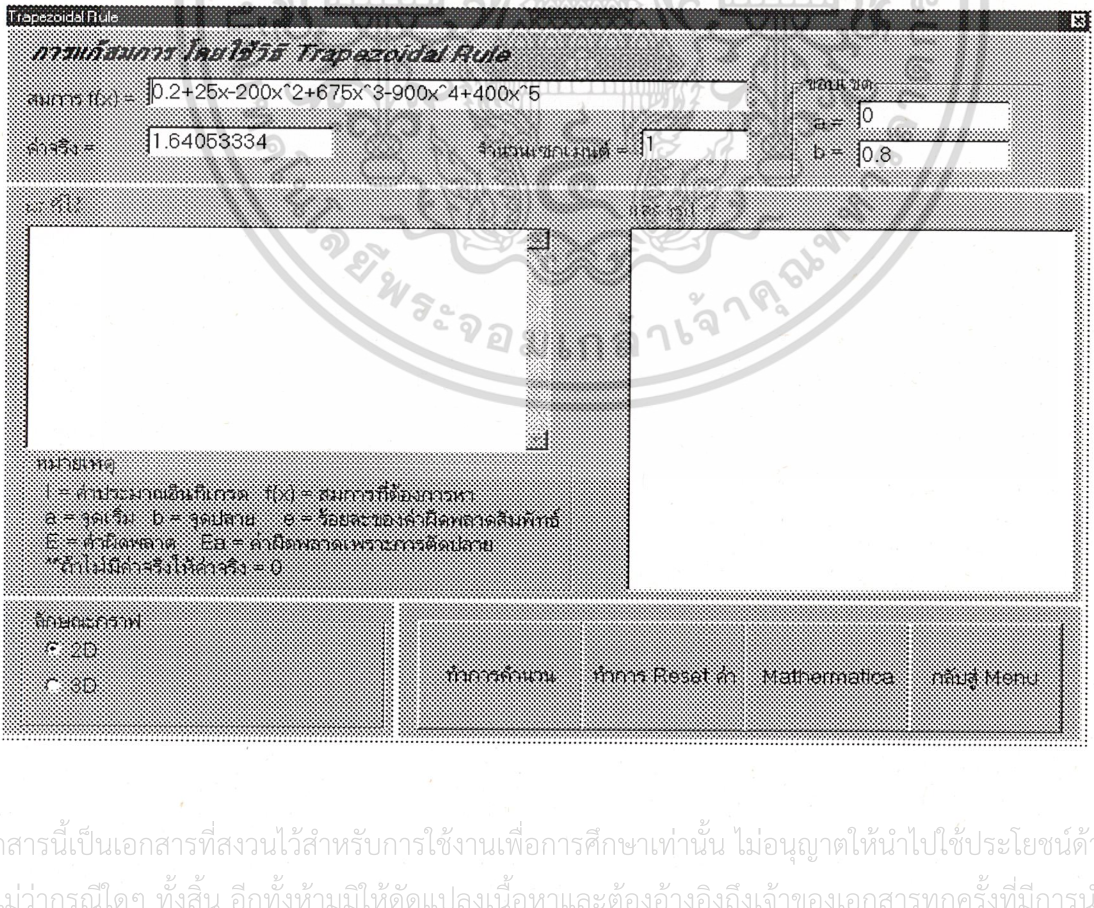
2. ทำการหา file MathKernel.exe ใน Mathematica โดยการใช้ ปุ่ม Browse… เมื่อหาได้แล้วก็ทำการคลิกที่ปุ่ม Connect
3. เมื่อปุ่ม Main Menu สามารถใช้ได้ก็ให้คลิกที่ปุ่ม Main Menu จะไปที่ Menu หลัก



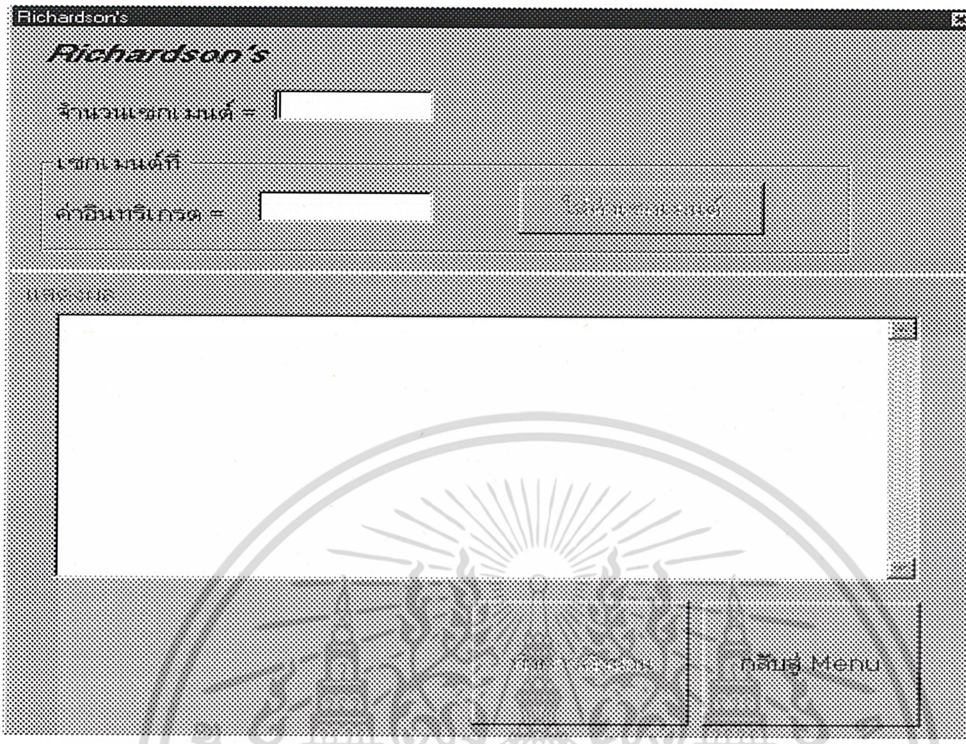
3.1 ใน Form นี้จะมีให้เลือก 5 หัวข้อ ถ้าเราคลิกที่ปุ่มแรก คือ Numerical Integration จะปรากฏ Form ดังนี้



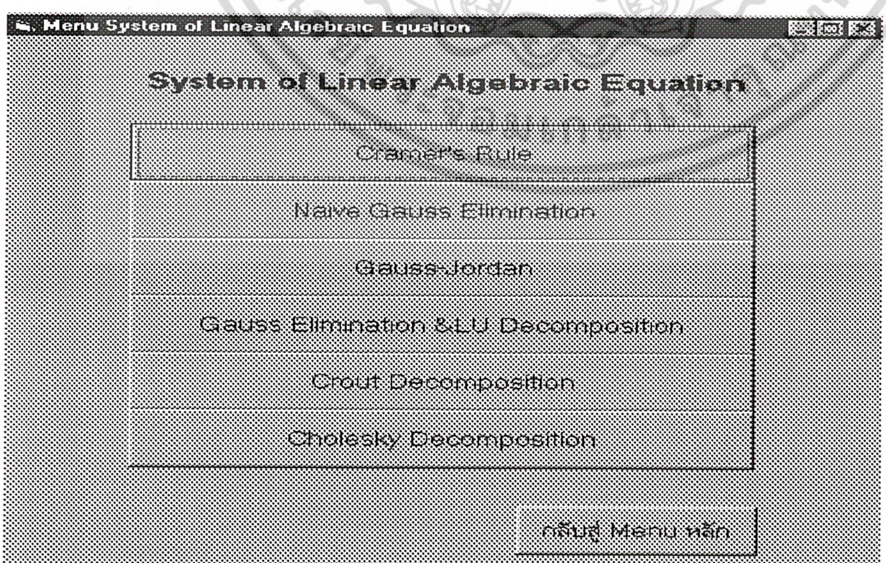
3.1.1 เมื่อคลิกที่ ปุ่ม Trapezodial Rule จะปรากฏหน้าจอดังนี้



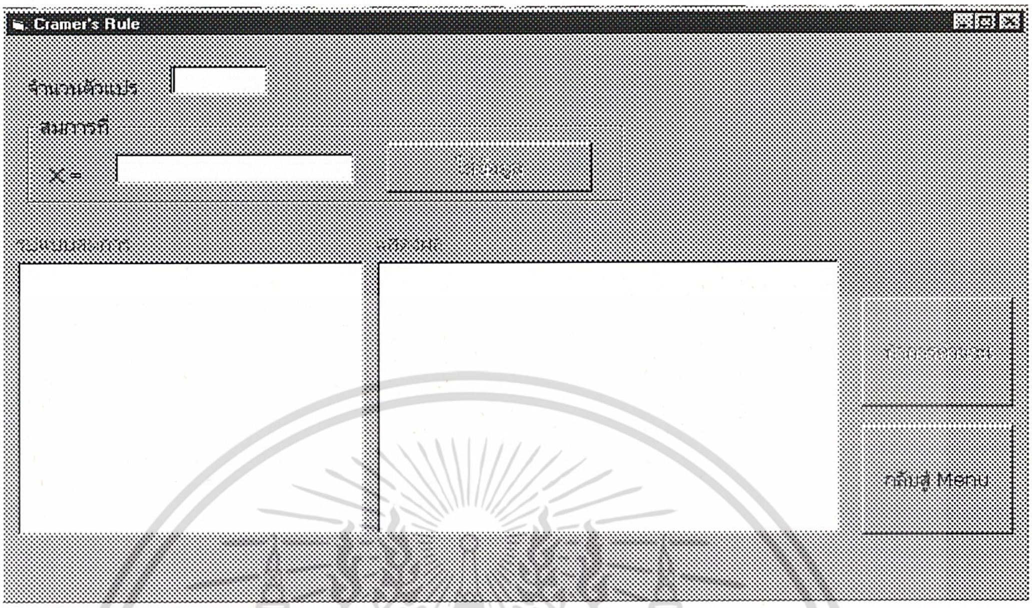
3.1.2 และใน หัวข้ออื่น ๆ ก็เป็น Form เดียวกัน ยกเว้น ปุ่ม Richardson's จะมีแบบ Form ดังนี้



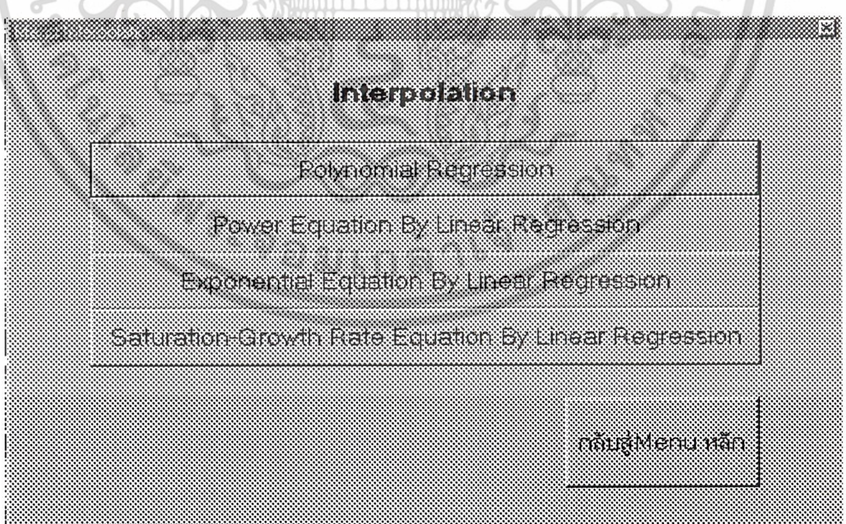
3.2 จาก Main Menu เมื่อเราคลิกปุ่ม System of Linear Algebralic Equation จะเห็นรูปแบบ Form ดังนี้



3.2.1 เมื่อกดปุ่มใด ๆ จะปรากฏ Form นี้คือ โดยจะใช้ Interface ในการรับค่าแบบเดียวกัน



3.3 จาก Main Menu เมื่อคลิกที่ปุ่ม Curve Fitting จะปรากฏ Menu ของ Menu ของ Curve Fitting



4. ถ้ากดปุ่ม About จะเป็นส่วนของผู้เขียนโปรแกรม จะเห็น Form About ดังนี้

5. และถ้าเรากดปุ่ม Exit ในข้อที่ 1 จะออกจากโปรแกรมนี้

บทที่ 4
การประเมินผล

ผลที่ได้จากการศึกษาและพัฒนาโปรแกรม Software Mathematica V.3 และVisual Basic V.5.0 แล้วนำมาทำการเชื่อมโยงกัน ในการสร้างแอพพลิเคชั่นที่สามารถเป็นการปฏิบัติการ (workshop) สำหรับวิชาวิเคราะห์เชิงตัวเลข จากการศึกษาและพัฒนาโปรแกรม สามารถประเมินผลได้ดังนี้

คุณสมบัติและความสามารถของโปรแกรม

1. Software Mathematica V.3 ที่นำมาใช้ สามารถคำนวณผลทางคณิตศาสตร์ได้รวดเร็วและถูกต้อง
2. Software Mathematica V.3 สามารถนำฟังก์ชันไป plot กราฟ 2 และ 3 มิติ ในขอบเขตที่กำหนดได้
3. Software Mathematica V.3 สามารถคำนวณฟังก์ชันทางคณิตศาสตร์ที่ยากต่อการหาผลเฉลย และยังมีชุดคำสั่งสำเร็จรูปในการหาผลเฉลย
4. Software Mathematica V.3 มีฟังก์ชันที่สามารถนำความสามารถของ Software Mathematica V.3 ไปใช้กับแอพพลิเคชั่นอื่นได้ เช่น ภาษา C , Visual Basic
5. Software Visual Basic V.5.0 ง่ายต่อการสร้าง interface
6. Software Visual Basic V.5.0 สามารถเชื่อมโยงกับ Software Mathematica V.3 ได้โดยใช้ฟังก์ชัน MathLink
7. Software Visual Basic V.5.0 มีเครื่องมือที่สะดวกต่อการใช้งานและเป็นแบบฟอร์มมาตราฐานบนวินโดว์

อย่างไรก็ตาม โปรแกรมที่พัฒนาขึ้นยังมีข้อจำกัดหลายประการดังนี้

1. จะต้องลง Software Mathematica V.3 ในเครื่องที่จะลงแอพพลิเคชั่นนี้
2. ไม่สามารถที่จะนำแอพพลิเคชั่นนี้ไปทำการ Run ที่ directory อื่น นอกจาก directory ที่กำหนดไว้ คือ c:\project\
3. สามารถใช้โปรแกรม ในโหมดกราฟฟิกส์ที่มีความละเอียดของหน้าจอ 800×600 ขึ้นไป และต้องใช้โหมดสีมากกว่า 250 ขึ้นไป
4. ไม่สามารถพิมพ์ output ออกทางเครื่องพิมพ์ได้
5. ไม่สามารถคัดลอกรูปภาพไปใช้ในแอพพลิเคชั่นอื่นได้

บทที่ 5
สรุปผลและข้อเสนอแนะ

สรุปผล

ผลที่ได้จากการศึกษาและพัฒนาโปรแกรม Numerical Analysis Workshop with Mathematica V.3 for Science/Engineer ที่สร้างจากการเชื่อมโยงระหว่าง Software Mathematica V.3 และ Software Visual Basic V.5.0 โดยที่ Software Visual Basic V.5.0 นั้นสามารถช่วยให้เราปรับแต่ง interface การรับค่า input และแสดงผล output ทั้งที่เป็น text mode และ graphic mode ได้ง่าย และ interface จะใช้ระบบเดียวกับวินโดว์ ซึ่งง่ายต่อการใช้งาน โดยที่เราจะนำความสามารถของ Software Mathematica V.3 ไปใช้ในการประมวลผลฟังก์ชันทางคณิตศาสตร์ เพราะ Software Mathematica V.3 นั้นมีคำสั่งที่ช่วยในการหาผลเฉลยทางคณิตศาสตร์ได้ง่าย แต่การรับค่าและการแสดงผลนั้นทำได้ยุ่งยาก เนื่องจากจะต้องไปแก้ไขไฟล์ใน source code จากนั้นนำเอาผลเฉลยที่ได้ไปแสดงโดยผ่าน interface ที่ได้สร้างไว้โดย Software Visual Basic V.5.0

ข้อเสนอแนะ

1. ควรจะสามารถพิมพ์ผลเฉลยที่แสดงออกมาได้
2. ควรจะ Run แอพพลิเคชั่นนี้ใน directory อื่น ๆ ได้
3. ควรจะสามารถคัดลอกรูปภาพไปใช้ในแอพพลิเคชั่นอื่นได้

ภาคผนวก

การติดตั้งระบบ

ฮาร์ดแวร์

- เครื่องคอมพิวเตอร์ที่มี CPU ตั้งแต่รุ่น PENTIUM 166 ขึ้นไป
- มีเนื้อที่ว่างในฮาร์ดดิสก์อย่างน้อย 50 เมกกะไบต์
- ควรมีหน่วยความจำ 16 เมกกะไบต์ขึ้นไป
- ไมโครซอฟท์เมาส์ หรือที่เข้ากันได้
- จอแสดงผลสี

ซอฟท์แวร์

- โปรแกรมระบบปฏิบัติการ ไมโครซอฟท์วินโดว์ 98
- Mathematica V.3
- Visual Basic V.5.0

บรรณานุกรม

ปราโมทย์ เดชะอำไพ, **ระเบียบวิธีเชิงตัวเลขในงานวิศวกรรม**, สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, พิมพ์ครั้งที่ 1, 2538

Stephen C.Chapra,Raymond P.Canale,**Numerical Methods for Engineers,**McGraw-Hill Book Company,1998

Richard L.Buchanan,Peter R.Yurner,**Numerical Analysis,**PWS Plblishing,1993

James L.Burden,J.Douglas Faires,**Numerical Methods and Analysis**,McGraw-Hill Book Company,1992

Erwin Kreyszig,**Advanced Engineering Mathematics**,John Wiley&Sons Inc.,1993

Stephen Wolfram,**The Mathematica Book**,3 ed.,Addision-Wesley Publishing Company