# CPE 332
# Computer Engineering Mathematics II

## Part III,
## Chapter 12 ODE

# Today Topics

- Chapter 12 Ordinary Differential Equation
  - HW 10 Due Today
  - HW 11 Due Wed 4 May, 12.00 Noon
    - ส่งห้องเลขาภาค
- Course Ends This Week
  - No Chapter 13(Curve Fitting)

# Differential Equation

- เป็นสมการทางคณิตศาสตร์ ที่ประกอบด้วย Derivative (หรือ Integral)
  - Solution ของสมการคือ Function ที่ไม่มี Derivative หรือ Integral ปรากฏอยู่
- รูปแบบทั่วไปคือ (First Order)
  - $dy/dx = f(x,y)$
  - กรณีพิเศษที่ $dy/dx = f(x)$ เราสามารถแก้สมการโดยใช้การ Integrate
    - $dy = f(x)dx$
    - $y = \int f(x)dx = F(x) + C$ : ค่า $C$ สามารถหาได้โดยการกำหนด Initial condition
  - ปกติการแก้สมการทั่วไปไม่สามารถใช้ Integrate ได้

# Example 1

- Solve for $\dfrac{dy}{dx} = 3x^2; \quad x = 2, y = 4$

# Example 1

- Solve for $\dfrac{dy}{dx} = 3x^2; \quad x = 2, y = 4$

$$dy = 3x^2 dx$$

$$y = \int 3x^2 dx$$

$$y = x^3 + C \text{ (General Solution)}$$

# Example 1

- Solve for $\dfrac{dy}{dx} = 3x^2; \quad x = 2, y = 4$

$$dy = 3x^2 dx$$

$$y = \int 3x^2 dx$$

$$y = x^3 + C \text{ (General Solution)}$$

$$\mathrm{x} = 2, \mathrm{y} = 4$$

$$4 = 2^3 + C$$

$$C = -4$$

$$y = x^3 - 4$$

# Differential Equation

- ค่า Order สูงสุดของ Derivative จะกำหนด Order ของ Differential Equation
  - จำนวน Initial Condition ที่จะต้องใช้จะเท่ากับค่าของ Order ของสมการ

# Example 2

$$F(x,y): xy^3 + 2y + x^2 = 4 \quad \text{Find } f(x,y) = \frac{d}{dx}F(x,y)$$

$$\frac{d}{dx} xy^3 + 2y + x^2 = \frac{d}{dx}4$$

$$[x\frac{d}{dx}y^3 + y^3\frac{dx}{dx}] + 2\frac{dy}{dx} + 2x\frac{dx}{dx} = 0$$

$$[3xy^2\frac{dy}{dx} + y^3] + 2\frac{dy}{dx} + 2x = 0$$

$$3xy^2\frac{dy}{dx} + 2\frac{dy}{dx} = -2x - y^3$$

$$\frac{dy}{dx}[3xy^2 + 2] = -2x - y^3$$

$$\frac{dy}{dx} = \frac{-2x - y^3}{[3xy^2 + 2]}$$

# Example 2

- แก้สมการ $\dfrac{dy}{dx} = f(x,y) = \dfrac{-2x - y^3}{[3xy^2 + 2]}$

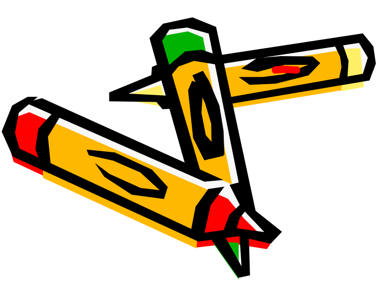
$$dy = \frac{-2x - y^3}{[3xy^2 + 2]} dx$$

$$y = \int \frac{-2x - y^3}{[3xy^2 + 2]} dx$$

Integrate ไม่ได้

# Differential Equation

- Tools ทางคณิตศาสตร์ที่สำคัญสำหรับใช้ในการแก้ปัญหา Differential Equation
  - Laplace Transform (one side/two side)
  - Fourier Transform ใช้ได้เช่นกัน
  - Z-Transform ใช้สำหรับแก้ปัญหา Discrete Version ของ Differential Equation
    - คือ Differential Equation ที่ได้จากการสุ่มตัวอย่างของตัวแปร
    - กรณีนี้เราเรียก Difference Equation



ทั้งหมดนี้อยู่ในเนื้อหาวิชา *CPE 308*

# Differential Equation

- บทนี้เราจะมาดู Numerical Method สำหรับใช้ในการแก้สมการ Differential Equation
  - เราจะจำกัดอยู่ที่ First Order และ Initial Condition ที่จุดตั้งต้น
  - เป็นสมการของ Ordinary Differential Equation
    - One Independent Variable
  - สามารถดัดแปลงสำหรับ Higher Order ได้
  - กรณีที่เป็น Boundary Condition จะต้องใช้วิธีอื่น
    - ศึกษาเพิ่มเติมได้จาก Reference

# Chapter 10

- Ordinary Differential Equation

บทนี้จะเป็นบทที่ 4 และเป็นบทสุดท้ายที่เราจะกล่าวใน Part ที่ 3 นี้ โดยจะเป็นหัวข้อเกี่ยวกับการนำคอมพิวเตอร์มาแก้สมการของ Differential Equation ซึ่งเนื้อหาจะครอบคลุมเฉพาะสมการที่ประกอบด้วยหนึ่ง Independent Variable เท่านั้น หรือเป็นสมการที่เราเรียก Ordinary Differential Equation(ODE) ซึ่งถ้าสมการมีมากกว่าหนึ่ง Independent Variable เราจะเรียก Partial Differential Equation(PDE)

ปกติการจัดสมการของ Differential Equation จะบ่งบอกเป็น Order ของสมการ ซึ่งคือค่า Order ของ Derivative ที่สูงที่สุดในสมการนั้น นอกจากนี้แล้ว สมการของ Differential Equation ยังจัดเป็นได้อีกหลายแบบ ที่สำคัญคือ Linear Constant Coefficient Differential Equation ซึ่งรายละเอียดและคำจำกัดความเหล่านี้ นักศึกษาควรจะได้เรียนมาบ้างแล้วในวิชา Calculus

# ODE

Solution ของสมการ Ordinary Differential Equation นั้นคือ Function ของ Independent Variable และค่า Parameter ที่ทำให้สมการเป็นจริง ยกตัวอย่าง Function ข้างล่าง

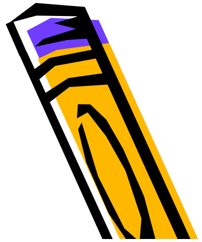
$$y = -0.5x^4 + 4x^3 - 10x^2 + 8.5x + 1$$

ซึ่งเป็น Polynomial ที่มี Order เท่ากับสี่ ถ้าเราทำการ Differentiate สมการดังกล่าว เราจะได้ Ordinary Differential Equation

$$\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5$$

สมการที่ได้จะอธิบายพฤติกรรมของ Polynomial ข้างบน แต่ในมุมมองที่ต่างกัน ซึ่งแทนที่สมการจะแสดงความสัมพันธ์ในค่าของ $y$ ต่อค่าของ $x$ แต่มันจะแสดงอัตราการเปลี่ยนแปลงของค่าของ $y$ ต่อการเปลี่ยนแปลงของ $x$ (ซึ่งก็คือค่า Slope) ที่จุดต่างๆของ $x$ รูปข้างล่างแสดง Plot ของ Function และ Derivative ของมัน

# ODE



จาก Differential Equation ที่ได้ ถ้าเราพยายามหา Solution ของมัน คือ Function เดิม ในกรณีนี้ เราทำได้โดยการ Integration ดังนี้

$$y = \int \left[-2x^3 + 12x^2 - 20x + 8.5\right] dx$$

$$= -0.5x^4 + 4x^3 - 10x^2 + 8.5x + C$$

ซึ่งจะอยู่ในรูปแบบที่เหมือนสมการดั้งเดิม ยกเว้นที่จุดเดียว คือในการทำการ Differentiate และ Integrate กลับมานั้น เราได้สูญเสียข้อมูลของค่า Constant ไป ซึ่งถ้าขาดค่านี้แล้วจะทำให้ Solution ที่ได้มีจำนวนที่ไม่จำกัด ขึ้นอยู่กับค่า C ที่เป็นไปได้ ซึ่งไม่จำกัดเช่นกัน(ดูรูปถัดไป)

อย่างไรก็ตาม คำตอบที่ถูกต้องจะมีคำตอบเดียว และในการที่จะหาคำตอบดังกล่าว เราต้องกำหนดสภาวะช่วย (Auxiliary Condition) สำหรับสมการของ First-Order ODE สภาวะช่วยดังกล่าวก็คือ Initial Condition ที่จะใช้ในการหาค่า Constant ยกตัวอย่างเช่นถ้าเรากำหนด Initial Condition ที่ $x = 0, y = 1$ เมื่อแทนค่าในสมการที่ได้

$$1 = -0.5(0)^4 + 4(0)^3 - 10(0)^2 + 8.5(0) + C$$

เราสามารถแก้สมการและหาค่า $C = 1$ และเราจะได้ Unique Solution และเมื่อแทนค่าที่ได้กลับในสมการของ Solution เราจะได้ คำตอบของ Differential Equation ที่ต้องการ

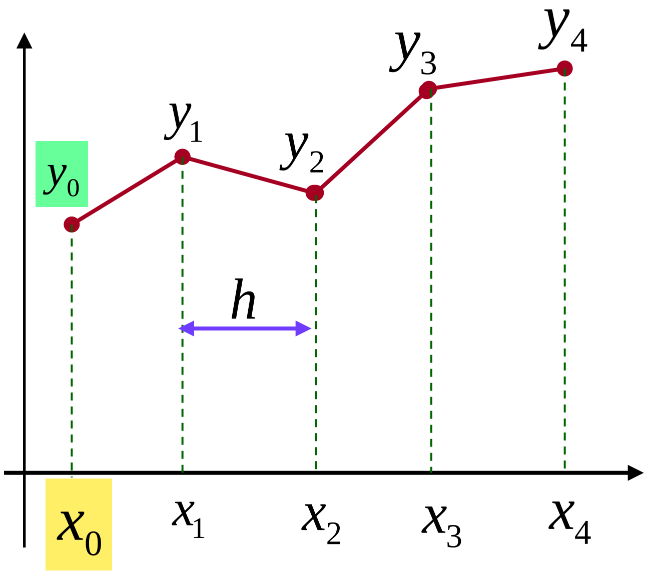
# ODE

การหา Solution ของ N-Order Differential Equation นั้น เราจะต้องการ Condition ทั้งหมด N ตัวในการหา Unique Solution ถ้าทุกๆ Condition ที่ได้ ถูกกำหนดไว้ที่ค่าของ Independent Variable ตัวเดียวกัน การแก้ปัญหาของเราจะเรียก Initial-Value Problem แต่ถ้าค่า Condition ถูกกำหนดที่ค่าต่างกันของ Independent Variable ปัญหานี้จะเรียก Boundary-Value Proble
ในบทนี้ จะคำนึงถึงปัญหาประเภท Initial-Value Problem เท่านั้น

# ODE: One Step Method

เราจะมุ่งความสนใจสำหรับการแก้ปัญหาสมการ Ordinary Differential Equation ที่อยู่ในรูป

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




ซึ่งการใช้วิธีการของ Numerical มาแก้ปัญหานั้น สมการที่ใช้จะอยู่ในรูปของ

ค่าใหม่ = ค่าเก่า + Slope X Step Size

หรือเขียนในลักษณะของสมการทางคณิตศาสตร์

$$y_{i+1} = y_i + \phi h$$

ซึ่งจากสมการ ค่า Estimate ของ Slope $\phi$ จะถูกใช้ในการ Extrapolate ในการหาค่าใหม่ $y_{i+1}$ จากค่าเก่า $y_i$ ในช่วงของระยะทาง $h$ ซึ่งสมการสามารถกระทำทีละ Step ในการคำนวณหาเส้นทางของ Solution ที่ต้องการ

# ODE: One Step Method

$$y_{i+1} = y_i + \phi h$$

ซึ่งจากสมการ ค่า Estimate ของ Slope $\phi$ จะถูกใช้ในการ Extrapolate ในการหาค่าใหม่ $y_{i+1}$ จากค่าเก่า $y_i$ ในช่วงของระยะทาง $h$ ซึ่งสมการสามารถกระทำทีละ Step ในการคำนวณหาเส้นทางของ Solution ที่ต้องการ

วิธีการดังกล่าว เรียกว่า One-Step Method ซึ่งในกรรมวิธีที่จะกล่าวต่อไป จะอยู่ในรูปของสมการนี้ เพียงแต่ความแตกต่างจะอยู่ในส่วนของการ Estimate ค่า Slope ซึ่งวิธีการที่ง่ายที่สุดก็คือการ Estimate จาก First Derivative ที่จุด $x_i$ โดยใช้วิธีที่กล่าวมาในบทก่อน วิธีนี้เราเรียก Euler's Method จะเป็นวิธีแรกที่เราจะศึกษา

วิธีอื่นๆที่เราจะกล่าวต่อไปนั้น จะแตกต่างที่การปรับปรุงการหาค่า Slope ให้ดียิ่งขึ้น ได้แก่ Heun's Method และ Polygon Method ในส่วนสุดท้ายของบท เราจะกล่าวถึงวิธีที่นิยมใช้กันมากที่สุด คือ Runge-Kutta Method

# Euler's Method

## 10.3 Euler's Method (Euler's Cauchy หรือ Point-Slope Method)

ค่า First Derivative ที่ได้จากการ Estimate โดยตรงของ Slope ที่จุด $x_i$ จะหาได้จาก

$$\phi = f(x_i, y_i)$$

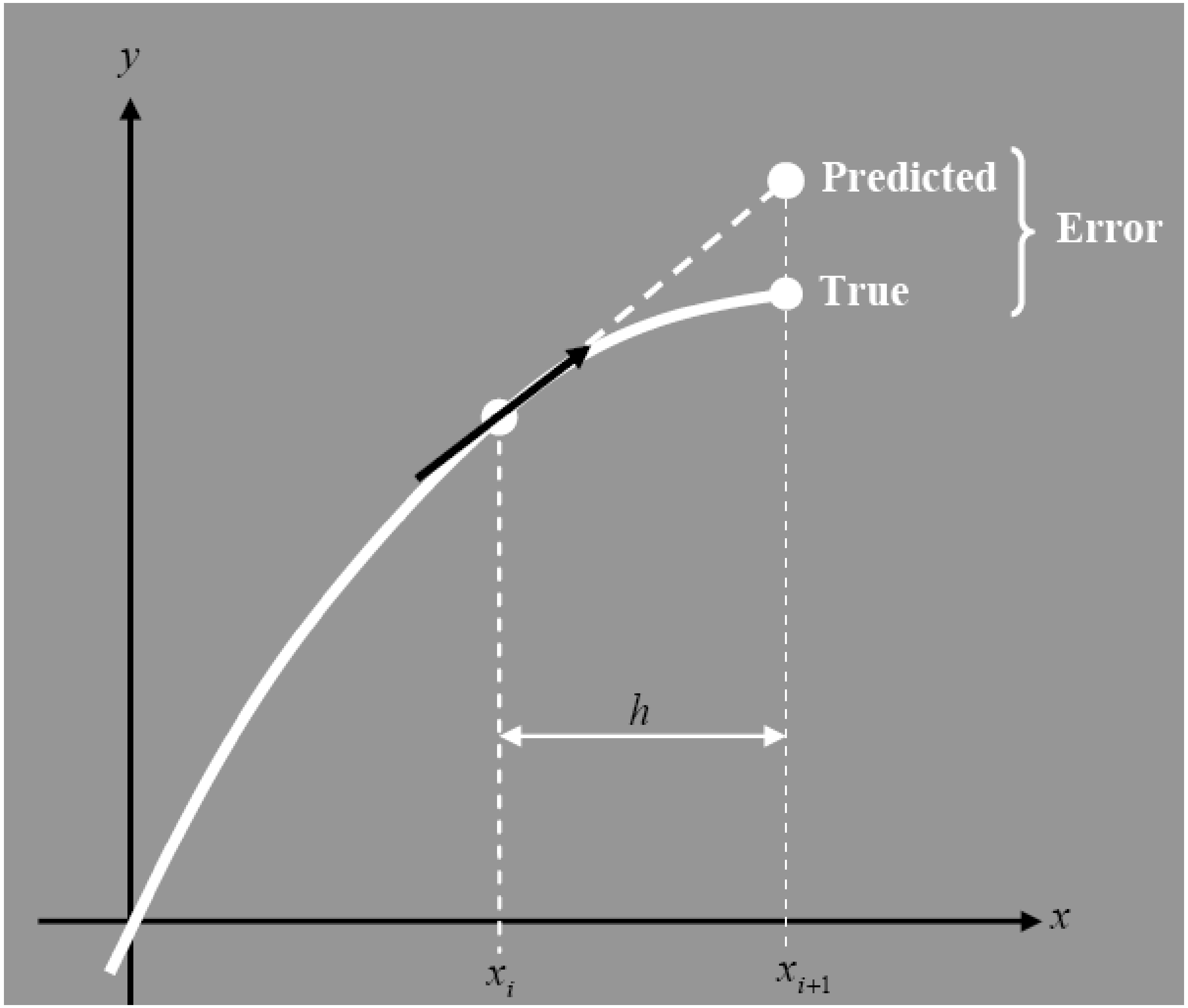
โดยที่ $f(x_i, y_i)$ คือค่าของสมการ Differential Equation ที่จุด $x_i$ และ $y_i$ ดังนั้นสมการของ Solution จะเป็น

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

สมการที่เขียนข้างบนรู้จักกันในนามของ Euler's Method (หรือ Euler-Cauchy Method หรือ Point-Slope Method) โดยที่ค่า $y$ ค่าใหม่หาได้จากการทำนายโดยใช้ค่า Slope จากค่าเก่า(ดูรูป) ทำการ Extrapolate ด้วย Step Size เท่ากับ $h$

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

สมการที่เขียนข้างบนรู้จักกันในนามของ Euler's Method (หรือ Euler-Cauchy Method หรือ Point-Slope Method) โดยที่ค่า $y$ ค่าใหม่หาได้จากการทำนายโดยใช้ค่า Slope จากค่าเก่า(ดูรูป) ทำการ Extrapolate ด้วย Step Size เท่ากับ $h$



# Euler's Method

**Example 10.1** จงใช้ Euler's Method ทำการหาค่า Integrate ของสมการ

$$f(x,y) = -2x^3 + 12x^2 - 20x + 8.5$$

จากจุด $x = 0$ ถึงจุด $x = 4$ โดยใช้ Step Size 0.5 ด้วยค่า Initial Condition $x = 0,\ y = 1$ หรือ $y(0) = 1$ ซึ่งคำตอบที่แท้จริงคือสมการที่แสดงก่อนหน้านี้ คือ $y = -0.5x^4 + 4x^3 - 10x^2 + 8.5x + 1$

**Answer:**

จาก $y(0) = 1$ เราได้ $y(0.5) = y(0) + f(0,1)0.5 = 5.25$

ซึ่ง Solution ที่แท้จริงของจุดนี้คือ $y(0.5) = -0.5(0.5)^4 + 4(0.5)^3 - 10(0.5)^2 + 8.5(0.5) + 1 = 3.21875$

เราได้ $E_t = 3.21875 - 5.25 = -2.03125, \quad e_t = -63.1\%$

จาก $y(0.5) = 5.25$ เราได้ $y(1.0) = y(0.5) + f(0.5,5.25)0.5 = 5.875$

True Solution $y(1) = 3.0$ และ $e_t = -95.8\%$

# Euler's Method

จากนั้นเราคำนวณต่อที่ $y(1.5), y(2), y(2.5), y(3), y(3.5), y(4)$ ผลลัพธ์ที่ได้แสดงดังตารางข้างล่าง ซึ่งจะสังเกตได้ว่า แม้ว่า Solution ที่ได้จะมีลักษณะการเปลี่ยนแปลงตาม True Solution แต่ Error ที่เกิด ค่อนข้างจะมาก

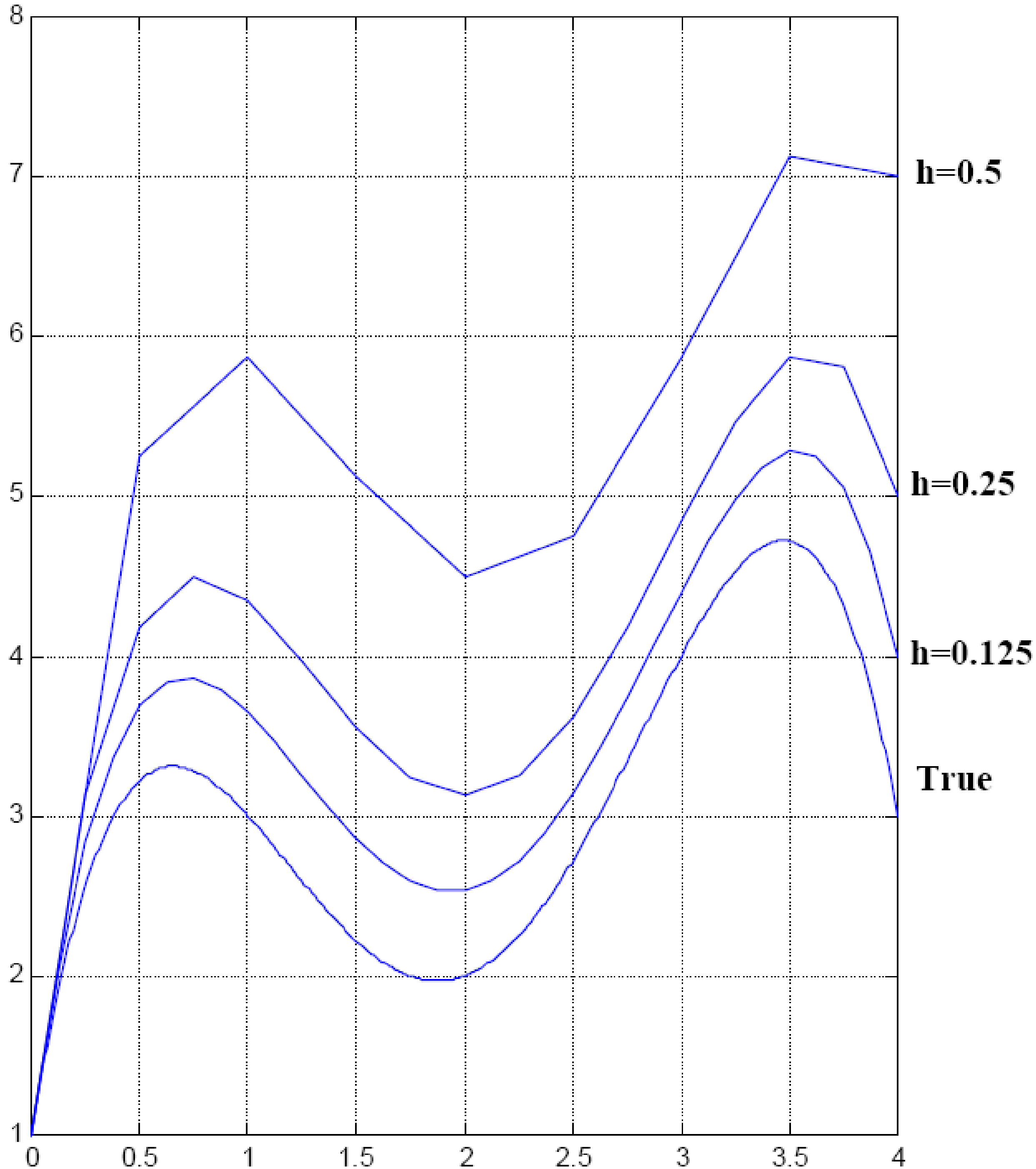
| | | | Relative Error, $e_t$ ,% | |
|---|---|---|---|---|
| $x$ | $y_{true}$ | $y_{Euler}$ | Global | Local(ดูต่อข้างล่าง) |
| 0.0 | 1.0000 | 1.0000 | 0.0000 | 0.0000 |
| 0.5 | 3.2188 | 5.2500 | -63.1068 | -63.1068 |
| 1.0 | 3.0000 | 5.8750 | -95.8333 | -28.1250 |
| 1.5 | 2.2188 | 5.1250 | -130.9859 | -1.4085 |
| 2.0 | 2.0000 | 4.5000 | -125.0000 | 20.3125 |
| 2.5 | 2.7188 | 4.7500 | -74.7126 | 17.2414 |
| 3.0 | 4.0000 | 5.8750 | -46.8750 | 3.9063 |
| 3.5 | 4.7188 | 7.1250 | -50.9934 | -11.2583 |
| 4.0 | 3.0000 | 7.0000 | -133.3333 | -53.1250 |

# Euler's Method

ในวิธีของ Euler's Method นั้น เราทำการ Estimate จนถึงเทอม $f(x,y)h = y'h$ ดังนั้นค่าของ Error จะอยู่ใน $O(h^2)$ และเราสามารถแสดงได้ว่าค่า Estimate Local Truncation Error จะอยู่ในรูป

$$E_a = \frac{f'(x,y)}{2}h^2 = O(h^2)$$

ค่า Truncation Error ที่เกิดสามารถลดลงได้โดยการลดขนาดของ Step Size ซึ่งถ้าเราลดขนาด Step Size ลงครึ่งหนึ่ง ค่าของ Local และ Global Error จะเหลือประมาณหนึ่งในสี่ รูปข้างบนแสดง Solution ที่ได้จากการลดของ Step Size ลง อย่างไรก็ตามแม้ว่าเราจะลด Step ลงถึง $h = 0.001$ ซึ่งเราต้องใช้การคำนวณทั้งหมด 4000 Step จาก $x = 0$ ถึง $x = 4$ ในกรณีนี้ ค่าของ Error ยังคงไม่ได้ถึง 0.1% ดังนั้นวิธีการของ Euler's Method จะใช้การคำนวณที่มากเกินไปถ้าเราต้องการ Error ต่ำๆ และเราจำเป็นต้องหาวิธีอื่นที่ดีกว่า อย่างไรก็ตาม วิธีของ Euler's Method นั้นง่าย และน่าสนใจสำหรับการหาค่าประมาณของ Solution ที่ต้องการ



8
7
6
5
4
3
2
1
0
0.5
1
1.5
2
2.5
3
3.5
4
h=0.5
h=0.25
h=0.125
True

# Heun and Polygon Method

**10.4 การปรับปรุงวิธีการของ Euler’s Method: Heun’s Method และ Polygon Method**

ส่วนใหญ่ Error ที่เกิดจาก Euler’s Method นั้นเป็นผลมาจากการหา Derivative ในส่วนต้นของช่วง และเป็น Derivative ที่จะใช้ตลอดช่วงนั้น วิธีการง่ายๆที่จะแก้ไขความผิดพลาดที่เกิด คือการปรับการหาค่า Derivative ซึ่งในส่วนนี้จะกล่าวสองวิธีง่ายๆที่ใช้การปรับปรุงจากการพิจารณากราฟของ Function ซึ่งทั้งสองวิธีนี้ ความจริงแล้วเป็นวิธีที่อยู่ในเทคนิคการหา Solution ที่เรียก Runge-Kutta Methods อย่างไรก็ตามรายละเอียดของ Runge-Kutta Methods นั้นจะขอกล่าวในส่วนหน้า

# Heun Method

**10.4.1 Heun's Method**

วิธีนี้จะพิจารณาค่า Derivative สองค่า ค่าแรกคือที่จุดเริ่มต้นของช่วง และค่าที่สองคือค่าที่จุดปลายของช่วง และค่า Derivative ทั้งสองจะถูกเฉลี่ย และนำมาใช้สำหรับช่วงนั้น ซึ่งวิธีนี้เราเรียก Heun's Method

จากที่กล่าวใน Euler's Method ค่า Slope ที่จุดเริ่มต้นของ Interval จะเป็น

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

และเรานำมาใช้ในการ Extrapolate หาค่า $y_{i+1}$

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

ซึ่งสมการนี้เราเรียก ***Predictor Equation*** สังเกตว่าเราใช้ Notation $y_{i+1}^0$ ซึ่งเป็น Solution ใน Euler's Method

ในการ Estimate ค่า Slope ที่ปลายของ Interval เราใช้ค่า $y_{i+1}^0$ ทำการ Estimate และเราได้

$$y'_{i+1} = f(x_{i+1}, y_{i+1}^0)$$

# Heun Method

เมื่อเรารวมค่า Slope ทั้งสอง และหาค่าเฉลี่ยของ Slope ในช่วงนั้น เราจะได้

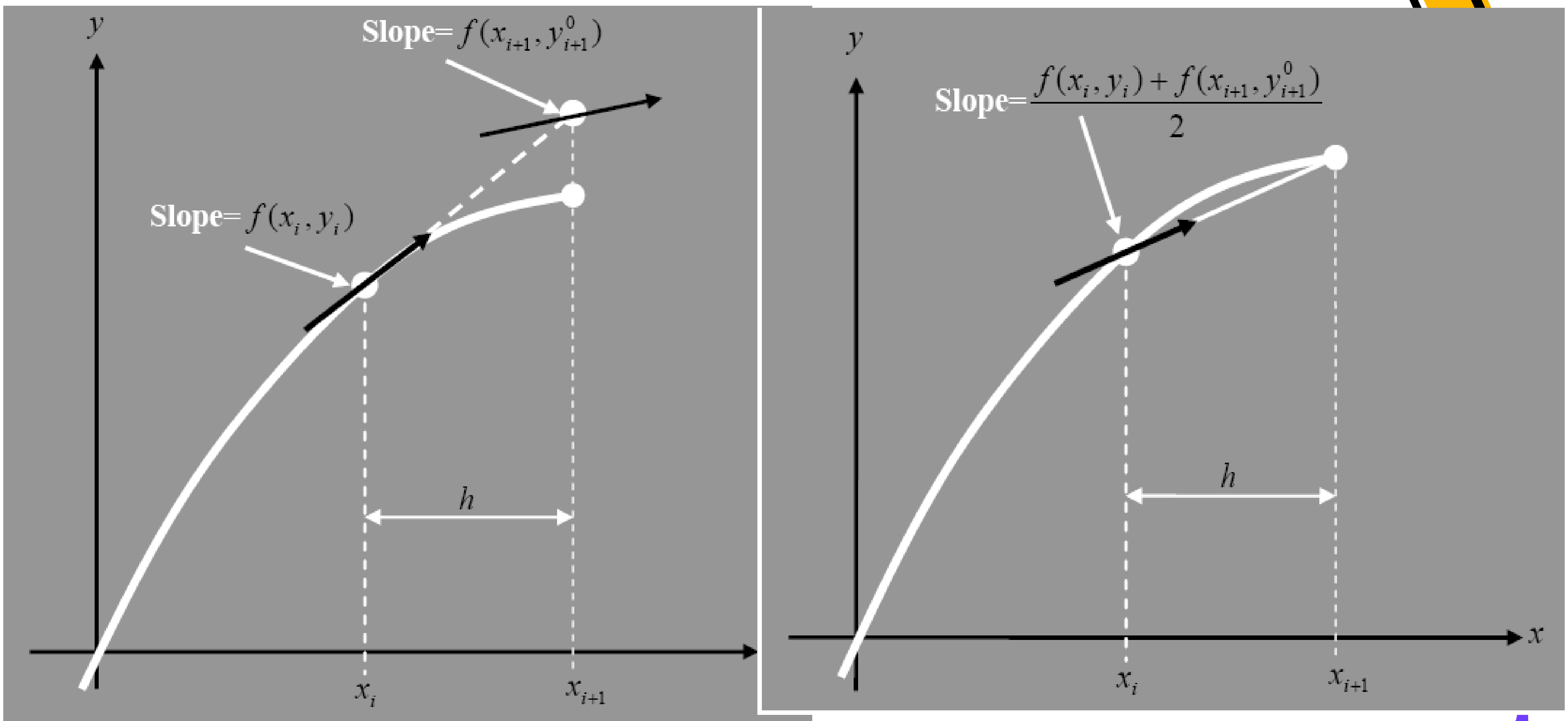
$$\bar{y}' = \frac{y_i' + y_{i+1}'}{2} = \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)}{2}$$

ซึ่งค่าดังกล่าวจะถูกใช้ในการ Extrapolate จาก $y_i$ ไปยัง $y_{i+1}$ โดยใช้สมการเส้นตรง ด้วยวิธีเดิมของ Euler's Method

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

และสมการข้างบนเราเรียก ***Corrector Equation***

# Heun Method



y
Slope= $f(x_{i+1}, y^0_{i+1})$
Slope= $f(x_i, y_i)$
h
$x_i$
$x_{i+1}$
y
Slope= $\dfrac{f(x_i, y_i) + f(x_{i+1}, y^0_{i+1})}{2}$
h
x
$x_i$
$x_{i+1}$

# Heun and Polygon Method

Heun's Method จัดว่าเป็นวิธีที่เรียก Predictor-Corrector Approach และยังเป็น One-Step Method ซึ่งใน Class ของ Multistep Method จะใช้หลักการนี้ทั้งนั้น อย่างไรก็ตาม วิธีของ Multistep Method จะไม่กล่าวในชั้นนี้ ดังนั้น Heun's Method จะเป็นวิธีเดียวของ Predictor-Corrector ที่เราจะกล่าวถึง และสรุปได้ดังนี้

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

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

สมการของ Corrector สามารถทำเป็น Iteration ได้ โดยนำผลที่ได้กลับมาป้อนสมการเดิมเพื่อที่จะได้ Solution ที่ดีขึ้น ดังนี้

**Iterative Corrector:** $$y_{i+1}^{j} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{j-1})}{2}h$$

# Heun Method

โดยที่ $y_{i+1}^{j}$ และ $y_{i+1}^{j-1}$ เป็นค่าของ Corrector ที่ Iteration $j$ และ $j-1$ ตามลำดับ และดังนั้นเราสามารถกำหนด Termination Criteria ของแต่ละ Step โดยใช้ Error Estimate ดังนี้

$$\textbf{Estimate Error: } |e_a| = \left|\frac{y_{i+1}^{j} - y_{i+1}^{j-1}}{y_{i+1}^{j}}\right| 100\%$$

ค่าของ Error ที่เกิดใน Heun's Method จะดีกว่า Euler's Method โดยมีค่า Local Error อยู่ที่ $O(h^3)$ และค่า Global Error อยู่ที่ $O(h^2)$ ในขณะที่ Global Truncation Error ของ Euler's Method อยู่ที่ $O(h)$

**Example 10.2** จงใช้ Heun's Method ทำการหาค่า Integrate ของสมการ $y' = 4e^{0.8x} - 0.5y$ จาก $x = 0$ ถึง 4 โดยใช้ขนาดของ Step Size = 1 ค่า Initial Condition คือ $f(0) = 2$

**Answer:**

คำตอบที่แท้จริงของสมการ(ผู้สนใจการแก้สมการสามารถศึกษาได้ในวิชา Calculus(Differential Equation)) คือ

$$y = \frac{4}{1.3}(e^{0.8x} - e^{-0.5x}) + 2e^{-0.5x}$$

การคำนวณใน Step ที่ 1 : Iteration ที่ 1

ค่าของ Predictor ที่ $x = 1$ หาได้จาก $y_1^0 = 2 + [4e^0 - 0.5(2)] \times 1 = 5$ ซึ่งคือค่าที่ได้จาก Euler's Method

ค่า Slope ที่ $(x_0, y_0)$ คำนวณได้เท่ากับ $y'_0 = 4e^0 - 0.5(2) = 3$

โดยที่ ค่า Estimate $y'_1 = f(x_1, y_1^0) = -4e^{0.8(1)} - 0.5(5) = 6.40216371$

และดังนั้นค่าเฉลี่ยของ Slope จะเป็น $\dfrac{3 + 6.40216371}{2} = 4.70108186$

ท้ายสุด สมการ Corrector สามารถคำนวณได้เป็น $y_1^1 = 2 + (4.70108186)(1) = 6.70108186$

เมื่อเทียบกับคำตอบที่แท้จริงคือ 6.19463138 เราได้ $e_t = -8.18\%$

# Heun Method

การคำนวณใน Step ที่ 1 : Iteration ที่ 2 และ 3 เราทำดังนี้

เราใช้ค่า $y_1^1$ ป้อนกลับในสมการเดิม และได้

$$y_1^2 = 2 + \frac{[3 + 4e^{0.8(1)} - 0.5(6.070108186)]}{2} \times 1 = 6.27581139\,;\ |e_t| = 1.31\%$$

$$y_1^3 = 2 + \frac{[3 + 4e^{0.8(1)} - 0.5(6.27581139)]}{2} \times 1 = 6.38212901\,;\ |e_t| = 3.03\%$$

สังเกตว่าใน Iteration ที่ 3 Error จะเพิ่มขึ้น นั่นหมายความว่าขบวนการ Iterative อาจจะไม่ Converge สำหรับ True Error แต่อย่างไรก็ตาม มันจะ Converge สำหรับ Estimate Error ตารางข้างล่างแสดงผลการ Run แต่ละ Step เปรียบเทียบการ Run 1 Iteration และ 15 Iteration สำหรับแต่ละ Step และภาคผนวกแสดง Program MATLAB สำหรับคำนวณ

# Heun Method

| | | Result 0f 1 and 15 Iterations and Error | | | |
|---|---|---|---|---|---|
| | | 1 Iteration | | 15 Iterations | |
| $x$ | $y_{true}$ | $y_{heun}$ | $\lvert e_t \rvert$, % | $y_{heun}$ | $\lvert e_t \rvert$, % |
| 0 | 2.00000000000000 | 2.00000000000000 | 0.00 | 2.00000000000000 | 0.00 |
| 1 | 6.19463137720937 | 6.70108185698494 | 8.17563546458700 | 6.36086548685535 | 2.68351899448886 |
| 2 | 14.84392190764649 | 16.31978193789828 | 9.94252084748262 | 15.30223665973187 | 3.08755836184566 |
| 3 | 33.67717176796817 | 37.19924889686475 | 10.45835188644487 | 34.74327609067792 | 3.16565871402471 |
| 4 | 75.33896260915857 | 83.33776733540077 | 10.61708901904872 | 77.73509619396161 | 3.18047063805967 |

# Improved Polygon Method

**10.4.2 The Improved Polygon Method (Modified Euler Method)**

วิธีการง่ายๆอีกวิธีหนึ่งที่สามารถนำมาปรับปรุงวิธีการของ Euler's Method เดิม คือการใช้วิธีของ Euler's Method แต่เราทำการ Predict ค่าของ $y$ ที่จุดกึ่งกลางของ Interval แทนที่จะเป็นจุดเริ่มต้น ดังแสดงในรูป ซึ่งวิธีการนี้ เราเรียก Improved Polygon หรือ Modified Euler Method

จากรูป เราใช้ Euler's Method ทำการ Predict ค่าของ $y$ ที่จุดกึ่งกลางของ Interval ได้เป็น

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

ซึ่งค่าที่ Predict ได้จะถูกใช้ในการ Estimate ค่า Slope ที่จุดกึ่งกลางดังนี้

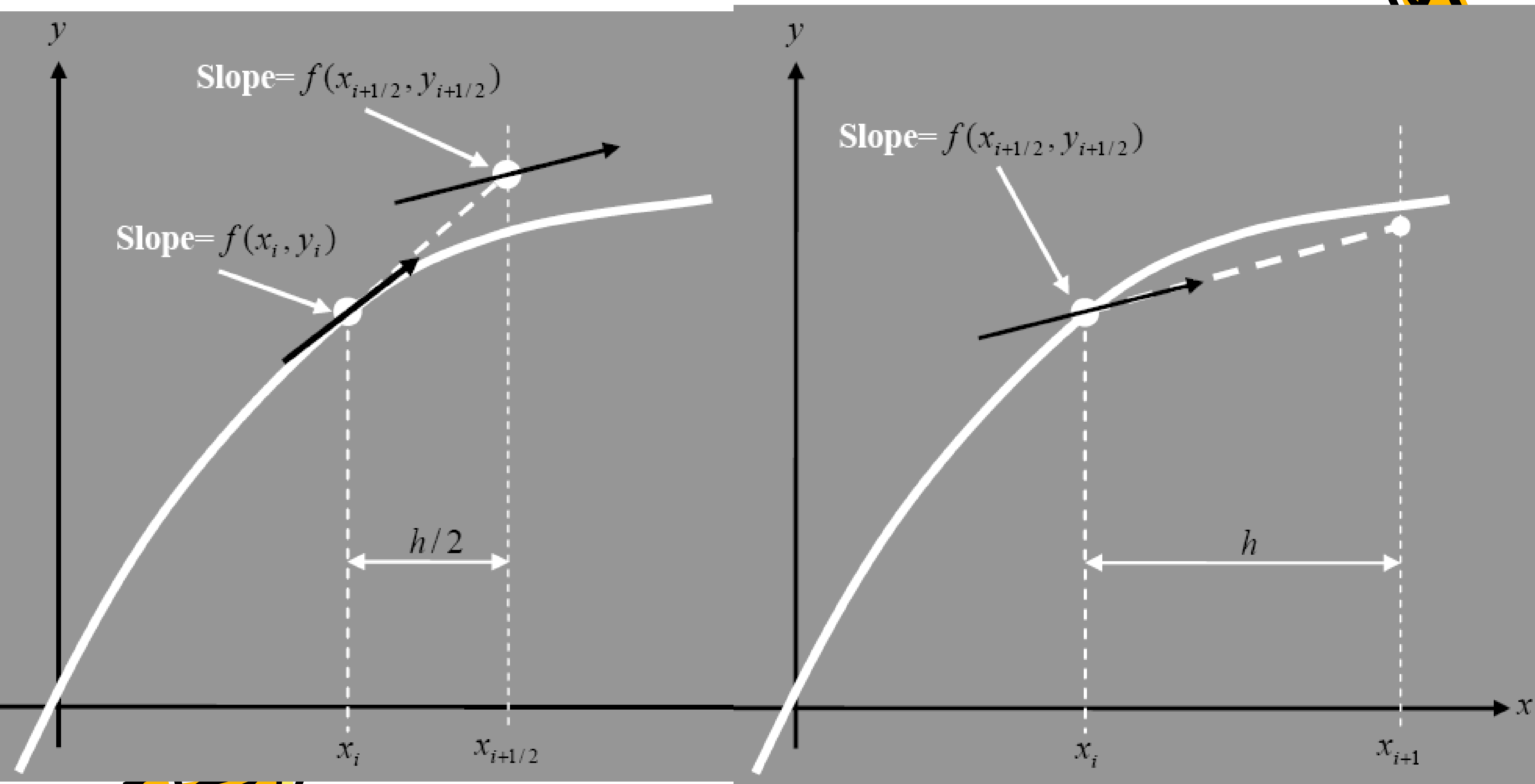
$$y'_{i+1/2} = f(x_{i+1/2}, y_{i+1/2})$$

จากนั้นค่า Slope ที่ได้จะถูกนำมาใช้ในการ Extrapolate แบบ Linear จาก $x_i$ ไปยัง $x_{i+1}$ ด้วย Euler's Method

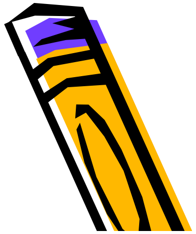
$$y_{i+1} = y_i + f(x_{i+1/2}, y_{i+1/2})h$$

ค่า Local Error และ Global Error ของ Improved Polygon Method พบว่าอยู่ใน $O(h^3)$ และ $O(h^2)$ ตามลำดับ

# Improved Polygon Method



# Runge-Kutta Method



กรรมวิธีของ Runge-Kutta(RK Method) เป็นวิธีที่ให้ความถูกต้องโดยไม่ต้องมีการคำนวณค่า Derivative ในระดับที่สูงๆ ซึ่งวิธีนี้สามารถเขียนเป็นรูปแบบทั่วไปได้ดังนี้

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

โดยที่ $\phi(x_i, y_i, h)$ เรียก ***Increment Function*** ซึ่งเป็นค่าที่แสดงถึงค่า Slope ของ Function โดย Increment Function นี้สามารถเขียนได้ในรูปของ

$$\phi = a_1k_1 + a_2k_2 + \cdots + a_nk$$

ค่า $a_i$ เป็นค่าคงที่ และค่า $k_i$ กำหนดดังนี้

$$\begin{aligned}
k_1 &= f(x_i, y_i) \\
k_2 &= f(x_i + p_1h,\ y_i + q_{11}k_1h) \\
k_3 &= f(x_i + p_2h,\ y_i + q_{21}k_1h + q_{22}k_2h) \\
&\ \vdots \\
k_n &= f(x_i + p_{n-1}h,\ y_i + q_{n-1,1}k_1h + q_{n-1,2}k_2h + \cdots + q_{n-1,n-1}k_{n-1}h)
\end{aligned}$$

สังเกตว่าค่า $k_i$ มีความสัมพันธ์กันแบบ Recurrence Relation ซึ่งเป็นผลให้วิธีการของ RK มีประสิทธิภาพเมื่อนำมาเขียนเป็นโปรแกรม

$$\phi = a_1k_1 + a_2k_2 + \cdots + a_nk$$

ค่า $a_i$ เป็นค่าคงที่ และค่า $k_i$ กำหนดดังนี้

$$\begin{aligned}
k_1 &= f(x_i, y_i) \\
k_2 &= f(x_i + p_1h,\ y_i + q_{11}k_1h) \\
k_3 &= f(x_i + p_2h,\ y_i + q_{21}k_1h + q_{22}k_2h) \\
&\ \vdots \\
k_n &= f(x_i + p_{n-1}h,\ y_i + q_{n-1,1}k_1h + q_{n-1,2}k_2h + \cdots + q_{n-1,n-1}k_{n-1}h)
\end{aligned}$$

ด้วยการเปลี่ยนวิธีการหาเทอมต่างๆตามสมการข้างบน จากค่า $n$ เราสามารถเปลี่ยนแปลงกรรมวิธีของ Runge-Kutta ได้หลายแบบ สังเกตว่าในกรณีของ First-Order Runge-Kutta ด้วยค่า $n = 1$ ที่จริงแล้วก็คือ Euler's Method ในการนำ Runge-Kutta Method ไปใช้ เมื่อเราเลือกค่าของ $n$ (หรือ Order ในกรณีที่มี Order ต่ำๆ) ค่าของ $a_i$, $p_j$ และ $q_{k,l}$ สามารถหาได้โดยตั้ง $y_{i+1} = y_i + \phi(x_i, y_i, h)h$ เท่ากับเทอมใน Taylor Series Expansion

เราจะเริ่มจากการศึกษา Second-Order Runge-Kutta Method ก่อน ซึ่งในกรณีนี้เราจะได้คำตอบที่แท้จริงถ้าคำตอบของสมการอยู่ในรูปสมการ Quadratic และในวิธีนี้จะมีค่า Local Truncation Error อยู่ที่ $O(h^3)$ และ Global Truncation Error อยู่ที่ $O(h^2)$ ต่อจากนั้น เราจะศึกษา Third-Order และ Forth-Order ซึ่งมีค่า Global Truncation Error อยู่ที่ $O(h^3)$ และ $O(h^4)$ ตามลำดับ

# Second Order Runge-Kutta Method

สมการของ Second-Order RK จะอยู่ในรูป

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

โดยที่

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

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

ค่า Constant ต่างๆ สามารถหาได้จากการตั้งสมการให้เท่ากับ Taylor Series Expansion

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h = y_i + f(x_i, y_i)h + f'(x_i, y_i)\frac{h^2}{2}$$

โดยที่ค่า $f'(x_i, y_i)$ สามารถหาได้จากกฎ Chain-Rule ของการหาค่า Derivative

# Second Order Runge-Kutta Method

$$f'(x_i, y_i) = \frac{df}{dx} + \frac{df}{dy}\frac{dy}{dx}$$

และเราได้

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h = y_i + f(x_i, y_i)h + \left(\frac{df}{dx} + \frac{df}{dy}\frac{dy}{dx}\right)\frac{h^2}{2}$$

จากนั้น เมื่อเราใช้ Taylor Series Expansion สำหรับ Two-Variable Function ซึ่งมีรูปแบบดังนี้

$$g(x+r, y+s) = g(x, y) + r\frac{dg}{dx} + s\frac{dg}{dy} + \cdots$$

กับสมการของ $k_2$ เราจะได้

$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h) = f(x_i, y_i) + p_1 h\frac{df}{dx} + q_{11} k_1 h\frac{df}{dy} + O(h^2)$$

# Second Order Runge-Kutta Method

และเมื่อนำไปแทนค่าในสมการ Second-Order RK และจัดเรียงเทอมที่เหมือนกัน เราได้

$$y_{i+1} = y_i + [a_1 f(x_i, y_i) + a_2 f(x_i, y_i)]h + \left[ a_2 p_1 \frac{df}{dx} + a_2 q_{11} f(x_i, y_i) \frac{df}{dy} \right] h^2 + O(h^3)$$

เมื่อเปรียบเทียบกับสมการของ Taylor Series Expansion ของ $y_{i+1}$ และทำการ Equate Term เราสรุปได้ว่า

$$a_1 + a_2 = 1$$

$$a_2 p_1 = 1/2$$

$$a_2 q_{11} = 1/2$$

# Second Order Runge-Kutta Method

เมื่อเปรียบเทียบกับสมการของ Taylor Series Expansion ของ $y_{i+1}$ และทำการ Equate Term เราสรุปได้ว่า

$$a_1 + a_2 = 1$$

$$a_2 p_1 = 1/2$$

$$a_2 q_{11} = 1/2$$

ชุดของสมการข้างบนประกอบด้วยสามสมการ แต่มี 4 Unknown และจะไม่มีคำตอบอันเดียว การหาคำตอบสามารถทำได้โดยเลือกค่า Constant อันหนึ่งก่อน จากนั้นใช้สามสมการข้างบนหาค่า Constant ที่เหลือ ดังนั้น Second-Order RK Method จะมีหลาย Variation และที่สำคัญมีดังนี้

# Second Order Runge-Kutta Method

1. Heun's Method with a Single Corrector: ถ้าเราสมมุติค่า $a_2 = 1/2$ เราแก้สมการได้ $a_1 = 1/2, p_1 = q_{11} = 1$ เมื่อนำไปแทนค่าในสมการของ RK เราได้

$$y_{i+1} = y_i + \left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)h$$
$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + h, y_i + hk_1)$$

ซึ่ง $k_1$ ที่จริงแล้วคือค่า Slope ที่ส่วนต้นของ Interval และ $k_2$ คือค่า Slope ที่ส่วนท้ายของ Interval และสมการดังกล่าวก็คือ Heun's Method ที่มี 1 Iteration

# Second Order Runge-Kutta Method

2. The Improved Polygon Method: ถ้าเราสมมุติค่า $a_2 = 1$ เราแก้สมการได้ $a_1 = 0, p_1 = q_{11} = 1/2$ เมื่อนำไปแทนค่าในสมการของ RK เราได้

$$y_{i+1} = y_i + k_2 h$$

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

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

และสมการดังกล่าวก็คือ Improved Polygon Method

# Second Order Runge-Kutta Method

3. Ralston's Method: ถ้าเราสมมุติค่า $a_2 = 2/3$ เราแก้สมการได้ $a_1 = 1/3, p_1 = q_{11} = 3/4$ เมื่อนำไปแทนค่าในสมการของ RK เราได้

$$y_{i+1} = y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)h$$

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

$$k_2 = f(x_i + \tfrac{3}{4}h, y_i + \tfrac{3}{4}hk_1)$$

สมการนี้รู้จักกันในนาม Ralston's Method ซึ่งจะให้ค่า Truncation Error ของ Second-Order RK ที่เป็น Minimum Bound

# Second Order Runge-Kutta Method

**Example 10.3** จงใช้ Second-Order RK Methods ทั้งสามวิธี เปรียบเทียบการหาค่า Integrate ของสมการ

$y' = f(x,y) = -2x^3 + 12x^2 - 20x + 8.5$ จาก $x = 0$ ถึง $4$ โดยใช้ขนาดของ Step Size = $0.5$ ค่า Initial Condition คือ $f(0) = 1$

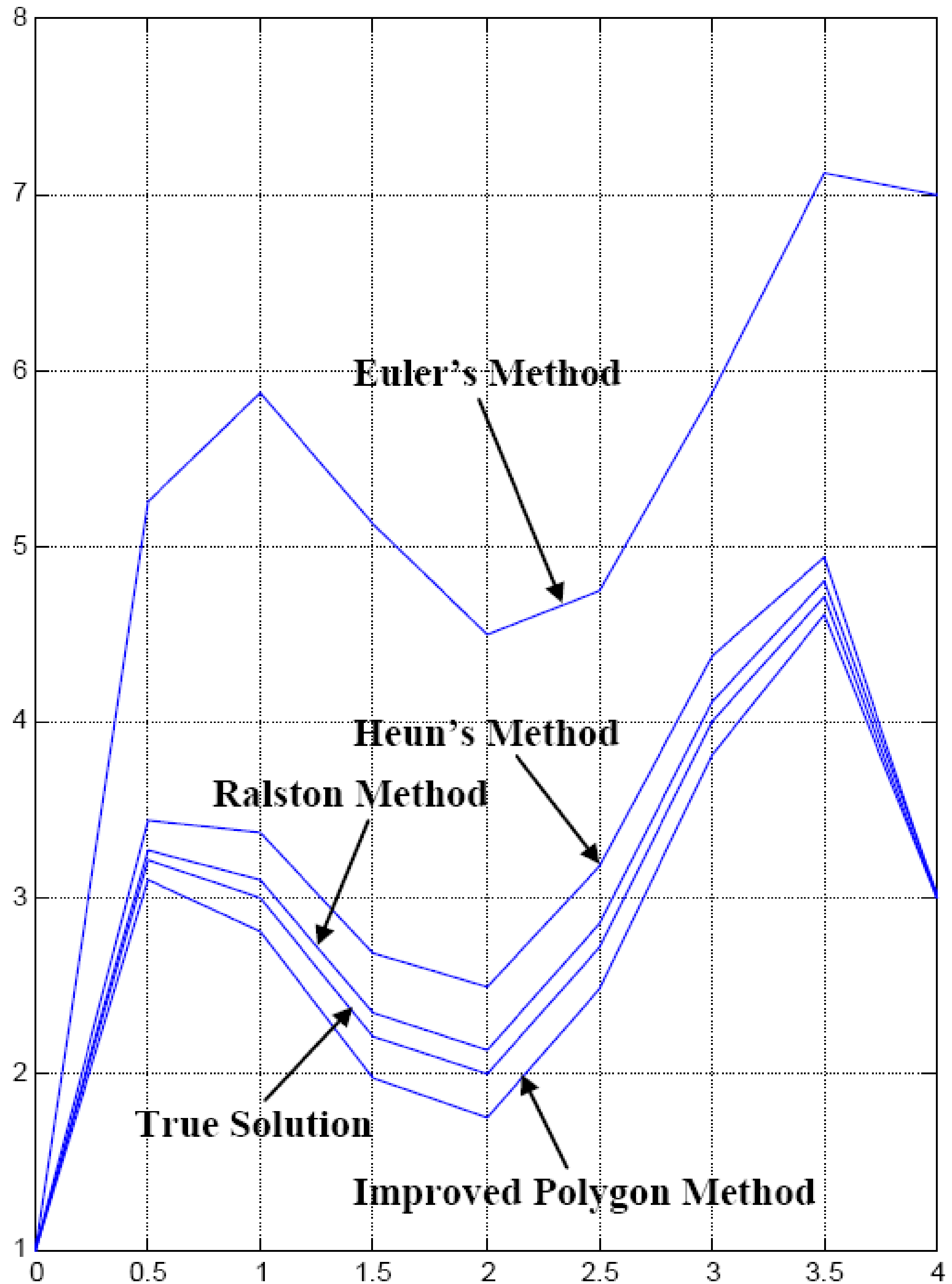
**Answer:**

คำตอบที่แท้จริงคือ(จากตัวอย่างที่ 1) $y = -0.5x^4 + 4x^3 - 10x^2 + 8.5x + 1$

ค่าคำตอบของทั้งสามวิธี และค่า Error สรุปได้ในตารางข้างล่าง สำหรับ Source Code ของ MATLAB ดูได้จากภาคผนวก

# Second Order Runge-Kutta Method

| $x$ | $y_{true}$ | Single Corrector Heun $y$ | $\lvert e_t \rvert,\%$ | Improved Polygon $y$ | $\lvert e_t \rvert,\%$ | Second-Order Ralston $y$ | $\lvert e_t \rvert,\%$ |
|---|---|---|---|---|---|---|---|
| 0.0 | 1.00000000000000 | 1.00000000000000 | 0 | 1.00000000000000 | 0 | 1.00000000000000 | 0 |
| 0.5 | 3.21875000000000 | 3.21875000000000 | 6.8 | 3.10937500000000 | 3.4 | 3.27734375000000 | 1.8 |
| 1.0 | 3.00000000000000 | 3.00000000000000 | 12.5 | 2.81250000000000 | 6.3 | 3.10156250000000 | 3.4 |
| 1.5 | 2.21875000000000 | 2.21875000000000 | 21.1 | 1.98437500000000 | 10.6 | 2.34765625000000 | 5.8 |
| 2.0 | 2.00000000000000 | 2.00000000000000 | 25.0 | 1.75000000000000 | 12.5 | 2.14062500000000 | 7.0 |
| 2.5 | 2.71875000000000 | 2.71875000000000 | 17.2 | 2.48437500000000 | 8.6 | 2.85546875000000 | 5.0 |
| 3.0 | 4.00000000000000 | 4.00000000000000 | 9.4 | 3.81250000000000 | 4.7 | 4.11718750000000 | 2.9 |
| 3.5 | 4.71875000000000 | 4.71875000000000 | 4.6 | 4.60937500000000 | 2.3 | 4.80078125000000 | 1.7 |
| 4.0 | 3.00000000000000 | 3.00000000000000 | 0 | 3.00000000000000 | 0 | 3.03125000000000 | 1.0 |



Euler's Method
Heun's Method
Ralston Method
True Solution
Improved Polygon Method

# Third Order Runge-Kutta Method

ในกรณีที่ $n = 3$ เราจะได้ Third-Order Runge-Kutta Methods และผลลัพธ์จะประกอบไปด้วยการหา 8 Unknown จาก 6 สมการ ดังนั้นเราจะต้องกำหนดค่าของสอง Unknown และหาค่าค่า Constant ที่เหลืออีก 6 ตัว และจะมีคำตอบได้มากมาย คำตอบหนึ่งที่นิยมใช้กันได้แก่ชุดสมการต่อไปนี้

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

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

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

$$k_3 = f(x_i + h, y_i - hk_1 + 2hk_2)$$

สังเกตว่า ถ้า Function ของสมการประกอบด้วยค่า $x$ อย่างเดียว ชุดสมการข้างบนก็คือ Simson 1/3 Rule ค่า Local Truncation Error และ Global Truncation Error ของ Third-Order Runge-Kutta Method อยู่ใน $O(h^4)$ และ $O(h^3)$ ตามลำดับ ในกรณีที่สมการเป็น Cubic Polynomial เราได้ Solution เป็น Quartic ดังนั้นคำตอบของ Third-Order จะให้ค่าที่แท้จริง

# Third Order Runge-Kutta Method

**Example 10.4** จงใช้ Third-Order RK Method หา Solution ของสมการ

$$\frac{dy}{dx} = 4e^{0.8x} - 0.5y$$

จากค่า $x = 0$ จนถึง $x = 1$ โดยใช้ Step Size = 1 กำหนดให้ $y(0) = 2$

**Answer**: คำตอบได้จากการ Run MATLAB Program ท้ายบทนี้

```
» [x,y,e]=o3rk(0,1,1,2)
x =
     0
     1
y =
   2.00000000000000   2.00000000000000
   6.19463137720937   6.17567668094419
e =
                  0
   0.30598586277341
```

# Forth Order Runge-Kutta Method

**10.5.3 Forth-Order Runge-Kutta Methods**

เช่นเดียวกับที่กล่าวในหัวข้อก่อน Forth-Order Runge-Kutta Method จะมี Variation ที่ไม่มีที่สิ้นสุด อย่างไรก็ตาม Forth-Order RK Method เป็นวิธีที่นิยมมากที่สุด ในที่นี้จะกล่าวกรรมวิธีที่เรียก Classical Forth-Order Runge-Kutta Method ซึ่งเป็นชุดของสมการต่อไปนี้

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

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

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

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

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

# Forth Order Runge-Kutta Method

**Example 10.5** จงใช้ Forth-Order RK Method หา Solution ของสมการ

$$y' = f(x,y) = -2x^3 + 12x^2 - 20x + 8.5$$

จากค่า $x = 0$ จนถึง $x = 1$ โดยใช้ Step Size = 0.5 กำหนดให้ $y(0) = 1$

**Answer**: คำตอบได้จากการ Run MATLAB Program ท้ายบทนี้

```
x =
                   0
   0.50000000000000
   1.00000000000000
y =
   1.00000000000000   1.00000000000000
   3.21875000000000   3.21875000000000
   3.00000000000000   3.00000000000000
e =
     0
     0
     0
```

สังเกตว่าในกรณีนี้เราจะได้คำตอบที่แท้จริง เนื่องจาก Solution เป็น Quartic และกรรมวิธีของ Forth-Order RK จะให้คำตอบที่แท้จริงของ Forth-Order Polynomial

# Higher Order Runge-Kutta Method

**10.5.4 Higher-Order Runge-Kutta Methods and System of ODEs**

ถ้าต้องการคำตอบที่ถูกต้องกว่านี้ เราสามารถใช้ Order ที่สูงกว่าของ Runge-Kutta Method ยกตัวอย่างเช่นในกรณีของ Fifth-Order Method ที่รู้จักกันในนามของ Butcher's Method ประกอบด้วยชุดของสมการดังนี้

$$y_{i+1} = y_i + \left[\tfrac{1}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6)\right]h$$

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

$$k_2 = f(x_i + \tfrac{1}{4}h, y_i + \tfrac{1}{4}hk_1)$$

$$k_3 = f(x_i + \tfrac{1}{4}h, y_i + \tfrac{1}{8}hk_1 + \tfrac{1}{8}hk_2)$$

$$k_4 = f(x_i + \tfrac{1}{2}h, y_i - \tfrac{1}{2}hk_2 + hk_3)$$

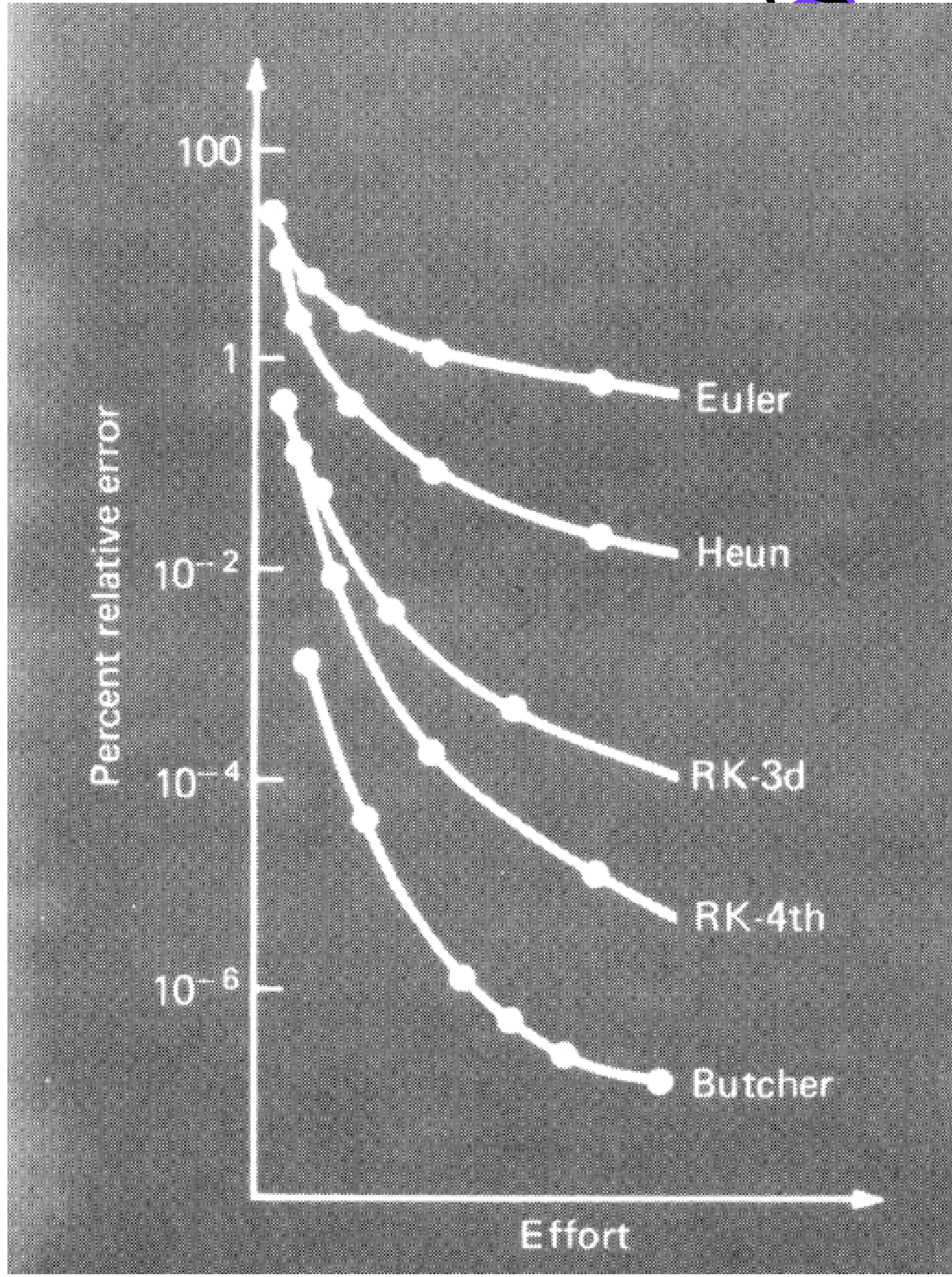
$$k_5 = f(x_i + \tfrac{3}{4}h, y_i + \tfrac{3}{16}hk_1 + \tfrac{9}{16}hk_4)$$

$$k_6 = f(x_i + h, y_i - \tfrac{3}{7}hk_1 + \tfrac{2}{7}hk_2 + \tfrac{12}{7}hk_3 - \tfrac{12}{7}hk_4 + \tfrac{8}{7}hk_5)$$

รายละเอียดจะไม่กล่าวถึง นักศึกษาที่สนใจสามารถเขียนโปรแกรม MATLAB สำหรับสมการข้างบนและทดลองนำไปใช้แก้สมการ ODE เพื่อเปรียบเทียบกับวิธีการที่กล่าวมาข้างต้น

# Comparison

$$dy / dx = 4e^{0.8x} - 0.5y$$

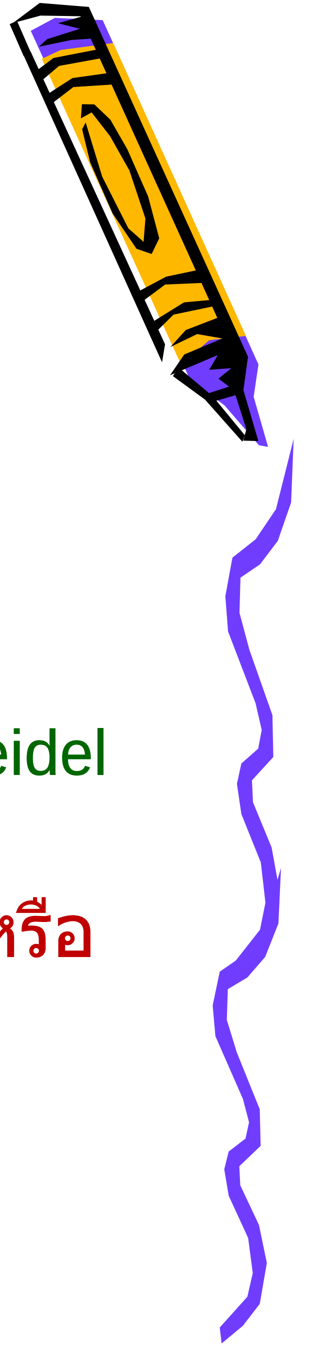


# Chapter 12 Homework (HW 11)

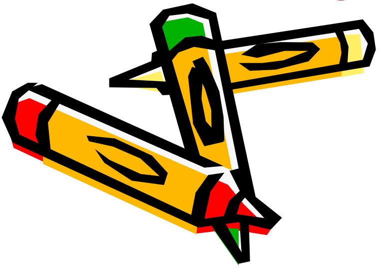
- Download HW 11(ODE)และทำใน Sheet
  - Option ถ้าใครส่งครบ 10 ครั้งและได้เต็ม ไม่ต้องส่ง
  - จะนับ 10 HW ที่คะแนนมากที่สุด
  - ส่งพุธที่ 4 พ.ค. ที่ห้องภาค 5-310 ก่อนเที่ยง
- ไม่มีบทที่ 13 Curve Fitting
- Course Ends

# Final Exam Preparation

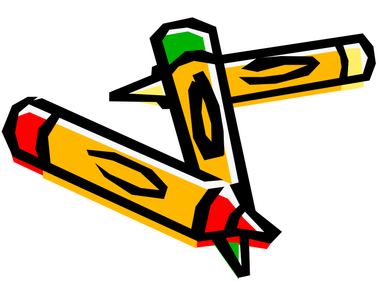
- สูตรจะให้มา
- ข้อสอบมี 7 ข้อ เลือกทำ 5 ข้อ 10x5 = 50 คะแนน เทียบเป็นคะแนนเก็บ 50 %
- 2 ข้อ เป็นเรื่องก่อน Midterm (Part 1 หนึ่งข้อ และ Part 2 หนึ่งข้อ)
- 5 ข้อเป็นเรื่องใหม่ ดังนี้



- **5 ข้อ เป็นเรื่องใหม่ หลัง MT ดังนี้**
  - 1. Taylor Series และการประมาณค่าของ Function รวมถึง Error
  - 2. Root of Function (Bisection หรือ Newton)
  - 3. Linear Equation 1 ข้อ
    - Gauss Elimination, Gauss Jordan, Gauss Seidel และ LU Decompositon
  - 4. Numerical Integration (Trapezoidal หรือ Simpson)
  - 5. ODE โดยใช้ 4th Order RK



# Formulas
## [illegible] MT+[illegible]



**Numerical Methods:**

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

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

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

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

( Secant Method Formula )

LU Decomposition:

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

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

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

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

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

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

Simpson's 1/3 Rule:

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

Simpson's 3/8 Rule:

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

Rhomberg Integration:

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

**ODE**

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

Heun's Method:

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

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

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

Ralston Method:

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

Classical Fourth Order Runge-Kutta Method:

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

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

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

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

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

Taylor Series Expansion:

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

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

Maclaurin Series:

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

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

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

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

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

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

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