

Received: 2 พ.ค. 2568

Revised: 26 ก.ค. 2568

Accepted: 29 ก.ค. 2568

การเปรียบเทียบความแม่นยำของสมการเชิงอนุพันธ์อันดับหนึ่ง ในวิธีออยเลอร์และ  
วิธีรุงเงอ-คุตตาโดยใช้จาวาสคริปต์

A Comparative Study on the Accuracy of Euler and Runge-Kutta Methods for  
Solving First-Order Differential Equations Using JavaScript

พรหมพิริยะ ศรีสุพรรณ<sup>1</sup> และ พิลาสลักณ์ ศรีแก้ว<sup>1\*</sup>

<sup>1</sup>สาขาวิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้

Phomphiriya Srisuphan<sup>1</sup> and Pilasluck Sornkaew<sup>1\*</sup>

<sup>1</sup>Mathematics, Faculty of Science, Maejo University

\*Corresponding author: pisorn@mju.ac.th

## Abstract

A first-order differential equation is a mathematical equation used in mathematics, science, and engineering, and it can be solved using various methods. One commonly used approach is numerical methods, which often rely on computer programs to enhance the accuracy of solutions. Mathematical software such as MATLAB and Scilab are frequently used to solve such problems. In this research, the researcher used the Euler method and the Runge-Kutta method, which are numerical methods used to solve first-order differential equations, to compare the accuracy of the two methods through JavaScript programming. The program displays actual values, estimated values and error values. The results show that Runger-Kutta method is more accurate than Euler method for actual values, estimated values and error values with efficient representation through example and display in HTML window.

**Keywords:** *Differential Equations; Euler; Runge-kutta; Javascript*

## บทคัดย่อ

สมการเชิงอนุพันธ์อันดับที่หนึ่ง เป็นสมการทางคณิตศาสตร์ วิทยาศาสตร์ และ วิศวกรรมศาสตร์ ที่มีการแก้ปัญหาได้หลากหลายวิธี วิธีเชิงตัวเลขเป็นวิธีหนึ่งที่น่าสนใจ และได้มีการใช้โปรแกรมเพื่อมาช่วยหาผลเฉลยที่ต้องการความละเอียดมากขึ้น โดยโปรแกรมทางคณิตศาสตร์ที่ใช้ในการแก้ปัญหา คือ โปรแกรม MATLAB โปรแกรม Scilab เป็นต้น ในการวิจัยนี้ นักวิจัยได้ใช้วิธีออยเลอร์ (Euler) และ วิธีรุงเงอ-คุตทา (Runge-Kutta) ซึ่งเป็นวิธีเชิงตัวเลขวิธีหนึ่งที่ใช้ในการแก้ปัญหาสมการเชิงอนุพันธ์อันดับหนึ่ง เพื่อเปรียบเทียบความแม่นยำของ 2 วิธี โดยผ่านการเขียนโปรแกรมจาวาสคริปต์ โดยให้โปรแกรมแสดงค่าจริง ค่าประมาณ และค่าความคลาดเคลื่อน ผลการศึกษาพบว่า วิธีรุงเงอ-คุตทาให้ความแม่นยำกว่าวิธีออยเลอร์ สำหรับการหาค่าจริง ค่าประมาณ และค่าความคลาดเคลื่อน ที่มีการแสดงผ่านตัวอย่างได้อย่างมีประสิทธิภาพและมีการแสดงผลที่หน้าต่าง html

**คำสำคัญ :** สมการเชิงอนุพันธ์อันดับหนึ่ง; วิธีออยเลอร์; วิธีรุงเงอ-คุตทา; จาวาสคริปต์

## บทนำ

ในการคำนวณทางคณิตศาสตร์มีบทบาทสำคัญในหลายสาขาวิชา เช่น วิทยาศาสตร์ วิศวกรรมศาสตร์ เศรษฐศาสตร์ และการวิเคราะห์ข้อมูล การใช้โปรแกรมคำนวณทางคณิตศาสตร์ช่วยให้การคำนวณที่ซับซ้อน ทำได้อย่างรวดเร็วและแม่นยำ ลดข้อผิดพลาดที่เกิดจากการคำนวณด้วยมือ และช่วยให้สามารถทำการวิเคราะห์ข้อมูลเชิงตัวเลขได้อย่างมีประสิทธิภาพ ปัจจุบันมีโปรแกรมและเครื่องมือหลายชนิดที่ถูกพัฒนา เพื่อรองรับการคำนวณทางคณิตศาสตร์ โดยแต่ละโปรแกรมมีคุณสมบัติที่แตกต่างกันตามลักษณะการใช้งาน ได้แก่ MATLAB, Python, Scilab และ Excel เป็นต้น การเลือกใช้โปรแกรมขึ้นอยู่กับลักษณะของการคำนวณและความต้องการของผู้ใช้วิธีออยเลอร์ (Euler Method) เป็นหนึ่งในวิธีการเชิงตัวเลขพื้นฐานที่ใช้สำหรับการแก้ปัญหสมการเชิงอนุพันธ์สามัญ (Ordinary Differential Equations หรือ ODEs) ซึ่งเป็นแกนสำคัญของแบบจำลองเชิงคณิตศาสตร์ ในงานวิจัยที่ครอบคลุมหลากหลายสาขา เช่น วิทยาศาสตร์ วิศวกรรมศาสตร์ เศรษฐศาสตร์ และการวิเคราะห์ข้อมูล วิธีนี้พัฒนาขึ้นโดย Leonhard Euler นักคณิตศาสตร์ผู้มีชื่อเสียง และได้กลายเป็นหนึ่งในเครื่องมือพื้นฐานที่ใช้ในการศึกษาระบบพลวัตที่ซับซ้อน

สมการรุงเงอ-คุตทา อันดับ 1 (RK1) มีลักษณะคล้ายกับสมการในระเบียบวิธีของออยเลอร์ โดยในวิธีนี้ค่าของผลเฉลยของเวลาถัดไปจะขึ้นอยู่กับผลรวมของค่าของผลเฉลย ณ ปัจจุบันกับความชันของช่วงเวลาปัจจุบัน ( $k_1$ ) ทำให้ผลเฉลยของวิธีนี้ไม่เสถียรและไม่ถูกต้องแม่นยำเท่าที่ควร RK2

เป็นสมการที่ปรับปรุงขึ้นมาจากวิธีของออยเลอร์โดยได้เพิ่มพจน์ของความชัน ( $k_2$ ) ณ ตำแหน่งตรงกลางของผลเฉลยในปัจจุบันกับ  $k_1$  เข้าไปด้วยทำให้ผลเฉลยมีความถูกต้องแม่นยำดีขึ้นจากเดิม RK3 ได้เพิ่มพจน์ของความชัน ( $k_3$ ) เพื่อเพิ่มความถูกต้องแม่นยำ ส่วน RK4 นั้นจะถูกนำมาใช้ค่อนข้างมากในทางปฏิบัติ สามารถคำนวณหาผลเฉลยโดยคอมพิวเตอร์ด้วยการใช้โปรแกรมภาษาต่าง ๆ ซึ่งมีไลบรารีที่นักพัฒนาทั่วโลกได้ทำไว้มากมายสามารถเข้าถึงได้จากสื่ออินเทอร์เน็ตทั่วไปทั้งยังสามารถนำเอาหลักการอื่นมารวมเพื่อประยุกต์ใช้ต่อได้อีกหลากหลายแนวทาง

JavaScript (JS) เป็นหนึ่งในภาษาการเขียนโปรแกรมที่ได้รับความนิยมอย่างแพร่หลาย โดยเฉพาะในการพัฒนาเว็บไซต์และแอปพลิเคชันบนเว็บ อย่างไรก็ตาม เมื่อพูดถึงการใช้งานทางคณิตศาสตร์ JavaScript มักไม่ได้เห็นนำมาใช้ทางคณิตศาสตร์เท่ากับภาษาอื่นๆ ถึงแม้ว่า JavaScript จะไม่ได้ถูกออกแบบมาเพื่อการคำนวณเชิงตัวเลขโดยเฉพาะ แต่ด้วยการพัฒนาไลบรารีเสริม ทำให้ JS มีศักยภาพในการใช้คำนวณ สมการทางคณิตศาสตร์ วิเคราะห์ ได้อย่างมีประสิทธิภาพ

ในปี 2010 มูชา เอช, อิบราฮิม ไชดู และ วาซิริ ได้ศึกษาเรื่องการวิเคราะห์และการหาอนุพันธ์แบบง่ายของ วิธีรุงเง-คุตทาดันดับ 4 ที่เกี่ยวข้องกับการหาอนุพันธ์ของวิธีรุงเง-คุตทาดันดับ 4 ซึ่งได้ทำการปรับปรุงการหาอนุพันธ์และการวิเคราะห์ที่มีอยู่ของวิธีรุงเง-คุตทาดันดับ 4 ให้เรียบง่ายขึ้นเพื่อให้สามารถอ้างอิงได้ง่ายและได้มีการแสดงกราฟ นอกจากนี้ ยังลดความซับซ้อนของวิธีการนี้โดยเสนอแนวทาง การหาอนุพันธ์ที่ละขั้นตอนเพื่อให้เข้าใจได้ดีขึ้น ต่อมา ดร. อาบีรุล อิสลาม ได้ศึกษาเรื่องการหาคำตอบที่แม่นยำของปัญหาค่าเริ่มต้นสำหรับสมการเชิงอนุพันธ์สามัญด้วยวิธีรุงเง-คุตทาดันดับ 4 ในปี 2015 ซึ่งงานวิจัยนี้ พิจารณาวีธีรุงเง-คุตทา อันดับ 4 สำหรับการแก้ปัญหาสมการเชิงอนุพันธ์สามัญ โดยวิธีที่นำเสนอนี้เป็นวิธีที่มีประสิทธิภาพค่อนข้างสูง เหมาะสำหรับการแก้ปัญหาสมการเชิงอนุพันธ์สามัญ ในงานวิจัยนี้มีการใช้ตัวอย่างหลายตัวอย่างเพื่อแสดงให้เห็นถึงความแม่นยำและความง่ายในการนำไปใช้ นอกจากนี้ผลลัพธ์จากการทดลองเชิงตัวเลขถูกนำมาเปรียบเทียบกับคำตอบเชิงวิเคราะห์ และได้ทำการวิเคราะห์ค่าความคลาดเคลื่อนของวิธีรุงเง-คุตทา ผลการวิจัยพบว่า ผลลัพธ์ที่ได้รับถือว่าดี แม้จะใช้ขนาดของระยะห่างระหว่างจุดมาก และสามารถปรับความแม่นยำได้โดยการลดขนาดของระยะห่างระหว่างจุด เพื่อให้ได้ค่าความแม่นยำที่ต้องการ ดังนั้นวิธีนี้จึงสามารถนำไปใช้ในการแก้ปัญหาเริ่มต้น เพื่อให้ได้ค่าความแม่นยำที่ต้องการได้ และในปี 2015 คาซิม อับบาส ฮุสเซน, ฟัดเซียห์ อิสมาอิล, นรศักดิ์ เสนุ และ ฟารานัก ราเบีย ก็ได้ทำการศึกษาเรื่องวิธีรุงเง-คุตทา แบบปรับปรุงอันดับ 4 สำหรับการแก้สมการเชิงอนุพันธ์สามัญอันดับสามแบบพิเศษ งานวิจัยนี้ได้สร้างวิธีรุงเง-คุตทา แบบปรับปรุงอันดับ 4 สำหรับการแก้สมการอนุพันธ์สามัญอันดับ 3 แบบพิเศษโดยตรง มีการเปรียบเทียบเชิงตัวเลขโดยใช้วิธีรุงเง-คุตทา ที่มีอยู่เดิม แก้ปัญหาโดยการลดสมการให้เป็นระบบของสมการอันดับหนึ่ง แล้วทำการคำนวณใช้วิธี RKD แบบตรงสำหรับ

การแก้สมการอนุพันธ์สามัญอันดับสามแบบพิเศษ ผลการวิจัยพบว่า ผลลัพธ์เชิงตัวเลขแสดงให้เห็นถึงความสามารถของ วิธี RKD4 แสดงให้เห็นถึงประสิทธิภาพและความแม่นยำของวิธีใหม่ รวมถึงค่าความคลาดเคลื่อนสัมบูรณ์

งานวิจัยใช้โปรแกรมจาวาสคริปต์ ซึ่งเป็นโปรแกรมที่ยังไม่ได้ถูกออกแบบมาเพื่อการคำนวณเชิงตัวเลขโดยเฉพาะทางคณิตศาสตร์ ดังนั้น ผู้วิจัยจึงสนใจที่จะทำการเปรียบเทียบความแม่นยำของสมการเชิงอนุพันธ์อันดับ 1 ของวิธีออยเลอร์และวิธีรุงเงอ-คุททา ซึ่งจะแสดงการประมาณค่าของผลเฉลย ผ่านการเขียนโปรแกรมจาวาสคริปต์ ที่มีการแสดงผลที่หน้าต่าง html ในที่นี้ผู้วิจัยสนใจเฉพาะสมการเชิงอนุพันธ์สามัญอันดับ 1 เท่านั้น

### วิธีการดำเนินการวิจัย

การหาคำตอบของสมการเชิงอนุพันธ์มี 2 ประเภท คือ ปัญหาค่าเริ่มต้น (Initial value problem) และ ปัญหาค่าขอบ (Boundary value problem) การใช้วิธีการวิเคราะห์เชิงตัวเลขเป็นหนึ่งวิธีที่ใช้ในการหาคำตอบ เช่น วิธีออยเลอร์ หรือวิธีรุงเงอ-คุททา อีกทั้งการใช้โปรแกรมเพื่อช่วยในการหาคำตอบ

#### การหาคำตอบโดยวิธีของออยเลอร์

จากสมการ.  $y' = f(x, y)$  ซึ่งมีเงื่อนไขเริ่มต้น  $y(x_0) = y_0$

จะได้  $y_1 = y(x_1) = y_0 + hf(x_0, y_0)$

$y_2 = y(x_2) = y_1 + hf(x_1, y_1)$

ดังนั้นเขียนเป็นสมการทั่วไป จะได้  $y_{n+1} = y_n + hf(x_n, y_n)$

#### การหาคำตอบวิธีรุงเงอ-คุททาอันดับ 4

เป็นการคำนวณหาค่า  $y(x_n)$  ที่ให้คำตอบมีค่าแม่นยำค่อนข้างสูง จากสมการ  $y' = f(x, y)$  ซึ่งมีเงื่อนไขเริ่มต้น  $y(x_0) = y_0$

ทำให้ได้  $y(x_{n+1}) = y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

โดยที่  $k_1 = hf(x_n, y_n)$

$k_2 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right)$

$k_3 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right)$

$k_4 = hf(x_n + h, y_n + k_3)$

### ความคลาดเคลื่อน (ERROR)

$$\text{ค่าความคลาดเคลื่อนสัมบูรณ์} = |x_{\text{mean}} - x_{\text{true}}|$$

$$\text{ค่าความคลาดเคลื่อนสัมพัทธ์} = \left| \frac{x_{\text{mean}} - x_{\text{true}}}{x_{\text{true}}} \right|$$

$$\text{เปอร์เซ็นต์ความคลาดเคลื่อน} = \text{ค่าความคลาดเคลื่อนสัมพัทธ์} * 100$$

### โปรแกรม JAVASCRIPT

คำสั่ง  $f(x,t)$  เป็นฟังก์ชันที่กำหนดให้เป็นสมการเชิงอนุพันธ์

exactSolution (t) เป็นฟังก์ชันที่ให้คำตอบจริง

การแสดงผล

หลังจากคำนวณค่าทั้งหมดแล้ว ผลลัพธ์จะถูกแสดงในตารางที่มีคอลัมน์

| ระยะห่าง | ค่าประมาณ | ค่าจริง | ค่าความคลาดเคลื่อน |
|----------|-----------|---------|--------------------|
| t1       | x1        | exactX1 | error1             |
| t2       | x2        | exactX2 | error2             |
| ...      |           |         |                    |

### ผลการวิจัยและอภิปรายผล

#### การเขียนโปรแกรม Javascript วิธีออยเลอร์และวิธีรุงเงอ-คุททา

```
<script>
let equationStr = prompt("สมการ f(x,t) เช่น '-x + 2*cos(t)':");
let exactSolutionStr = prompt("ค่าจริง x(t) เช่น 'sin(t) + cos(t)':");

const f = (x, t) => math.evaluate(equationStr, { x, t });
const exactSolution = (t) => math.evaluate(exactSolutionStr, { t });

const a = Number(prompt("a (ค่าเริ่มต้น t): "));
const b = Number(prompt("b (ค่าสุดท้าย t): "));
const N = Number(prompt("N (จำนวนช่วงแบ่ง): "));
const h = (b - a) / N;
let x0 = Number(prompt("ค่าเริ่มต้น x0: "));

let x_euler = x0;
let x_rk = x0;

const tpoints = [], eulerPoints = [], rkPoints = [], exactPoints = [], eulerErrorPoints = [], rkErrorPoints = [];
let resultsHTML = "";

for (let i = 0; i <= N; i++) {
  let t = a + i * h;
  let exactX = exactSolution(t);

  let eulerError = Math.abs((x_euler - exactX) / exactX);
  let rkError = Math.abs((x_rk - exactX) / exactX);

  tpoints.push(t);
  eulerPoints.push(x_euler);
  rkPoints.push(x_rk);
  exactPoints.push(exactX);
  eulerErrorPoints.push(eulerError);
  rkErrorPoints.push(rkError);
}
```

```

resultsHTML += `<tr>
  <td>${t.toFixed(3)}</td>
  <td>${x_euler.toFixed(6)}</td>
  <td>${x_rk.toFixed(6)}</td>
  <td>${exactX.toFixed(6)}</td>
  <td>${eulerError.toExponential(6)}</td>
  <td>${rkError.toExponential(6)}</td>
</tr>`;

// Euler Method
x_euler += h * f(x_euler, t);

// Runge-Kutta 4th Order
const k1 = h * f(x_rk, t);
const k2 = h * f(x_rk + 0.5 * k1, t + 0.5 * h);
const k3 = h * f(x_rk + 0.5 * k2, t + 0.5 * h);
const k4 = h * f(x_rk + k3, t + h);
x_rk += (k1 + 2 * k2 + 2 * k3 + k4) / 6;
}

document.getElementById("results").innerHTML = resultsHTML;

Plotly.newPlot('plot', [
  { x: tpoints, y: eulerPoints, mode: 'lines', type: 'scatter', name: 'Euler' },
  { x: tpoints, y: rkPoints, mode: 'lines', type: 'scatter', name: 'Runge-Kutta' },
  { x: tpoints, y: exactPoints, mode: 'lines', type: 'scatter', name: 'Exact Solution', line: { dash: 'dot' } },
], {
  title: 'Euler vs Runge-Kutta vs Exact Solution',
  xaxis: { title: 't' },
  yaxis: { title: 'x(t)' }
});
</script>

```

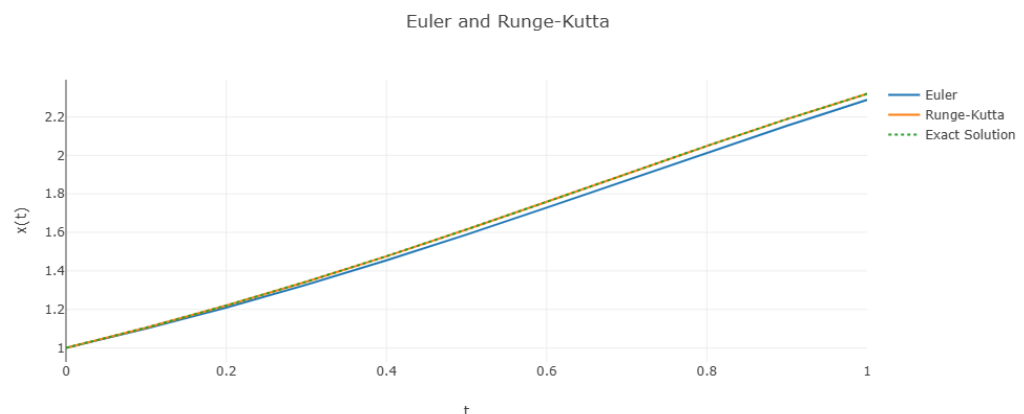
### ตัวอย่างที่ 1

พิจารณาปัญหาค่าเริ่มต้นที่กำหนดโดยสมการ  $x'(t) = x \cos(t)$

โดยมีเงื่อนไขเริ่มต้น  $x(0) = 1$  บนช่วง  $0 \leq t \leq 1$

ค่าจริง  $x(t) = e^{\sin(t)}$

ผลจากโปรแกรม



รูปที่ 1 : กราฟแสดงการเปรียบเทียบวิธีออยเลอร์กับวิธีรุงเงอ-คุตทาดันดับ 4

ตารางที่1: ตารางการเปรียบเทียบวิธีออยเลอร์กับวิธีรุงเงอ-คุตทาดันดับ 4

## Numerical Results

| t     | Euler .  | Runge-Kutta . | Exact Solution | Euler Error | Runge-Kutta Error |
|-------|----------|---------------|----------------|-------------|-------------------|
| 0.000 | 1.000000 | 1.000000      | 1.000000       | 0.000000e+0 | 0.000000e+0       |
| 0.100 | 1.100000 | 1.104987      | 1.104987       | 4.513022e-3 | 7.659357e-8       |
| 0.200 | 1.209450 | 1.219778      | 1.219779       | 8.467191e-3 | 1.496028e-7       |
| 0.300 | 1.327985 | 1.343825      | 1.343825       | 1.178768e-2 | 2.158809e-7       |
| 0.400 | 1.454852 | 1.476122      | 1.476122       | 1.440943e-2 | 2.730793e-7       |
| 0.500 | 1.588853 | 1.615146      | 1.615146       | 1.627945e-2 | 3.198712e-7       |
| 0.600 | 1.728288 | 1.758818      | 1.758819       | 1.735898e-2 | 3.560056e-7       |
| 0.700 | 1.870929 | 1.904496      | 1.904497       | 1.762527e-2 | 3.821927e-7       |
| 0.800 | 2.014026 | 2.049008      | 2.049009       | 1.707305e-2 | 3.998561e-7       |
| 0.900 | 2.154344 | 2.188741      | 2.188742       | 1.571567e-2 | 4.108100e-7       |
| 1.000 | 2.288261 | 2.319776      | 2.319777       | 1.358591e-2 | 4.169330e-7       |

จากสมการเชิงอนุพันธ์อันดับ 1 โดยการเขียนโปรแกรมจาวาสคริปต์ทำให้ได้ผลดังตาราง ที่มีการแสดงผล ระยะห่างระหว่างจุด ค่าประมาณโดยวิธีออยเลอร์ ค่าประมาณโดยวิธีรุงเงอ-คุตทา ค่าจริง ค่าความคลาดเคลื่อนของออยเลอร์และค่าความคลาดเคลื่อนของรุงเงอ-คุตทา ผลที่ได้แสดงให้เห็นว่าวิธีรุงเงอ-คุตทาให้ค่าใกล้เคียงกับค่าจริงมากกว่าวิธีออยเลอร์ ทุกระยะห่างระหว่างจุด

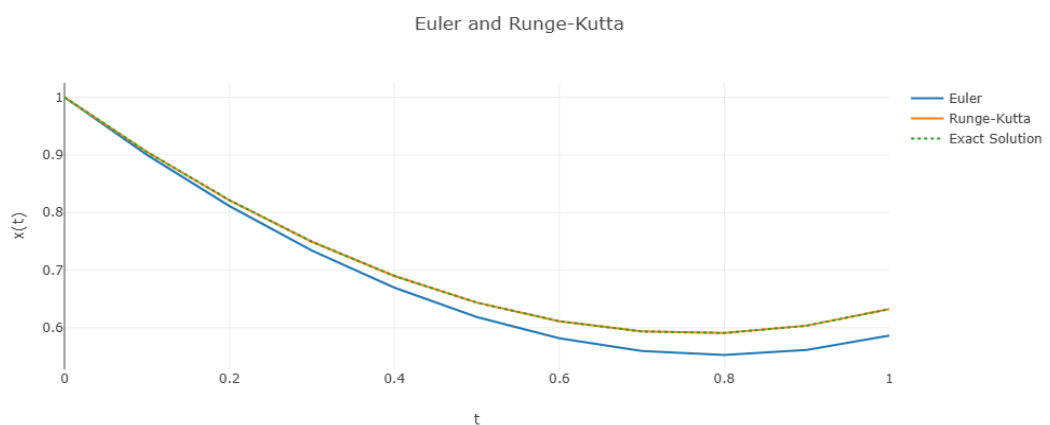
### ตัวอย่างที่ 2

พิจารณาปัญหาค่าเริ่มต้นที่กำหนดโดยสมการ  $x'(t) = t^2 - x$

โดยมีเงื่อนไขเริ่มต้น  $x(0) = 1$  บนช่วง  $0 \leq t \leq 1$

ค่าจริง  $x(t) = t^2 - 2t - e^{-t} + 2$

ผลจากโปรแกรม



รูปที่ 2 : กราฟแสดงการเปรียบเทียบวิธีออยเลอร์กับวิธีรุงเงอ-คุตทาดันดับ 4

**ตารางที่2:** ตารางการเปรียบเทียบวิธีออยเลอร์กับวิธีรุงเงอ-คุททาด้าน 4

## Numerical Results

| t     | Euler .  | Runge-Kutta . | Exact Solution | Euler Error | Runge-Kutta Error |
|-------|----------|---------------|----------------|-------------|-------------------|
| 0.000 | 1.000000 | 1.000000      | 1.000000       | 0.000000e+0 | 0.000000e+0       |
| 0.100 | 0.900000 | 0.905163      | 0.905163       | 5.703486e-3 | 1.396095e-7       |
| 0.200 | 0.811000 | 0.821269      | 0.821269       | 1.250412e-2 | 3.025961e-7       |
| 0.300 | 0.733900 | 0.749182      | 0.749182       | 2.039796e-2 | 4.886539e-7       |
| 0.400 | 0.669510 | 0.689680      | 0.689680       | 2.924538e-2 | 6.943299e-7       |
| 0.500 | 0.618559 | 0.643470      | 0.643469       | 3.871255e-2 | 9.117553e-7       |
| 0.600 | 0.581703 | 0.611189      | 0.611188       | 4.824252e-2 | 1.128090e-6       |
| 0.700 | 0.559533 | 0.593415      | 0.593415       | 5.709651e-2 | 1.326583e-6       |
| 0.800 | 0.552580 | 0.590672      | 0.590671       | 6.448856e-2 | 1.489716e-6       |
| 0.900 | 0.561322 | 0.603431      | 0.603430       | 6.978234e-2 | 1.603665e-6       |
| 1.000 | 0.586189 | 0.632122      | 0.632121       | 7.266202e-2 | 1.662057e-6       |

จากสมการเชิงอนุพันธ์อันดับ 1 โดยการเขียนโปรแกรมจาวาสคริปต์ทำให้ได้ผลดังตาราง ที่มี การแสดงผล ระยะห่างระหว่างจุด ค่าประมาณโดยวิธีออยเลอร์ ค่าประมาณโดยวิธีรุงเงอ-คุททา ค่าจริง ค่าความคลาดเคลื่อนของออยเลอร์และค่าความคลาดเคลื่อนของรุงเงอ-คุททา ผลที่ได้แสดงให้เห็นว่าวิธีรุงเงอ-คุททาให้ค่าใกล้เคียงกับค่าจริงมากกว่าวิธีออยเลอร์ ทุกระยะห่างระหว่างจุด

### บทสรุป

จาก 2 ตัวอย่างข้างต้น ซึ่งเป็นสมการเชิงอนุพันธ์สามัญอันดับ 1 โดยการเขียนโปรแกรมจาวาสคริปต์ เพื่อแสดงผลเฉลย ที่ทำให้ได้ว่าวิธีรุงเงอ-คุททา ให้ความแม่นยำมากกว่าวิธีออยเลอร์ สำหรับการหาค่าจริง ค่าประมาณ และค่าความคลาดเคลื่อน ที่มีประสิทธิภาพและมีการแสดงผลที่หน้าต่าง html จากรูปที่ 1 ในตัวอย่างที่ 1 และรูปที่ 2 ในตัวอย่างที่ 2 และจากข้อมูลที่แสดงค่าความคลาดเคลื่อนในตัวอย่างที่ 1 และ ตัวอย่างที่ 2 โดยมีระยะห่างระหว่างจุดเท่ากันทั้ง 2 ตัวอย่าง

### ข้อเสนอแนะ

ควรมีการเปรียบเทียบสมการเชิงอนุพันธ์ทุกประเภท โดยการเขียนโปรแกรมจาวาสคริปต์

## รายการอ้างอิง

ความคลาดเคลื่อน. (ม.ป.ป.). <https://www.foodnetworksolution.com/wiki/word/7240>

สอน Javascript พื้นฐาน. (ม.ป.ป.). <https://www.mindphp.com/>

Dharma Raj Paudel, Mohan Raj Bhatt. (2023). *Comparative study of Euler's method and Runge-Kutta method to solve an ordinary differential equation through a computational approach*. **Academic Journal of Mathematics Education**. Vol.6. 1

Kasim Abbas Hussain, Fudziah Ismail, Norazak Senu & Faranak Rabiei. (2015). *Fourth Order Improved Runge-Kutta Method for Directly Solving Special Third-Order Ordinary Differential Equations*. **Iran J Sci Technol Trans Sci**. 41:429–43

Md. Amirul Islam. (2015). *Accurate Solutions of Initial Value Problems for Ordinary Differential Equations with the Fourth Order Runge Kutta Methods*. **Journal of Mathematics Research**. Vol.7. 3

Musa H,Ibrahim Saidu, M. Y. Waziri. (2010). *A Simplified Derivation and Analysis of Fourth Order Runge Kutta Method*. **International Journal of Computer Applications**. (0975-8887).

Runge-Kutta methods. (n.d.). [https://en.wikipedia.org/wiki/Runge-Kutta\\_methods](https://en.wikipedia.org/wiki/Runge-Kutta_methods)