# บทที่ 1
# บทนำ

## 1.1 ที่มาและความสำคัญของโครงงาน

โครงงานนี้เริ่มต้นมาจากการที่สมาชิกในกลุ่มโครงงานเริ่มมีความสนใจในเรื่องปฏิสัมพันธ์แรงโน้มถ่วงระหว่างวัตถุในท้องฟ้า (Celestial Bodies) ซึ่งไม่จำกัดอยู่แค่ในระบบสุริยะ ถึงเรื่องที่ได้กล่าวมานี้จะเป็นที่รู้มามากมายแล้ว สามารถสืบค้นหาในหนังสือได้ ฯลฯ อย่างไรก็ตามการจำลองนี้ไม่ได้ทำเพื่อเรื่องของ “การรู้” แต่เป็นเพื่อการจำลองให้เห็นภาพ ดาวเคราะห์นี้จะมีผลกระทบต่อดาวเคราะห์นั้นอย่างไร ฯลฯ ดังนั้นสมาชิกในกลุ่มจึงมีการได้ทำแบบจำลองนี้ขึ้นมา เพื่อศึกษาความซับซ้อนของระบบแรงโน้มถ่วง อย่างเช่นระบบ Trinary ซึ่งจากปัญหา N-Body จะทำให้เห็นได้ว่า เมื่อมีวัตถุหลายชิ้นที่ต่างส่งแรงโน้มถ่วงต่อกัน การทำนายว่าแต่จะวัตถุจะเคลื่อนไหวอย่างไรต่อเป็นสิ่งที่ทำยาก ซึ่งการจำลองนี้สามารถนำไปต่อยอดในการศึกษาทางดาราศาสตร์และฟิสิกข์ได้ในอนาคต

## 1.2 วัตถุประสงค์

1. เพื่อจำลองการเคลื่อนที่ของดาวเคราะห์รอบดาวฤกษ์ศูนย์กลาง
2. เพื่อศึกษาพฤติกรรมของระบบแรงโน้มถ่วงที่ซับซ้อน
3. เพื่อแสดงให้เห็นถึงความสัมพันธ์ระหว่างมวล ตำแหน่ง และความเร็วของวัตถุต่าง ๆ ที่มีผลต่อการเคลื่อนที่ของระบบทั้งหมด

## 1.3 ขอบเขตการศึกษาค้นคว้า

1. ขอบเขตด้านเนื้อหา การศึกษาครั้งนี้คลอบคลุมเนื้อหาเกี่ยวกับปฏิสัมพันธ์แรงโน้มถ่วงระหว่างวัตถุในท้องฟ้าตามความสัมพันธ์แรงโน้มถ่วงระหว่างวัตถุ/การผกผันกับกำลังสองของระยะห่างระหว่างกัน การเคลื่อนที่ของวัตถุในสุญญากาศ และกฎความโน้มถ่วงสากลของนิวตัน

2. ขอบเขตด้านเวลา โครงงานนี้เริ่มจัดทำและศึกษาเมื่อเดือนตุลาคม และจะแล้วเสร็จภายในเดือนพฤศจิกายน

# บทที่ 2
# เอกสารและงานวิจัยที่เกี่ยวข้อง

การศึกษาค้นคว้าครั้งนี้ ผู้ศึกษาได้ทำการศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องโดย แบ่งเป็น หัวข้อดังนี้

1. หลักการของแรงโน้มถ่วง (Newton)

2. หลักการของแรงโน้มถ่วง (Einstein)

3. การจำลองแรงโน้มถ่วงและปฏิสัมพันธ์ระหว่างวัตถุในอวกาศ

    3.1 แบบจำลองทางคณิตศาสตร์ (Mathematical Model)

    3.2 วิธีการเชิงตัวเลข (Numerical Method)

    3.3 การจำลองโดยการใช้ภาษา Python

4. งานวิจัยที่เกี่ยวข้อง

## 2.1 หลักการของแรงโน้มถ่วง (Newton)

แรงโน้มถ่วง (Gravity) เป็นหนึ่งในแรงพื้นฐาน ซึ่งประกอบไปด้วย แรงแม่เหล็กไฟฟ้า แรงนิวเคลียร์อย่างเข้มและแรงนิวเคลียร์อย่างอ่อน โดยแรงโน้มถ่วงเป็นแรงที่ดึงดูดอย่างเดียว ไม่สามารถผลักได้

นิวตันได้กล่าวเป็นแรงโน้มถ่วงเป็นแรงที่กระทำต่อระหว่างสองวัตถุที่มีมวล ดึงดูดเข้าหากัน โดยจะสามารถหาค่าของแรงนี้จากกฎแรงโน้มถ่วงสากลของนิวตัน ซึ่งเขียนเป็นสมการได้ว่า

$$F = G\frac{m_1 m_2}{r^2}$$

โดยที่

- $F$ คือ แรงโน้มถ่วงระหว่างสองวัตถุ (N)
- $G$ คือ ค่าคงที่แรงโน้มถ่วงสากล
- $m_1$ และ $m_2$ คือ มวลของวัตถุ
- $r$ คือระยะห่างระหว่างสองวัตถุนั้น

กฎนี้เป็นกฎสากล มีความหมายว่าทุกอย่างต้องเป็นไปตามกฎนี้ เช่นแอปเปิ้ลตกใส่หัว ดวงจันทร์โคจรรอบโลก

## 2.2 หลักการของแรงโน้มถ่วง (Einstein)

ในปี 1915 ไอสไตน์ได้คิดค้นทฤษฎีสัมพัทธภาพทั่วไป (General Relativity) โดยทฤษฎีได้มีการนิยามแรงโน้มถ่วงว่าแรงที่เกิดขึ้นเพราะกาลอวกาศโค้งตามมวล (Curvature of Space Time) กล่าวง่ายๆ ได้ว่าแทนที่จะดึงดูดวัตถุเข้าหากัน วัตถุจะมีการเคลื่อนที่ไปตามการโค้งของกาลอวกาศ

ทฤษฎีนี้ถูกพิสูจน์โดยการทดลองเช่น Eddington Experiment (1919) ซึ่งเป็นการสังเกตขอบของดวงจันทร์ตอนมีสุริยุปราคาเพราะจะทำให้เห็นดาวข้างๆ เปรียบเทียบกับดาวนั้นตอนไม่มีสุริยุปราคา โดยจะสังเกตเห็นว่าตำแหน่งดาวข้างๆนั้นจะไม่อยู่ที่เดิมเมื่อเปรียบเทียบกับตอนที่ไม่มีสุริยุปราคา สาเหตุเป็นเพราะมวลของดวงอาทิตย์ทำให้อวกาศบิดเบียว ซึ่งส่งผลให้แสงที่เดินทางมาจากดาวนั้นโค้งก่อนแสงมาถึงโลก

ความสัมพันธ์ระหว่างวัตถุและความโค้งของอวกาศถูกอธิบายโดย Einstein Field Equation (EFE)

$$G_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi g}{c^4} T_{\mu\nu}$$

โดยที่

- $G_{\mu\nu}$ คือ Einstein Tensor ซึ่งเป็นการแสดงให้เห็นว่ากาลอวกาศโค้งงอ บิดเบี้ยวอย่างไร
- $\Lambda$ คือ Cosmological Constant ซึ่งแทนพลังงานที่ทำให้จักรวาลขยายเร็วขึ้น
- $g_{\mu\nu}$ คือ Metric Tensor ซึ่งเป็นสิ่งที่ใช้คำณวนระยะทางและเวลาจากวัตถุสองอย่าง
- $c$ คือ ความเร็วแสง มีค่าประมาณ $3 * 10^8$
- $\frac{8\pi g}{c^4}$ คือ Conversion Factor ซึ่งมีค่าประมาณ $2.07665 * 10^{-43}$
- $T_{\mu\nu}$ คือ Stress Energy Tensor ซึ่งเป็นการแสดงให้เห็นว่า ณ ตอนนั้นว่าวัตถุนั้นเป็นอย่างไร (มวล, พลังงาน ความดัน ฯลฯ)

Tensor คือ สิ่งที่ใช้เก็บข้อมูลที่มีขนาดเดียวกันและชนิดเดียวกัน ซึ่งสามารถมีกี่มิติก็ได้ ซึ่ง Tensor เป็นแนวคิดที่ต่อยอดมาจาก Scalar (0 มิติ), Vector (1 มิติ), Matrix (2 มิติ)

EFE เป็นสิ่งที่สามารถอธิบายปรากฏการณ์ที่กฎของนิวตันไม่สามารถอธิบายได้เช่น การโค้งของแสง แต่เมื่อสนามโน้มถ่วงนั้นอ่อนแอ่ เช่นใน ระบบสุริยะ และความเร็วช้ากว่าความเร็วแสงอย่างมาก ทำให้กาลอวกาศแทบไม่มีการโค้งงอเลย EFE จะลดรูปเป็นกฎของนิวตัน

ถึงการที่จำลองระบบเหล่านี้ด้วยทฤษฎีของไอน์สไตน์จะมีความถูกต้องมากกว่า หรือแม่นยำมากกว่า แต่การจำลองจะมีความซับซ้อนทางการคำณวนมากกว่าทำให้ต้องมีการใช้ Supercomputers ในการจำลอง จึงได้มีการตัดสินใจใช้กฎของนิวตันมาจำลองแทน เพราะมีความซับซ้อนน้อยกว่าและทำให้สามารถมีการใช้เครื่องคำณวน (คอมพิวเตอร์) ที่ไม่ต้องดีเท่า Supercomputers ได้

## 2.3 การจำลองแรงโน้มถ่วงและปฏิสัมพันธ์ระหว่างวัตถุในอวกาศ

เพื่อการที่จะเข้าใจแรงโน้มถ่วงและปฏิสัมพันธ์ระหว่างวัตถุในอวกาศได้ดีขึ้น จึงได้มีการสร้างแบบจำลองขึ้นมา โดยแบบจำลองนี้จะเน้นไปที่การจำลองแบบที่ทุกอย่างสามารถมีปฏิสัมพันธ์กันได้ หรือเรียกอีกอย่างได้ว่าการจำลอง N-Body ซึ่งจะมีความแม่นยำมากกว่าการจำลองเช่น Two-Body ที่จำลองแค่สองวัตถุ

### 2.3.1 แบบจำลองทางคณิตศาสตร์ (Mathematical Model)

เพื่อให้แบบจำลองแม่นยำ จึงต้องมีการสร้างแบบจำลองทางคณิตศาสตร์ที่สามารถควบคุมปฏิสัมพันธ์ระหว่างวัตถุได้ถูกต้อง และแม่นยำ โดยการใช้กลศาสตร์นิวตัน

เริ่มด้วยการหาแรงดึงดูดของแต่ละวัตถุ โดยข้อมูลที่ต้องมีในการหาแรงดึงดูดนั้นคือระยะทาง ซึ่งสามารถหาได้ด้วยสูตรพีทาโกรัส

$$r = \sqrt{x^2 + y^2}$$

โดยที่

- $x$ คือระยะทางในแกน X ซึ่งหาได้จากการลบตำแหน่งในแกน X ของวัตถุนั้น กับวัตถุนี้
- $y$ คือระยะทางในแกน Y ซึ่งหาได้จากการลบตำแหน่งในแกน Y ของวัตถุนั้น กับวัตถุนี้

และเมื่อได้ข้อมูลระยะทางแล้วจึงสามารถใช้กฎแรงโน้มถ่วงสากลของนิวตันหาแรงดึงดูดได้

$$F = G\frac{m_1 m_2}{r^2}$$

โดยตัวแปรแต่ละตัวได้มีการอธิบายใน 2.1

แต่แรงที่ได้รับมาจากกฎแรงโน้มถ่วงสากลของนิวตันเป็นแค่ตัวเลข แรงนั้นเป็นเวกเตอร์จึงต้องเปลี่ยนเป็นเวกเตอร์โดยการหาทิศทางที่แรงดึงดูดนั้นดึงไป ซึ่งหาได้ด้วย

$$\theta = atan2(x, y)$$

โดยที่

- $\theta$ คือทิศทางที่แรงนั้นถูกดึงไป จะมีค่าเป็นเรเดียน
- $atan2(x, y)$ เป็นฟังค์ชั่นที่หามุมของเวกเตอร์จาก (0, 0) ถึง (x, y)

แล้วจึงเปลี่ยนเป็นเวกเตอร์โดยการแยกแรงเป็นแรงในทิศ x และทิศ y โดยการใช้สูตร

$$Force_x = Fcos(\theta)$$
$$Force_y = Fsin(\theta)$$

เมื่อรวม $Force_x$ และ $Force_y$ จะได้ $\vec{F}$ หรือเขียนเป็นอีกรูปแบบได้ $\vec{F} = \langle F_x, F_y \rangle$ ซึ่งเป็นแรงในรูปเวกเตอร์ ซึ่งจะช่วยในการตัดสินใจว่าวัตถุนั้นจะอยู่ที่ไหนในเฟรมต่อไปในแบบจำลอง

### 2.3.2 วิธีการเชิงตัวเลข (Numerical Method)

วิธีการทางคณิตศาสตร์ที่ใช้ประมาณค่าคำตอบของปัญหาทางคณิตศาสตร์ เมื่อคำตอบนั้นไม่สามารถหาได้ด้วยการวิเคราะห์ โดยวิธีเหล่านี้มักใช้คอมพิวเตอร์มาช่วยในการคำณวน ยกตัวอย่างเช่นสมการ $x^2 = 2$ สมการนี้จะมีคำตอบในเชิงวิเคราะห์คือ $x = \sqrt{2}$ แต่จะมีคำตอบในเชิงตัวเลขเป็น $x = 1.415$

วิธีการที่ใช้ในแบบจำลองนี้ จะเป็นวิธี Numerical Integration หรือ Quadrature ซึ่งเป็นชนิดของ Numerical Method โดยปัญหาเหล่านี้จะเป็นการแก้โดยการอินทิเกรต (ปริพันธ์)

$$\int_a^b f(x)dx$$

ในแบบจำลองนี้ Numerical Method มีหน้าที่ในการประมาณหาตำแหน่งต่อไปของวัตถุที่อยู่ในแบบจำลอง

โดยมีหลากหลายวิธีในการประมาณค่าอินทิกรัลให้ได้ความแม่นยำตามที่ต้องการ โดยในโครงงานนี้จะมีการกล่าวถึง 2 แบบ

แบบแรกที่จะกล่าวถึงคือ Euler Method ซึ่งเป็นวิธีการเชิงตัวเลขที่แบบจำลองนี้เคยใช้ โดยวิธีนี้เป็นวิธีการประมาณการเคลื่อนที่ของวัตถุอย่างต่อเนื่องภายใต้อิทธิพลของแรงโน้มถ่วง

เมื่อรู้ตำแหน่ง $(r)$ และความเร็ว $(v)$ ของวัตถุที่เวลาใดเวลาหนึ่ง วิธีการนี้จะแบ่งการคำนวณออกเป็นช่วงเวลาเล็กๆ $(\Delta t)$ ในแต่ละขั้น แล้วจึงใช้ความเร่ง $(a)$ ที่คำนวณมาจากแรงโน้มถ่วงปัจจุบัน เพื่อประเมินตำแหน่งต่อไป:

ความเร็วใหม่: ความเร็วใหม่ $(v_{i+1})$ จะถูกประมาณโดยประเมินว่าความเร่งปัจจุบันคงที่ตลอดช่วงเวลา $\Delta t$ :

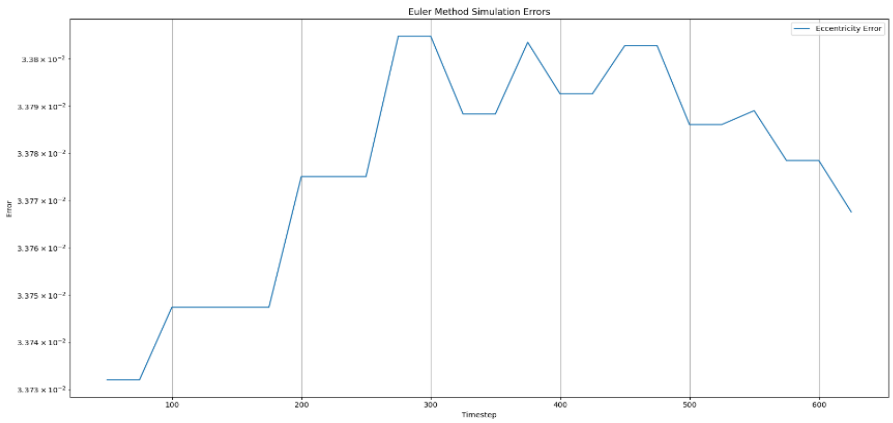
$$v_{i+1} = v_i + a_i \Delta t$$

ตำแหน่งใหม่: ตำแหน่งใหม่ $(r_{i+1})$ จะถูกประมาณโดยใช้ความเร็วปัจจุบัน:

$$r_{i+1} = r_i + v_i \Delta t$$

โดยวิธีนี้มีข้อบกพร่อง คือเป็นวิธี First-Order Integrator หมายความว่าวิธีนี้ใช้ค่าความเร่งที่ ณ จุดเริ่มต้นของช่วงเวลาเท่านั้นในการคำนวณความเร็วและตำแหน่งใหม่ ทำให้ไม่สามารถพิจารณาการเปลี่ยนแปลงของแรงโน้มถ่วงและความเร่งที่เกิดขึ้นอย่างต่อเนื่องขณะที่วัตถุเคลื่อนที่

ดังนั้นความแม่นยำของวิธีนี้ จะเกี่ยวข้องกับขนาดของช่วงเวลานั้นๆ เมื่อช่วงเวลานั้นเล็กลงการประมาณความเร็วและตำแหน่งก็จะมีความแม่นยำมากขึ้น เนื่องจากวิธีนี้จะคำณวนค่าความเร่ง ณ แค่จุดเริ่มต้นของช่วงเวลาเท่านั้น อย่างไรก็ตามเพราะการที่ประเมินความเร่งปัจจุบันจะคงที่ตลอดช่วงเวลานั้นๆ ความผิดพลาดเล็กน้อยจะสะสมไปเรื่อยๆตามเวลา ทำให้วงโคจรนั้นเคลื่อนออกไปจากวงโคจรจริงๆ ไปทีละเล็กน้อย




ภาพ 2-1 (ความผิดพลาดความเยื้องศูนย์กลางของ Euler Method เมื่อเปรียบเทียบ Δt ที่ 6hr/step และ 48hr/step)

สังเกตจากกราฟนี้ ความผิดพลาดของ Euler Method จะผันผวนอยู่ที่ประมาณ 0.3373 ถึง 0.338 และมีแนวทางที่จะเติบโตขึ้นไปเรื่อยๆ ถึงการที่กราฟนี้จะลดลงไปในช่วงเวลา 500 ถึง 600 ก็ตาม ไม่ได้เป็นการบ่งชี้ถึงการที่ Euler Method จะถูกต้องขึ้นตามเวลา แต่เป็นผลกระทบเฉพาะซึ่งเกิดขึ้นเมื่อวัตถุนั้นอยู่ในตำแหน่งเฉพาะที่ เมื่อการจำลองนี้จบลง

แบบสองที่จะกล่าวถึงคือ Runge-Kutta 4th Order (RK4) ซึ่งเป็นวิธีการเชิงตัวเลขที่แบบจำลองนี้กำลังใช้ โดยวิธีนี้จะประมาณตำแหน่งใหม่ได้ดีกว่า Euler Method มาก

RK4 จะมีการลดข้อผิดพลาดโดยการประมาณความเร่ง 4 ครั้งในหนึ่งช่วงเวลา ต่างกับ Euler Method ที่ประมาณความเร่งแค่ 1 ครั้งในหนึ่งช่วงเวลา (ซึ่งทำให้ต้องประมาณว่าความเร่งนั้นคงที่ตลอดช่วงเวลานั้น)

โดยการประมาณนั้นจะต้องมีการหาค่าความชัน หรืออัตราการเปลี่ยนแปลงของความเร็วและความเร่ง ซึ่งวิธีหานั้นจะเป็นการหาอนุพันธ์ (Differentiation) จากกฎของที่สองของนิวตัน

$$F_{net} = ma \Rightarrow a = \frac{F_{net}}{m}$$

โดยที่:

- $F_{net}$ คือผลรวมของแรง
- $m$ คือมวล
- $a$ คือความเร่ง

เป็นเรื่องที่ทราบกันดีกว่า ความเร่ง คืออัตราการเปลี่ยนแปลงของความเร็วกับเวลา ($ax, ay$)

แล้วจึงหาอัตราการเปลี่ยนแปลงหรืออนุพันธ์ของตำแหน่งของวัตถุนั้นๆ ซึ่งตามปกติ จะมีค่าเป็นความเร็วกับเวลา ($vx, vy$)

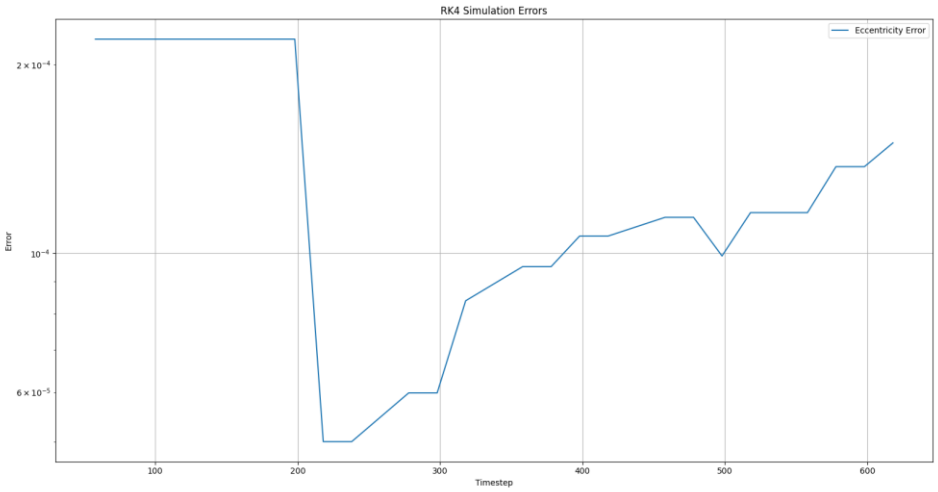
เมื่อได้ทั้งสี่ตัวแปรแล้ว จึงนำตัวแปรเหล่านั้นมาประมาณหาค่าความเร็วและความเร่ง:

1. $k_1$ คือค่าความชันของตัวแปรทั้งสี่ที่จุดเริ่มต้นของช่วงเวลา ($t$)
2. $k_2$ คือค่าประมาณความชันที่คำนวณที่จุด $t + \frac{\Delta t}{2}$ (จุดครึ่งทาง) โดยใช้ค่าตัวแปรจาก $k_1$
3. $k_3$ คือค่าประมาณความชันที่คำนวณที่จุด $t + \frac{\Delta t}{2}$ (จุดครึ่งทาง) โดยใช้ค่าตัวแปรจาก $k_2$
4. $k_4$ คือค่าประมาณความชันที่คำณวนที่จุด $t + \Delta t$ (จุดสุดท้าย) โดยใช้ค่าตัวแปรจาก $k_3$

แล้วจึงนำค่าความชันที่ได้มานั้นไปถัวเฉลี่ยถ่วงน้ำหนักเพื่อหาความชันเฉลี่ย:

$$= \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

การเฉลี่ยนี้ให้ความสำคัญกับ $k_2$ และ $k_3$ มากกวา เพราะการประมาณที่จุดครึ่งทางมักให้ค่าความชันที่ถูกตองกวา




ภาพ 2-2 (ความผิดพลาดความเยื้องศูนย์กลางของ Runge-Kutta 4th Order เมื่อเปรียบเทียบ Δt ที่ 6hr/step และ 48hr/step)

สังเกตจากกราฟนี้ ความผิดพลาดของ RK4 จะผันผวนอยู่ที่ประมาณ 0.0006 ถึง 0.002 และมีแนวทางที่จะเติบโตขึ้นไปเรื่อยๆ เหมือนวิธีการเชิงตัวเลขอื่นๆ ซึ่งเมื่อเปรียบเทียบความผิดพลาดของวิธีนี้กับ Euler Method จะเห็นได้ว่าความผิดพลาดของ RK4 จะน้อยกว่า Euler Method เป็นอย่างมากที่ 0.0006 กับ 0.33 และ 0.002 กับ 0.38 เนื่องจากที่ Euler Method เป็น First Order Integrator ในขณะที่ RK4 เป็น Fourth Order Integrator

### 2.3.3 การจำลองโดยการใช้ภาษา Python

ในส่วนนี้จะมีเป็นการกล่าวถึงและอธิบายโปรแกรมที่ได้มีการเขียนด้วยภาษา Python เพื่อการจำลองปฏิสัมพันธ์ระหว่างวัตถุ

ก่อนจะเริ่มเขียนโปรแกรมอะไรใน Python เราจะต้องมีการเรียกใช้ไลบรารี่ก่อน โดยการใช้คำสั่ง import โดยสิ่งที่จะเรียกใช้มี pygame, math, matplotlib.pyplot เพื่อจะได้สามารถเรียกใช้โมดูลของไลบรารี่เหล่านี้

หลังจากนั้นจึงมีการสร้าง class ขึ้นมา โดย class เปรียบเสมือนเป็นพิมเขียว (blueprint) ซึ่งจะใช้เพื่อการสร้างวัตถุต่างๆในแบบจำลองนี้ ซึ่งจะเริ่มด้วยการกำหนดตัวแปรหลายๆตัว เช่น AU (Astronomical Unit) , G (Gravitational Constant), SCALE, TIMESTEP (dt )หลังจากกำหนดตัวแปรพวกนี้แล้วจึงสร้างฟังค์ชั่น __init__ เป็นฟังค์ชั่นกำหนดค่าเริ่มต้นของออบเจ็กส์ โดย Parameter ที่เขียนไว้ในฟังค์ชั่นนี้จะเป็นสิ่งที่จะใช้เป็น Argument เมื่อเรียกใช้ class นี้ เช่น ให้ class นี้มีชื่อว่า person และในฟังค์ชั่น __init__ มี Parameter เป็น (self, name, age) โดยที่ self นั้นเป็นออบเจ็กส์ เมื่อมีการเรียกใช้ class นี้จะเรียกเป็น person(“John”, 18)

โดยภายใน class นี้จะมีหลายฟังค์ชั่น เช่นฟังค์ชั่น draw ซึ่งเป็นฟังค์ชั่นที่ใช้เพื่อการ “วาด” ลงบนหน้าจอโดยใช้ไลบรารี่ pygame โดยใช้โมดูล pygame.draw และฟังค์ชั่นอื่น เป็นฟังค์ชั่นเพื่อคำนวณด้านฟิสิกส์ซึ่งได้กล่าวไว้ใน 2.3.1 แบบจำลองทางคณิตศาสตร์ และคำนวณวิธีการเชิงตัวเลข Runge-Kutta 4$^{th}$ Order เพื่อคำนวณตำแหน่งต่อไปของวัตถุนั้นๆ ซึ่งได้กล่าวไว้ใน 2.3.2 วิธีการเชิงตัวเลข

แล้วจึงสร้างฟังค์ชั่น main ซึ่งเป็นฟังค์ชั่นที่ควบคุมทุกอย่าง โดยจะเริ่มด้วยกันสร้างตัวแปรของวัตถุต่างๆ โดยการใช้ class ที่ได้สร้างไว้ เช่น sun = Planet(“Argument”) เมื่อ Planet คือชื่อของ class หลังจากนั้นจะมีการสร้าง while loop เพื่อเริ่มโปรแกรมหลักโดยใช้ไลบรารี่ pygame ซึ่งภายในจะมี for loop เพื่อใช้วนอ่านข้อมูลของแต่ละวัตถุในแบบจำลอง โดยใน loop นี้จะมีการใช้คำสั่ง .draw ไปเรื่อยเพื่อใช้เป็นการอัพเดทข้อมูลตำแหน่งใหม่ของวัตถุนั้นๆ และเพื่อคำนวณข้อมูลต่างๆ เพื่อไปใช้ทำกราฟหาข้อมูลโดยใช้ไลบรารี่ matplotlib.plt อย่างเช่นการคำนวณว่าวัตถุนั้นโคจรครบไปกี่รอบแล้วโดยการใช้ฟังค์ชั่น math.atan2 เพื่อหามุมของวัตถุเมื่อเทียบกับดาวนั้นๆ ซึ่งจะหาโดยการหาว่ามุมของวัตถุนั้นมีค่าเท่ากับ 0 เรเดียนหรือปล่าว

ตอนท้ายของฟังค์ชั่น main จะเป็นการใช้โมดูลของ pygame เพื่ออัพเดทแต่ละเฟรมหรือ dt ที่ผ่านไป

หลังจากนั้นจะเป็นการนำข้อมูลที่ได้มา แล้วเปลี่ยนเป็นกราฟโดยไลบรารี่ matplotlib.plt ซึ่งจะมีการอธิบายเพิ่มใน บทที่ 4 ผลการดำเนินงาน

## 2.4 งานวิจัยที่เกี่ยวข้อง

Kyle Emmi, Braden Oh, Madi Wyatt ได้มีการจัดทำงานวิจัยเรื่อง Orbital Dynamics ในหัวข้อ Modeling Orbital Mechanics Through Various Methods in Python (2018) โดยการวิจัยจะมีความคล้ายโครงงานนี้ ที่การวิจัยนี้จะเป็นทดลองการจำลองกลศาสตร์โคจร แต่ต่างกันที่เปรียบเทียบความถูกต้องโดยใช้วิธี Euler Method และ Equation-Derived Trajectory หรือเรียกได้อีกแบบว่าเป็นการใช้การอนุรักษ์โมเมนตัมเชิงมุมเพื่อสร้างวิถีโคจร และเป็นการจำลอง Two Body โดยผลที่ได้จากการจำลองทั้งสองนั้น คือวิธีทั้งสองนั้นสร้างวงโคจรที่คล้ายกันมาก แต่เพื่อรู้ข้อผิดพลาดเฉลี่ยในตำแหน่งได้มีการคำนวณรูปแบบวงโคจรที่แตกต่างกัน 10 แบบ ซึ่งข้อผิดพลาดที่เฉลี่ยมานั้นได้ช่วงตั้งแต่ประมาณ 129.7 กม. ถึง 2.99 ล้านกม.

Tailin Zhu ได้มีการทำแบบจำลองของระบบสุริยะแบบ N-Body โดยการวิจัยนี้มีชื่อว่า N-body Simulations of the Solar System with CPU-based Parallel Methods การวิจัยนี้ เพื่อเรียนรู้พลวัตของระบบโคจรภายใต้แรงโน้มถ่วงของนิวตันโดยการอินทิเกรตเชิงตัวเลขของเซ็ท N (N จาก N-Body) โดยการใช้วิธีการเชิงตัวเลข Verlet 2$^{nd}$ Order และ dt ที่ 0.1 วัน ซึ่งจะสามารถคำนวณได้เร็วกว่า RK4 แต่แลกกับข้อผิดพลาดที่ต่ำลง วัตถุในแบบจำลองนี้จะมี ดวงอาทิตย์, 30 วัตถุหลัก และ 10,000 ดาวเคราะห์น้อย

ซึ่งวัตถุที่จำลองนี้จะมีเยอะมาก และจะต้องคำนวณตำแหน่งวัตถุแต่ละวัตถุเพราะเป็นแบบจำลอง N-Body โดย Tailin Zhu ได้ใช้ “Parallel Programming” เป็นการเขียนโปรแกรมที่จะแบ่งงานใหญ่หนึ่งเป็นงานย่อยๆ แล้วประมวลงานย่อยเหล่านั้นไปพร้อมๆกัน ซึ่งวิธีนี้จะลดเวลาการคำนวณแบบจำลองนี้ไปเยอะมาก โดย Tailin Zhu ได้ใช้วิธี 2 แบบ OpenMP และ MPI ซึ่งเขาได้ใช้คู่กับ BlueCrystal Phase3 Supercomputer ที่มหาวิทยาลัย Bristol

โดยผลลัพท์ที่เขาได้คือ ข้อผิดพลาดของวงโคจรโลกคือ ±0.02% ซึ่งถือว่ายอมรับได้ แต่ข้อผิดพลาดของวงโคจรดวงจันทร์นั้นประมาณ 70 เท่าของโลก และเขาได้ข้อสรุปว่าแบบจำลองที่มีวัตถุมากกว่า 10,000 วัตถุนั้นวิธี MPI จะเร็วกว่า แต่ในทางกลับกันถ้าแบบจำลองมีวัตถุน้อยกว่า 10,000 วัตถุ วิธี OpenMP จะเร็วกว่า

# บทที่ 3
# วิธีการดำเนินโครงงาน

ในการจัดทำโครงงานการสร้างแบบจำลองและการแสดงภาพของพลวัตวงโคจรหลายตัวด้วย Python นี้ ได้มีขั้นตอนดังนี้

## 3.1 การคัดเลือกหัวข้อและศึกษาค้นคว้า

3.1.1 คัดเลือกหัวข้อจากการปรึกษาสมาชิกในโครงงาน

3.1.2 ศึกษาจากงานวิจัยที่เกี่ยวข้อง หนังสือ เว็บไซต์ และคำปรึกษาจากครูที่ปรึกษา

## 3.2 ขั้นตอนการดำเนินงาน

3.2.1 คัดเลือกภาษาที่จะใช้เพื่อการเขียนโปรแกรมนี้

3.2.2 ใช้ซอฟต์แวร์ Visual Studio Code หรือ PyCharm คู่กับ Git เพื่อการเขียนโปรแกรม

3.2.3 คัดเลือกวิธีการเชิงตัวเลขเพื่อใช้ในโปรแกรมนี้

3.2.4 เมื่อเขียนโปรแกรมเสร็จแล้วจึงเก็บข้อมูล อย่างเช่นความเร็ว ระยะห่างจากดวงอาทิตย์ และความเยื้องศูนย์

3.2.5 แล้วจึงใช้ข้อมูลที่ได้มานำไปทำกราฟโดยไลบรารี่ matploblib และโมดูล pyplot

3.2.6 จัดทำเอกสารรายงานบนซอฟต์แวร์ MS Word

3.2.7 เมื่อทำเอกสารรายงานเสร็จ จึงเสนอให้ครูปรึกษาตรวจสอบ

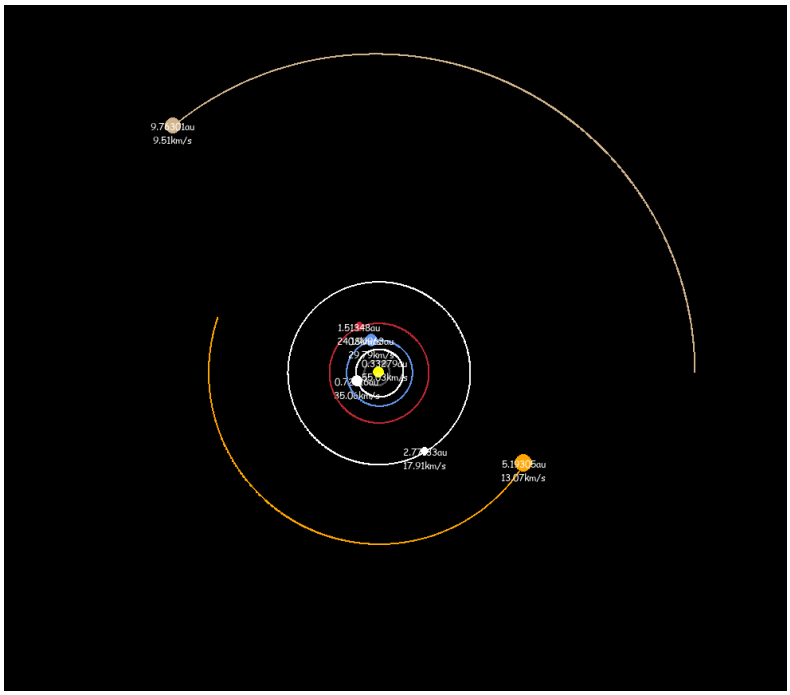
3.2.8 จัดทำเสนอโครงงาน โดยการทำบอร์ดเสนอโครงงาน

# บทที่ 4
# ผลการดำเนินงาน

การจัดทำโครงงานนี้ มีวัตถุประสงค์เพื่อเรียนรู้และเข้าใจพลวัตของวงโคจรแบบ N-Body เพื่อให้ผู้จัดทำโครงงาน และผู้ที่ได้เรียนรุ้เกี่ยวกับโครงงานนี้ สามารถนำไปประยุกต์ต่อในด้านการศึกษาได้ ซึ่งมีผลการดำเนินโครงงานตามนี้

## 4.1 ผลลัพท์ของโครงงาน

โครงงานนี้ได้สำเร็จลุล่วงในการจำลองพลวัตของวงโคจรแบบ 2 มิติ ที่สามารถใช้งานได้ แบบจำลองนี้จะคำนวณตำแหน่งใหม่ของวัตถุนั้นทุกๆ $\Delta t$ โดยการใช้สูตรแรงโน้มถ่วงสากลของนิวตัน โดยจะสามารถเห็นวงโคจร ความเร็ว ระยะห่างจากดวงอาทิตย์ (au) ของวัตถุนั้นๆ ได้ ซึ่งช่วยให้สามารถมองเห็นการเคลื่อนไหวในทั้งเชิงตัวเลขและเชิงภาพได้ โดยแบบจำลองจะมีลักษณะตามภาพ 4-1



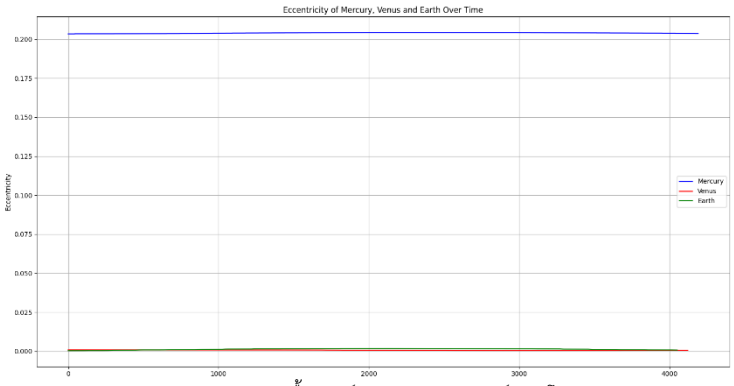

ภาพ 4-1 (แบบจำลอง)

โดยดาวเคราะห์แต่ละดาวเคราะห์จะถูกแทนด้วยสี อย่างเช่น สีขาวถัดจากดวงอาทิตย์ที่อยู่ตรงกลางคือดาวพุธ และ วัตถุสีขาวระหว่างสีแดง (ดาวอังคาร) และสีส้ม (ดาวพฤหัสบดี) คือดาวเคราะห์แคระ Ceres

## 4.2 การแสดงผลลัพท์ในรูปแบบกราฟโดยการใช้ Matplotlib

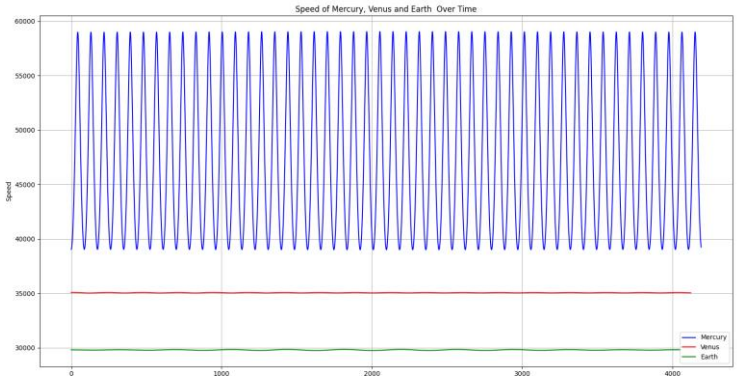
แบบจำลองนี้สามารถให้ข้อมูลหลายอย่างได้เช่น ความเร็ว ระยะห่างจากดวงอาทิตย์ ความเยื้องศูนย์ของวัตถุนั้นๆ โดยที่ข้อมูลเหล่านั้นจะถูกนำมาแสดงเป็นกราฟโดยใช้โมดูล pyplot ของไลบรารี่ matplotlib

แต่ละกราฟนั้นจะแสดงแค่ข้อมูลของดาวพุธ ดาวศุกร์ และโลก โดยข้อมูลที่จะถูกรวบรวมเมื่อใช้วิธีการเชิงตัวเลข Runge-Kutta $4^{th}$ Order ซึ่งข้อมูลแกน X จะเป็น $\Delta t$ เสมอ



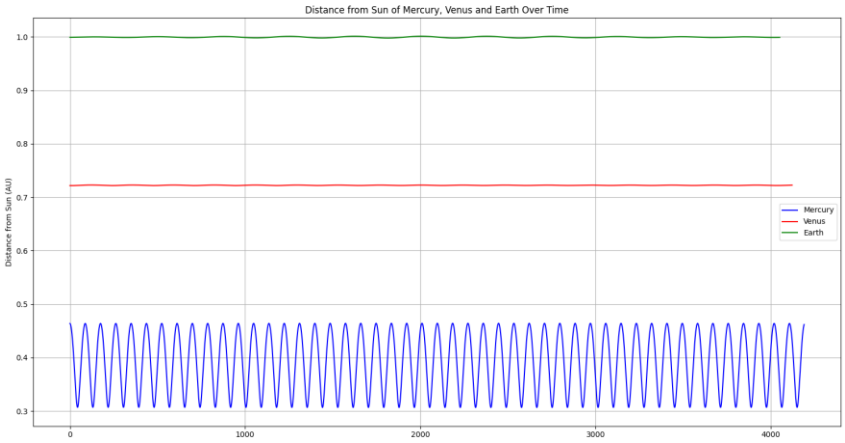

ภาพ 4-2 (ความเยื้องศูนย์ของดาวพุธ ดาวศุกร์ และโลก)

สังเกตจากกราฟดาวพุธนั้นจะมีค่าความเยื้องศูนย์สูงกว่า ดาวศุกร์และโลก เพราะวงโคจรของดาวพุธนั้นมีลักษณะเป็นวงรี โดยที่เมื่อความเยื้องศูนย์มีค่าเท่ากับ 0 วงโคจรจะเป็นวงกลม เมื่อมีค่าเท่ากับ 1 วงโคจรเป็นพาราโบลา และเมื่อมีค่ามากกว่า 1 วงโคจรจะเป็นไฮเพอร์โบลา (กล่าวเพิ่มใน 5.2)




ภาพ 4-3 (ความเร็วของดาวพุธ ดาวศุกร์ และโลก)

สังเกตจากกราฟนั้นจะเห็นได้ว่าดาวพุธจะสั่นไหว (Oscillate) ขึ้นลงไปมา ซึ่งเป็นเพราะที่ดาวพุธมีความเยื้องศูนย์สูงกว่าดาวเคราะห์อื่นๆในแบบจำลอง ที่ 0.2 หรือ 20% ทำให้วงโคจรเป็นวงรี โดยเมื่อดาวพุธเข้าใกล้ดาวมากขึ้นความเร็วจะสูงขึ้น และเมื่อออกห่างจากดาวมากขึ้นความเร็วจะลดลง




ภาพ 4-4 (ระยะห่างจากดวงอาทิตย์ของดาวพุธ ดาวศุกร์ และโลก)

# บทที่ 5
# สรุปผลของโครงงาน

การทำโครงงานนี้ ทำให้สมาชิกในกลุ่มของโครงงานได้มีความรู้มากขึ้น ซึ่งเกี่ยวข้องกับฟิสิกข์ ในเรื่องดาราศาสตร์ แรงโน้มถ่วง พลวัตวงโคจร และการเขียนโปรแกรมด้วยภาษา Python

ผู้จัดทำได้มีการทำโครงงานตามลำดับที่กล่าวในบทที่ 3 จนกระทั่งเสร็จสิ้นลุล่วงไปด้วยดี ซึ่งสามารถสรุปผลการดำเนินงาน อภิปรายผล และข้อเสนอแนะตามลำดับดังต่อไปนี้

## 5.1 สรุปผลการโครงงาน

การจัดทำโครงงานในครั้งนี้ ผู้จัดทำได้ทำโครงงานวิทยาศาสตร์ การสร้างแบบจำลองและการแสดงภาพของพลวัตวงโคจรหลายตัว ดวย Python เพื่อการเรียนรู้ ดังรายละเอียดต่อไปนี้

### 5.1.1 วัตถุประสงค์โครงงาน

1. เพื่อจำลองการเคลื่อนที่ของดาวเคราะห์รอบดาวฤกษ์ศูนย์กลาง
2. เพื่อศึกษาพฤติกรรมของระบบแรงโน้มถ่วงที่ซับซ้อน
3. เพื่อแสดงให้เห็นถึงความสัมพันธ์ระหว่างมวล ตำแหน่ง และความเร็วของวัตถุต่าง ๆ ที่มีผลต่อการเคลื่อนที่ของระบบทั้งหมด

### 5.1.2 ผลของโครงงาน

โครงงานนี้ ได้มีการสร้างแบบจำลองที่สามารถทำนายตำแหน่งของวัตถุในแบบจำลองได้อย่างแม่นยำซึ่งมีความผิดพลาดแค่ ±0.006% [ภาพ 2-2] โดยการใช้ภาษา Python ในการโปรแกรมขึ้นมาและใช้ Runge-Kutta $4^{th}$ Order ซึ่งทำให้แบบจำลองมีความแม่นยำเป็นอย่างมาก

โครงงานนี้เปรียบเสมือนบทนำสู่ความซับซ้อนของกลศาสตร์การโคจร ซึ่งแบบจำลองนี้เป็นแค่แบบจำลองเบื้องต้นที่ไม่ได้คำนวณทฤษฎีสัมพันธภาพทั่วไปของไอน์สไตน์ แต่คำนวณแค่กฎแรงโน้มถ่วงของนิวตัน นอกจากนี้แบบจำลองยังช่วยให้เห็นความสัมพันธ์ระหว่างมวล ตำแหน่ง และความเร็วของวัตถุ ซึ่งทำให้สามารถเข้าใจพฤติกรรมของระบบโคจรได้ ถึงแม้ว่าจะเป็นแบบจำลองพื้นฐานที่ใช้กฎแรงโน้มถ่วงนิวตัน แต่ก็สามารถใช้เป็นฐาน เพื่อให้ผู้ที่ได้อ่าน หรือศึกษาเกี่ยวกับโครงงานนี้ไปปรับใช้ต่อได้ในอนาคต เช่น การเปลี่ยนไปใช้แรงโน้มถ่วงแบบไอน์สไตน์

## 5.2 อภิปรายผล

โดยหัวข้อย่อยนี้จะกล่าวถึงข้อมูลที่ได้รับจากแบบจำลองแล้วจึงนำมาพล็อตเป็นกราฟในหัวข้อย่อย 4.2 ในบทที่ 4

จากรูป 4-2 ความเยื้องศูนย์ (e) เป็นค่าที่ใช้อธิบาย ความเบี่ยงเบนของวงโคจรจากวงกลม โดยมีค่าแตกต่างกันไปตามรูปแบบวงโคจร เช่น วงโคจรรอบดวงอาทิตย์ จะมีค่าระหว่าง 0 ถึง >1

- e = 0 วงโคจรเป็นวงกลม
- 0 < e < 1 วงโคจรเป็นวงรี
- e = 1 วงโคจรเป็นพาราโบลา
- e > 1 วงโคจรเป็นไฮเพอร์โบลา

เมื่อสังเกตจากกราฟจะเห็นว่าดาวพุธ (Mercury) จะมีค่าอยู่ประมาณ 0.2 ส่งผลให้ดาวพุธมีวงโคจรเป็นวงรี โดยสาเหตุนั้นเป็นเพราะระยะห่างจากดวงอาทิตย์ ทำให้แรงโน้มถ่วงจากดวงอาทิตย์ส่งผลต่อดาวพุธมากขึ้น และการที่วงโคจรเข้าไปใกล้ดาวศุกร์และดาวพฤหัสบดีทำให้ถูกดึง

เมื่อวงโคจรนั้นเป็นวงรีจะทำให้มีตำแหน่งที่ชื่อว่า Periapsis และ Apoapsis หรือ จุดต่ำสุดในวงโคจร และจุดสูงสุดในวงโคจร ซึ่งวัตถุนั้นจะเร็วขึ้นเมื่อเข้าใกล้ Periapsis และช้าขึ้นเมื่อเข้าใกล้ Apoapsis ตามกฎของนิวตันที่ระยะห่างจะแปรผกผันกับแรง

ความเยื้องศูนย์จะหาได้โดยการหา ผลต่างของจุดสูงสุดและจุดต่ำสุดในวงโคจรหารกับผลบวกของจุดสูงสุดและจุดต่ำสุดในวงโคจร ซึ่งการที่ในภาพ 4-2 เส้นของดาวพุธเป็นเส้นตรงก็เป็นอีกหลักฐานที่พิสูจน์ได้ว่าแบบจำลองนี้มีความแม่นยำ

## 5.3 ข้อเสนอแนะ

เพื่อให้ผู้ที่ได้เรียนรู้เกี่ยวกับโครงงานนี้แล้วอยากนำไปต่อยอด ผู้จัดทำจึงได้มีการจัดทำข้อเสนอแนะเพื่อการต่อยอดต่อไปนี้

1. จำลองเป็นแบบจำลอง 3 มิติโดยการใช้ Pygame ควบกับ PyOpenGL
2. ใช้เป็นสื่อการสอน เพื่อการสอนฟิสิกข์ดาราศาสตร์
3. ใช้วิธีการเชิงตัวเลขอื่นๆ เช่น Velocity Verlet หรือ RK45
4. การปรับค่าคงที่ (มวล, ความเร็ว ฯลฯ) ในเรียลไทม์
    a. ใช้ PCBs เช่น Raspberry Pi เพื่อการปรับค่าคงที่โดยการใช้ลูกบิด หรือสวิตซ์