



โปรแกรมจำลองการทำงานของสนามแรงที่โปรแกรมได้

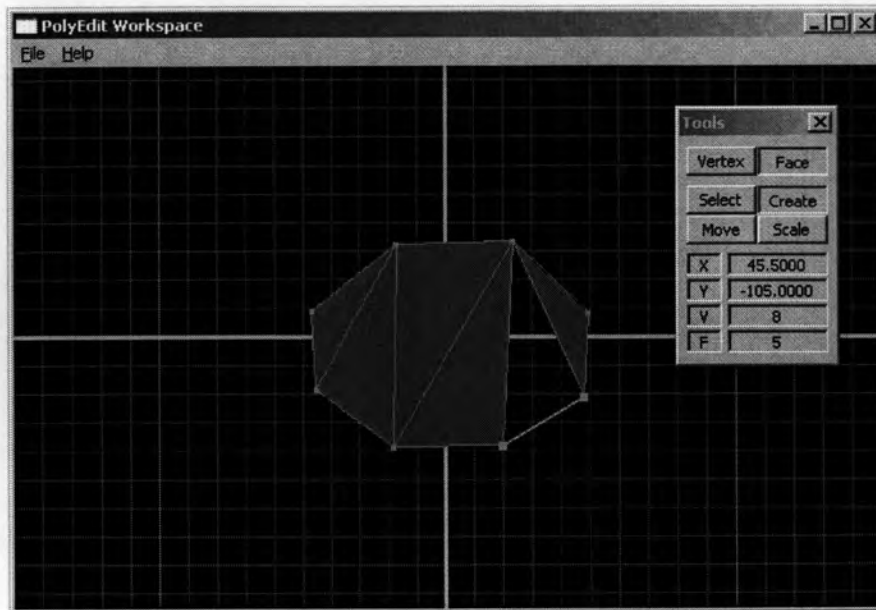
เนื่องจากงานวิจัยในวิทยานิพนธ์นี้ ต้องทำการศึกษาโดยต้องทำการทดสอบผลของสนามแรงรูปแบบต่างๆ ที่มีต่อการจัดวัตถุตั้งแต่เริ่มต้นจนกระทั่งเข้าสู่ภาวะสมดุลที่มีเสถียรภาพ อย่างไรก็ตาม อุปกรณ์ที่สามารถสร้างสนามแรงได้จริงตามทฤษฎีนั้นยังคงเป็นไปได้ยากดังที่ได้นำเสนอมา เพื่อการนี้ จึงจำเป็นต้องพัฒนาโปรแกรมจำลองสถานการณ์สำหรับทดสอบสนามแรงที่โปรแกรมได้แบบต่างๆ เพื่อเป็นเครื่องมือช่วยในการทดสอบ อีกทั้งยังสามารถเปรียบเทียบประสิทธิภาพของสนามแรงแต่ละรูปแบบภายใต้การควบคุมปัจจัยภาวะแวดล้อมต่างๆ เช่น ความหนาแน่นของวัตถุ, สมบัติความเสียดทาน, รูปร่างวัตถุ ตลอดจนคุณสมบัติตอนเริ่มต้นของวัตถุ (ตำแหน่ง, ทิศทาง, ความเร็ว, ความเร็วเชิงมุม) ในการทดสอบกับสนามแรงรูปแบบต่างๆ ให้มีค่าเหมือนๆ กัน เพื่อให้ได้ผลลัพธ์ที่ชัดเจน

เนื้อหาในบทนี้ จะได้นำเสนอถึงการพัฒนาโปรแกรมจำลองสถานการณ์สำหรับการจัดวัตถุภายใต้สนามแรงที่ใช้ในวิทยานิพนธ์นี้ ซึ่งเป็นการพัฒนาเพิ่มเติมมาจากโปรแกรมจำลองสถานการณ์เดิมที่ได้ออกแบบและพัฒนาไว้ใน [16] โดยจะได้อธิบายถึงโครงสร้างโดยรวม รูปแบบข้อมูลที่ใช้ประมวลผล และการพัฒนาส่วนประมวลผลเพิ่มเติม เพื่อการไปใช้งานจริงในการทดสอบในบทถัดไป

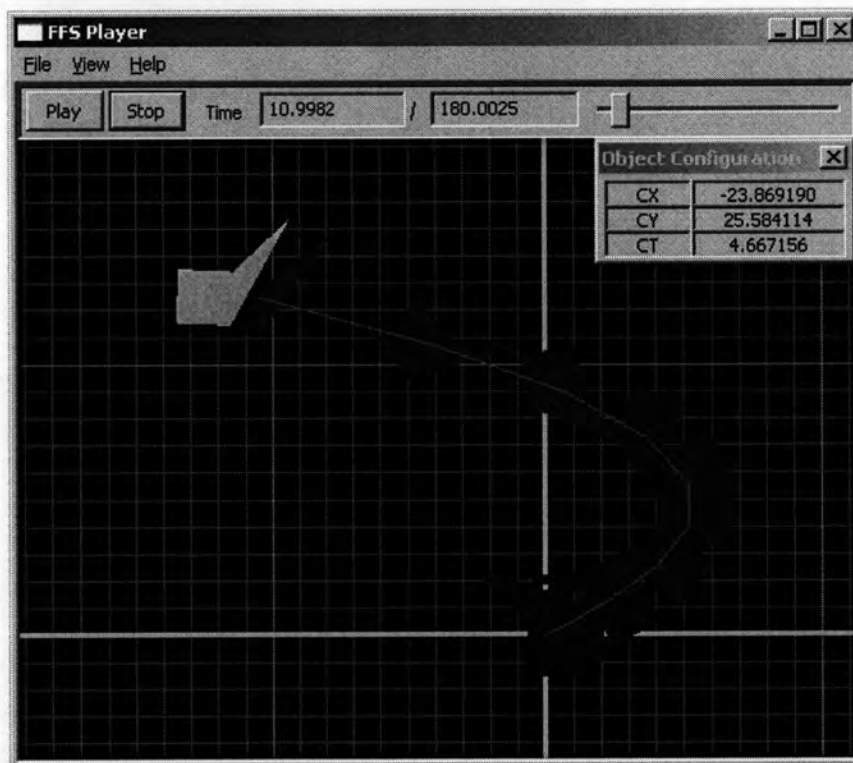
3.1 โครงสร้างโดยรวมของโปรแกรม

ดังที่ได้กล่าวไปข้างต้นแล้วว่า ตัวโปรแกรมจำลองการทำงานของสนามแรงที่จะใช้สำหรับการทดสอบต่างๆ ในวิทยานิพนธ์ฉบับนี้นั้น เป็นการพัฒนาต่อเนื่องมาจากโปรแกรมจำลองสถานการณ์การจัดวัตถุภายใต้สนามแรงที่เคยมีการออกแบบและพัฒนาไว้ใน [16] โดยโครงสร้างของโปรแกรมเดิมนั้น แบ่งออกเป็น 3 ส่วนใหญ่ คือ

- **ส่วนสร้างข้อมูลวัตถุ (Object Editor)** ส่วนนี้จะเป็นส่วนที่ใช้ออกแบบวัตถุโดยใช้การวาดวัตถุผ่าน Graphic User Interface แล้วจึงทำการแปลงรูปวัตถุออกมาเป็นข้อมูลวัตถุที่จะป้อนให้กับส่วนประมวลผล ซึ่งผู้ใช้สามารถปรับเปลี่ยนแก้ไขรูปร่างวัตถุผ่านโปรแกรมส่วนนี้ โดยไม่ต้องยุ่งยากกับการจัดการข้อมูลวัตถุโดยตรง ดังตัวอย่างในรูปที่ 3.1
- **ส่วนประมวลผลการจัดวัตถุ** ส่วนนี้จะทำการรับข้อมูลวัตถุ ข้อมูลสนามแรง และข้อมูลตำแหน่งเริ่มต้นของวัตถุ ตลอดจนช่วงเวลาที่ต้องการจำลองการเคลื่อนที่ ทำการประมวลผลออกมาเป็นตำแหน่งและลักษณะของวัตถุในแต่ละหน่วยย่อยของเวลาตั้งแต่เริ่มต้นจนถึงสิ้นสุดช่วงของเวลาที่ต้องการ ส่วนนี้จะทำงานในลักษณะที่เป็นแบบ Command-Line (คือ ไม่มีการแสดงผลเป็นกราฟิกออกมา) เพื่อให้สะดวกในการทำการสอบหลายๆ การทดลองต่อเนื่องกัน โดยเมื่อประมวลผลเสร็จ ก็จะได้ผลลัพธ์เพื่อนำไปประมวลผล รวมถึงเป็นข้อมูลป้อนให้กับส่วนแสดงผลกราฟิกต่อไป
- **ส่วนแสดงผล (Playback)** เป็นส่วนที่นำผลการคำนวณที่ได้จากส่วนประมวลผล ซึ่งอยู่ในรูปของข้อมูลพิกัดตำแหน่งและลักษณะของวัตถุในแต่ละหน่วยย่อยของเวลา มาแสดงผลออกเป็นภาพกราฟิก 3 มิติ โดยสามารถที่จะแสดงผลแบบ Movie คือสามารถ Play, Pause, Seek ไปยังตำแหน่งต่างๆ หรือ Play ที่ speed ต่างๆ ได้ ดังตัวอย่างในรูปที่ 3.2



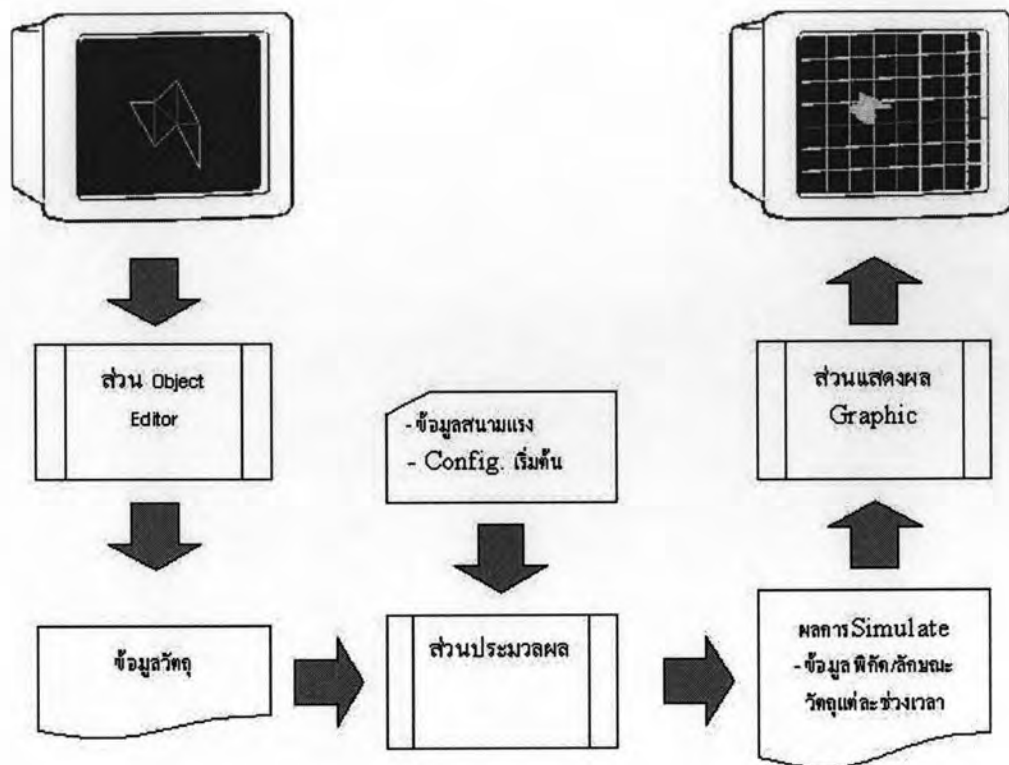
รูปที่ 3.1: ตัวอย่างหน้าจอของโปรแกรมส่วนสร้างข้อมูลวัตถุ



รูปที่ 3.2: ตัวอย่างหน้าจอของโปรแกรมส่วนแสดงผลการจัดวัตถุ

อนึ่ง โครงสร้างโดยรวมของโปรแกรมจำลองสถานการณ์ สามารถแสดงออกมาได้ดังแผนภาพในรูปที่ 3.3

ทั้งนี้ โปรแกรมจำลองที่ใช้ในการทดสอบการจัดวัตถุในวิทยานิพนธ์นี้ ยังคงยึดโครงสร้างการทำงาน ของโปรแกรมโดยรวมไว้ตามเดิม เพียงแต่เพื่อการทำทดลองให้มีประสิทธิภาพ จึงได้ทำการปรับปรุงการทำงานของส่วนประมวลผลขึ้นใหม่ เพื่อให้มีความแม่นยำในการคำนวณ ตลอดจนความเร็วที่ใช้



รูปที่ 3.3: โครงสร้างโดยรวมของโปรแกรมจำลองสถานการณ์

ในการจำลองสถานการณ์แต่ละกรณีศึกษาให้มีความซับซ้อน ดังจะได้อธิบายถึงการปรับปรุงของส่วนประมวลผลในหัวข้อต่อไป

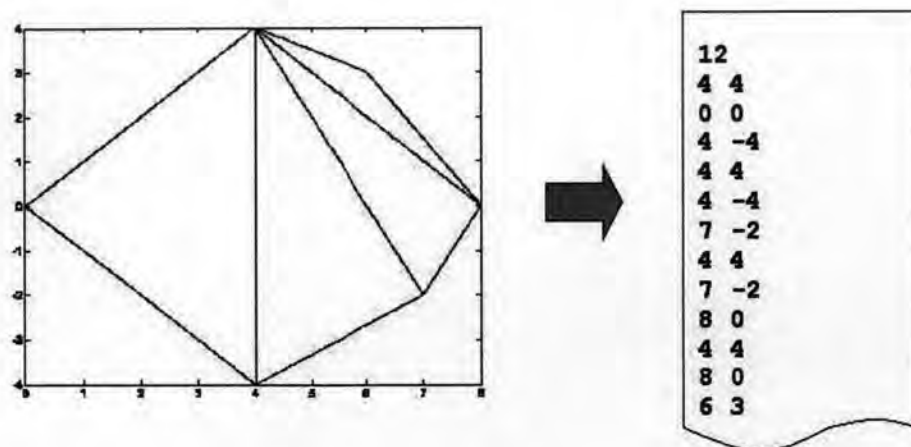
3.2 รูปแบบข้อมูลที่ป้อนให้กับโปรแกรมจำลองสถานการณ์

3.2.1 รูปแบบข้อมูลวัตถุ

ข้อมูลวัตถุในที่นี้ คือ ข้อมูลที่เป็นผลลัพธ์จากส่วน Object Editor ที่จะใช้เป็นข้อมูลป้อนให้กับส่วนประมวลผล เพื่อนำไปใช้ประมวลผล การจัดวัตถุ ร่วมกับข้อมูลสนามแรง และข้อมูลตัวแปรภาวะแวดล้อมต่างๆ โดยข้อมูลของวัตถุจะอยู่ในรูปของไฟล์ข้อความ ที่มีโครงสร้างดังต่อไปนี้ คือ

- **บรรทัดแรก** เป็นเลขจำนวนเต็ม n บอกถึงจำนวนของจุดยอดของสามเหลี่ยมทั้งหมดที่ประกอบกันเป็นตัววัตถุ
- **อีก n บรรทัดถัดมา** แต่ละบรรทัดจะประกอบด้วยจำนวนจริง 2 ค่า คั่นด้วยช่องว่าง ซึ่งแทนค่าพิกัด (x, y) ของแต่ละจุดยอดของสามเหลี่ยม ทั้งนี้ ทุกๆ 3 บรรทัดจะประกอบเป็นสามเหลี่ยม 1 รูป

ตัวอย่างของข้อมูลวัตถุเป็นดังในรูปที่ 3.4 ทั้งนี้ สาเหตุที่ต้องแบ่งวัตถุออกเป็นรูปสามเหลี่ยมย่อยๆ ก็เพื่อให้สะดวกในการทำการคำนวณค่าอินทิเกรตของฟังก์ชันต่างๆ เหนือวัตถุให้มีประสิทธิภาพ (ซึ่งจะได้อธิบายในหัวข้อต่อไป)



รูปที่ 3.4: ตัวอย่างข้อมูลวัตถุ (ซ้าย) รูปวัตถุตัวอย่าง (ขวา) ข้อมูลวัตถุที่ได้

3.2.2 รูปแบบข้อมูลสนามแรง

ในโปรแกรมสถานการณ์ตัวเต็มใน [16] นั้น ได้ออกแบบให้การป้อนข้อมูลสนามแรง อยู่ในรูปของนิพจน์ของ แรงในแนวแกน X และแกน Y โดยมี x, y เป็นตัวแปร ประกอบเข้ากับสัญลักษณ์ทางคณิตศาสตร์อื่นๆ โดยที่ในตัวโปรแกรมส่วนประมวลผล จะมีส่วนที่แปลงนิพจน์เหล่านี้ให้เป็นฟังก์ชันการคำนวณที่คอมพิวเตอร์เข้าใจ ก่อนจะนำไปประมวลผล ทั้งนี้ ข้อดีของวิธีการออกแบบข้อมูลสนามแรงในรูปแบบนี้ เพื่อให้สามารถจำลองสถานการณ์การจัดวัตถุด้วยสนามแรงแบบไหนก็ได้ ขอเพียงให้สามารถเขียนฟังก์ชันของแรงเนื่องจากสนามแรงในแนวแกน X และ Y ได้ ก็สามารถทำการคำนวณได้ ดังตัวอย่างข้อมูลสนามแรงสำหรับ Elliptic Field และ Unit Radial & Constant Field ในรูปที่ 3.5 (ก),(ข) ตามลำดับ

1 $-x \quad ; \quad -2y$	1 $-x/\sqrt{x^2+y^2} \quad ; \quad -y/\sqrt{x^2+y^2} \quad -0.4$
(ก)	(ข)

รูปที่ 3.5: ตัวอย่างข้อมูลสนามแรงสำหรับโปรแกรมรูปแบบเต็ม (ก) Elliptic Field ($\xi = 1, \eta = 2$) (ข) Unit Radial Field ($h = 1$) & Constant Field ศีตทางตามแกน Y ($c = -0.4$)

อย่างไรก็ตาม วิธีการออกแบบดังกล่าว มีข้อเสียตรงที่วิธีการคำนวณที่มีความยืดหยุ่นมากเกินไป ทำให้ประสิทธิภาพในการคำนวณต่ำ ตลอดจนการคำนวณแรงลัทธิและทอร์กลัทธิแต่ละครั้งใช้เวลานาน เนื่องจากต้องใช้วิธีการประมาณการเชิงตัวเลข (รายละเอียดอยู่ในหัวข้อ 3.3) นอกจากนั้น โปรแกรมตัวเต็มยังมีข้อจำกัดที่สามารถป้อนข้อมูลสนามแรงได้เพียงชุดเดียวต่อการทดสอบแต่ละครั้ง โดยที่สนามแรงที่ป้อนเข้าไปดังกล่าว จะทำงานตลอดเวลาที่มีการประมวลผล ไม่สามารถเปลี่ยนเป็นสนามแรงชุดอื่นระหว่างการทดสอบได้

ด้วยเหตุดังกล่าว ในวิทยานิพนธ์จึงได้ทำการปรับปรุง รูปแบบ ข้อมูล สนาม แรง ที่จะป้อนให้กับโปรแกรมส่วนประมวลผลเสียใหม่ควบคู่ไปพร้อมๆ กับการปรับปรุงการทำงานของส่วนประมวลผล โดยแทนที่จะป้อนเป็นฟังก์ชันของสนามแรงใดๆ ก็ได้ นั้น ข้อมูลสนามแรงแบบใหม่จะมีเพียงรหัสบอกรูปแบบของสนามแรง และพารามิเตอร์ของสนามแรงนั้นๆ แทน โดยที่ในตัวส่วนประมวลผลจะมีฟังก์ชันการ

คำนวณที่เกี่ยวข้องกับสนามแรงเหล่านั้นบรรจุไว้ภายในล่วงหน้า นอกจากนี้ ยังสามารถกำหนดสนามแรงให้มีมีการเปลี่ยนชุดสนามแรงระหว่างการจัดวัตถุโดยการกำหนดเวลาทำงานของสนามแรงแต่ละชุดได้อีกด้วย ซึ่งมีรูปแบบดังต่อไปนี้

- **บรรทัดแรก** เป็นเลขจำนวนเต็ม n บอกถึงจำนวนครั้งที่มีการเปลี่ยนสนามแรงระหว่างการจัดวัตถุ
- **ส่วนที่เหลือ** จะถูกแบ่งออกเป็น n ชุดย่อยๆ ซึ่งแต่ละชุดจะมีโครงสร้างเป็น
 - **บรรทัดแรกของชุด** ประกอบด้วยจำนวนเต็ม m บอกถึงจำนวนรูปแบบสนามแรงในชุดนั้นๆ ตามด้วยจำนวนจริง 2 ตัว บอกเวลาเริ่มต้นและเวลาสิ้นสุดในการใช้สนามแรงชุดนั้น
 - **อีก m บรรทัดที่เหลือในชุด** จะเป็นข้อมูลสนามแรงพื้นฐานแต่ละสนามแรงในช่วงเวลานั้นๆ ซึ่งมีรูปแบบดังที่ปรากฏในตาราง 3.1

ตารางที่ 3.1: ตัวอย่าง Format ข้อมูลสนามแรงพื้นฐานที่ใช้ในไฟล์ข้อมูลสนามแรง

สนามแรง	รูปแบบข้อมูล	คำอธิบาย
Elliptic Field	$E < \xi > < \eta > < x_c > < y_c > < \theta >$	(x_c, y_c) คือจุดศูนย์กลางสนามแรง θ คือมุมที่แกนระนาบสนามแรงทำกับแกนระนาบของสิ่งแวดล้อม หน่วยเป็นองศา
Unit Radial Field	$U < h > < x_c > < y_c >$	(x_c, y_c) คือจุดศูนย์กลางสนามแรง
Constnat Field	$C < c > < \theta >$	θ คือมุมที่แกนระนาบสนามแรงทำกับแกนระนาบของสิ่งแวดล้อม หน่วยเป็นองศา

ตัวอย่างข้อมูลสนามแรงแบบใหม่ สำหรับ Elliptic Field และ Unit Radial & Constant Field ดังตัวอย่างในรูปที่ 3.5 (ก),(ข) โดยให้ทำงานชุดละ 100 วินาที ตามลำดับ สามารถเขียนได้ดังตัวอย่างในรูปที่ 3.6

```

2
1 0.0 100.0
E 1.0 2.0 0 0 0
2 100.0 200.0
U 1.0 0 0
C 0.4 270

```

รูปที่ 3.6: ตัวอย่างข้อมูลสนามแรงสำหรับโปรแกรมรูปแบบใหม่ ซึ่งประกอบด้วย Elliptic Field ($\xi = 1, \eta = 2$) และ Unit Radial Field ($h = 1$) & Constant Field ทิศทางตามแกน Y ($c = -0.4$) ทำงานชุดละ 100 วินาที ตามลำดับ

3.2.3 รูปแบบข้อมูลสภาพแวดล้อมอื่นๆ

นอกเหนือจากข้อมูลวัตถุและข้อมูลของสนามแรงที่ใช้แล้ว ยังมีข้อมูลสภาพแวดล้อมอื่นๆ ที่จำเป็นต่อการจำลองสถานการณ์ เช่น ความหนาแน่นของวัตถุ, สัมประสิทธิ์ความเสียดทาน, configuration เริ่มต้น

ของวัตถุ ฯลฯ ซึ่งข้อมูลเหล่านี้ จะถูกป้อนให้กับโปรแกรมส่วนประมวลผลในรูปแบบของ command-line parameter ดังมีรายละเอียดในตาราง 3.2

ตารางที่ 3.2: รูปแบบ Command-Line สำหรับข้อมูลสภาพแวดล้อมอื่นๆ ในการจำลองสถานการณ์

Command	คำอธิบาย
$-d < density >$	ความหนาแน่นต่อพื้นที่ของวัตถุ
$-k < coefficient >$	ค่าสัมประสิทธิ์ความเสียดทานแบบ Viscous
$-p < x > < y >$	ตำแหน่งของศูนย์กลางมวลของวัตถุ ณ ตอนเริ่มต้น (x, y)
$-v < v_x > < v_y >$	ความเร็วของวัตถุ ณ ตอนเริ่มต้น [ปกติเป็น $(0, 0)$]
$-o < \theta >$	ทิศทางของระนาบวัตถุเทียบกับระนาบของพื้นที่สนามแรง ณ ตอนเริ่มต้น
$-w < \omega >$	ความเร็วเชิงมุมรอบศูนย์กลางมวลของวัตถุ ณ ตอนเริ่มต้น [ปกติจะเป็น 0]

3.3 การคำนวณเชิงพื้นที่ : การอินทิเกรตเหนือพื้นที่วัตถุ

การคำนวณค่าต่างๆ ที่เกี่ยวข้องกับบริเวณที่มีวัตถุอยู่นั้น จะอยู่ในรูปของการอินทิเกรตเหนือพื้นที่วัตถุทั้งสิ้น ไม่ว่าจะเป็นการหาพื้นที่, Moment of Inertia, แรงลัพท์-ทอร์กลัพท์เนื่องจากสนามแรง หรือแม้กระทั่ง Lifted Potential Energy ของวัตถุ

ทั้งนี้ เรืองจากวัตถุที่เราทำการวิจัย มีลักษณะเป็นรูปของสามเหลี่ยม n รูป ประกอบกับขึ้นเป็นวัตถุ ดังนั้น การอินทิเกรตฟังก์ชันใดๆ เหนือวัตถุ จึงมีค่าเทียบเท่ากับผลรวมของการอินทิเกรตฟังก์ชันนั้นเหนือพื้นที่สามเหลี่ยมย่อยๆ เหล่านั้น กล่าวคือ

$$\int_P \int G(x, y) dx dy = \sum_{i=1}^n \int_{T_i} \int G(x, y) dx dy$$

รูปแบบของข้อมูลวัตถุที่กล่าวมาแล้วในหัวข้อ 3.2.1 ก็ได้แนวคิดในการออกแบบมาจากวิธีการอินทิเกรตเหนือรูปสามเหลี่ยมทีละรูปนี้ เพื่อให้ง่ายต่อการประมวลผลของโปรแกรม

ในส่วนของการอินทิเกรตฟังก์ชันเหนือสามเหลี่ยมแต่ละรูปนั้น ในโปรแกรมจำลองสถานการณ์ตัวเดิม [16] นั้น ได้เลือกใช้ Extended Simpson Rule ในการประมาณค่าทั้งในแนวแกน X และแกน Y ซึ่งมีขั้นตอนดังต่อไปนี้ คือ

- สำหรับการอินทิเกรตเหนือสามเหลี่ยม T_i ใดๆ ถ้าให้

$$\begin{aligned} \int_{T_i} \int G(x, y) dx dy &= \int_{y_{\min}(T_i)}^{y_{\max}(T_i)} \left[\int_{x_{\min}(y)}^{x_{\max}(y)} G(x, y) dx \right] dy \\ &= \int_{y_{\min}(T_i)}^{y_{\max}(T_i)} G^*(y) dy \end{aligned} \quad (3.1)$$

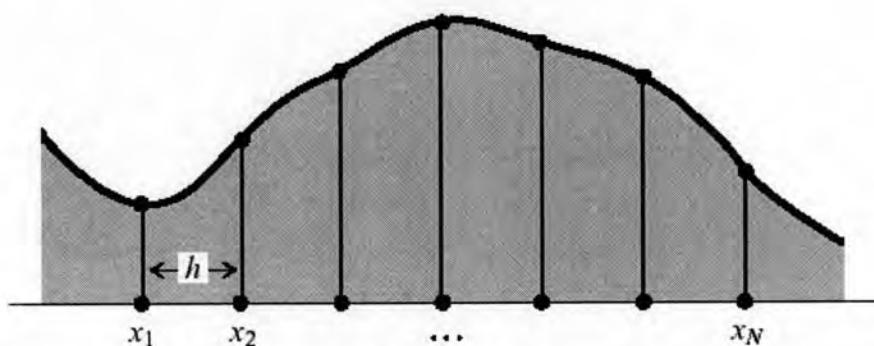
โดยที่ $G^*(y)$ คือผลการอินทิเกรตตามแนวแกน X เฉพาะส่วนที่อยู่ในสามเหลี่ยม T_i ณ ค่า y ใดๆ

- ในการหาค่า $G^*(y)$ ใดๆ นั้น กระทำได้โดยการแบ่งช่วง $(x_{\min}(y), x_{\max}(y))$ ออกเป็นส่วนๆ โดยให้แต่ละส่วนมีขนาด h เท่าๆ กัน ดังตัวอย่างในรูปที่ 3.7 สมมติให้มี $2m + 1$ ช่วง จากนั้นใช้ Extended

Simpson Rule ในการประมาณค่าของ $G^*(y)$ ซึ่งจะได้ผลเป็น

$$\begin{aligned} G^*(y) &= \int_{x_{\min}(y)}^{x_{\max}(y)} G(x, y) dx \\ &= \frac{h}{3} (G(x_{\min}(y), y) + G(x_{\max}(y), y)) + \frac{2h}{3} \sum_{k=1}^{m-1} G(x_{\min}(y) + 2kh, y) \\ &\quad + \frac{4h}{3} \sum_{k=1}^{m-1} G(x_{\min}(y) + (2k-1)h, y) \end{aligned} \quad (3.2)$$

ทั้งนี้ ค่าความคลาดเคลื่อนจากการประมาณค่าดังกล่าว จะมีค่าเป็น ซึ่งถ้ามีค่ามากเกินกว่าขอบเขตที่ยอมรับได้ ก็จะทำให้การแบ่งส่วนให้ h มีค่าน้อยลง แล้วประมาณค่าใหม่ จนกว่าค่าความคลาดเคลื่อนจะอยู่ในขอบเขตที่ยอมรับได้



รูปที่ 3.7: ตัวอย่างของการหา $G^*(y)$ ณ ค่า y ใดในการอินทิเกรตฟังก์ชัน $G(x, y)$ เหนือสามเหลี่ยม T_i ใดๆ

- ในส่วนของการอินทิเกรตตามสมการ (3.1) ก็ทำการแบ่งช่วง $(y_{\min}(T_i), y_{\max}(T_i))$ เป็นส่วนๆ แล้วใช้ Extended Simpson Rule เช่นเดียวกัน ซึ่งจะได้เป็น

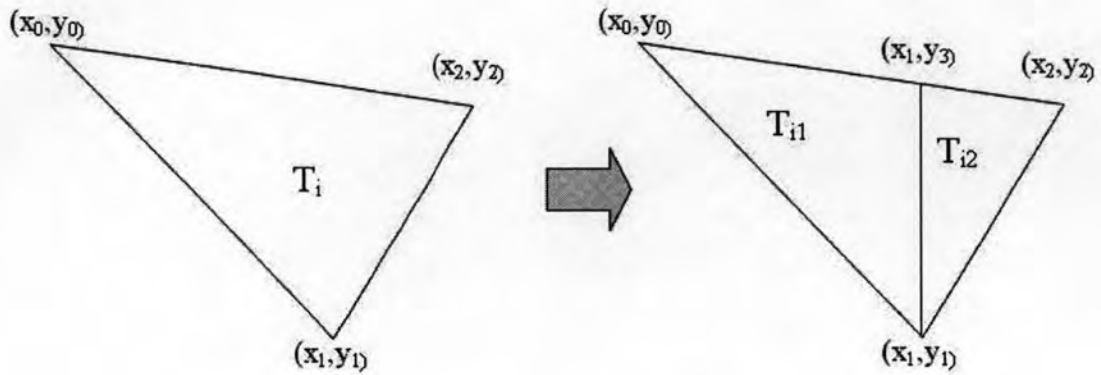
$$\int_{y_{\min}(T_i)}^{y_{\max}(T_i)} G^*(y) dy = \frac{h}{3} (G^*(y_{\min}) + G^*(y_{\max})) + \frac{2h}{3} \sum_{k=1}^{m-1} G^*(y_{\min} + 2kh) + \frac{4h}{3} \sum_{k=1}^{m-1} G^*(y_{\min} + (2k-1)h)$$

ในทำนองเดียวกัน ถ้าค่าความคลาดเคลื่อนจากการประมาณค่าเกินกว่าขอบเขตที่ยอมรับได้ ก็จะทำให้การแบ่งส่วนให้ถี่ขึ้น (h มีค่าน้อยลง) แล้วประมาณค่าใหม่เช่นเดียวกัน

ด้วยวิธีดังกล่าวนี้ มีข้อดีตรงที่ง่ายต่อการอินทิเกรตฟังก์ชันใดๆ เนื่องจากสามารถใช้ตัวฟังก์ชัน $G(x, y)$ นั้นๆ ในการประมาณค่าได้โดยตรง อย่างไรก็ตาม วิธีดังกล่าวนี้มีข้อด้อยในเรื่องของค่าความคลาดเคลื่อนจากวิธีการคำนวณดังกล่าว รวมไปถึงเวลาที่เสียไปกับการแบ่งส่วนของสี่เหลี่ยมคางหมูเพื่อให้ระดับความคลาดเคลื่อนอยู่ในระดับที่ยอมรับได้ ซึ่งทำให้การอินทิเกรตแต่ละครั้งต้องใช้เวลาไปกับการทำงานหลายขั้นตอนดังกล่าว

ดังนั้น ในวิทยานิพนธ์นี้ จึงได้ทำการเปลี่ยนแปลงวิธีการคำนวณค่าอินทิเกรตเหนือพื้นที่วัตถุเสียใหม่ เพื่อให้มีความแม่นยำและความเร็วมากขึ้น โดยมีขั้นตอนดังต่อไปนี้

- สำหรับสามเหลี่ยม T_i ใดๆ เราทำการแบ่งออกเป็น สามเหลี่ยม 2 รูป โดยให้ด้านที่เป็นด้านร่วมระหว่างสามเหลี่ยมสองรูปนั้น อยู่ในแนวขนานกับแกน Y ของระนาบสนามแรง ดังตัวอย่างในรูป 3.8

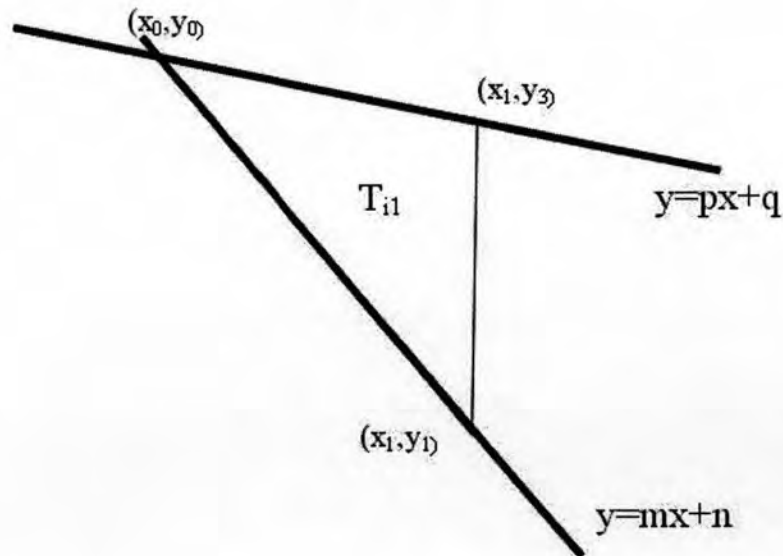


รูปที่ 3.8: ตัวอย่างการแบ่งสามเหลี่ยมเป็น 2 ส่วน

ซึ่งจะทำให้

$$\int_{T_i} \int G(x, y) dx dy = \int_{T_{i1}} \int G(x, y) dx dy + \int_{T_{i2}} \int G(x, y) dx dy$$

- ในสามเหลี่ยมแต่ละรูป เขียนด้านอีก 2 ด้านที่เหลือให้อยู่ในรูปของ $y = f(x)$ โดยที่ $f(x)$ เป็นฟังก์ชันเชิงเส้น ดังตัวอย่างของสามเหลี่ยม T_{i1} ในรูป 3.9



รูปที่ 3.9: ตัวอย่างการหาสมการกำกับด้านของสามเหลี่ยมที่ไม่ใช่ด้านที่ถูกแบ่ง

- ในกรณีดังกล่าวนี้ ถ้าพิจารณาการอินทิเกรตเหนือสามเหลี่ยม T_{i1} ก็สามารเขียนได้เป็น

$$\begin{aligned}
 \int_{T_{i1}} \int G(x, y) dx dy &= \int_{x_0}^{x_1} \left[\int_{mx+n}^{px+q} G(x, y) dy \right] dx \\
 &= \int_{x_0}^{x_1} [G^*(x, y)|_{mx+n}^{px+q}] dx \\
 &= \int_{x_0}^{x_1} H(x) dx \\
 &= H^*(x)|_{x_0}^{x_1} \tag{3.3}
 \end{aligned}$$

ซึ่งถ้าเราสามารถหา $H^*(x)$ สำหรับ $G(x, y)$ ใดๆ ได้ เราก็สามารถแทนที่การอินทิเกรตเหนือสามเหลี่ยม T_i ด้วยวิธี Simpson Quadrature ของเดิม ด้วยการอินทิเกรตเหนือสามเหลี่ยม T_{i1}, T_{i2} ด้วยสมการ (3.3) เพียงขั้นตอนเดียว

ด้วยวิธีการใหม่ที่น่าเสนอนี้ จะเห็นว่าการอินทิเกรตเหนือวัตถุทำได้แม่นยำและรวดเร็วกว่าวิธีการเดิมเป็นอย่างมาก เพียงแต่ความยุ่งยากในการเขียนโปรแกรมจะเพิ่มมากขึ้น เนื่องจากเราจำเป็นต้องเตรียม $H^*(x)$ สำหรับ $G(x, y)$ ใดๆ ไว้ล่วงหน้า เช่นสำหรับสนามแรงหนึ่งๆ จะต้องหา $H^*(x)$ สำหรับทั้งแรงลัพธ์, ทอร์กลัพธ์ ตลอดจน Lifted Potential Energy เตรียมไว้เพื่อบรรจุลงในตัวโปรแกรม ดังตัวอย่างในตาราง 3.3 เป็นต้น ทำให้ตัวโปรแกรมสูญเสียคุณสมบัติความยืดหยุ่นในการจำลองสถานการณ์สำหรับสนามแรงใดๆ ก็ได้ (เนื่องจากการป้อน $G(x, y)$ จากภายนอกโปรแกรม) ไป แต่ทั้งนี้ ภายใต้การทำการทดสอบกับสนามแรงที่มีรูปแบบจำกัด ก็สามารถเตรียม $H^*(x)$ สำหรับสนามแรงทุกรูปแบบที่จะทำการทดสอบเอาไว้ล่วงหน้าทั้งหมดได้ และเมื่อต้องการทำการทดสอบสนามแรงรูปแบบใหม่ ก็เพียงแต่หา $H^*(x)$ ที่เกี่ยวข้องกับสนามแรงนั้นๆ เพิ่มลงไปโปรแกรมได้เสมอ

3.4 การคำนวณเชิงเวลา : การประมาณการเคลื่อนที่และการหมุนของวัตถุ

จากที่ทราบอยู่แล้วว่า ตลอดเวลาที่วัตถุอยู่ภายใต้สนามแรง แรงแลัพท์และทอร์กลัพธ์กระทำต่อวัตถุ จะส่งผลให้วัตถุเคลื่อนที่ และ/หรือ หมุนไป ซึ่งก็หมายถึงการเปลี่ยนแปลงของ ตำแหน่ง, ทิศทาง, ความเร็ว และความเร็วเชิงมุมของวัตถุที่กำหนดให้

- $\mathbf{p}(t) = (x(t), y(t))$ คือตำแหน่งของศูนย์กลางมวลของวัตถุที่เวลา t
- $\theta(t)$ คือทิศทางของแกนระนาบของวัตถุ อ้างอิง จากแกนระนาบของสนามแรงที่เวลา t
- $\mathbf{v}(t) = (v_x(t), v_y(t))$ คือความเร็วของวัตถุที่เวลา t
- $\omega(t)$ คือความเร็วเชิงมุมรอบศูนย์กลางมวลของวัตถุที่เวลา t

ตารางที่ 3.3: ตัวอย่างของ $H^*(x)$ สำหรับ $G(x, y)$ บางส่วนที่ใช้ในการจำลองสถานการณ์

ผลลัพธ์ที่ต้องการ	$G(x, y)$	$H^*(x)$
พื้นที่วัตถุ	1	$\frac{\pi}{2}(-2n + 2q + (p - m)x)$
Moment of Inertia	$\frac{\rho}{2}((x - x_c)^2 + (y - y_c)^2)$	$\frac{\rho x}{12}((-4(n - q)(3x_c^2 + 3y_c^2 + n^2 + nq + q^2 - 3y_c(n + q)) - 6(mn^2 + x_c^2(m - p) + y_c^2(m - p) - 2x_c(n - q) - pq^2 + y_c(-2mn + 2pq))x + 4((-1 - m^2)n + 2x_c(m - p) + y_c(m - p)(m + p) + (1 + p^2)q)x^2 - (3m + m^3 - p(3 + p^2))x^3)$
Constant Field : F_x	$c \cos(\theta_c)$	$\frac{cx}{2}(-2n + 2q + (p - m)x) \cos(\theta_c)$
Constant Field : F_y	$c \sin(\theta_c)$	$\frac{cx}{2}(-2n + 2q + (p - m)x) \sin(\theta_c)$
Constant Field ทอร์กลัพธ์	$\begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} \times \begin{bmatrix} c \cos(\theta_c) \\ c \sin(\theta_c) \end{bmatrix}$	$\frac{cx}{6}(((m - p)(m + p)x^2 + 3(mn + y_c(p - m) - pq)x + 3(n - q)(-2y_c + n + q)) \cos(\theta_c) + (3x_c(2n - 2q + mx - px) + x(-3n + 3q - 2mx + 2px)) \sin(\theta_c))$
Constant Field L.P.E.	$c(x \cos(\theta_c) + y \sin(\theta_c))$	$\frac{cx}{6}(x(-3n + 3q - 2mx + 2px) \cos(\theta_c) + (-3n^2 - 3m^2x + 3q^2 + (p^2 - m^2)x^2 + 3pqx) \sin(\theta_c))$

ซึ่งความสัมพันธ์ของค่าต่างๆ ดังกล่าวข้างต้น สามารถอธิบายตามหลักกลศาสตร์ได้ด้วยสมการเชิงอนุพันธ์ 4 สมการ คือ

$$\frac{dp(t)}{dt} = v(t) \quad (3.4)$$

$$\frac{dv(t)}{dt} = a(t) = \frac{\mathbf{F}(t) + \mathbf{F}_d(t)}{\rho A} \quad (3.5)$$

$$\frac{d\theta(t)}{dt} = \omega(t) \quad (3.6)$$

$$\frac{d\omega(t)}{dt} = \alpha(t) = \frac{\mathbf{T}(t) + \mathbf{T}_d(t)}{I_z} \quad (3.7)$$

โดยที่ค่าของ $\mathbf{F}(t)$, $\mathbf{T}(t)$ หรือก็คือ แรงลัพธ์/ทอร์กลัพธ์ เนื่องจากสนามแรงที่เวลา t นั้น สามารถหาได้จากวิธีการที่ได้นำเสนอไปในหัวข้อ 3.3 เมื่อวัตถุอยู่ที่ configuration $(x(t), y(t), \theta(t))$ ส่วน $\mathbf{F}_d(t)$, $\mathbf{T}_d(t)$ หรือแรงและทอร์กเนื่องจากแรงเสียดทาน ก็สามารถหาได้จากสมการ (2.10), (2.11)

จากที่เวลา $t = 0$ ซึ่งเราสามารถกำหนด $p(0), v(0), \theta(0), \omega(0)$ เป็นเงื่อนไขเริ่มต้นของวัตถุนั้น เป้าหมายที่ต้องการคือการหาค่าของ $p(t), v(t), \theta(t), \omega(t)$ ที่เวลา t ใดๆ นั่นคือ เราต้องหาค่าผลเฉลยของสมการ (3.4)-(3.7) ซึ่งเป็นที่แน่นอนอยู่แล้วว่าเราไม่สามารถหาได้โดยตรง ดังนั้น จึงต้องหาค่าผลเฉลยด้วยระเบียบวิธีเชิงตัวเลขแทน

ในโปรแกรมจำลองสถานการณ์ที่สร้างขึ้นนี้ ได้เลือกใช้วิธี Runge-Kutta-Merson ในการหาค่าผลเฉลยของทั้ง 4 สมการ ซึ่งเป็นวิธีที่นิยมในการหาค่าผลเฉลยของสมการเชิงอนุพันธ์โดยทั่วไป ทั้งนี้ โดยผสมผสานเข้ากับการปรับขนาดช่วงระยะ iteration ในการประมาณค่าแต่ละครั้งเพื่อควบคุมระดับของความคลาดเคลื่อนให้อยู่ในระดับที่ยอมรับได้ตลอดเวลา ซึ่งกระบวนการประมาณค่าแต่ละครั้ง สามารถอธิบายได้ด้วย Algorithm 2

Algorithm 2 $RKM(x(t_n), f(x, t), h, \epsilon)$

Require: $x(t_n)$: base value of $x(t)$ at n^{th} iteration;

Require: $f(x, t)$: ODE Function of $\dot{x}(t)$;

Require: h : iteration stepsize;

Require: ϵ : acceptable error threshold;

```

1:
2: {ODE :  $\dot{x}(t) = f(x, t)$ }
3: repeat
4:    $k_1 = hf(x(t_n), t_n)$ ;
5:    $k_2 = hf(x(t_n) + \frac{k_1}{3}, t_n + \frac{h}{3})$ ;
6:    $k_3 = hf(x(t_n) + \frac{k_1+k_2}{6}, t_n + \frac{h}{3})$ ;
7:    $k_4 = hf(x(t_n) + \frac{k_1+3k_3}{8}, t_n + \frac{h}{2})$ ;
8:    $x_e = x(t_n) + \frac{k_1-3k_3+4k_4}{2}$ ;
9:    $k_5 = hf(x_e, t_n + h)$ ;
10:   $x(t_n + h) = x(t_n) + \frac{k_1+4k_4+k_5}{6}$ ;
11:   $Err = \frac{x_e - x(t_n+h)}{5}$ ;
12:  if  $|Err| \geq \epsilon$  then
13:     $newH = \frac{h}{2}$ ; {Decrease iteration stepsize}
14:     $h = newH$ ;
15:  end if
16: until  $|Err| < \epsilon$ 
17: { Increase "next" stepsize if  $Err < \frac{\epsilon}{64}$  }
18: if  $|Err| < \frac{\epsilon}{64}$  then
19:    $nextH = 2h$ ;
20: end if
21:
22: Return  $[x(t_n + h), nextH, Err]$ ;

```

ทั้งนี้ ค่าที่ได้คืนมาจาก Algorithm 2 นั้น ประกอบไปด้วย

- $x(t_n + h)$: ผลลัพธ์ของการประมาณค่า
- $nextH$: ขนาดของช่วงระยะห่างในการประมาณค่า (iteration step) ที่เหมาะสมสำหรับการประมาณค่าครั้งถัดไป
- Err : ค่าความคลาดเคลื่อนของการประมาณค่าครั้งปัจจุบัน

ซึ่งโดยการประมวลผลเริ่มจากที่เวลา $t = 0$ ไปเรื่อยๆ ผลที่ได้ก็คือผลลัพธ์ที่สนามแรงกระทำต่อวัตถุ นั่นเอง ทั้งนี้ เวลาที่ใช้ในการประมวลผล ตลอดจนช่วงเวลาสำหรับแต่ละ iteration ตอนเริ่มต้น รวมถึงค่า

ระดับความคลาดเคลื่อนที่ยอมรับได้นั้น จะถูกกำหนดด้วย Command-Line Parameter ซึ่งมีรายละเอียดดังในตาราง 3.4 ทั้งนี้ โปรแกรมจะประมวลผลไปเรื่อยๆ จนกระทั่งถึงเวลาสิ้นสุด หรือจนกว่าวัตถุหยุดนิ่ง และเข้าสู่สมดุลย์ตามเงื่อนไขในสมการ (2.14)-(2.17)

ตารางที่ 3.4: รูปแบบ Command-Line สำหรับพารามิเตอร์ควบคุมการคำนวณเชิงเวลา

Command	คำอธิบาย
$-ti < time >$	เวลาเริ่มต้น
$-tl < time >$	เวลาสิ้นสุด
$-ts < time >$	ช่วงเวลาสำหรับแต่ละ iteration ตอนเริ่มต้น
$-to < time >$	ช่วงเวลาสำหรับเก็บข้อมูลโดยละเอียด (ดูหัวข้อ 3.5)

3.5 รูปแบบของผลลัพธ์ที่ได้จากส่วนประมวลผล

ผลลัพธ์จากส่วนประมวลผลในโปรแกรมเดิมนั้น ใช้เป็นเพียงข้อมูลป้อนให้กับส่วนแสดงผล (Playback) เพียงเท่านั้น ซึ่งจะประกอบไปด้วย

- ส่วนแรก จะเป็นข้อมูลของวัตถุ ซึ่งมีลักษณะเดียวกับข้อมูลวัตถุในหัวข้อ 3.2.1
- ส่วนถัดมา จะเป็นข้อมูล configuration ของวัตถุที่เวลาต่างๆ โดยในแต่ละบรรทัดจะประกอบไปด้วย

$$\langle t \rangle \langle x(t) \rangle \langle y(t) \rangle \langle \theta(t) \rangle$$

ซึ่งจะบันทึกทุกๆ Iteration ของการคำนวณเชิงเวลา โดยบรรทัดสุดท้ายจะจบลงด้วย $t = -1$

- ส่วนสุดท้าย เป็น ข้อมูล ของ ตัวแปรสภาวะแวดล้อม และ ตัว พารามิเตอร์อื่นๆ ทำ กำหนด ให้ แก่ โปรแกรมส่วนประมวลผล

ซึ่งทั้งหมดจะถูกบันทึกลงในไฟล์ข้อความธรรมดา ดังตัวอย่างผลลัพธ์ในรูป 3.10 เป็นต้น

อย่างไรก็ตาม ในการปรับปรุงโปรแกรมขึ้นใหม่นั้น เพื่อการวิเคราะห์พฤติกรรมของวัตถุภายใต้สนามแรงโดยละเอียดยิ่งขึ้น จึงได้เพิ่มการบันทึกผลลัพธ์ลงในอีกไฟล์หนึ่งต่างหากจากไฟล์ผลลัพธ์รูปแบบเดิม ซึ่งข้อมูลแต่ละบรรทัดจะมีรูปแบบเป็น

$$\langle t \rangle \langle x(t) \rangle \langle y(t) \rangle \langle \theta(t) \rangle \langle v_x(t) \rangle \langle v_y(t) \rangle \langle \omega(t) \rangle \langle F_x(t) \rangle \langle F_y(t) \rangle \langle T(t) \rangle \langle U(t) \rangle$$

ทั้งนี้ ข้อมูลผลลัพธ์โดยละเอียดดังกล่าว ไม่ได้บันทึกในทุก Iteration เหมือนผลลัพธ์แบบเดิมเนื่องจากปริมาณข้อมูลจะมีมากเกินไป โดยจะบันทึกทุกๆ ช่วงที่เวลาผ่านไป มากกว่าเวลาที่กำหนดไว้ด้วยพารามิเตอร์ $-to$ ที่กล่าวถึงในหัวข้อที่ผ่านมา

ตัวอย่างผลลัพธ์โดยละเอียดที่สร้างขึ้นใหม่นี้ เป็นดังรูปที่ 3.11

```

sampleOutput.txt - Notepad
File Edit Format View Help
3
-5.001825 -3.333236
4.998175 -3.333236
-0.001825 6.666764
0.000000 40.000000 60.000000 1.047198
0.200000 40.000369 60.014966 1.047188
0.400000 40.001177 60.047690 1.047169
0.600000 40.002457 60.099602 1.047137
0.800000 40.004247 60.172243 1.047093
1.000000 40.006567 60.267279 1.047035
1.200000 40.009522 60.386511 1.046963
1.400000 40.013096 60.531863 1.046874
1.600000 40.017360 60.705502 1.046768
1.800000 40.022368 60.909639 1.046643
2.000000 40.028178 61.146749 1.046497
-1
Force Expressions: 1
XD | 0.05*(-5/sqrt((x-30)^2+(y-20)^2+0.1)-(5+5)*(x-30)-2.3)
YO | 0.05*(-5/sqrt((x-30)^2+(y-20)^2+0.1)-(5+5)*(y-20))
simulation output file name [orient5_62d.fff]
force expression file name [in10d.exp]
polygon triangle file name [in3.ply]
initial position [6.000000 -14.000000]
initial velocity [0.000000 0.000000]
initial orientation [1.540000]
initial angular velocity [0.000000]
simulation duration [300.000000]
initial simulation timestep [0.640000]
integration tolerance [0.000001]
integration maximum iteration [12]
RK maximum error [0.500000]
RK minimum error [0.050000]
zero threshold [0.000001]
rigid body density [1.000000]
friction coefficient [0.500000]
discrete force field [1]
    
```

รูปที่ 3.10: ตัวอย่างผลลัพธ์จากโปรแกรมจำลองสถานการณ์ตัวเต็ม

t	Px	Py	0	Vx	Vy	W	Fx	Fy	T	LPE
1.000000	-25.485652	-7.104624	3.144568	31.800489	37.050713	0.068610	10981.200071	6127.738228	2318.621694	180209.749821
2.000000	8.239760	20.726066	3.234368	30.056271	10.444809	0.102800	-3579.370220	-17876.231702	1614.478532	221552.355203
3.000000	26.717140	11.108056	3.344301	4.926031	-24.501980	0.109319	-11521.766587	-9480.697905	684.214870	228453.233494
4.000000	19.138407	-12.222046	3.444106	-17.634924	-14.649742	0.086768	-8253.437892	10541.515004	-190.107626	164700.975045
5.000000	-1.679818	-11.678795	3.512933	-20.338553	14.020609	0.049564	724.421512	10070.373467	-790.744692	80794.772279
6.000000	-15.768004	5.761905	3.542628	-6.184946	0.010438	0.010438	6799.951814	-4969.642961	-1045.932164	89291.932724
7.000000	-19.678638	10.274641	3.526436	9.214333	-6.211400	-0.020959	5898.312605	-8961.977462	-992.593162	107228.027813
8.000000	-1.164067	-1.371157	3.505088	13.302104	-12.720927	-0.038368	845.254017	1182.623325	-725.829216	12486.016935
9.000000	8.396861	-7.398435	3.462397	5.783119	-1.014307	-0.043803	-3854.021162	6898.650389	-350.543945	66118.283446
10.000000	9.394653	-1.239756	3.421333	-4.400216	9.710088	-0.036756	-4051.443909	1069.289225	9.986007	40994.438079
11.000000	2.270603	5.576877	3.391181	-9.433893	1.986806	-0.028937	-379.197684	-4809.194072	274.803992	35024.372911
12.000000	-4.810882	2.430841	3.376245	-4.722182	-6.632255	-0.007128	2074.680209	-2148.350647	405.676914	28976.123149
13.000000	-8.235185	-3.428267	3.376110	1.778885	-3.342459	0.006151	2489.923591	2355.865173	406.864588	34761.837793
14.000000	-2.337081	-2.812088	3.388970	5.181873	3.971549	0.014598	1007.866079	2425.426027	311.756296	23894.173977
15.000000	2.407327	1.746833	3.403413	3.586365	3.896169	0.017378	-1038.189742	-1806.643710	167.443766	23867.769100
16.000078	4.009671	2.875745	3.420105	-0.444972	-1.928497	0.018325	-1729.170785	-2221.580181	20.775593	27629.489866
17.000078	2.011202	-0.567974	3.439300	-3.077911	-3.207756	0.010177	-867.330847	489.877569	-92.811653	20212.447941
18.000078	-1.067049	-2.068031	3.440105	-2.682477	8.524132	0.003954	860.165065	1783.387612	-154.966776	23392.618229
19.000078	-2.499824	-0.162503	3.441172	-0.164635	8.524249	-0.001691	1078.052893	140.159032	-164.336290	22661.026218
20.000078	-1.876385	1.486828	3.437338	1.757078	0.235334	-0.005323	679.384842	-1282.130772	-132.392992	22789.755629
21.000078	0.360981	0.839588	3.431275	1.781698	-1.778837	-0.006828	-185.673038	-465.394663	-77.386999	21454.648743
22.000078	1.509732	-0.949122	3.424552	0.387727	-0.741917	-0.006327	-650.640755	818.617547	-18.308239	22179.974096
23.000703	0.489627	-0.866757	3.419076	-0.966523	1.109538	-0.004466	-500.211287	875.077552	29.821092	21782.510085
24.000703	-0.020216	0.515624	3.411828	-1.151948	0.861218	-0.002016	9.718028	-444.725143	58.390073	21415.587417
25.000703	-0.878303	0.639460	3.414930	-0.419259	-0.579336	0.000269	378.865970	-551.551281	65.731220	21643.671095
26.000703	-0.006703	-0.201761	3.416142	0.486219	-0.801545	0.001898	352.117231	174.019058	55.608920	21462.034422
27.000703	-0.121860	-0.530761	3.418493	0.770844	0.203272	0.002656	52.552234	487.781677	34.944412	21426.407724
28.000703	0.489627	-0.000160	3.421176	0.369315	0.481281	0.002588	-211.151878	0.139768	11.361433	21352.349921
29.000703	0.883694	0.393259	3.423470	-0.221594	0.030215	0.001925	-238.780598	-339.186310	-8.799991	21430.454058
30.000703	0.161270	0.111012	3.424923	-0.483308	-0.473259	0.000978	-69.547485	-95.748799	-21.652894	21311.599130
31.000703	-0.257608	-0.260109	3.425433	-0.293290	-0.157938	0.000045	111.092487	224.343592	-26.048367	21244.174577

รูปที่ 3.11: ตัวอย่างผลลัพธ์ละเอียดจากโปรแกรมจำลองสถานการณ์ที่ปรับปรุงใหม่