



# การสร้างภาพเคลื่อนไหวเชิงฟิสิกส์ Physics Based Computer Animation

ณัฐพงศ์ ชินธเนศ

ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
จุฬาลงกรณ์มหาวิทยาลัย

การสร้างภาพเคลื่อนไหวเชิงฟิสิกส์

Physics Based Computer Animation

ณัฐพงศ์ ชินธเนศ

ชื่อหนังสือ การสร้างภาพเคลื่อนไหวเชิงฟิสิกส์

ชื่อผู้แต่ง ญัฐพงศ์ ชินธเนศ

ข้อมูลทางบรรณานุกรมของสำนักหอสมุดแห่งชาติ  
National Library of Thailand Cataloging in Publication Data  
การสร้างภาพเคลื่อนไหวเชิงฟิสิกส์  
ญัฐพงศ์ ชินธเนศ  
ISBN (e-Book) 978-616-594-158-7

จำนวนหน้า 124 หน้า

แต่งและจัดทำโดย: ญัฐพงศ์ ชินธเนศ

Ph.D. (Computer Science)	University of California, Berkeley พ.ศ. 2550
M.Sc. (Computer Science)	University of California, Berkeley พ.ศ. 2547
B.S.E. (Computer Science/ Electrical Engineering)	University of Michigan, Ann Arbor พ.ศ. 2545

รองศาสตราจารย์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

สงวนลิขสิทธิ์

## คำนำ

การสร้างภาพเคลื่อนไหวเชิงฟิสิกส์เป็นศาสตร์ที่มีความสำคัญในการสร้างภาพยนตร์ เกม คอมพิวเตอร์ และ โปรแกรมจำลองสถานการณ์ต่างๆ เช่น การขับเครื่องบิน การหนีไฟ การใช้เครื่องมือต่างๆ เป็นต้น ในปัจจุบันยังไม่พบว่ามีตำราหรือหนังสือภาษาไทยที่รวบรวมวิธีการที่ใช้ในการการสร้างภาพเคลื่อนไหวเชิงฟิสิกส์ ซึ่งหากผู้ที่มีความรู้ด้านคณิตศาสตร์ และฟิสิกส์เนื้อหา ระดับมัธยมศึกษาตอนปลาย คณิตศาสตร์แคลคูลัส 1, 2 และ 3 ของระดับชั้นอุดมศึกษาปีที่ 2 และมีความสามารถในการเขียนโปรแกรม เมื่อได้อ่านเนื้อหาภาคทฤษฎีและคำอธิบายวิธีการ ต่างๆจากหนังสือเล่มนี้แล้วก็จะสามารถนำไปเขียนโปรแกรมสร้างภาพเคลื่อนไหวเชิงฟิสิกส์แบบ เบื้องต้นได้ ในการเขียนหนังสือเล่มนี้ผู้เขียนได้รวบรวมทฤษฎีและวิธีสร้างภาพเคลื่อนไหวเชิง ฟิสิกส์แบบต่างๆ จากตำราต่างประเทศและงานวิจัยที่ผ่านมา ตลอดจนประสบการณ์ตรงจาก งานวิจัยของผู้เขียนเองมาเรียบเรียงให้ผู้อ่านได้ทั้งความรู้พื้นฐานที่จำเป็นต่อการสร้าง ภาพเคลื่อนไหวเชิงฟิสิกส์และวิธีการเขียนโปรแกรมการสร้างภาพดังกล่าวได้ หนังสือเล่มนี้จึง ประกอบด้วย บทที่เกี่ยวกับการทบทวนคณิตศาสตร์ ระบบอนุภาค การจำลองวัตถุแข็ง การ จำลองทางฟิสิกส์ด้วยวิธี ไดนามิกเชิงตำแหน่ง การจำลองทางฟิสิกส์ด้วยตาราง การจำลองทาง ฟิสิกส์แบบผสมตารางและอนุภาค การจำลองตัวละครด้วยฟิสิกส์ โดยมีภาพประกอบเพื่อให้ ผู้อ่านสามารถติดตามและทำความเข้าใจวิธีการหลายๆแบบในการจำลองภาพได้ง่ายขึ้นอีกทั้งยัง สามารถเลือกได้ว่าวิธีการใดเหมาะสมกับการนำไปใช้ในกรณีใด ในบทสุดท้ายผู้เขียนได้รวบรวม และสังเคราะห์ข้อมูลจากงานวิจัยของผู้เขียนเองและงานวิจัยที่สำคัญอื่นๆ เพื่อเป็นตัวอย่างของ การศึกษาวิจัยค้นคว้าวิธีการสร้างภาพเคลื่อนไหวเชิงฟิสิกส์ เพื่อให้ผู้ที่สนใจในศาสตร์นี้สามารถ ศึกษาเนื้อหาขั้นสูงของการสร้างภาพเคลื่อนไหวเชิงฟิสิกส์ในแต่ละด้านได้อย่างหลากหลายขึ้น

ในการพิมพ์ครั้งที่สองนี้ผู้เขียนได้มีการปรับปรุงเนื้อหาตามข้อเสนอแนะของผู้ทรงคุณวุฒิ ที่ได้ให้ความกรุณาอ่านประเมินหนังสือเล่มนี้เพื่อให้ผู้อ่านสามารถติดตามและทำความเข้าใจได้ ง่ายขึ้น มีการเพิ่มคำอธิบายหัวข้อต่างๆให้สมบูรณ์ขึ้น แก้ไขคำที่พิมพ์ผิด และปรับปรุงรูปเล่มให้มี ขนาดกะทัดรัดขึ้น ผู้เขียนหวังเป็นอย่างยิ่งว่าผู้ที่ได้อ่านหนังสือเล่มนี้จะได้รับความรู้ทั้งทางทฤษฎี และทางการปฏิบัติในการเขียนโปรแกรมการสร้างภาพเคลื่อนไหวเชิงฟิสิกส์ได้

ณัฐพงศ์ ชินธเนศ

10 กันยายน 2565

## กิตติกรรมประกาศ

ผู้เขียนขอกราบขอบพระคุณครูและอาจารย์ที่ได้ให้วิชาความรู้และอบรมสั่งสอนตั้งแต่ อนุบาลจนสำเร็จการศึกษาขั้นสูงได้แก่คุณครูและคณาจารย์จากโรงเรียนอนุบาลโกศลวิทยา อนุบาล ประถมและมัธยม โรงเรียนสาธิตจุฬาลงกรณ์มหาวิทยาลัย อาจารย์สอนภาษา Pascal และ Assembly ที่ BBC อาจารย์ จิตรทัศน์ ฝึกเจริญผล และอาจารย์ที่สอนค่ายคอมพิวเตอร์ โอลิมปิกวิชาการทุกท่านที่กรุณาถ่ายทอดความรู้ด้านคอมพิวเตอร์ให้ ดร. สุรพันธ์ เมฆนาวิน ดร. อภินทร อุณากุล ดร. วิวัฒน์ วงศ์รวาวิภัทร์ และอาจารย์ท่านอื่นๆ จาก NECTEC ที่ช่วยให้ คำแนะนำด้านการทำวิจัยและโครงการต่างๆในขณะที่ผู้เขียนเรียนอยู่ในชั้นมัธยมศึกษา อาจารย์ ที่ จุฬาลงกรณ์มหาวิทยาลัย University of Michigan, Ann Arbor และ University of California, Berkeley ที่กรุณาถ่ายทอดความรู้ทางด้านคอมพิวเตอร์กราฟฟิกโดยเฉพาะอย่างยิ่ง Prof. James F. O'Brien และ Prof. Ken Goldberg ที่กรุณาเป็นที่ปรึกษาวิทยานิพนธ์ระดับปริญญาเอก ขอขอบคุณเพื่อนนักเรียน และเพื่อนร่วมงานที่ Pixar Animation Studio, NVIDIA Corporation และ จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ร่วมงานกันและแลกเปลี่ยนความรู้ ประสบการณ์ ตลอดจนข้อคิดเห็นในโอกาสต่างๆซึ่งเป็นประโยชน์ต่อการเขียนหนังสือเล่มนี้มาก

กราบขอบพระคุณ ศ. ดร. บุญเสริม กิจศิริกุล ศ. ดร. วิวัฒน์ วัฒนาวุฒิ รศ. ดร. สมชาย ประสิทธิ์จตุระกุล ที่กรุณาให้คำแนะนำเกี่ยวกับการเขียนหนังสือเล่มนี้ กราบขอบพระคุณ รศ. ดร. พันธุ์ปิติ เปี่ยมสง่า Associate Professor Dr. บัณฑิต เลขานุกิจ และ Professor Dr. ดนุพล ฌ หนองคาย ที่กรุณาสละเวลามาช่วยประเมินคุณภาพหนังสือเล่มนี้รวมทั้งให้คำแนะนำในการ ปรับปรุงที่มีคุณค่ายิ่ง ผู้เขียนขอขอบพระคุณ ผศ. ดร. อรรถวิทย์ สุตแสง หัวหน้าภาควิชา วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่กรุณาให้ให้ความ สนับสนุนในทุกด้านเสมอมา

ท้ายที่สุดขอกราบขอบพระคุณบุพการี รศ.ดร. ไถ้ออน และ ศ.ดร.พญ. วิไล ชินธเนศ ที่ให้ การอบรมเลี้ยงดูเป็นอย่างดี และขอบคุณนายแพทย์ ธีระ (พี่ชาย) และ สมาชิกในครอบครัว ได้แก่ คุณสุนันท์ คุณปิยะนันท์ ด.ช. ณดิณณ์ ด.ช. ณณาน และ ด.ญ. ณวาริน สำหรับการส่งเสริม ความเสียสละและเป็นกำลังใจในการทำงานวิจัยและเขียนหนังสือเล่มนี้

ณัฐพงศ์ ชินธเนศ

10 กันยายน 2565

# สารบัญ

บทนำ.....	1
ความหมายของภาพเคลื่อนไหวเชิงฟิสิกส์และศาสตร์ที่เกี่ยวข้อง .....	2
คณิตศาสตร์.....	2
ฟิสิกส์ .....	2
คอมพิวเตอร์ .....	3
ตัวอย่างของภาพเคลื่อนไหวเชิงฟิสิกส์ และการประยุกต์ใช้ในด้านต่างๆ.....	3
บทที่ 1: ทบทวนคณิตศาสตร์ .....	6
บทนำ.....	6
สเกลาร์ และ เวกเตอร์.....	6
เมทริกซ์.....	11
การคูณเมทริกซ์ .....	13
ควอเทอร์เนียน .....	15
ควอเทอร์เนียนสำหรับการหมุน .....	17
อนุพันธ์.....	19
ปฏิยานุพันธ์.....	21
สมการเชิงอนุพันธ์สามัญ .....	22
สมการเชิงอนุพันธ์ย่อย .....	23
บทสรุป.....	23
บทที่ 2: ระบบอนุภาค.....	24
ขั้นตอนการทำงานของระบบอนุภาค.....	25
ตัวปล่อยอนุภาค .....	25

การปรับปรุงอนุภาค .....	26
การแสดงผล .....	30
บทสรุป.....	32
บทที่ 3: การจำลองวัตถุแกร่ง.....	33
ระบบสมการเชิงอนุพันธ์สามัญของการเคลื่อนไหวของวัตถุแกร่ง .....	35
การชน, การสัมผัส และข้อจำกัดต่างๆ .....	38
การติดตามลำดับ.....	44
การปรับปรุงการติดตามลำดับ.....	46
ประเด็นอื่นๆเกี่ยวกับการจำลองวัตถุแกร่ง .....	48
บทสรุป.....	49
บทที่ 4: การจำลองทางฟิสิกส์ด้วยวิธีไดนามิกเชิงตำแหน่ง.....	50
พื้นฐานของการจำลองวิธีไดนามิกเชิงตำแหน่ง .....	50
การจำลองวัตถุอ่อนนุ่มด้วยไดนามิกเชิงตำแหน่ง.....	52
ข้อจำกัดการรอ .....	52
ข้อจำกัดพื้นที่ของสามเหลี่ยม .....	53
ข้อจำกัดปริมาตรของจตุรมุข .....	54
ข้อจำกัดของการชนระหว่างอนุภาคกับสามเหลี่ยม .....	54
ข้อจำกัดของการชนระหว่างอนุภาคกับสิ่งแวดล้อม .....	54
ข้อจำกัดของการชนระหว่างอนุภาคกับอนุภาค .....	54
ข้อจำกัดของแรงเสียดทานระหว่างอนุภาคกับอนุภาค .....	55
การจำลองของเหลวด้วยไดนามิกเชิงตำแหน่ง .....	55
บทสรุป.....	57
บทที่ 5: การจำลองทางฟิสิกส์ด้วยตาราง .....	58

พื้นฐานของการจำลองด้วยตาราง.....	58
สมการของของไหล .....	59
การจำลองน้ำด้วยสมการคลื่น .....	60
การจำลองควีน.....	63
บทสรุป.....	69
บทที่ 6: การจำลองทางฟิสิกส์แบบผสมตารางและอนุภาค .....	70
วิธี Fluid-Implicit Particle (FLIP) .....	70
บทสรุป.....	74
บทที่ 7: การจำลองตัวละครด้วยฟิสิกส์ .....	75
แอนิเมชันของตัวละครแบบพื้นฐานโดยไม่ใช้ฟิสิกส์.....	75
พื้นฐานของโมชันแคปเจอร์ .....	78
โมชันแคปเจอร์แบบแม่เหล็ก (Magnetic Motion Capture).....	78
โมชันแคปเจอร์แบบกล (Mechanics Motion Capture).....	78
โมชันแคปเจอร์แบบแสง (Optical Motion Capture).....	78
พื้นฐานของการจำลองตัวละครด้วยฟิสิกส์ .....	80
บทสรุป.....	83
บทที่ 8: คอมพิวเตอร์แอนิเมชันเชิงฟิสิกส์ขั้นสูง.....	84
ระบบอนุภาคขั้นสูง.....	84
การจำลองวัตถุแกร่งขั้นสูง.....	84
การจำลองทางฟิสิกส์ด้วยวิธีไดนามิกเชิงตำแหน่งขั้นสูง .....	88
การจำลองทางฟิสิกส์ด้วยตารางขั้นสูง และ การจำลองทางฟิสิกส์แบบผสมตารางและอนุภาคขั้นสูง .....	94
การจำลองตัวละครด้วยฟิสิกส์ขั้นสูง.....	101
การใช้ปัญญาประดิษฐ์ในคอมพิวเตอร์แอนิเมชันเชิงฟิสิกส์ขั้นสูง.....	102

บทสรุป.....	104
บรรณานุกรม.....	106
ดัชนี / Index .....	116
ประวัติผู้เขียน.....	122

## บทนำ

ในปัจจุบันได้มีการใช้คอมพิวเตอร์ในการสร้างภาพเคลื่อนไหว (Animation) สำหรับวัตถุประสงค์ต่างๆ มากมายเช่น งานโฆษณา ภาพยนตร์ และเกม โดยภาพเคลื่อนไหวส่วนใหญ่จะสร้างโดยศิลปินด้วยโปรแกรมต่างๆ ซึ่งต้องใช้เวลา และความสามารถเฉพาะตัวสูงมาก แต่ว่ามีภาพเคลื่อนไหวบางประเภทเช่น การระเบิด กลุ่มควัน ไฟไหม้ น้ำ คลื่น ตึกถล่ม การยับของเสื้อผ้า และผม เช่นในภาพประกอบ 1 ซึ่งการเคลื่อนไหวเหล่านี้มักเป็นไปตามกฎทางฟิสิกส์ ซึ่งมีความซับซ้อนแต่มีกฎเกณฑ์ชัดเจน และผู้รับชมมักมีโอกาสดูได้เห็นเหตุการณ์คล้ายกันนี้ในชีวิตประจำวัน หรือจากข่าวต่างๆอยู่บ่อยๆ ทำให้ศิลปินสร้างภาพเคลื่อนไหวประเภทนี้ด้วยความยากลำบาก จึงทำให้หลายสิบปีที่ผ่านมาได้มีการวิจัยจำนวนมากที่เกี่ยวกับการเขียนโปรแกรม เพื่อให้คอมพิวเตอร์สามารถสร้างภาพเคลื่อนไหวประเภทนี้ได้



ภาพประกอบ 1: ตัวอย่างการเคลื่อนไหวที่เกิดจากฟิสิกส์ ระเบิดควัน ไฟ คลื่น ตึกถล่ม เสื้อผ้า ผม เครดิต: Unsplash.com

## ความหมายของภาพเคลื่อนไหวเชิงฟิสิกส์และศาสตร์ที่เกี่ยวข้อง

ภาพเคลื่อนไหวที่พยายามจะเลียนแบบปรากฏการณ์การเคลื่อนไหวในเชิงฟิสิกส์ เรียกโดยรวมว่า ภาพเคลื่อนไหวเชิงฟิสิกส์ (Physics Based Computer Animation) ซึ่งแบ่งเป็นประเภทหลักๆ 2 ประเภทด้วยกัน คือแบบแอคทีฟ (Active) ที่เป็นการเคลื่อนไหวที่มีเจตจำนงเช่นการเคลื่อนไหวของร่างกายของคนและสัตว์ ซึ่งบางที่เรียกว่าการเคลื่อนที่ปฐมภูมิ (Primary Motion) และแบบพาสซีฟ (Passive) ซึ่งเป็นการเคลื่อนที่ที่ปราศจากเจตจำนง เช่น ไฟ ควัน น้ำ หรือแม้แต่ เส้นผม เสื้อผ้า ที่ขยับตามร่างกายของคน หรือสัตว์ ซึ่งบางที่เรียกว่าการเคลื่อนที่ทุติยภูมิ (Secondary Motion) โดยเนื้อหาส่วนใหญ่ ของหนังสือเล่มนี้จะพูดถึง ภาพเคลื่อนไหวเชิงฟิสิกส์แบบพาสซีฟเป็นหลัก ในการสร้างภาพเคลื่อนไหวเชิงฟิสิกส์นั้น จะต้องใช้ความรู้จากศาสตร์ต่างๆหลายสาขาโดยเฉพาะคณิตศาสตร์ ฟิสิกส์ และคอมพิวเตอร์โดยความรู้เบื้องต้นที่เกี่ยวข้องในแต่ละสาขามีดังนี้

### คณิตศาสตร์

เป็นเครื่องมือที่ใช้ในการคำนวณต่างๆ โดยเนื้อหาหลักที่จำเป็นต้องมีความรู้พื้นฐานคือ เวกเตอร์ (Vector) เมทริกซ์ (Matrix), ควอเทอร์เนียน (Quaternion) อนุพันธ์ (Derivative) ปฏิยานุพันธ์ (Integrate) และ วิธีเชิงตัวเลข (Numerical Method) ซึ่งจะถูกนำมาใช้ในการเขียน สมการ อสมการ สมการอนุพันธ์ ต่างๆจากกฎทางฟิสิกส์เพื่อทำการหาคำตอบ และนำคำตอบ มาใช้แสดงผล หรือเพื่อพิสูจน์ความถูกต้อง และความแม่นยำของคำตอบเป็นต้น

### ฟิสิกส์

เป็นเครื่องมือในการสร้างแบบจำลอง (Model) จากปรากฏการณ์ ที่เกิดขึ้นในธรรมชาติโดยการเขียนความสัมพันธ์ทางคณิตศาสตร์ของปริมาณต่างๆ เช่น มวล แรง ความเร่ง ความร้อน ความหนาแน่น เป็นต้น โดยความรู้ทางฟิสิกส์ ที่มีการใช้งานอย่างแพร่หลายในการสร้างภาพเคลื่อนไหวเชิงฟิสิกส์ที่ได้แก่ กฎการเคลื่อนที่ของนิวตัน โมเมนตัม งานและพลังงาน การหมุน กลศาสตร์ของไหล และความร้อนเป็นต้น

## คอมพิวเตอร์

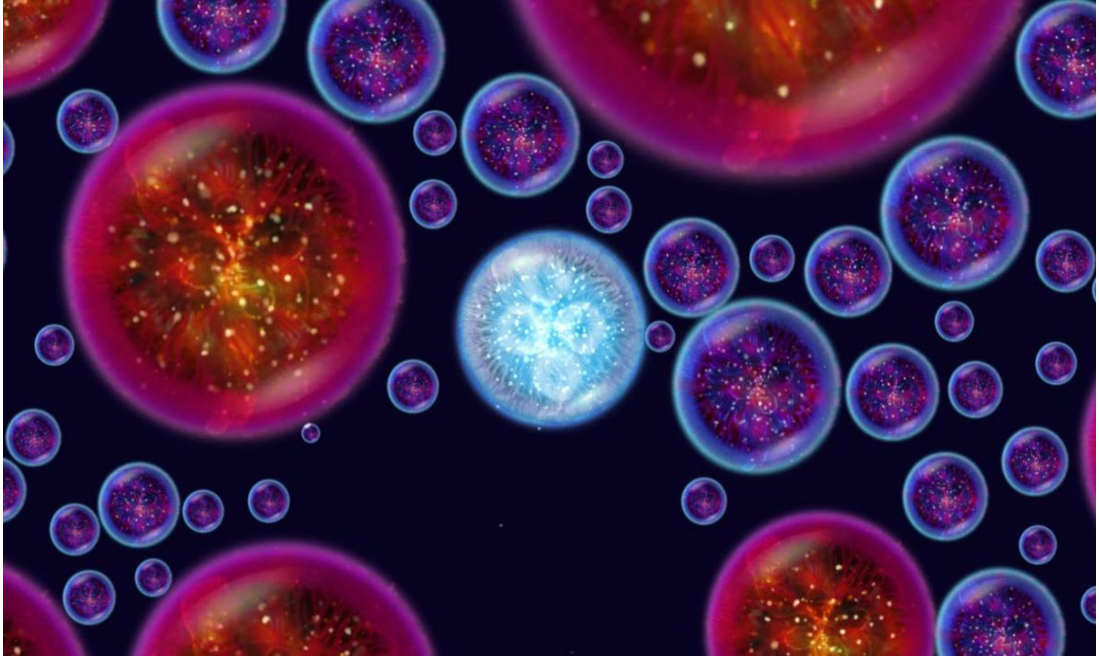
การจะสร้างภาพเคลื่อนไหวเชิงฟิสิกส์ได้นั้นจะต้องมีการคำนวณด้วยคอมพิวเตอร์ซึ่งต้องอาศัยการเขียนโปรแกรม การออกแบบโครงสร้างข้อมูลที่เหมาะสม การออกแบบอัลกอริทึมที่มีประสิทธิภาพ โดยโปรแกรมที่ใช้ในการคำนวณนั้นโดยมากจะต้องมีการคำนวณที่ซับซ้อน และทำกับปริมาณข้อมูลที่ค่อนข้างมากซึ่งทำให้จำเป็นที่จะต้องมีโปรแกรมที่มีประสิทธิภาพบนเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูง โดยอาจจะต้องใช้ Multi Threading หรือ Graphics Processing Unit (GPU) ในการคำนวณด้วย

## ตัวอย่างของภาพเคลื่อนไหวเชิงฟิสิกส์ และการประยุกต์ใช้ในด้านต่างๆ

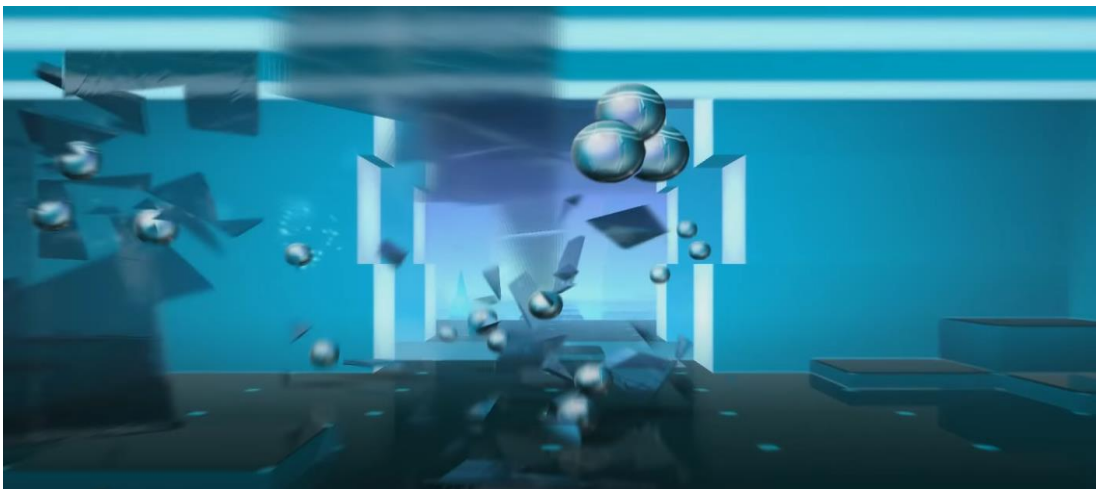
ภาพเคลื่อนไหวเชิงฟิสิกส์มีการใช้งานอย่างกว้างขวางในวงการภาพยนตร์ เกม และสื่อการสอนต่างๆ ดังตัวอย่างในภาพประกอบที่ 2-6 ในบทนี้



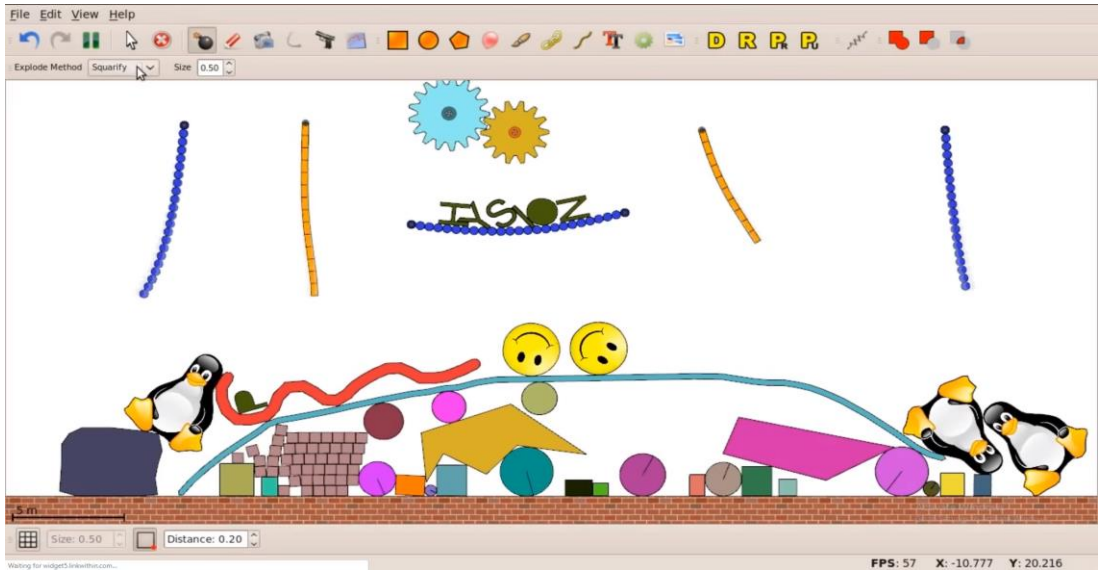
ภาพประกอบ 2: การจำลองน้ำที่ฉีดจากสายยางด้วยฟิสิกส์ของของเหลวเพื่อดับไฟ และการจำลองวัตถุแกร่งเช่น หิน ซึ่งสามารถมีผลกระทบกับน้ำได้ ภาพจากเกม Sprinkle, เครดิต: ©2011 Mediocre Games



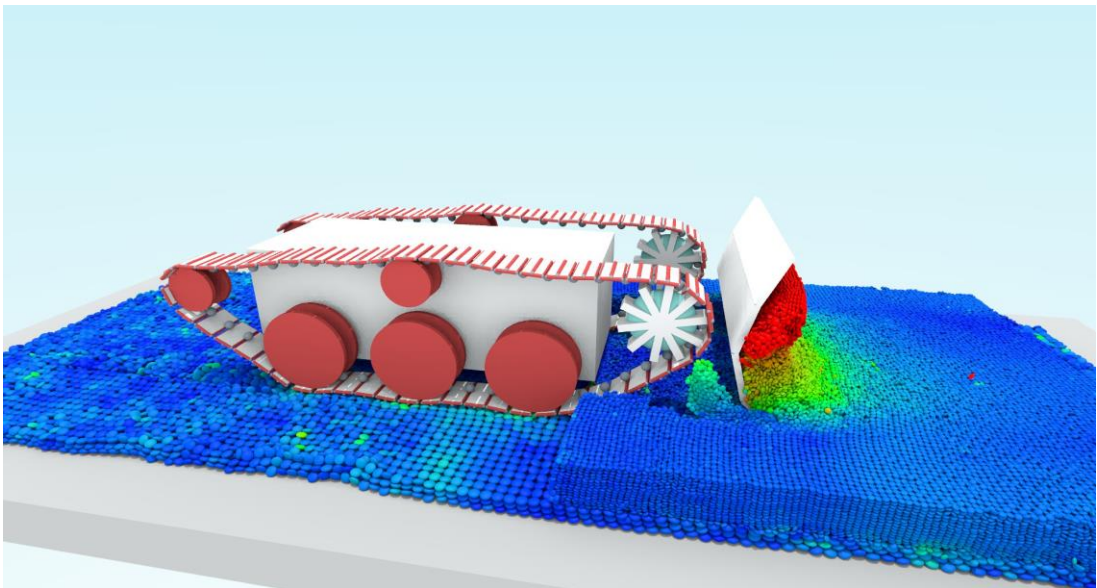
ภาพประกอบ 3: การจำลองกฎการเคลื่อนที่ของนิวตันของวงกลมขนาดต่างๆจำนวนมาก โดยวงกลมต่างๆสามารถชนและแลกเปลี่ยนโมเมนตัมกันได้ ภาพจากเกม Osmos, เครดิต: ©2009 Hemisphere Games



ภาพประกอบ 4: การจำลองการแตก และการเคลื่อนที่ของวัตถุแกร่ง โดยเมื่อเกิดการชนวัตถุสามารถแตกหักได้ และวัตถุแกร่งถูกแรงโน้มถ่วงดูดให้ตกลงสู่พื้น ภาพจากเกม Smash Hit, เครดิต: ©2014 Mediocre AB



ภาพประกอบ 5: การจำลองวัตถุแข็ง 2 มิติเช่นกล่อง วงกลม รูปเพนกวิน รูปเฟือง รวมไปถึงวัตถุแข็งที่มีข้อต่อเชื่อมต่อกันเป็นเชือก เพื่อสร้างสื่อการสอน ภาพจากโปรแกรม Physion เครดิต: ©2015 Physion



ภาพประกอบ 6: การจำลองรถถังที่มีล้อสายพานผลัดหน้าดิน จำลองด้วย Chrono รถไถนั้นจำลองด้วยวัตถุแข็งที่มีข้อต่อประเภทต่างๆ เพื่อเชื่อมเป็นสายพานและล้อ ดินนั้นจำลองด้วยระบบอนุภาคที่สามารถชนกันได้ เครดิต:

©2018 Project Chrono

# บทที่ 1: ทบทวนคณิตศาสตร์

## บทนำ

คณิตศาสตร์มีบทบาทสำคัญในการสร้างภาพเคลื่อนไหวเชิงฟิสิกส์อย่างมากทั้งในแง่การทำความเข้าใจกฎ

ทางฟิสิกส์ การแปลงกฎทางฟิสิกส์เป็น สมการ อสมการ สมการอนุพันธ์

รวมถึงการหาคำตอบเพื่อนำคำตอบไปแสดงผล จึงจำเป็นที่ผู้ศึกษาจะต้องมีความรู้ทางคณิตศาสตร์พื้นฐานเป็น

อย่างดี แต่ในขณะที่เดียวกันก็ไม่จำเป็นที่ผู้ศึกษาจะต้องมีความเข้าใจ คณิตศาสตร์ในระดับลึกซึ่งไปเสียทั้งหมด

ดังนั้นผู้เขียนจึงจะนำเสนอเนื้อหาทางคณิตศาสตร์เบื้องต้นที่จำเป็นต้องใช้บ่อยๆ ไว้ในบทนี้

ส่วนเนื้อหาทางคณิตศาสตร์ที่ซับซ้อนอื่นๆที่จำเป็นต้องใช้สำหรับเนื้อหาบางส่วนของหนังสือ ก็จะถูกกล่าวถึง

ในส่วนนั้นๆ พร้อมกับหนังสืออ้างอิงที่ไว้ให้ผู้ศึกษาที่สนใจสามารถอ่านเพิ่มเติมได้

โดยเนื้อหาทางคณิตศาสตร์ที่จะกล่าวถึง ในบทนี้คือเรื่อง เวกเตอร์ (Vector) เมทริกซ์ (Matrix)

ควอเทอร์เนียน(Quaternion) สมการเชิงอนุพันธ์สามัญ (Ordinary Differential Equation)

สมการเชิงอนุพันธ์ย่อย (Partial Differential Equation) ผู้อ่านที่มีความคุ้นเคยกับเนื้อหาเหล่านี้ดี สามารถข้าม

ไปยังบทต่อไปได้

## สเกลาร์ และ เวกเตอร์

สเกลาร์ (Scalar) เป็นจำนวนจริงที่มีขนาดแต่ไม่มีทิศทาง ในขณะที่ เวกเตอร์ (Vector) เป็นรายการของจำนวน

จริงที่มีทั้งขนาดและทิศทาง ตัวอย่างของปริมาณในเชิงฟิสิกส์ที่เป็นสเกลาร์คือ มวล (Mass) อัตราเร็ว (Speed)

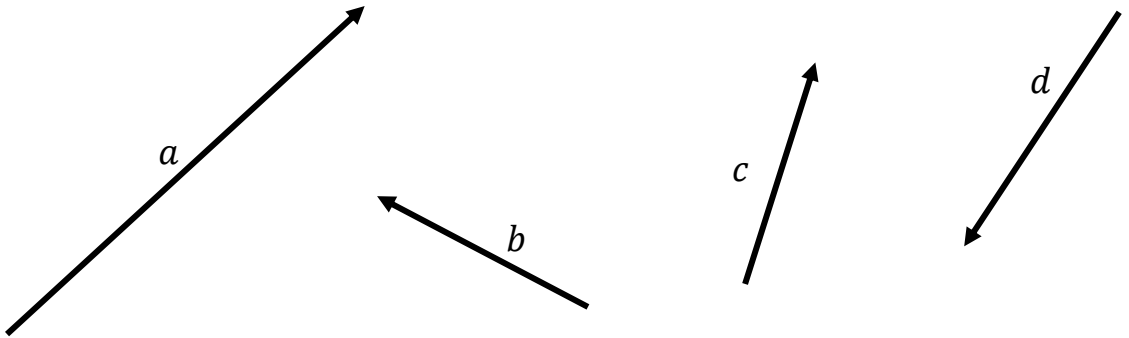
พลังงาน (Energy) ความดัน (Pressure) เป็นต้น ตัวอย่างของปริมาณในเชิงฟิสิกส์ที่เป็นเวกเตอร์คือ การกระจัด

(Displacement) ความเร็ว (Velocity) ความเร่ง (Acceleration) ความเร็วเชิงมุม (Angular Velocity) แรง

(Force) แรงบิด (Torque) และโมเมนตัม (Momentum) เป็นต้น

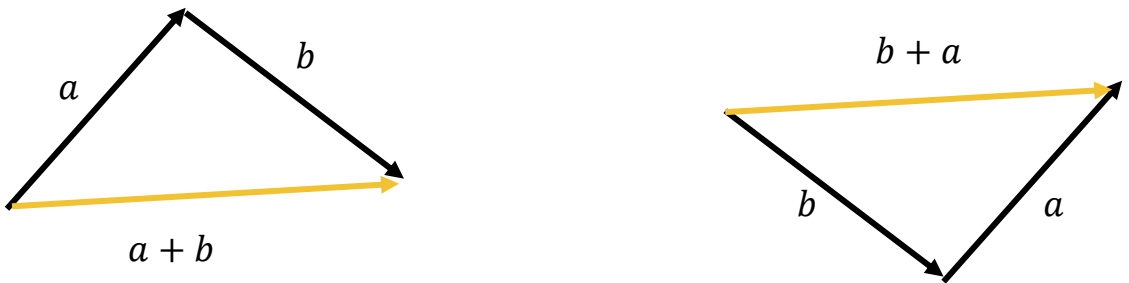
เวกเตอร์สามารถถูกระบุได้ด้วยลูกศร โดยที่ทิศทางของลูกศรจะเป็นตัวระบุทิศทางของเวกเตอร์

และความยาวของลูกศรระบุขนาดของเวกเตอร์ ตัวอย่างของเวกเตอร์อยู่ในภาพประกอบ 7



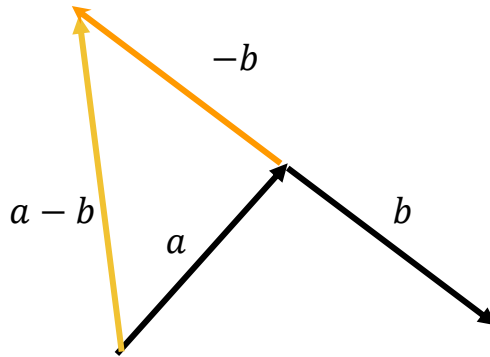
ภาพประกอบ 7: ตัวอย่างของเวกเตอร์ ซึ่งประกอบไปด้วยทิศทางและขนาด

ในหนังสือเล่มนี้ปริมาณที่เป็นเวกเตอร์จะถูกระบุด้วยตัวเอียงเช่น  $x, v, a, b$  เป็นต้น โดยสามารถนำเวกเตอร์สองอันมาบวกกันได้เช่น  $a + b$  ด้วยการนำลูกศรมาต่อกันโดยไม่จำเป็นว่าจะนำอันไหนมาก่อน กล่าวคือ  $a + b = b + a$  ดังตัวอย่างในภาพประกอบ 8



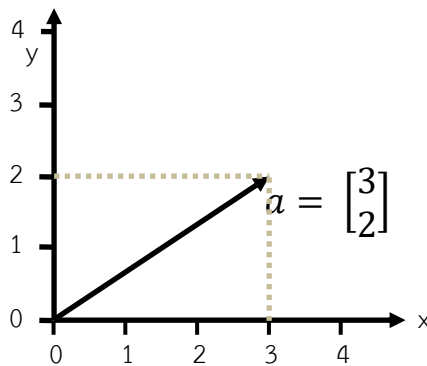
ภาพประกอบ 8: การบวกของเวกเตอร์โดยการนำลูกศรมาต่อกัน  $a + b = b + a$

และสามารถทำการลบเวกเตอร์ได้โดยการนำเวกเตอร์ที่ต้องการลบมากลับทิศทาง แล้วนำไปต่อกับเวกเตอร์แรก ตัวอย่างเช่น  $a - b$  ในภาพประกอบ 9



ภาพประกอบ 9 : การลบของเวกเตอร์โดยการนำเวกเตอร์ที่ต้องการลบมากลับทิศแล้วนำมาต่อกัน

โดยทั่วไปแล้วเวกเตอร์สามารถระบุด้วยรายการของจำนวนจริงได้เช่น  $\mathbf{a} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$  โดยจำนวนจริงระบุพิกัด (Coordinate) เมื่อเทียบกับแกนอ้างอิง (Axis) ซึ่งโดยทั่วไปแล้วสำหรับเวกเตอร์ 2 มิติแกนอ้างอิงที่มักจะใช้คือ แกน x และแกน y ซึ่งแกน x เป็นแกนแนวนอนที่ชี้ไปทางขวา และแกน y คือแกนแนวตั้งที่ชี้ขึ้น โดยในหนังสือเล่มนี้ก็ยึดถือขนบ (Convention) นี้เช่นกัน ตัวอย่างเวกเตอร์  $\mathbf{a} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$  แกน x และ แกน y แสดงอยู่ในภาพประกอบ 10

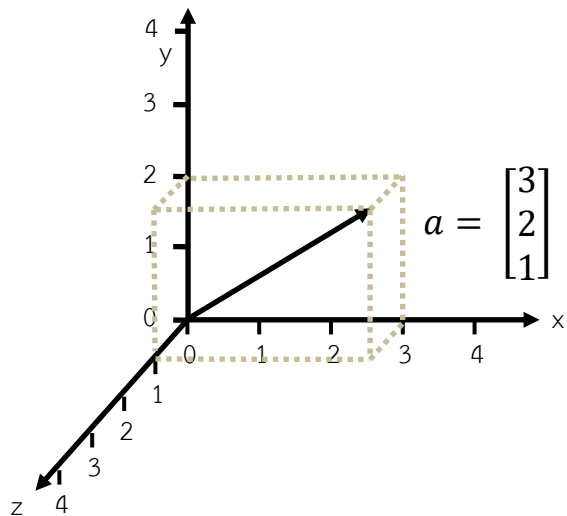


ภาพประกอบ 10: ตัวอย่างเวกเตอร์ แกน x และ แกน y

มิติของเวกเตอร์คือความยาวของรายการของจำนวนจริงที่ใช้ระบุเวกเตอร์ โดยจะมีกี่มิติก็ได้ แต่โดยทั่วไปแล้วที่พบบ่อยที่สุดคือเวกเตอร์ 2 มิติ และ 3 มิติ โดยเวกเตอร์ 3 มิติจะมีแกน z เพิ่มเข้ามาซึ่งคือแกนที่พุ่งออกจากกระดาษ\*

และเวกเตอร์จะถูกระบุได้ด้วยรายการของจำนวนจริง 3 ตัวเช่น  $\begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$  ตัวอย่างของเวกเตอร์ 3 มิติดูได้จาก

ภาพประกอบ 11



ภาพประกอบ 11: ตัวอย่างเวกเตอร์ 3 มิติ แกน x แกน y และ แกน z

โดยถ้าแกน x บวกชี้ไปทางขวา แกน y บวกชี้ขึ้น และแกน z ชี้ออกมา จะเรียกว่า Right Hand Coordinate System เนื่องจาก  $\mathbf{x} \times \mathbf{y} = \mathbf{z}$  ด้วยนियามการ Cross ด้วยกฎมือขวาซึ่งหนังสือเล่มนี้ใช้ แต่หนังสือและซอฟต์แวร์บางตัวแกน z ที่เป็นบวกจะพุ่งเข้าไป  $\mathbf{x} \times \mathbf{y} = -\mathbf{z}$  ซึ่งเรียกว่า Left Hand Coordinate System

ขนาดของเวกเตอร์  $\mathbf{a}$  แทนด้วย  $|\mathbf{a}|$  คือความยาวของเวกเตอร์ซึ่งคำนวณได้โดยนำส่วนของเวกเตอร์ในแกน

ต่างๆมายกกำลังแล้วบวกกัน จากนั้นก็นำมาหารากที่สองเช่น  $\left\| \begin{bmatrix} 3 \\ 2 \end{bmatrix} \right\| = \sqrt{3^2 + 2^2} = \sqrt{9 + 4} = \sqrt{13}$ ,

$$\left\| \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \right\| = \sqrt{3^2 + 2^2 + 1^2} = \sqrt{9 + 4 + 1} = \sqrt{14} \text{ เป็นต้น โดยเรียกเวกเตอร์ที่มีขนาด}$$

เป็น 1 ว่าเวกเตอร์หน่วย (Unit Vector)

$$\text{เวกเตอร์หน่วยที่ชี้ไปทางบวกของแกน } x, y, z \text{ มักถูกเรียกว่า } i = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, j = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, k = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

การเอาเวกเตอร์มาคูณกับจำนวนจริงหรือสเกลาร์ทำได้โดยเอาแต่ละส่วนของเวกเตอร์มาคูณกับสเกลาร์นั้น เช่น

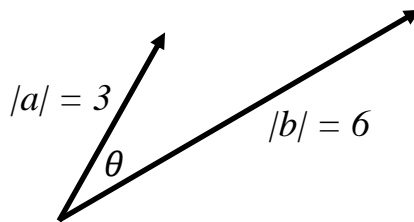
$$7.2 \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 7.2 \times 3 \\ 7.2 \times 2 \end{bmatrix} = \begin{bmatrix} 21.6 \\ 14.4 \end{bmatrix} \text{ เป็นต้น}$$

การคูณเวกเตอร์ด้วยเวกเตอร์มีสองแบบคือ การคูณเชิงสเกลาร์ (Dot Product) และการคูณเชิงเวกเตอร์ (Cross Product) การคูณเชิงสเกลาร์ (Dot Product) ของ  $a$  และ  $b$  แทนด้วย  $a \cdot b$  นั่นคือผลบวกของการเอาแต่ละ

$$\text{ส่วนของเวกเตอร์ทั้งสองอันมาคูณกันเช่น } \begin{bmatrix} 2 \\ 5 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 7 \\ 1 \\ 6 \end{bmatrix} = 2 \times 7 + 5 \times 1 + 4 \times 6 = 14 +$$

$$5 + 24 = 43 \text{ เป็นต้น}$$

โดย  $a \cdot b$  สามารถคำนวณได้อีกวิธีโดยการใช้สูตร  $a \cdot b = |a||b| \cos \theta$  โดยที่  $\theta$  คือมุมระหว่างสองเวกเตอร์ตัวอย่างเช่นใน ภาพประกอบ 12



ภาพประกอบ 12: การคูณเชิงสเกลาร์ (Dot Product) ของ  $a$  และ  $b$  โดยที่  $\theta = 30^\circ$  ดังนั้น

$$a \cdot b = |a||b| \cos \theta = 3 \times 6 \times \cos 30^\circ = 18 \times \frac{\sqrt{3}}{2} = 9\sqrt{3}$$

การคูณเชิงสเกลาร์นั้นสามารถทำได้กับเวกเตอร์ที่มีมิติก็ได้ โดยการคูณเชิงสเกลาร์มีคุณสมบัติดังนี้

1.  $a \cdot b = b \cdot a$
2.  $a \cdot (b + c) = a \cdot b + a \cdot c$

3.  $(a + b) \cdot (c + d) = a \cdot c + a \cdot d + b \cdot c + b \cdot d$
4.  $k(a \cdot b) = (ka) \cdot b = a \cdot (kb)$  เมื่อ  $k$  เป็นสเกลาร์
5.  $a \cdot a = |a|^2$
6.  $a \cdot b = 0$  ถ้า  $a$  และ  $b$  ตั้งฉากกัน
7.  $i \cdot i = j \cdot j = k \cdot k = 1$
8.  $i \cdot j = i \cdot k = k \cdot j = 0$

การคูณเวกเตอร์อีกแบบคือการคูณเชิงเวกเตอร์ (Cross Product) ซึ่งนิยามกับเวกเตอร์ 3 มิติโดยเขียนได้ด้วย

$$a \times b \text{ โดยนิยามด้วยสูตรดังต่อไปนี้ ถ้า } a = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \text{ และ } b = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \text{ แล้ว } a \times b =$$

$$\begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix} \text{ โดยการคูณเชิงเวกเตอร์มีคุณสมบัติดังต่อไปนี้}$$

1.  $a \times b = -b \times a$
2.  $a \times (b + c) = a \times b + a \times c$
3.  $k(a \times b) = (ka) \times b = a \times (kb)$  เมื่อ  $k$  เป็นสเกลาร์
4.  $a \times 0 = 0 \times a = 0$
5.  $a \times b = 0$  ถ้า  $a$  และ  $b$  ขนานกัน
6.  $|a \times b| = |a||b| \sin \theta$  เมื่อ  $\theta$  คือมุมระหว่างสองเวกเตอร์
7.  $i \times i = j \times j = k \times k = 0$
8.  $i \times j = k, j \times k = i, k \times i = j$
9.  $j \times i = -k, k \times j = -i, i \times k = -j$

## เมทริกซ์

เมทริกซ์ (Matrix) คือตารางของจำนวนจริง ในหนังสือนี้จะเขียนโดยใช้อักษรตัวใหญ่เอียงเช่น  $A =$

$$\begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \end{bmatrix} \text{ โดยทั่วไปแล้วสามารถเขียนเมทริกซ์ที่มีขนาด } m \text{ แถว } n \text{ คอลัมน์ได้โดย}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

องค์ประกอบ (element) ในเมทริกซ์สามารถระบุได้ด้วย  $a_{ij}$  โดยที่  $i = 1, 2, \dots, m$  ระบุแถว และ  $j = 1, 2, \dots, n$  ระบุคอลัมน์ ซึ่งเรียกเมทริกซ์ที่มีขนาด  $m$  แถว  $n$  คอลัมน์ว่าเป็นเมทริกซ์ขนาด  $m \times n$  ส่วนเมทริกซ์ที่มี 1 คอลัมน์เรียกว่าเวกเตอร์คอลัมน์ (หรือเพียงเวกเตอร์เท่านั้นก็ได้) เมทริกซ์ที่มี 1 แถวเรียกว่าเวกเตอร์แถว และเมทริกซ์ที่มีจำนวนคอลัมน์เท่ากับจำนวนแถวเรียกว่าเมทริกซ์จัตุรัส (Square Matrix)

เมทริกซ์  $A$  และ  $B$  จะเท่ากันก็ต่อเมื่อ  $A$  และ  $B$  มีขนาดเท่ากันและ  $\forall 1 \leq i \leq m, 1 \leq j \leq n a_{ij} = b_{ij}$

และนิยามเมทริกซ์สลับเปลี่ยน (Transpose) ด้วยการสลับแถวและคอลัมน์เช่นถ้า  $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$  แล้ว

เมทริกซ์สลับเปลี่ยนของ  $A$  คือ  $A^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \end{bmatrix}$  โดยถ้า  $A$  เป็นเมทริกซ์ขนาด  $m \times n$  แล้ว  $A^T$  จะเป็นเมทริกซ์ขนาด  $n \times m$  โดยเมทริกซ์สลับเปลี่ยน มีคุณสมบัติดังนี้

1.  $A^{TT} = A$
2.  $(kA)^T = kA^T$  เมื่อ  $k$  เป็นสเกลาร์

เมทริกซ์สมมาตร (Symmetric Matrix) คือเมทริกซ์ที่มีคุณสมบัติ  $A = A^T$  ว่า ซึ่งจะต้องเป็นเมทริกซ์จัตุรัสเสมอ

สำหรับเมทริกซ์จัตุรัสที่ทุกๆช่องเป็น 0 นอกจากช่องบนเส้นทแยงมุมจะเรียกว่าเมทริกซ์แนวทแยง (Diagonal Matrix) โดยสามารถระบุเมทริกซ์ประเภทนี้ได้โดย

$$\text{diag}(a_1, a_2, \dots, a_n) = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_n \end{bmatrix}$$

เมทริกซ์พิเศษที่พบบ่อยๆ คือเมทริกซ์เอกลักษณ์ (Identity Matrix) ซึ่งเป็นเมทริกซ์จัตุรัส ที่มีค่า 1 ตรงช่องบนเส้นทแยงมุมและเป็น 0 ตรงช่องที่เหลือโดยจะใช้สัญลักษณ์  $I$  ในการแทนเมทริกซ์เอกลักษณ์

$$I = \text{diag}(1,1, \dots, 1) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

เมทริกซ์  $A$  และ  $B$  ที่มีขนาดเท่ากันสามารถนำมาบวกหรือลบกันได้โดย  $C = A \pm B$  จะมี  $c_{ij} = a_{ij} \pm b_{ij}$

การบวกและลบของเมทริกซ์มีคุณสมบัติดังนี้

1.  $A + B = B + A$
2.  $(A \pm B) \pm C = A \pm (B \pm C)$
3.  $(A \pm B)^T = A^T \pm B^T$

### การคูณเมทริกซ์

สเกลาร์กับเมทริกซ์สามารถคูณกันได้เช่น ถ้า  $k$  เป็นสเกลาร์และ  $A$  เป็นเมทริกซ์แล้ว  $B = kA$  คำนวณได้โดย

$$b_{ij} = ka_{ij}$$

เมทริกซ์  $A$  และ  $B$  สามารถนำมาคูณกันได้ก็ต่อเมื่อจำนวนคอลัมน์ของ  $A$  เท่ากับจำนวนแถวของ  $B$  เท่านั้น นั่นคือถ้า  $A$  มีขนาด  $m \times p$  และ  $B$  มีขนาด  $p \times n$  แล้ว  $C = AB$  จะมีขนาด  $m \times n$  โดย  $C$  สามารถคำนวณได้โดย

$$c_{ij} = \sum_{k=1}^p a_{ik}b_{kj}$$

เมทริกซ์จัตุรัส  $A$  และ  $B$  ที่มีขนาดเท่ากันสามารถคูณกันได้เสมอแต่โดยทั่วไปแล้ว  $AB \neq BA$

การคูณเมทริกซ์กับเวกเตอร์นั้นเป็นกรณีพิเศษของการคูณเมทริกซ์ เช่น

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 \times 7 + 2 \times 8 + 3 \times 9 \\ 4 \times 7 + 5 \times 8 + 6 \times 9 \end{bmatrix} = \begin{bmatrix} 7 + 16 + 27 \\ 28 + 40 + 54 \end{bmatrix} = \begin{bmatrix} 50 \\ 122 \end{bmatrix}$$

เป็นต้น

การคูณของเมทริกซ์มีคุณสมบัติดังนี้

1.  $A(BC) = (AB)C$
2.  $(AB)^T = B^T A^T$
3.  $C(A + B) = CA + CB$

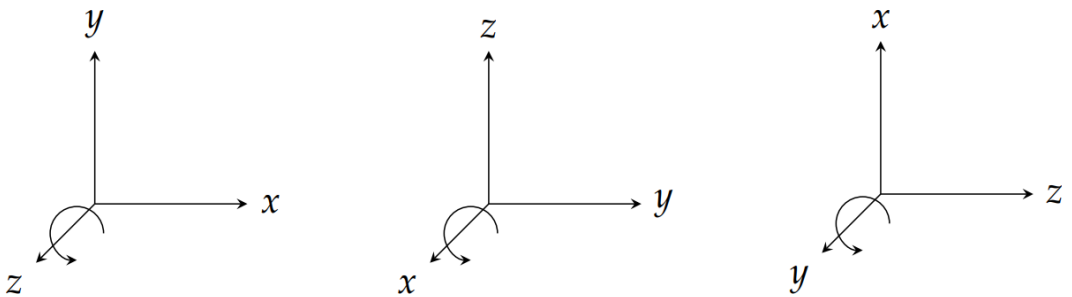
เมทริกซ์จัตุรัส  $A$  บางอันสามารถหาเมทริกซ์ผกผัน (Inverse Matrix) ได้ซึ่งเมทริกซ์ผกผันสามารถเขียนได้ด้วย  $A^{-1}$

ซึ่งเมทริกซ์ผกผันมีคุณสมบัติคือ  $AA^{-1} = I$  แต่ไม่ใช่เมทริกซ์จัตุรัสทุกอันจะสามารถหาเมทริกซ์ผกผันได้

เมทริกซ์ที่จะพบบ่อยในหนังสือเล่มนี้คือเมทริกซ์ขนาด  $3 \times 3$  เนื่องจากว่าสามารถนำมาคูณกับเวกเตอร์ 3 มิติ

ได้ โดยเมทริกซ์ที่พบบ่อยคือเมทริกซ์ที่ใช้ในการหมุนเวกเตอร์รอบแกนต่างๆ เช่น การหมุนรอบแกน  $z, x, y$  ที่

แสดงในภาพประกอบ 13



ภาพประกอบ 13: การหมุนรอบแกน  $z, x, y$

เมทริกซ์ที่ใช้ในการหมุนเวกเตอร์รอบแกน  $z, x, y$  ด้วยมุม  $\theta$  คือ

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

$$\begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

ถ้า  $k = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}$  เป็นเวกเตอร์หน่วยแล้วการหมุนเวกเตอร์รอบแกน  $k$  ด้วยมุม  $\theta$  สามารถใช้เมทริกซ์

$$R = I + \sin \theta K + (1 - \cos \theta)K^2 \text{ โดยที่ } K = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix}$$

โดยสูตรนี้เรียกว่า Rodrigues' rotation formula

ถ้ามีเมทริกซ์ที่ใช้ในการหมุน  $R_1, R_2, R_3$  จะสามารถเขียนเมทริกซ์ที่ใช้คำนวณการหมุน 1 ตามด้วย 2 ตามด้วย 3 ได้โดยผลคูณของเมทริกซ์  $R_3 R_2 R_1$

รอย (Trace) ของเมทริกซ์จัตุรัสสามารถคำนวณได้โดยนำค่าของสมาชิกบนเส้นทแยงมุมของเมทริกซ์มาบวกกัน

$$\text{tr}(A) = a_{11} + a_{22} + \dots + a_{nn}$$

โดยรอยของเมทริกซ์มีคุณสมบัติดังนี้

1.  $\text{tr}(A + B) = \text{tr}(A) + \text{tr}(B)$
2.  $\text{tr}(kA) = k \text{tr}(A)$  เมื่อ  $k$  เป็นสเกลาร์
3.  $\text{tr}(A) = \text{tr}(A^T)$
4.  $\text{tr}(AB) = \text{tr}(BA)$
5.  $\text{tr}(ABC) = \text{tr}(CAB) = \text{tr}(BCA)$

เนื้อหาของเมทริกซ์และเวกเตอร์นั้นมีมาก โดยหนังสือเล่มนี้จะมีเนื้อหาเฉพาะส่วนที่จำเป็นต่อการเข้าใจการจำลองทางฟิสิกส์พื้นฐานเท่านั้น ผู้ที่สนใจสามารถศึกษาได้เพิ่มเติมจากหนังสือ [1]

## ควอเทอร์เนียน

ควอเทอร์เนียน (Quaternion) ถูกคิดค้นโดย William Rowan Hamilton คือรายการของจำนวนจริง 4 ตัวที่มักเขียนด้วย

$q = q_0 + q_1 i + q_2 j + q_3 k$  โดยที่  $q_0, q_1, q_2, q_3$  เป็นจำนวนจริง และนิยามของสัญลักษณ์  $i, j, k$  เป็นดังนี้

$$i^2 = j^2 = k^2 = -1$$

$$ij = k, ji = -k$$

$$jk = i, kj = -i$$

$$ki = j, ik = -j$$

โดยการ บวก ลบ และคูณ จะทำโดยใช้การบวก ลบ และคูณ ของจำนวนจริงร่วมกับกฎข้างบนเช่น

$$\begin{aligned} & (q_0 + q_1i + q_2j + q_3k) + (p_0 + p_1i + p_2j + p_3k) \\ & = (q_0 + p_0) + (q_1 + p_1)i + (q_2 + p_2)j + (q_3 + p_3)k \end{aligned}$$

และ

$$\begin{aligned} & (q_0 + q_1i + q_2j + q_3k)(p_0 + p_1i + p_2j + p_3k) \\ & = (q_0p_0 + q_1p_1i^2 + q_2p_2j^2 + q_3p_3k^2) + \\ & \quad (q_0p_1i + q_1p_0i + q_2p_3jk + q_3p_2kj) + \\ & \quad (q_0p_2j + q_2p_0j + q_1p_3ik + q_3p_1ki) + \\ & \quad (q_0p_3k + q_3p_0k + q_1p_2ij + q_2p_1ji) \\ & = (q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3) + \\ & \quad (q_0p_1 + q_1p_0 + q_2p_3 - q_3p_2)i + \\ & \quad (q_0p_2 + q_2p_0 - q_1p_3 + q_3p_1)j + \\ & \quad (q_0p_3 + q_3p_0 + q_1p_2 - q_2p_1)k \end{aligned}$$

ค่าผกผัน (Conjugate) ของควอเทอร์เนียน  $q = q_0 + q_1i + q_2j + q_3k$  คือ

$$q^* = q_0 - q_1i - q_2j - q_3k$$

และขนาด (Norm) คือ

$$|q| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = qq^*$$

และทุกๆควอเทอร์เนียนที่ไม่ใช่ 0 มีค่าผกผัน (Inverse) เสมอซึ่งคำนวณได้โดย

$$q^{-1} = \frac{q^*}{|q|^2}$$

ซึ่งมีคุณสมบัติคือ  $qq^{-1} = 1$

ควอเทอร์เนียนสำหรับการหมุน

เวกเตอร์ 3 มิติ  $\mathbf{a} = \begin{bmatrix} \mathbf{a}_x \\ \mathbf{a}_y \\ \mathbf{a}_z \end{bmatrix}$  สามารถเขียนในรูปควอเทอร์เนียนบริสุทธิ์ (pure quaternion) ได้เป็น

$q_a = 0 + \mathbf{a}_x \mathbf{i} + \mathbf{a}_y \mathbf{j} + \mathbf{a}_z \mathbf{k}$  ซึ่งส่วนที่เป็นจำนวนจริงมีค่าเป็น 0 และสามารถเขียนการหมุน  $R$  ที่

แทนการหมุนรอบแกน  $\mathbf{r} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}$  ที่เป็น เวกเตอร์หน่วยไป  $\theta$  องศาในรูปของควอเทอร์เนียนหน่วย (unit

quaternion)  $q_R$  ซึ่งมีขนาดเป็น 1 ( $|q_R| = 1$ ) ได้โดย  $q_R = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} (r_x \mathbf{i} + r_y \mathbf{j} + r_z \mathbf{k})$  ซึ่งจะสามารถหมุนเวกเตอร์  $\mathbf{a}$  ด้วยการหมุน  $R$  ได้ด้วย

$$q_R q_a q_R^*$$

โดยจะได้ผลลัพธ์เป็นควอเทอร์เนียนบริสุทธิ์เสมอ ซึ่งสามารถแปลงกลับเป็นเวกเตอร์ได้

นอกจากนั้นควอเทอร์เนียนหน่วย  $q$  ใดๆ จะสามารถแปลงเป็นเมทริกซ์หมุน  $M$  ได้เสมอ ด้วยการเขียนเมทริกซ์ที่

เมื่อมาคูณกับเวกเตอร์  $\mathbf{a}$  แล้วได้ผลลัพธ์เหมือนกับ  $q q_a q^*$  ดังนี้

$$M = 2 \begin{bmatrix} q_0^2 + q_1^2 - 0.5 & q_1 q_2 - q_0 q_3 & q_0 q_2 + q_1 q_3 \\ q_0 q_3 + q_1 q_2 & q_0^2 + q_2^2 - 0.5 & q_2 q_3 - q_0 q_1 \\ q_1 q_3 - q_0 q_2 & q_0 q_1 + q_2 q_3 & q_0^2 + q_3^2 - 0.5 \end{bmatrix}$$

และถ้ามีเมทริกซ์หมุน

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

แล้วจะสามารถหาควอเทอร์เนียนหน่วยที่แทนการหมุนนี้ได้โดยวิธีการดังต่อไปนี้ [2]

จากรอยของเมทริกซ์

$$\begin{aligned} \text{tr}(M) &= m_{11} + m_{22} + m_{33} \\ &= 2(3q_0^2 + q_1^2 + q_2^2 + q_3^2 - 1.5) \\ &= 2(3q_0^2 + (1 - q_0^2) - 1.5) \\ &= 2(2q_0^2 - 0.5) \\ &= 4q_0^2 - 1 \end{aligned}$$

ดังนั้นถ้า  $\text{tr}(M) > 0$  จะสามารถหา  $q_0, q_1, q_2, q_3$  ได้โดย

$$s = 2\sqrt{\text{tr}(M) + 1}$$

$$q_0 = 0.25s$$

$$q_1 = \frac{(m_{32} - m_{23})}{s}$$

$$q_2 = \frac{(m_{13} - m_{31})}{s}$$

$$q_3 = \frac{(m_{21} - m_{12})}{s}$$

ถ้า  $\text{tr}(M) < 0$  แล้วจะต้องระวังการหารด้วย 0 และระวังการหารด้วยจำนวนจริงที่มีค่าใกล้ 0 โดยจะต้อง

พิจารณาจากขนาดของค่า  $m_{11}, m_{22}, m_{33}$  โดยถ้าค่า  $m_{11}$  มีค่ามากที่สุดจะใช้สูตร

$$s = 2\sqrt{1 + m_{11} - m_{22} - m_{33}}$$

$$q_0 = \frac{(m_{32} - m_{23})}{s}$$

$$q_1 = 0.25s$$

$$q_2 = \frac{(m_{12} + m_{21})}{s}$$

$$q_3 = \frac{(m_{13} + m_{31})}{s}$$

ถ้าค่า  $m_{22}$  มีค่ามากที่สุดจะใช้สูตร

$$s = 2\sqrt{1 + m_{22} - m_{11} - m_{33}}$$

$$q_0 = \frac{(m_{13} - m_{31})}{s}$$

$$q_1 = \frac{(m_{12} + m_{21})}{s}$$

$$q_2 = 0.25s$$

$$q_3 = \frac{(m_{23} + m_{32})}{s}$$

ถ้าค่า  $m_{33}$  มีค่ามากที่สุดจะใช้สูตร

$$s = 2\sqrt{1 + m_{33} - m_{11} - m_{22}}$$

$$q_0 = \frac{(m_{21} - m_{12})}{s}$$

$$q_1 = \frac{(m_{13} + m_{31})}{s}$$

$$q_2 = \frac{(m_{23} + m_{32})}{s}$$

$$q_3 = 0.25s$$

## อนุพันธ์

อนุพันธ์ (Derivative) ของฟังก์ชัน (Function) คืออัตราการเปลี่ยนแปลงของค่าของฟังก์ชันเมื่อเทียบการ

เปลี่ยนแปลงของค่าของพารามิเตอร์ (Parameter) อนุพันธ์ของฟังก์ชัน  $y(t)$  คือ  $\frac{dy(t)}{dt}$  หรือบางทีเขียนสั้นๆ

ด้วย  $\frac{dy}{dt}$  หรือ  $y'$

นิยาม โดย

$$\frac{dy(t)}{dt} = \lim_{h \rightarrow 0} \frac{y(t+h) - y(t)}{h}$$

ซึ่งคือการเปลี่ยนค่าของ  $y$  หารด้วยการเปลี่ยนค่าของ  $t$  เมื่อการเปลี่ยนค่าของ  $t$  มีค่าใกล้ศูนย์มากๆ

ตัวอย่างของอนุพันธ์คือ

$$y(t) = t^k$$

$$\frac{dy(t)}{dt} = \lim_{h \rightarrow 0} \frac{(t+h)^k - t^k}{h}$$

$$= \lim_{h \rightarrow 0} \frac{\sum_{j=0}^k \binom{k}{j} t^{k-j} h^j - t^k}{h}$$

$$= \lim_{h \rightarrow 0} \frac{\sum_{j=1}^k \binom{k}{j} t^{k-j} h^j + t^k - t^k}{h}$$

$$\begin{aligned}
&= \lim_{h \rightarrow 0} \frac{\sum_{j=1}^k \binom{k}{j} t^{k-j} h^j}{h} \\
&= kt^{k-1} + \lim_{h \rightarrow 0} \frac{\sum_{j=2}^k \binom{k}{j} t^{k-j} h^j}{h} \\
&= kt^{k-1}
\end{aligned}$$

อีกตัวอย่างคือ

$$\begin{aligned}
y(t) &= e^t \\
\frac{dy(t)}{dt} &= \lim_{h \rightarrow 0} \frac{e^{(t+h)} - e^t}{h} \\
&= \lim_{h \rightarrow 0} \frac{e^h e^t - e^t}{h} \\
&= \lim_{h \rightarrow 0} \frac{(e^h - 1)}{h} e^t \\
&= e^t
\end{aligned}$$

จากนิยามของ  $e$  ที่ว่า  $e$  คือจำนวนจริงที่  $\lim_{h \rightarrow 0} \frac{(e^h - 1)}{h} = 1$

คุณสมบัติของอนุพันธ์เมื่อ  $u, v$  เป็นฟังก์ชัน  $k$  เป็นค่าคงที่

1.  $(ku)' = ku'$
2.  $(u \pm v)' = u' \pm v'$
3.  $(uv)' = u'v + uv'$
4.  $\left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$
5.  $\frac{dy(x(t))}{dt} = \frac{dy(x)}{dx} \frac{dx(t)}{dt}$  เมื่อ  $y$  เป็นฟังก์ชันของ  $x$  และ  $x$  เป็นฟังก์ชันของ  $t$  เรียกว่ากฎลูกโซ่

(Chain Rules)

ถ้าฟังก์ชันใดสามารถหาค่าของอนุพันธ์ได้ทุกค่า  $t$  จะเรียกฟังก์ชันนั้นว่าฟังก์ชันที่สามารถหาอนุพันธ์ได้ (Differentiable Function) ส่วนฟังก์ชันที่ไม่สามารถหาอนุพันธ์ได้ทุกที่ จะเรียกว่าฟังก์ชันที่ไม่สามารถหาอนุพันธ์ได้ (Non-Differentiable Function)

สามารถนิยามอนุพันธ์ของฟังก์ชันที่มีค่าเป็นเวกเตอร์ (Vector Function) ได้ในลักษณะเดียวกันเช่น อนุพันธ์ของฟังก์ชัน  $y(t)$  เมื่อ  $y(t)$  เป็นเวกเตอร์คือ  $\frac{dy(t)}{dt}$

$$\frac{dy(t)}{dt} = \lim_{h \rightarrow 0} \frac{y(t+h) - y(t)}{h}$$

อนุพันธ์อันดับสอง (Second Order Derivative) คืออนุพันธ์ของอนุพันธ์เขียนแทนได้ด้วย  $\frac{d^2y(t)}{dt^2}$  หรือ  $y''$  ซึ่งมีค่าเป็น

$$\frac{d^2y(t)}{dt^2} = \lim_{h \rightarrow 0} \frac{y'(t+h) - y'(t)}{h}$$

โดยทั่วไปแล้วจะสามารถนิยามอนุพันธ์อันดับที่  $n$  ซึ่งเขียนแทนด้วย  $\frac{d^ny(t)}{dt^n}$  หรือ  $y^{(n)}(t)$  (ซึ่งคืออนุพันธ์ของอนุพันธ์ลำดับที่  $n - 1$ ) ได้โดยวิธีคล้ายๆกัน

นอกจากนี้ยังสามารถประมาณค่าของฟังก์ชันได้โดยการใช้อนุพันธ์เช่น

$$y(t+a) \approx y(t) + ay'(t)$$

และถ้ารู้ค่าอนุพันธ์อันดับอื่นด้วยก็จะสามารถประมาณค่าของฟังก์ชันได้แม่นยำมากขึ้นด้วย

$$y(t+a) \approx y(t) + ay'(t) + \frac{a^2}{2!}y''(t) + \frac{a^3}{3!}y^{(3)}(t) + \dots + \frac{a^n}{n!}y^{(n)}(t) + \dots$$

ซึ่งเรียกว่าอนุกรมเทย์เลอร์ (Taylor Series)

## ปฏิยานุพันธ์

ปฏิยานุพันธ์ (Integral) คือตัวกระทำตรงข้าม (Inverse Operator) ของอนุพันธ์เช่นถ้าอนุพันธ์ของ  $y(t)$  คือ  $x(t)$  แล้วปฏิยานุพันธ์ของ  $x(t)$  ก็คือ  $y(t)$  นั่นเองโดยสามารถเขียนสมการได้เป็น

$$x(t) = y'(t)$$

$$y(t) = \int x(t) dt$$

โดยปฏิยานุพันธ์ตัวบนเป็นปฏิยานุพันธ์แบบไม่มีขอบเขต (Indefinite Integral)

สามารถนิยามปฏิยานุพันธ์แบบมีขอบเขต (Definite Integral) ได้โดยการกำหนดขอบเขตขึ้นมาเช่น

$$\int_2^5 x(t)dt \text{ โดยถ้า } y(t) = \int x(t) dt \text{ แล้ว } \int_2^5 x(t)dt = y(5) - y(2)$$

โดยทั่วไปแล้วถ้ารู้  $x(t)$  ก็ไม่ได้หมายความว่าจะสามารถหาปฏิยานุพันธ์แบบมีขอบเขตของ  $x(t)$  ได้เสมอแต่

สามารถประมาณค่าได้โดยใช้วิธีการเชิงตัวเลข (Numerical Method) เช่น

$$\int_2^5 x(t)dt \approx \sum_{i=0}^{n-1} x(2 + i\Delta t)\Delta t \text{ เมื่อ } n \text{ เป็นจำนวนเต็มบวกและ } \Delta t = \frac{5-2}{n}$$

โดยค่าที่ประมาณจะแม่นยำมากขึ้นเมื่อ  $n \rightarrow \infty$  ซึ่ง  $\Delta t \rightarrow 0$

และยังสามารถประมาณค่าได้อีกหลายแบบเช่น

$$\int_2^5 x(t)dt \approx \sum_{i=1}^n x(2 + i\Delta t)\Delta t \text{ เมื่อ } n \text{ เป็นจำนวนเต็มบวกและ } \Delta t = \frac{5-2}{n}$$

หรือ

$$\int_2^5 x(t)dt \approx \sum_{i=0}^{n-1} x(2 + (i + 0.5)\Delta t)\Delta t \text{ เมื่อ } n \text{ เป็นจำนวนเต็มบวกและ } \Delta t = \frac{5-2}{n}$$

เนื้อหาที่เกี่ยวข้องกับอนุพันธ์และปฏิยานุพันธ์นั้นมีมาก ผู้ที่สนใจสามารถศึกษาเพิ่มเติมได้จากหนังสือคณิตศาสตร์

แคลคูลัส (Calculus) [3]

### สมการเชิงอนุพันธ์สามัญ

สมการเชิงอนุพันธ์สามัญ (Ordinary Differential Equation) คือสมการของฟังก์ชันพหามิตอร์เดียวที่มีอนุพันธ์

อันดับต่างๆของพหามิตอร์นั้นเข้ามาเกี่ยวข้องเช่น

$$y''(t) + y'(t) = 0$$

โดยสมการเชิงอนุพันธ์สามัญอาจจะเกี่ยวข้องกับหลายฟังก์ชันก็ได้ตราบใดที่ยังมีเพียงอนุพันธ์ของพารามิเตอร์เพียงอันเดียวมาเกี่ยวข้องเช่น

$$x'(t) + 2y''(t) = 3$$
$$y'(t) + x'(t) + x'(t)y''(t) = 2$$

ซึ่งสามารถเรียกสมการเชิงอนุพันธ์สามัญหลายๆอันที่จะต้องเป็นจริงพร้อมกันว่า ระบบสมการเชิงอนุพันธ์สามัญ (System of Ordinary Differential Equations) เนื้อหาของการแก้สมการเชิงอนุพันธ์สามัญ และระบบสมการเชิงอนุพันธ์สามัญนั้นมีมากโดยหนังสือนี้จะกล่าวถึงเฉพาะส่วนที่เกี่ยวข้องกับการจำลองทางฟิสิกส์พื้นฐานเท่านั้น ผู้ที่สนใจสามารถศึกษาได้เพิ่มเติมจากหนังสือเกี่ยวกับ Differential Equation [4]

### สมการเชิงอนุพันธ์ย่อย

ฟังก์ชันโดยทั่วไปแล้วอาจจะมีหลายพารามิเตอร์ก็ได้เช่น  $y(s, t)$  ซึ่งสามารถเขียนอนุพันธ์ของฟังก์ชันเมื่อเทียบกับแต่ละพารามิเตอร์ได้โดยเรียกว่าอนุพันธ์ย่อย (Partial Derivative) เช่น  $\frac{\partial y(s,t)}{\partial s}$  และ  $\frac{\partial y(s,t)}{\partial t}$  และเรียกสมการเชิงอนุพันธ์ที่ขึ้นกับอนุพันธ์ย่อยเหล่านี้ว่า สมการเชิงอนุพันธ์ย่อย (Partial Differential Equation) ในหนังสือเล่มนี้จะกล่าวถึงเนื้อหาที่เกี่ยวข้องโดยตรงกับการจำลองทางฟิสิกส์พื้นฐานเท่านั้น ผู้ที่สนใจสามารถศึกษาได้เพิ่มเติมจากหนังสือ Partial Differential Equation [5]

### บทสรุป

เนื้อหาในบทนี้กล่าวถึงคณิตศาสตร์พื้นฐานต่างๆที่จำเป็นต่อการทำความเข้าใจวิธีการในการสร้างภาพเคลื่อนไหวเชิงฟิสิกส์โดยเนื้อหาครอบคลุมถึง เวกเตอร์ เมทริกซ์ คิวเตอร์เนียน อนุพันธ์ ปฏิยานุพันธ์ สมการเชิงอนุพันธ์สามัญ และสมการเชิงอนุพันธ์ย่อย ซึ่งใช้ในการจัดเก็บคุณสมบัติทางฟิสิกส์ และการอธิบายกฎทางฟิสิกส์ด้วยสมการและอสมการ และนำมาใช้วิธีทางตัวเลขเพื่อที่จะเขียนเป็นโปรแกรมคอมพิวเตอร์เพื่อใช้ในการสร้างภาพเคลื่อนไหวเชิงฟิสิกส์ต่อไป โดยผู้สนใจสามารถหาศึกษาเพิ่มเติมได้จากเอกสารอ้างอิงต่างๆ

## บทที่ 2: ระบบอนุภาค

การจำลองทางฟิสิกส์แบบพื้นฐานที่สุดคือการจำลองระบบอนุภาค (Particle System) ซึ่งประกอบด้วยอนุภาค (Particle) จำนวนมาก ซึ่งอนุภาคจะแทนสิ่งที่มีขนาดเล็กเช่น ละอองน้ำ เกล็ดหิมะ สะเก็ดไฟ เศษหิน หรือกลุ่มของสสารที่ประมาณการว่าเคลื่อนที่ไปด้วยกันเช่นลูกไฟ กลุ่มควันขนาดเล็กเป็นต้น โดยแต่ละอนุภาคเคลื่อนที่เป็นอิสระต่อกันและมีคุณลักษณะ (Attribute) เหมือนกัน คุณลักษณะที่พบบ่อยในระบบอนุภาคคือ

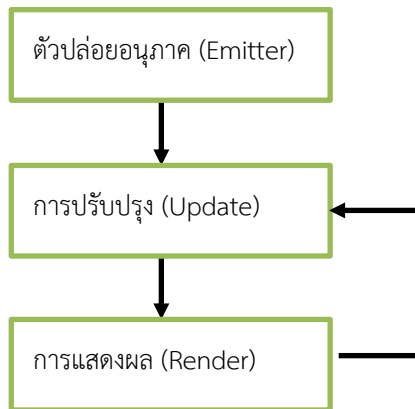
- ตำแหน่ง (Position) ซึ่งเป็นเวกเตอร์ 3 มิติ
- ความเร็ว (Velocity) ซึ่งเป็นเวกเตอร์ 3 มิติ
- เวลาชีวิต (Life Time) เป็นสเกลาร์ ใช้ระบุว่าอนุภาคนี้เหลือเวลาเท่าไรก่อนที่จะโดนลบ มักมีไว้เพื่อไม่ให้มีจำนวนอนุภาคที่จะต้องคำนวณมากเกินไป
- สี (Color) คือสีของอนุภาค
- ความทึบแสง (Opacity) ใช้ในการระบุความทึบแสงของอนุภาคเพื่อใช้ในการแสดงผล มักสัมพันธ์กับเวลาชีวิตโดยมากความทึบแสงจะค่อยๆลดลง (Fade Out) เมื่อเวลาชีวิตใกล้จะหมด
- ขนาด (Size) ระบุขนาดของอนุภาคที่จะใช้ในการแสดงผล

ระบบอนุภาคไม่จำเป็นต้องมีทุกลักษณะข้างบน และบางคุณลักษณะอาจจะสามารถคำนวณได้จากคุณลักษณะอื่น โดยไม่ต้องทำการเก็บไว้ต่างถ้าเช่น ความทึบแสง และขนาดอาจจะคำนวณได้จากเวลาชีวิตเป็นต้น ในหนังสือเล่มนี้จะกล่าวถึงอนุภาคใน 3 มิติโดยถ้าผู้อ่านต้องการใช้เพียง 2 มิติก็เพียงกำหนดให้ค่า  $z$  เป็น 0 เสมอ

ระบบอนุภาคมีการใช้งานครั้งแรกๆในภาพยนตร์เรื่อง Star Trek II: The Wrath of Kahn ในปี 1983 งานวิจัยที่กล่าวถึงระบบอนุภาคครั้งแรกก็ตีพิมพ์ในปีเดียวกัน [6] นับแต่นั้นเป็นต้นมาระบบอนุภาคถูกใช้อย่างแพร่หลายในการสร้างวิชวลเอฟเฟค (Visual Effect) ต่างๆด้วยคอมพิวเตอร์ ที่มาทดแทนสเปเชียลเอฟเฟค (Special Effect) ที่ใช้การถ่ายทำจริงของ ควัน ไฟ สะเก็ดไฟ ฝน หิมะ ละอองน้ำ และการระเบิดต่างๆ

## ขั้นตอนการทำงานของระบบอนุภาค

การทำงานของระบบอนุภาคประกอบด้วยขั้นตอนต่างๆ ดังภาพประกอบ 14



ภาพประกอบ 14: ขั้นตอนการทำงานของระบบอนุภาค

โดยตัวปล่อยอนุภาค (Emitter) จะผลิตอนุภาคออกมา และกำหนดค่าเริ่มต้นต่างๆของคุณลักษณะของอนุภาค จากนั้นอนุภาคแต่ละอันก็ต้องผ่านการถูกปรับปรุง (Update) ในแต่ละขั้นตอนเวลา (Time Step) เพื่อเปลี่ยนตำแหน่ง ความเร็ว และลักษณะอื่นๆ รวมถึงถูกลบเมื่อเวลาชีวิตหมดไป จากนั้นก็จะทำการแสดงผล (Render) เพื่อให้ผู้ใช้งานมองเห็นได้ โดยจะดูรายละเอียดของแต่ละขั้นตอนในลำดับถัดไป

### ตัวปล่อยอนุภาค

โดยทั่วไปแล้วตัวปล่อยอนุภาคจะทำหน้าที่สร้างอนุภาคออกมา โดยตำแหน่งเริ่มต้นของอนุภาคจะอยู่ในบริเวณที่กำหนดใกล้ๆกับตำแหน่งของตัวปล่อยอนุภาค และบริเวณที่จะสร้างอนุภาคอาจจะเป็นรูปร่างต่างๆ เช่น ลูกบาศก์ (Cube) กล่อง (Box) ทรงกลม (Sphere) ทรงกระบอก (Cylinder) หรือโคน (Cone) โดยตำแหน่งเริ่มต้นของ อนุภาคมักจะถูกสุ่มให้อยู่ในบริเวณดังกล่าว ซึ่งการสุ่มในคอมพิวเตอร์จะใช้ตัวสร้างตัวเลขสุ่มหลอก (Pseudo Random Number Generator)

ค่าเริ่มต้นของคุณลักษณะอื่นๆของอนุภาคก็จะถูกสุ่มให้อยู่ภายในช่วงที่กำหนดเช่นกัน เหตุที่ต้องอาศัยการสุ่มนั้น มาจากการที่จะพยายามไม่ให้อนุภาคมีระเบียบแบบแผนมากเกินไปซึ่งจะดูไม่สวยงาม โดยคุณลักษณะที่พบบ่อย มักจะถูกสุ่มดังนี้

1. ความเร็ว มักสุ่มให้มีค่าใกล้เคียงความเร็วเฉลี่ย (Average Velocity) ถ้าอนุภาคถูกยิงออกมา หรือ สุ่มทิศทางและขนาดถ้าอนุภาคแทนการระเบิด
2. เวลาชีวิต มักสุ่มให้มีค่าใกล้เคียงกับค่าเฉลี่ยค่าหนึ่ง
3. สี มักสุ่มให้เป็นสีใดสีหนึ่งจากกลุ่มของสี
4. ความทึบแสง มักสุ่มให้มีค่าใกล้เคียงกับค่าเฉลี่ยค่าหนึ่ง
5. ขนาด มักสุ่มให้มีค่าใกล้เคียงกับค่าเฉลี่ยค่าหนึ่ง

### การปรับปรุงอนุภาค

การเปลี่ยนแปลงค่าของคุณลักษณะของอนุภาคมักจะถูกปรับปรุงโดยใช้กฎทางฟิสิกส์ซึ่งกฎการเคลื่อนที่ที่ใช้อย่าง แพร่หลายคือกฎของนิวตัน [7] [8]

1. วัตถุที่หยุดนิ่งจะพยายามหยุดนิ่งอยู่กับที่ トラบที่ไม่มีแรงภายนอกมากกระทำ ส่วนวัตถุที่เคลื่อนที่ จะเคลื่อนที่เป็นเส้นตรงด้วยความเร็วคงที่ トラบที่ไม่มีแรงภายนอกมากกระทำเช่นกัน
2. ความเร่งของวัตถุจะแปรผันตามแรงที่กระทำต่อวัตถุ แต่จะแปรผกผันกับมวลของวัตถุ
3. แรงที่วัตถุที่หนึ่งกระทำต่อวัตถุที่สองย่อมเท่ากับแรงที่วัตถุที่สองกระทำต่อวัตถุที่หนึ่ง แต่ทิศทางตรงข้ามกัน

ซึ่งกฎข้อที่ 2 จะเป็นข้อที่ถูกนำมาใช้บ่อยที่สุดในการจำลองทางฟิสิกส์โดยเขียนเป็นสมการทางคณิตศาสตร์ได้ดังนี้

$$f = ma$$

$f$  คือแรงมีหน่วยเป็นนิวตัน (N),  $m$  คือมวลมีหน่วยเป็นกิโลกรัม (kg) และ  $a$  คือความเร่งมีหน่วยเป็นเมตรต่อวินาทีกำลังสอง ( $m/s^2$ ) ซึ่งความเร่ง  $a$  คืออนุพันธ์ของความเร็ว  $v$  ซึ่งมีหน่วยเป็นเมตรต่อวินาที ( $m/s$ ) และความเร็วเป็นอนุพันธ์ของการกระจัดหรือตำแหน่ง  $x$  ซึ่งมีหน่วยเป็นเมตรดังนั้นถ้ารู้ตำแหน่งเริ่มต้น ความเร็วเริ่มต้น และรู้แรงที่กระทำกับอนุภาค ณ เวลาต่างๆ ก็จะสามารถหาความเร็ว และเวลาใดๆ ได้เสมอกล่าวคือ

$$a(t) = \frac{f(t)}{m}$$

$$v(t_1) = v(t_0) + \int_{t=t_0}^{t=t_1} a(t) dt$$

$$x(t_1) = x(t_0) + \int_{t=t_0}^{t=t_1} v(t) dt$$

ซึ่งถ้าแรงที่กระทำเป็นค่าคงที่เช่นแรงดึงดูด (โดยประมาณ) แล้วจะสามารถหาสูตรของการคำนวณความเร็วและตำแหน่ง ณ เวลาใดๆ ได้ด้วยการหาปฏิยานุพันธ์ เช่นการเคลื่อนที่แบบโพรเจกไทล์ (Projectile) แต่โดยทั่วไปแล้วในการจำลองระบบอนุภาคแรงจะไม่เป็นค่าคงที่ จึงจำเป็นที่จะต้องใช้วิธีทางตัวเลขในการคำนวณ แรงที่กระทำกับอนุภาคมักจะประกอบด้วย แรงดึงดูด ( $mg$ ) แรงต้านอากาศและแรงจากภายนอกอื่นๆ( $f_{ext}$ ) ที่อาจขึ้นกับตำแหน่งและความเร็ว

$$f(x, v, t) = mg + f_{ext}(x, v, t)$$

ดังนั้นความเร่ง ณ เวลา  $t$  คือ

$$a(x, v, t) = g + \frac{f_{ext}(x, v, t)}{m}$$

โดยถ้ารู้ความเร็วและตำแหน่งของอนุภาค ณ เวลา  $t_0$  ( $v(t_0)$  และ  $x(t_0)$ ) แล้วต้องการหาความเร็ว และตำแหน่งของอนุภาค ณ เวลา  $t_0 + \Delta t$  ซึ่ง  $t_0$  และ  $\Delta t$  มีหน่วยเป็นวินาที จะสามารถใช้วิธีการออยเลอร์แบบไปข้างหน้า (Forward Euler Method) ในการประมาณค่าของ  $v(t_0 + \Delta t)$  และ  $x(t_0 + \Delta t)$  ได้ดังนี้

$$v(t_0 + \Delta t) = v(t_0) + \Delta t a(x(t_0), v(t_0), t_0)$$

$$x(t_0 + \Delta t) = x(t_0) + \Delta t v(t_0)$$

หลังจากได้ค่าของ  $v(t_0 + \Delta t)$  และ  $x(t_0 + \Delta t)$  แล้วก็สามารถใช้สูตรข้างต้นในการหาค่าของ  $v(t_0 + 2\Delta t)$  และ  $x(t_0 + 2\Delta t)$  ณ เวลาต่างๆในอนาคตได้ ซึ่งถ้า  $\Delta t$  มีขนาดเล็กพอแล้วก็จะได้ค่าที่มีความแม่นยำ แต่ทว่ายิ่ง  $\Delta t$  มีขนาดเล็กเท่าไร ก็ยิ่งจะต้องใช้จำนวนครั้งในการคำนวณมากขึ้นเพื่อที่จะหาค่าของ  $v(t_1)$  และ  $x(t_1)$  เนื่องจากจะต้องใช้การคำนวณ  $\frac{t_1 - t_0}{\Delta t}$  ครั้ง ซึ่งในทางปฏิบัติอาจจะต้องใช้เวลาในการคำนวณมาก จึงจำเป็นที่จะต้องใช้วิธีการประมาณที่แม่นยำกว่าวิธีการออยเลอร์แบบไปข้างหน้า ซึ่งวิธีหนึ่งที่ทำให้ได้ง่ายคือวิธีการออยเลอร์แบบกึ่งนัย (Semi-Implicit Euler Method) โดยการปรับเพียงเล็กน้อย

$$v(t_0 + \Delta t) = v(t_0) + \Delta t a(x(t_0), v(t_0), t_0)$$

$$x(t_0 + \Delta t) = x(t_0) + \Delta t v(t_0 + \Delta t)$$

และสามารถเพิ่มความแม่นยำของการคำนวณโดยใช้กฎจุดกลาง (Mid Point Rule)

$$v\left(t_0 + \frac{\Delta t}{2}\right) = v(t_0) + \frac{\Delta t}{2} a(x(t_0), v(t_0), t_0)$$

$$x\left(t_0 + \frac{\Delta t}{2}\right) = x(t_0) + \frac{\Delta t}{2} v\left(t_0 + \frac{\Delta t}{2}\right)$$

$$v(t_0 + \Delta t) = v(t_0) + \Delta t a\left(x\left(t_0 + \frac{\Delta t}{2}\right), v\left(t_0 + \frac{\Delta t}{2}\right), t_0 + \frac{\Delta t}{2}\right)$$

$$x(t_0 + \Delta t) = x(t_0) + \Delta t v\left(t_0 + \frac{\Delta t}{2}\right)$$

อีกวิธีหนึ่งที่นิยมใช้ในการปรับปรุงตำแหน่งก็คือการปฏิยานุพันธ์แบบเวอเลย์ (Verlet Integration) ซึ่งใช้ในกรณีที่ไม่เก็บความเร็วของอนุภาค แต่เก็บตำแหน่งของอนุภาคที่สองลำดับเวลาที่ติดกันแทน  $x(t_0 - \Delta t)$  และ  $x(t_0)$  เพื่อที่จะคำนวณ  $x(t_0 + \Delta t)$  โดยจะทำตามขั้นตอนดังนี้

$$\tilde{v} = \frac{x(t_0) - x(t_0 - \Delta t)}{\Delta t}$$

$$x(t_0 + \Delta t) = 2x(t_0) - x(t_0 - \Delta t) + \Delta t^2 a(x(t_0), \tilde{v}, t_0)$$

ซึ่งวิธีการประมาณการเหล่านี้แท้จริงแล้วคือการแก้สมการเชิงอนุพันธ์สามัญของ ความเร่ง ความเร็ว และตำแหน่งของอนุภาคนั้นเอง ซึ่งจริงๆ แล้วมีวิธีการอีกมากที่สามารถทำได้ โดยผู้ที่สนใจสามารถศึกษาเพิ่มเติมเกี่ยวกับวิธีการเชิงตัวเลขเหล่านี้รวมถึง การวิเคราะห์ การพิสูจน์ความแม่นยำ และการพิสูจน์ความถูกต้องได้จากหนังสือ Numerical Methods for Ordinary Differential Equations [9] ในทางปฏิบัติแล้วการจำลองระบบอนุภาคมักใช้วิธีการออยเลอร์แบบกึ่งนัย หรือกฎจุดกลาง หรือการปฏิยานุพันธ์แบบเวอเลย์ ในการคำนวณปรับปรุงความเร็วและตำแหน่ง

$f_{\text{ext}}(x, v, t)$  ซึ่งสามารถเปลี่ยนความเร็วของอนุภาคที่พบในทางปฏิบัตินั้นมีหลายอย่างด้วยกัน ที่พบบ่อยๆคือ

1. แรงแลม (Wind Force)

$$f_{\text{ext}}^{\text{wind}}(x, v, t) = f^{\text{wind}} \text{ มักมีค่าคงที่}$$

2. แรงดูด (Attractor Force) เป็นแรงที่ดูดอนุภาคให้เข้าสู่จุดจุดหนึ่ง

$$f_{\text{ext}}^{\text{attract}}(x, v, t) = \frac{-k^{\text{attract}}}{|x-x^{\text{attract}}|^2 + \epsilon} \frac{x-x^{\text{attract}}}{|x-x^{\text{attract}}|}$$

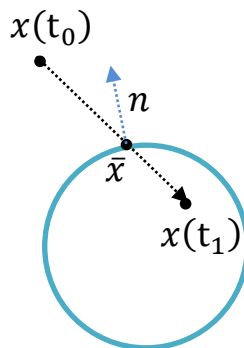
โดย  $k^{\text{attract}}$  และ  $x^{\text{attract}}$  คือค่าคงที่ของแรงดูด และจุดศูนย์กลางการดูด

3. แรงต้านอากาศ (Drag) หรือแรงหน่วง (Damping) เป็นแรงที่ทำให้อนุภาคเคลื่อนที่ช้าลง

$$f_{\text{ext}}^{\text{drag}}(x, v, t) = -k^{\text{drag}}v$$

โดย  $k^{\text{drag}}$  คือค่าคงที่ของแรงต้านอากาศ

นอกจากจะเคลื่อนที่เนื่องจากแรงต่างๆแล้ว อนุภาคยังชนกับสิ่งกีดขวาง (Collider) ต่างๆด้วย ซึ่งจะต้องปรับตำแหน่ง และความเร็วของอนุภาคให้เหมาะสมเช่น ถ้าตอนเริ่มของช่วงเวลาอนุภาคอยู่ที่ตำแหน่ง  $x(t_0)$  หลังจากการปรับตำแหน่งและความเร็วแล้วอนุภาคอยู่ที่ตำแหน่ง  $x(t_1)$  และมีความเร็ว  $v(t_1)$  แต่ทว่า ส่วนของเส้นตรง  $x(t_0)$  ไปยัง  $x(t_1)$  นั้นมีการผ่านสิ่งกีดขวางให้ตำแหน่ง  $\bar{x}$  เป็นตำแหน่งที่อนุภาคชนกับสิ่งกีดขวางครั้งแรก และให้  $n$  เป็นเวกเตอร์หน่วยที่ตั้งฉากกับผิวของสิ่งกีดขวาง ณ จุด  $\bar{x}$  ดังภาพประกอบ 15



ภาพประกอบ 15: อนุภาคชนกับสิ่งกีดขวาง

วิธีการปรับตำแหน่ง และความเร็วของอนุภาคเพื่อให้เกิดการชนขึ้น สามารถทำได้ตามขั้นตอนดังต่อไปนี้

1. คำนวณเวลาชน  $t_c \approx \frac{|\bar{x}-x_0|}{|x_1-x_0|} \Delta t + t_0$
2. ให้  $x(t_c) = \bar{x}$
3. ให้  $v(t_c) = -\epsilon(v(t_1) \cdot n)n + \mu(v(t_1) - (v(t_1) \cdot n)n)$

- โดยที่  $0 \leq \epsilon \leq 1$  คือค่าสัมประสิทธิ์การกระดอน (Restitution Coefficient) ซึ่งเป็นค่าที่ระบุว่าการชนสูญเสียพลังงานขนาดไหนโดยถ้าเป็น 0 คือการชนแบบไม่ยืดหยุ่น (Inelastic Collision) ซึ่งสูญเสียพลังงานมากที่สุด และถ้าเป็น 1 จะเป็นการชนแบบยืดหยุ่น (Elastic Collision) ซึ่งสูญเสียพลังงานน้อยที่สุด และ  $0 \leq \mu \leq 1$  คือค่าสัมประสิทธิ์แรงเสียดทาน
- ทำการปรับปรุงตำแหน่งและความเร็วจากเวลา  $t_c$  ไปยังเวลา  $t_1$  ด้วยวิธีปกติโดยใช้  $\Delta t = t_1 - t_c$

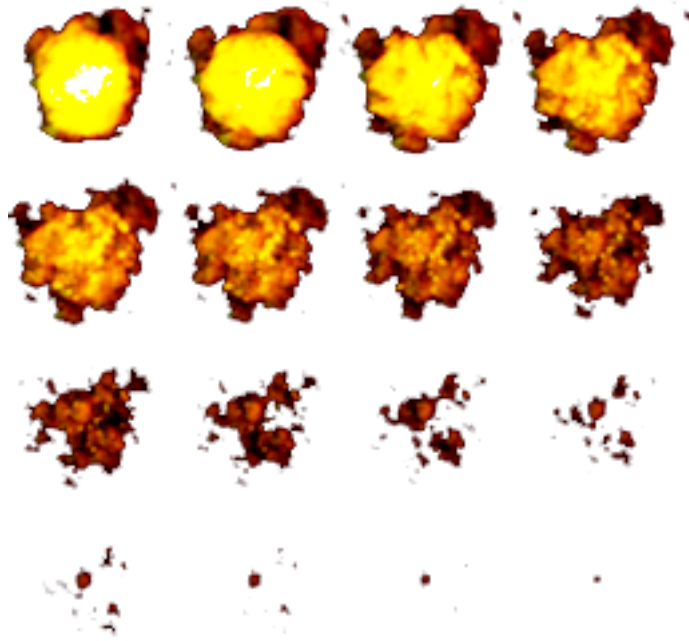
ถ้าใช้การปฏิยานุพันธ์แบบเวกเตอร์ ก็สามารถจัดการกับการชนได้โดยคำนวณความเร็ว  $v(t_1) = \frac{x(t_1) - x(t_0)}{\Delta t}$  แล้วทำตามขั้นตอนที่ 1-3 ด้านบน จากนั้นปรับให้  $x(t_c - \Delta t) = x(t_c) - \Delta t v(t_c)$  แล้วทำตามขั้นตอนที่ 4

การปรับปรุงคุณลักษณะอื่น ๆ มักทำได้โดยง่ายเช่น

- เวลาชีวิตของอนุภาค ทำได้โดยลบ  $\Delta t$  ออกจากเวลาที่เหลือถ้าเวลาชีวิตเหลือน้อยกว่าหรือเท่ากับ 0 ก็ให้ลบอนุภาคทิ้ง
- ความทึบแสงอาจจะค่อยๆ เพิ่มขึ้นเรื่อยๆ จนถึง 1 หรือค่อยๆ ลดลงเรื่อยๆ จนถึง 0 หรือไม่เปลี่ยนแปลงขึ้นกับลักษณะการใช้งาน

#### การแสดงผล

อนุภาคมักถูกแสดงผลด้วยการวาดรูปขนาดเล็กที่จุดศูนย์กลางอยู่ที่ตำแหน่งของอนุภาค เรียกรูปขนาดเล็กนี้ว่าสไปรต์ (Sprite) ซึ่งมักจะถูกเก็บเป็นรูปเล็กหลายรูปภายในรูปใหญ่ รวมเรียกว่าแผ่นสไปรต์ (Sprite Sheet) ตัวอย่างของแผ่นสไปรต์ที่แทนอนุภาคระเบิดแสดงในภาพประกอบ 16



ภาพประกอบ 16: ตัวอย่างแผ่นสไลด์ที่เก็บรูปที่ใช้แสดงผลอนุภาคระเบิด เครดิต: ©Explosion by Cuzco

<https://opengameart.org/content/explosion>

โดยส่วนใหญ่แล้วจะนำขนาด ความทึบแสง และสีของอนุภาคมาปรับการแสดงผลของสไลด์ด้วย บางครั้งอาจจะมี การนำความเร็วของอนุภาคมาทำการ หมุน ย่อ ยืด สไลด์เพื่อให้เห็นผลเหมือนกับมีการม้วนเนื่องจากการ เคลื่อนที่ (Motion Blur) เช่นสะเก็ดไฟที่จะเห็นเป็นแท่งที่วางแนวตามทิศทางการเคลื่อนที่ดังแสดงใน ภาพประกอบ 17



ภาพประกอบ 17: การแสดงผลอนุภาคด้วยการยึดตามทิศของความเร็วเพื่อจำลองการมั่วเนื่องจากการเคลื่อนที่

เครดิต: ©Trond Hille, Houdini Particles RnD - Weld Sparks v05

## บทสรุป

บทนี้กล่าวถึงการจำลองฟิสิกส์ด้วยระบบอนุภาคซึ่งสามารถทำได้ง่าย และสามารถใช้งานได้หลากหลาย โดยแต่ละอนุภาคเก็บคุณสมบัติทางฟิสิกส์ต่างๆ เช่นตำแหน่ง ความเร็ว ความเร่ง เวลาชีวิต เป็นต้น โดยอนุภาคจะถูกสร้างจากตัวปล่อยอนุภาคซึ่งจะต้องอาศัยการสุ่มเพื่อกำหนดค่าเริ่มต้นต่างๆ จากนั้น ในแต่ละขั้นตอนเวลาคุณสมบัติต่างๆของอนุภาคจะถูกปรับโดยอาศัยวิธีการการปฏิยานุพันธ์เชิงตัวเลข ตัวอย่างเช่นตำแหน่งจะถูกปรับปรุงโดยความเร็ว ซึ่งความเร็วจะถูกปรับปรุงโดยความเร่ง และความเร่งนั้นจะถูกคำนวณจากแรงต่างๆ ที่กระทำกับอนุภาค และจะต้องจัดการกับการชนด้วย หลังจากนั้นคุณสมบัติต่างๆของอนุภาคจะถูกนำมาใช้ในการแสดงผลต่อไป

### บทที่ 3: การจำลองวัตถุแข็ง

ในบทที่แล้วได้กล่าวถึงการจำลองทางฟิสิกส์ด้วยอนุภาคซึ่งแต่ละอนุภาคเป็นอิสระต่อกัน และไม่มีขนาดในเชิงฟิสิกส์ซึ่งใช้แทนวัตถุขนาดเล็กๆ ในบทนี้จะพิจารณาวัตถุแข็ง (Rigid Body) ซึ่งเป็นวัตถุที่มีขนาด น้ำหนัก และรูปร่างที่คงที่ไม่บวมไม่งอเมื่อเกิดการชนขึ้น วัตถุหลายอย่างที่พบเห็นในชีวิตประจำวันสามารถสมมติว่าเป็นวัตถุแข็งได้ในสถานการณ์ทั่วไปเช่น หิน โด๊ยะ แก้ว ไม้ อีฐ และอื่นๆ ในทางปฏิบัติแล้ววัตถุหลายอย่างที่ในโลกความจริงอาจจะนิ่ม เช่น รองเท้า ลูกบอล ผลไม้ ก็สามารถถูกจำลองด้วยวัตถุแข็งได้ในเกม หรือ ภาพยนตร์ โดยที่ยังดูสมจริงอยู่ถ้ามองจากระยะที่ไกลพอ

คุณลักษณะของวัตถุแข็งประกอบด้วย

1. ตำแหน่ง (Position) ในเนื้อที่โลก (World Space) ของวัตถุแข็งเป็นเวกเตอร์ 3 มิติ มักแทนด้วย  $\mathbf{x}(t)$  หน่วยเป็นเมตร (m)
2. การวางแนว (Orientation) ในเนื้อที่โลกเป็นเมทริกซ์หมุน  $3 \times 3$  มักแทนด้วย  $R(t)$
3. ความเร็ว (Velocity) เป็นเวกเตอร์ 3 มิติ มักแทนด้วย  $\mathbf{v}(t)$  หน่วยเป็นเมตรต่อวินาที (m/s)
4. ความเร็วเชิงมุม (Angular Velocity) เป็นเวกเตอร์ 3 มิติ มักแทนด้วย  $\boldsymbol{\omega}(t)$  ซึ่ง  $\frac{\boldsymbol{\omega}(t)}{|\boldsymbol{\omega}(t)|}$  คือแกนหมุน และ  $|\boldsymbol{\omega}(t)|$  คือความเร็วที่หมุนหน่วยเป็นเรเดียนต่อวินาที (rad/s)
5. มวล (Mass) ของวัตถุเป็นสเกลาร์หน่วยเป็นกิโลกรัม (kg) แทนด้วย  $m$
6. โมเมนต์ความเฉื่อย (Moment of Inertia) แทนด้วย  $I(t)$  เป็นเมทริกซ์ขนาด  $3 \times 3$  มีหน่วยเป็นกิโลกรัมเมตรกำลังสอง ( $\text{kgm}^2$ ) คือปริมาณที่ระบุความเฉื่อยของวัตถุในการที่จะรักษาความเร็วเชิงมุมปัจจุบันเอาไว้ ทำหน้าที่คล้ายมวลของความเร็วเชิงมุม โดยจะนิยามและเสนอวิธีคำนวณโมเมนต์ความเฉื่อยในภายหลัง

เนื่องจากวัตถุแข็งสามารถเคลื่อนที่และหมุนได้เท่านั้น ไม่สามารถเปลี่ยนรูปร่างได้ จึงสามารถกล่าวถึงรูปร่างของวัตถุแข็งในเนื้อที่ร่างกาย (Body Space) ซึ่งคงที่ไม่เปลี่ยนแปลงได้ และเพื่อให้การคำนวณต่างๆ ทำได้ง่ายขึ้น

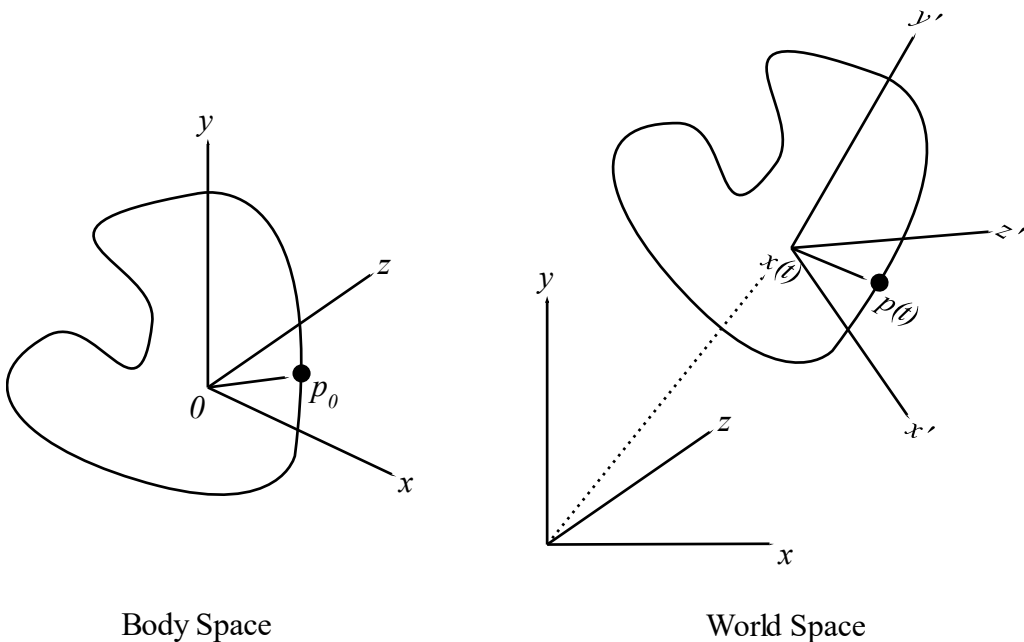
จะถือว่าจุดศูนย์กลางมวล (Center of Mass) นั้นอยู่ที่พิกัด  $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$  เสมอ โดยจุดศูนย์กลางมวลนั้นคือจุดที่จะใช้

พิจารณาการเคลื่อนที่ และการหมุนของวัตถุแข็งเป็นการเคลื่อนที่ของจุดศูนย์กลางมวล และการหมุนรอบจุดศูนย์กลางมวลได้ โดยจะกล่าวถึงรายละเอียดของวิธีการหาจุดศูนย์กลางมวลในเชิงคณิตศาสตร์ในภายหลัง

จากนิยามของจุดศูนย์กลางมวล ถ้ารู้ตำแหน่งของจุด  $p_0$  บนวัตถุแข็งในเนื้อที่ร่างกายแล้ว ก็จะสามารถคำนวณหาตำแหน่งของจุดนั้นในเนื้อที่โลกได้โดย

$$p(t) = R(t)p_0 + x(t)$$

ดังในภาพประกอบ 18



ภาพประกอบ 18: พิกัดในเนื้อที่ร่างกาย (Body Space) และเนื้อที่โลก (World Space) ดัดแปลงจาก [93]

และถ้ามีเวกเตอร์ที่อยู่ในเนื้อที่ร่างกาย  $s_0$  ก็จะสามารถที่จะคำนวณหาเวกเตอร์นั้นในเนื้อที่โลกได้ด้วย

$$s(t) = R(t)s_0$$

เวกเตอร์ในเนื้อที่ร่างกายที่น่าสนใจเป็นพิเศษก็คือแกนอ้างอิง  $x$  ( $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ ),  $y$  ( $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ ), และ  $z$  ( $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ ) ซึ่งถ้าเมทริกซ์

$$R(t) = \begin{bmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{bmatrix}$$

แกนอ้างอิง  $x, y, z$  ในเนื้อที่โลกจะเป็น  $\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}, \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}, \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix}$

### ระบบสมการเชิงอนุพันธ์สามัญของการเคลื่อนไหวของวัตถุแกร่ง

ความเร็วของจุดศูนย์กลางมวลของวัตถุแกร่งนั้น มีความสัมพันธ์กับตำแหน่งของจุดศูนย์กลางมวลของวัตถุแกร่ง เช่นเดียวกับของอนุภาคกล่าวคือ

$$\mathbf{x}'(t) = \mathbf{v}(t)$$

ส่วนความเร็วเชิงมุมนั้น มีความสัมพันธ์กับการวางแนวของวัตถุแกร่งดังนี้ [10]

$$\mathbf{R}'(t) = \boldsymbol{\omega}^\times(t)\mathbf{R}(t)$$

โดยที่ถ้า  $\boldsymbol{\omega}(t) = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$  แล้ว  $\boldsymbol{\omega}^\times(t) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$  โดยถ้านำเมทริกซ์

$\boldsymbol{\omega}^\times(t)$  มาคูณกับเวกเตอร์ใดก็ตาม จะได้ผลเหมือนกับการนำ  $\boldsymbol{\omega}(t)$  มาคูณเชิงเวกเตอร์กับเวกเตอร์นั้น

ถ้าวัตถุแกร่งอยู่ภายในปริมาตร  $\Omega_0$  และ ณ ตำแหน่ง  $q_0$  มีความหนาแน่น  $\rho(q_0)$  กิโลกรัมต่อลูกบาศก์เมตร ( $\text{kg/m}^3$ ) จะสามารถคำนวณมวลของวัตถุแกร่ง ได้โดยใช้การปริพันธ์ภายในปริมาตรดังนี้

$$m = \iiint_{q_0 \in \Omega_0} \rho(q_0) dV$$

และนิยามจุดศูนย์กลางมวลของวัตถุแกร่งว่าอยู่ที่

$$\frac{\iiint_{q_0 \in \Omega_0} \rho(q_0) q_0 dV}{m}$$

ซึ่งการที่ถือว่าจุดศูนย์กลางมวลอยู่ที่  $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$  นั้นแท้จริงแล้วคือการระบุให้  $\frac{\iiint_{q_0 \in \Omega_0} \rho(q_0) q_0 dV}{m} = \mathbf{0}$  ในเนื้อที่

ร่างกายนั่นเอง ซึ่งจะทำให้จุดศูนย์กลางมวลของวัตถุแกร่งในเนื้อที่โลกนั้นอยู่ที่  $\mathbf{x}(t)$  เสมอ เนื่องจากจุดศูนย์กลางมวลของวัตถุแกร่งในเนื้อที่โลกคือ

$$\begin{aligned} & \frac{\iiint_{q_0 \in \Omega_0} \rho(q_0)(x(t) + R(t)q_0) dV}{m} \\ &= \frac{\iiint_{q_0 \in \Omega_0} \rho(q_0) dV}{m} x(t) + R(t) \frac{\iiint_{q_0 \in \Omega_0} \rho(q_0)q_0 dV}{m} \\ &= \frac{m}{m} x(t) + R(t) \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = x(t) \end{aligned}$$

นั่นเอง

โมเมนตัม (Momentum) ของอนุภาคนั้นนิยามโดย  $p(t) = mv(t)$  ซึ่งโมเมนตัมของวัตถุแกว่งมีค่าเท่ากับ โมเมนตัมนั้นสัมพันธ์กับแรงที่กระทำกับวัตถุด้วยความสัมพันธ์  $p'(t) = f(t)$  โดยที่  $f(t)$  คือแรงที่กระทำกับวัตถุ ณ เวลา  $t$

โมเมนตัมเชิงมุม (Angular Momentum) ของวัตถุแกว่งนั้นมีนิยามคล้ายๆกัน โดย  $l(t) = I(t)\omega(t)$  และ โมเมนตัมเชิงมุมนั้นสัมพันธ์กับแรงบิด (Torque)  $\tau(t)$  ด้วยความสัมพันธ์  $l'(t) = \tau(t)$  [10]

โดยถ้าวัตถุมีแรง (ในเนื้อที่โลก) กระทำที่ตำแหน่ง  $q_0$  (ในเนื้อที่โลก) เท่ากับ  $f(q_0, t)$  แล้ว  $\tau(t)$  จะคำนวณได้โดย

$$\tau(t) = \iiint_{q_0 \in \Omega_0} (R(t)q_0) \times f(q_0, t) dV \quad \text{โดยที่} \quad F(t) = \iiint_{q_0 \in \Omega_0} f(q_0, t) dV$$

$I(t)$  นั้นสามารถคำนวณได้จากนิยาม

$$I(t) = \iiint_{q_0 \in \Omega_0} \rho(q_0) \begin{bmatrix} q_y^2 + q_z^2 & -q_x q_y & -q_x q_z \\ -q_x q_y & q_x^2 + q_z^2 & -q_y q_z \\ -q_x q_z & -q_y q_z & q_x^2 + q_y^2 \end{bmatrix} dV$$

โดยที่  $q = R(t)q_0$

ซึ่งถ้านิยาม

$$I_0 = \iiint_{q_0 \in \Omega_0} \rho(q_0) \begin{bmatrix} q_{0y}^2 + q_{0z}^2 & -q_{0x}q_{0y} & -q_{0x}q_{0z} \\ -q_{0x}q_{0y} & q_{0x}^2 + q_{0z}^2 & -q_{0y}q_{0z} \\ -q_{0x}q_{0z} & -q_{0y}q_{0z} & q_{0x}^2 + q_{0y}^2 \end{bmatrix} dV$$

ซึ่งไม่ขึ้นกับเวลาแล้ว ก็จะสามารถคำนวณหา  $I(t) = R(t)I_0R^T(t)$  [10]

ตำแหน่ง การวางแนว ความเร็ว และความเร็วเชิงมุม สามารถนำมาเขียนรวมกันเป็นสถานะ (State) ของวัตถุ แกร่งได้เป็น

$$Y(t) = \begin{bmatrix} x(t) \\ R(t) \\ v(t) \\ \omega(t) \end{bmatrix}$$

และสามารถนำสมการอนุพันธ์มาเขียนรวมกันได้เป็น

$$Y'(t) = \begin{bmatrix} x(t) \\ R(t) \\ v(t) \\ \omega(t) \end{bmatrix}' = \begin{bmatrix} v(t) \\ \omega^\times(t)R(t) \\ \frac{1}{m}f(t) \\ I^{-1}\tau(t) \end{bmatrix}$$

โดยที่  $I(t) = R(t)I_0R^T(t)$  และสามารถใช้วิธีการออยเลอร์แบบไปข้างหน้า หรือ วิธีการออยเลอร์แบบกึ่งนัย หรือ กฎจุดกลางในการที่จะคำนวณ  $Y(t_0 + \Delta t)$  จาก  $Y(t_0)$  ได้อย่างตรงไปตรงมา แต่ก็จะมีปัญหาหนึ่งคือ เมื่อทำการปฏิยานุพันธ์ด้วยวิธีตัวเลขไปหลายครั้ง  $R(t)$  มักจะไม่เป็นเมทริกซ์หมุนอีกต่อไป เนื่องจากความผิดพลาดทางตัวเลข (Numerical Error) ซึ่งถ้านำ  $R(t)$  มาใช้ในการแสดงผลจะทำให้วัตถุแกร่งนั้นบิดเบี้ยว ซึ่งวิธีแก้ที่ทำได้ก็คือปรับให้เมทริกซ์  $R(t)$  นั้นเป็นเมทริกซ์หมุนในแต่ละขั้นตอนเวลาด้วยวิธีที่อธิบายในงานวิจัย [11] ซึ่งเป็นการหาเมทริกซ์หมุนที่ใกล้ที่สุดกับเมทริกซ์ใดๆ

การทำให้  $R(t)$  เป็นเมทริกซ์หมุนนั้นใช้เวลาค่อนข้างมากและซับซ้อน เราจึงสามารถใช้ควอเตอร์เนียน  $q(t)$  เพื่อเก็บการวางแนวแทนที่เมทริกซ์  $R(t)$  และแทนที่สมการอนุพันธ์  $R'(t) = \omega^\times(t)R(t)$  ด้วย

$$q'(t) = \frac{1}{2} \omega^p(t)q(t)$$

[10] โดย  $\omega^p(t)$  คือควอเตอร์เนียนบริสุทธิ์ของ  $\omega(t)$  และเมื่อต้องใช้  $R(t)$  ในการคำนวณใดๆ ก็หาค่าของ  $R(t)$  ด้วยการแปลงจากควอเตอร์เนียน  $q(t)$  เป็นเมทริกซ์ และในทุกๆ ขั้นตอนเวลาจะสามารถปรับให้  $q(t)$  เป็นควอเตอร์เนียนที่แทนการหมุนได้ด้วยการทำงานให้  $q(t)$  เป็นควอเตอร์เนียนหน่วยด้วยการ

$$q(t) \leftarrow \frac{q(t)}{|q(t)|}$$

ณ ตอนนี้สามารถจำลองการเคลื่อนที่ของวัตถุแรงแรงที่ไม่มีการชนกัน (Collision) ไม่มีการสัมผัสกัน (Contact) และไม่มีข้อจำกัด (Constraint) ต่อกันเช่นเชื่อมกันด้วยข้อต่อ (Joint) ได้แล้ว ในลำดับถัดไปจะกล่าวถึงวิธีในการจำลองวัตถุแรงแรงที่มีการชนกัน การสัมผัสกัน และมีข้อจำกัดต่างๆ

การชน, การสัมผัส และข้อจำกัดต่างๆ

ในการจำลองอนุภาคนั้น จะสามารถจัดการกับการชนระหว่างอนุภาคกับสิ่งกีดขวางด้วยการหาเวลาที่เกิดการชนขึ้น แล้วทำการปรับตำแหน่งและความเร็ว จากนั้นก็ทำการปฏิยานุพันธ์ในช่วงเวลาที่เหลือ ซึ่งสามารถทำได้ง่ายและไม่ใช้เวลาในการคำนวณมากเนื่องจากแต่ละอนุภาคนั้นเป็นอิสระต่อกัน แต่สำหรับวัตถุแรงแรงที่อาจมีการชนและสัมผัสกันนั้น โดยทั่วไปแล้วถ้าหากทำการหาเวลาชน แก๊ซ และปฏิยานุพันธ์ต่อนั้น จะทำให้การคำนวณใช้เวลามากเนื่องจากในแต่ละขั้นตอนเวลานั้น อาจเกิดการชนจำนวนมาก และการแก้ไขการชน ก็อาจทำให้เกิดการชนใหม่ขึ้นมาก็ได้

ในทางปฏิบัติแล้วจึงมักใช้วิธีอื่นในการจัดการกับการชน ซึ่งจริงๆ แล้วการชนเป็นเพียงหนึ่งในวิธีที่วัตถุแรงแรงมีปฏิสัมพันธ์ต่อกัน (Interaction) เท่านั้น นอกเหนือจากการชนแล้ว วัตถุแรงแรงยังสามารถสัมผัสหรือแตะกัน หรืออาจจะมีข้อจำกัดต่อกันเช่นเชื่อมกันด้วยข้อต่อได้ ซึ่งทำให้การจำลองวัตถุแรงแรงนั้นมีความหลากหลาย และใช้ประโยชน์ได้อย่างกว้างขวาง

ถ้ามองโดยผิวเผินแล้ว การชน การสัมผัส และข้อจำกัดระหว่างวัตถุแฉ่ง อาจดูไม่มีความสัมพันธ์กันเท่าไรหรอก แต่แท้จริงแล้วสามารถทำการจำลองสิ่งเหล่านี้ด้วยวิธีการแบบเดียวกันได้ โดยมองการชนและการสัมผัสเป็นข้อจำกัดแบบหนึ่ง วิธีที่จะใช้ในการจำลองวัตถุแฉ่งที่มีปฏิสัมพันธ์ต่อกัน คือวิธีการตามลำดับ (Sequential Impulses) [12]

วิธีการตามลำดับนั้นไม่รับประกันว่าวัตถุแฉ่งจะไม่ทับกัน (Overlap, Intersect) หรือไม่แทรกกัน (Penetrate) แต่จะพยายามให้การแทรกกัน (Penetration) นั้นน้อยที่สุดเท่าที่จะทำได้ และการปฏิยานุพันธ์จะทำงานทั้งช่วงเวลาเท่านั้น ไม่มีการแบ่งแยกย่อยช่วงเวลาเพื่อจัดการกับเหตุการณ์การชนแต่ละอันเป็นพิเศษ จึงทำให้ประสิทธิภาพของวิธีการตามลำดับนั้นดี และได้ผลที่ยอมรับได้ในทางปฏิบัติ

ในแต่ละขั้นตอนเวลานั้นวิธีการตามลำดับประกอบด้วยขั้นตอนหลัก 3 ขั้นตอนคือ

1. ทำการรวบรวมแรงต่างๆ และทำการปฏิยานุพันธ์ความเร็ว และความเร็วเชิงมุม เพื่อให้ได้ความเร็ว และความเร็วเชิงมุมชั่วคราว (Tentative)
2. ทำการปรับแก้ความเร็ว และความเร็วเชิงมุมของวัตถุแฉ่งต่างๆ ด้วยแรงคลโดยทำไปที่ละข้อจำกัดและทำการหลายๆรอบจนกว่าจะเป็นไปตามข้อจำกัดทุกข้อ หรือจนกว่าจะพิจารณาไปครบจำนวนรอบที่กำหนด
3. ทำการปฏิยานุพันธ์ตำแหน่ง และการวางแนวโดยใช้ความเร็วและความเร็วเชิงมุมที่ปรับแก้แล้ว

ก่อนที่จะอธิบายแต่ละขั้นตอนของวิธีการตามลำดับอย่างละเอียดนั้น จะต้องทำความเข้าใจเกี่ยวกับข้อจำกัดก่อน

ข้อจำกัดในความหมายที่พิจารณาอยู่ในที่นี้คือสเกลาร์ฟังก์ชัน (Scalar Function)  $c(x)$  ของบางส่วนของสถานะของวัตถุแฉ่งต่างๆ ที่จำลองอยู่ โดยสามารถจัดกลุ่มของข้อจำกัดได้เป็นข้อจำกัดที่เกี่ยวข้องกับตำแหน่ง (รวมถึงการวางแนว) และข้อจำกัดที่เกี่ยวข้องกับความเร็ว (รวมถึงความเร็วเชิงมุม) โดยในเบื้องต้นจะพิจารณาข้อจำกัดที่เกี่ยวข้องกับตำแหน่งก่อนเช่น

$$c(x) = 0$$

ตัวอย่างเช่น การระบุว่าตำแหน่งจะต้องอยู่บนเส้นโค้ง (Curve) เส้นหนึ่งทีนิยามด้วยฟังก์ชันโดยนัย (Implicit

Function) เช่น  $c\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = x^2 + y^2 + z^2 - 5^2 = 0$  คือข้อจำกัดที่  $\mathbf{x}$  จะต้องอยู่บนทรงกลมที่

จุดศูนย์กลางอยู่ที่จุด  $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$  รัศมี 5

ถ้า  $c(\mathbf{x}) = 0$  ตลอดเวลาแล้ว  $\dot{c}(\mathbf{x}) = 0$  ด้วย (เพราะ  $c(\mathbf{x})$  ไม่เปลี่ยนค่า) โดยที่  $\dot{c}(\mathbf{x})$  คืออนุพันธ์

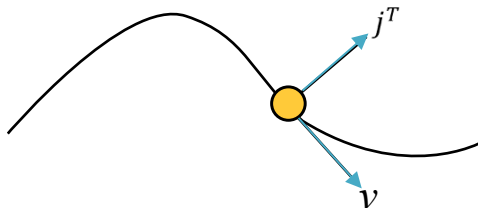
ของ  $c$  เมื่อเทียบกับเวลา และจากกฎลูกโซ่สามารถเขียน  $\dot{c}(\mathbf{x}) = \frac{dc}{dt} = \frac{dc}{dx} \frac{dx}{dt} = \frac{dc}{dx} \mathbf{v} = j\mathbf{v}$  โดยที่

$j = \frac{dc}{dx}$  เรียกว่าจาโคเบียน (Jacobian) ซึ่งเป็นเวกเตอร์แถว 3 มิติเช่นถ้า  $\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$  แล้ว  $j = \frac{dc}{dx} =$

$$\left[ \frac{dc}{dx} \quad \frac{dc}{dy} \quad \frac{dc}{dz} \right]$$

สามารถมอง  $j^T$  เป็นเวกเตอร์ 3 มิติได้และ  $j^T$  นั้นตั้งฉากกับ  $\mathbf{v}$  เสมอ

ดังแสดงในภาพประกอบ 19



ภาพประกอบ 19: ข้อจำกัดที่ทำให้อนุภาคอยู่บนเส้นโค้งเสมอ โดย  $j^T$  จะตั้งฉากกับ  $\mathbf{v}$  เสมอ ดัดแปลงจาก [12]

โดยการที่วัตถุแรงแรงจะเคลื่อนที่ตามข้อจำกัดนี้ได้จะต้องมีแรง  $f_c$  ที่กระทำกับวัตถุแรงแรงและจากหลักการงานเสมือน (Principle of Virtual Work) ที่กล่าวว่าแรงจากข้อจำกัดจะต้องให้ไม่ก่อให้เกิดงาน (Work) หรือในทางคณิตศาสตร์คือ

$$f_c \cdot \mathbf{v} = 0$$

ซึ่งเนื่องจากหลักการงานเสมือนต้องเป็นจริงไม่ว่า  $\mathbf{v}$  จะเป็นเท่าไรก็ตาม (ตราบใดที่ยังถูกต้องตามข้อจำกัด

$c(\mathbf{x}) = 0$ )  $f_c$  จึงจะต้องอยู่ในรูป  $f_c = j^T \lambda$  โดยที่  $\lambda$  เป็นสเกลาร์เนื่องจาก

$$f_c \cdot \mathbf{v} = (j^T \lambda) \cdot \mathbf{v} = \lambda(j^T \cdot \mathbf{v}) = 0$$

และถ้ามีข้อจำกัด  $c_1, c_2, \dots, c_k$  ระหว่างตำแหน่ง  $x_1, x_2, \dots, x_n$  ก็สามารถนำมาเขียนเป็นเวกเตอร์ของข้อจำกัดได้

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad c(x) = \begin{bmatrix} c_1(x) \\ c_2(x) \\ \vdots \\ c_k(x) \end{bmatrix}$$

โดยที่  $\dot{c} = Jv$  และ  $f_c = J^T \lambda$

และ

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \quad \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_k \end{bmatrix} \quad J = \begin{bmatrix} \frac{dc_1}{dx_1} & \frac{dc_1}{dx_2} & \dots & \frac{dc_1}{dx_n} \\ \frac{dc_2}{dx_1} & \frac{dc_2}{dx_2} & \dots & \frac{dc_2}{dx_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dc_k}{dx_1} & \frac{dc_k}{dx_2} & \dots & \frac{dc_k}{dx_n} \end{bmatrix}$$

จะเห็นได้ว่า  $J$  ทำหน้าที่แปลงจากความเร็วในระบบของตำแหน่ง ( $v$ ) เป็นความเร็วในระบบของข้อจำกัด ( $\dot{c}$ )

และ  $J^T$  ทำหน้าที่แปลงจากแรงในระบบของข้อจำกัด ( $\lambda$ ) เป็นแรงในระบบของตำแหน่ง ( $F_c$ )

ตัวอย่างของข้อจำกัดเช่น ถ้าในระบบมีวัตถุแกร่ง 2 ชิ้นที่มีตำแหน่ง  $x_1(t)$  และ  $x_2(t)$  และกำหนดว่า

ข้อจำกัดคือระยะทางระหว่างวัตถุแกร่ง 2 ชิ้นนี้จะต้องเป็น 1 เสมอ จะสามารถเขียนข้อจำกัดได้เป็น

$$c(x) = c(x_1, x_2) = ||x_1 - x_2|| - 1$$

$$\dot{c} = \frac{d(||x_1 - x_2|| - 1)}{dt}$$

$$\dot{c} = \frac{d(\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z - z_2)^2} - 1)}{dt}$$

$$\dot{c} = \frac{1}{2||x_1 - x_2||} \frac{d((x_1 - x_2)^2 + (y_1 - y_2)^2 + (z - z_2)^2)}{dt}$$

$$\dot{c} = \frac{((x_1 - x_2)(v_{1x} - v_{2x}) + (y_1 - y_2)(v_{1y} - v_{2y}) + (z_1 - z_2)(v_{1z} - v_{2z}))}{\|x_1 - x_2\|}$$

$$\dot{c} = \frac{1}{\|x_1 - x_2\|} \begin{bmatrix} (x_1 - x_2) & (y_1 - y_2) & (z_1 - z_2) & -(x_1 - x_2) & -(y_1 - y_2) & -(z_1 - z_2) \end{bmatrix} \begin{bmatrix} v_{1x} \\ v_{1y} \\ v_{1z} \\ v_{2x} \\ v_{2y} \\ v_{2z} \end{bmatrix}$$

$$\dot{c} = \frac{1}{\|x_1 - x_2\|} \begin{bmatrix} (x_1 - x_2)^T & -(x_1 - x_2)^T \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$\text{จะเห็นได้ว่า } j = \frac{1}{\|x_1 - x_2\|} \begin{bmatrix} (x_1 - x_2)^T & -(x_1 - x_2)^T \end{bmatrix}$$

โดยทั่วไปแล้ว  $c$  อาจไม่ได้เป็นแต่เพียงข้อจำกัดระหว่างตำแหน่ง แต่อาจจะเป็นข้อจำกัดระหว่างการวางแนวด้วยก็ได้ ซึ่งก็จะทำให้ความเร็วเชิงมุมนั้นปรากฏอยู่ใน  $\dot{c}$  ด้วย และ  $c$  ก็อาจจะเป็นฟังก์ชันของเวลาได้ ยกตัวอย่างเช่นมอเตอร์ที่ระบุให้องศาในการหมุนเพิ่มขึ้นเรื่อยๆ ซึ่ง  $c$  จะอยู่ในรูป  $c(x, t)$  และสามารถเขียนให้  $\dot{c}$  อยู่ในรูป  $\dot{c} = jv + b(t)$  โดยที่  $b(t)$  เป็นฟังก์ชันของเวลาได้เสมอ

ข้อจำกัดอาจจะไม่จำเป็นต้องอยู่ในรูปของสมการแต่อาจจะอยู่ในรูปของอสมการก็ได้เช่น

$$c(x, t) \geq 0$$

ซึ่งก็จะสามารถทำให้ข้อจำกัดนี้เป็นจริงได้ด้วยการตรวจสอบว่า  $c(x, t) \leq 0$  หรือไม่ ถ้าใช่ก็ใส่ข้อจำกัด

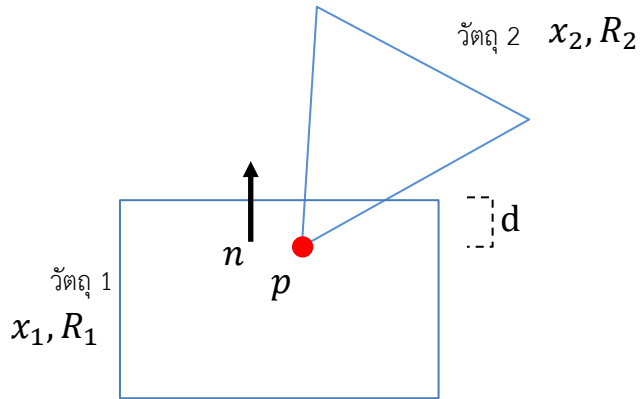
$c(x, t) \geq 0$  เข้าไปในการติดตามลำดับ แต่ถ้าไม่ใช่ก็ข้ามข้อจำกัดนี้ไป

นอกจากนี้ยังสามารถกำหนดขอบเขตของค่าของ  $\lambda$  ได้เช่น  $0 \leq \lambda \leq \infty$  ซึ่งจะต้องใช้ในการสร้างข้อจำกัดของการชนกันของวัตถุแรงที่ว่าแรงที่กระทำให้วัตถุแรงแสองอันไม่ซ้อนทับกันนั้นจะต้องเป็นแรงผลักเท่านั้น

(ห้ามดูด) หรือต้องการกำหนดให้แรงที่ข้อจำกัดสามารถกระทำได้นั้นไม่เกินที่  $\lambda_k$  นิวตัน ( $-\lambda_k \leq \lambda \leq \lambda_k$ )

ซึ่งวิธีที่ทำให้ข้อกำหนดนี้เป็นจริง ก็คือเพียงปรับค่าของ  $\lambda$  ให้อยู่ในช่วงที่กำหนดถ้าค่าของ  $\lambda$  ที่แก้ออกมาได้มันอยู่นอกช่วง โดยวิธีนี้เรียกว่าการยึด (Clamping)

ข้อจำกัดที่สำคัญมากในการจำลองวัตถุแกร่งก็คือข้อจำกัดที่ทำให้วัตถุไม่ซ้อนทับกัน โดยจะต้องสร้างข้อจำกัดของการชน/สัมผัสระหว่างทุกๆ คู่ของวัตถุต่างๆ ที่มีโอกาสชนกันขึ้นดังแสดงในภาพประกอบ 20



ภาพประกอบ 20: ข้อจำกัดที่ทำให้วัตถุไม่ซ้อนทับกัน

ให้จุด  $p$  คือจุดที่วัตถุ 1 และ 2 โกลักันมากที่สุด (ถ้ายังไม่ซ้อนทับกัน) หรือซ้อนทับกันมากที่สุด (ถ้าซ้อนทับกันแล้ว), พิกัดในเนื้อที่ร่างกายของจุด  $p$  ในวัตถุที่ 1 คือ  $p_1$  และพิกัดในเนื้อที่ร่างกายของจุด  $p$  ในวัตถุที่ 2 คือ  $p_2$  และเวกเตอร์หน่วยตั้งฉากกับพื้นผิวที่ชี้จากวัตถุ 1 ไปยังวัตถุ 2 คือ  $n$  และให้ระยะทางระหว่างวัตถุ 1 และวัตถุ 2 ตามทิศทาง  $n$  คือ  $d$  แล้ว ข้อจำกัดที่ทำให้วัตถุ 1 และ 2 ไม่ซ้อนทับกันคือ

$$\begin{aligned} c(x_1, R_1, x_2, R_2) &= d = (x_2 + R_2 p_2 - x_1 - R_1 p_1) \cdot n \geq 0 \\ \dot{c} &= \dot{d} = (v_2 + \omega_2 \times (R_2 p_2) - v_1 - \omega_1 \times (R_1 p_1)) \cdot n \\ &= (v_2 + \omega_2 \times (p - x_2) - v_1 - \omega_1 \times (p - x_1)) \cdot n \\ &= [-n \quad -(p - x_1) \times n \quad n \quad (p - x_2) \times n] \begin{bmatrix} v_1 \\ \omega_1 \\ v_2 \\ \omega_2 \end{bmatrix} \end{aligned}$$

ข้อจำกัดอันนี้จะทำการปรับให้วัตถุที่ชนจนซ้อนทับกันกลับมาแตะกันพอดี ซึ่งคือการชนแบบไม่ยืดหยุ่น ซึ่งถ้าต้องการให้การชนนั้นยืดหยุ่น จะต้องทำการปรับ  $\dot{c}$  จาก

$$\dot{c} = v_n$$

เมื่อ  $v_n$  คือความเร็วสัมพัทธ์ของจุด  $p$  ของวัตถุ 1 และ 2 ในทิศทาง  $n$  คำนวณจากความเร็วสัมพัทธ์  $v_{rel}$

$$v_{rel} = (v_2 + \omega_2 \times (p - x_2) - v_1 - \omega_1 \times (p - x_1))$$

$$v_n = v_{rel} \cdot n$$

ให้เป็น

$$\dot{c} = v_n + \varepsilon v_n^-$$

โดยที่  $v_n^-$  คือ  $v_n$  ที่คำนวณค่าก่อนจะเริ่มทำการปรับ และ  $0 \leq \varepsilon \leq 1$  คือค่าสัมประสิทธิ์การกระดอนซึ่งเมื่อทำการลดตามลำดับแล้วจะได้แรง  $0 \leq \lambda_n \leq \infty$

การจำลองแรงเสียดทานนั้นทำได้โดยการเลือกเวกเตอร์ที่ตั้งฉากกับ  $n$  มาสองเวกเตอร์ และสองเวกเตอร์นั้นตั้งฉากกันเองด้วย  $t_1, t_2$  แล้วเพิ่มข้อจำกัดให้การกระจัดสัมพัทธ์ในทิศ  $t_1, t_2$  เป็น 0 แล้วจำกัดขนาดของแรงไม่ให้เกิน  $\mu\lambda_n$  ซึ่ง  $\mu$  คือสัมประสิทธิ์แรงเสียดทานดังนี้

$$\begin{aligned} \dot{c}_{t_1} &= v_{rel} \cdot t_1 & \dot{c}_{t_2} &= v_{rel} \cdot t_2 \\ -\mu\lambda_n &\leq \lambda_{t_1} \leq \mu\lambda_n & -\mu\lambda_n &\leq \lambda_{t_2} \leq \mu\lambda_n \end{aligned}$$

จะสังเกตได้ว่าจะประมาณการแรงเสียดทานไม่ให้ขนาดเกิน  $\mu\lambda_n$  ในทิศ  $t_1, t_2$  เท่านั้นซึ่งอาจทำให้แรงเสียดทานรวมจริงๆแล้วอาจจะมีได้มากถึง  $\sqrt{(\mu\lambda_n)^2 + (\mu\lambda_n)^2} = \sqrt{2}\mu\lambda_n$  ซึ่งคือการประมาณการโคจรของแรงเสียดทานด้วยพีระมิดฐานสี่เหลี่ยมนั่นเอง ถ้าต้องการจำกัดแรงเสียดทานให้ใกล้เคียงกับ  $\mu\lambda_n$  มากขึ้นก็จะต้องใช้เวกเตอร์ที่ตั้งฉากกับ  $n$  จำนวนมากขึ้นซึ่งถ้าใช้  $k$  เวกเตอร์ที่มุมห่างกัน  $\frac{180}{k}$  องศาจะทำให้เป็นการประมาณการโคจรของแรงเสียดทานด้วยพีระมิดฐาน  $2k$  เหลี่ยม แต่ก็จะทำให้การคำนวณใช้เวลามากขึ้น เนื่องจากมีจำนวนของข้อจำกัดมากขึ้นด้วย

## การลดตามลำดับ

ขั้นตอนแรกของการลดตามลำดับคือการคำนวณความเร็ว และความเร็วเชิงมุมชั่วคราวซึ่งสำหรับระบบที่มีวัตถุแกร่งอยู่  $n$  ชิ้น ณ เวลา  $t$  ที่มีตำแหน่ง  $(x_i)$  ความเร็ว  $(v_i)$  การวางแนว  $(R_i)$  ความเร็วเชิงมุม  $(\omega_i)$  มวล  $(m_i)$  และโมเมนต์ความเฉื่อยในระบบพิกัดโลก  $(I_i)$  สำหรับ  $1 \leq i \leq n$  และขนาดของขั้นตอนเวลา  $\Delta t$  ทำได้โดยพิจารณาจากกฎข้อที่ 2 ของนิวตันโดย

$$\bar{v} = v + \Delta t M^{-1} f^{ext}$$

$$\text{โดยที่ } \mathbf{v} = \begin{bmatrix} v_1 \\ \omega_1 \\ v_2 \\ \omega_2 \\ \vdots \\ v_n \\ \omega_n \end{bmatrix} \quad M = \begin{bmatrix} m_1 E^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_2 E^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m_n E^3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_n \end{bmatrix} \quad \mathbf{f}^{\text{ext}} = \begin{bmatrix} f_1^{\text{ext}} \\ \tau_1^{\text{ext}} \\ f_2^{\text{ext}} \\ \tau_2^{\text{ext}} \\ \vdots \\ f_n^{\text{ext}} \\ \tau_n^{\text{ext}} \end{bmatrix}$$

เมื่อ  $E^3$  คือเมทริกซ์เอกลักษณะขนาด  $3 \times 3$ ,  $f_i^{\text{ext}}$  และ  $\tau_i^{\text{ext}}$  คือแรงและแรงบิดภายนอกที่กระทำกับวัตถุ  
 แกร่ง  $i$

ซึ่งเนื่องจาก  $M$  เป็นเมทริกซ์บล็อกในแนวทแยง (Block Diagonal)  $M^{-1}$  จึงคำนวณได้ง่ายคือ

$$M^{-1} = \begin{bmatrix} \frac{1}{m_1} E^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_1^{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m_2} E^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_2^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{m_n} E^3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_n^{-1} \end{bmatrix}$$

ขั้นตอนที่สองของการคลตามลำดับ คือการหาค่าของการคลที่ทำให้ข้อจำกัดเป็นจริง ด้วยวิธีดังต่อไปนี้

จากการที่  $f_c = J^T \lambda$  ดังนั้นข้อจำกัดคือ  $\dot{c} = Jv^* + b = 0$  และ  $v^* = \bar{v} + \Delta t M^{-1} f_c$  เมื่อ  
 $v^*$  คือความเร็วหลังการแก้ข้อจำกัดแล้ว จะสามารถแก้หา  $\lambda$  ได้โดย

$$J(\bar{v} + \Delta t M^{-1} J^T \lambda) + b = 0$$

$$J\bar{v} + \Delta t J M^{-1} J^T \lambda + b = 0$$

$$\Delta t J M^{-1} J^T \lambda = -(b + J\bar{v})$$

$$\lambda = -\frac{1}{\Delta t} (J M^{-1} J^T)^{-1} (b + J\bar{v})$$

หลังจากหา  $\lambda$  ได้แล้วก็ทำการเปลี่ยนค่าของความเร็วด้วย  $v^* = \bar{v} + \Delta t M^{-1} J^T \lambda$  และทำการปรับ  $\bar{v} \leftarrow v^*$

โดยทั่วไปแล้วถ้าพิจารณาทุกๆข้อจำกัดพร้อมๆกัน  $(JM^{-1}J^T)^{-1}$  จะต้องใช้เวลาในการคำนวณมากเนื่องจากจะต้องทำการหาเมทริกซ์ผกผันของเมทริกซ์ขนาดใหญ่ วิธีการลดตามลำดับจึงประมาณการด้วยการพิจารณาทีละข้อจำกัด กล่าวคือทำการหา  $\lambda$  โดยเสมือนหนึ่งว่าทั้งระบบมีแค่วัตถุแรงที่เกี่ยวกับข้อจำกัดนั้นเท่านั้น แล้วทำการแก้หา  $\lambda$  ที่เกี่ยวข้อง และทำการเปลี่ยนค่าของความเร็วที่เกี่ยวข้อง ซึ่งการคำนวณจะง่ายกว่ามากเนื่องจากแต่ละข้อจำกัดมักจะเกี่ยวข้องกับวัตถุเพียงไม่กี่ชิ้นเท่านั้น ผู้ที่สนใจสามารถดูสูตรของเมทริกซ์ผกผัน  $(JM^{-1}J^T)^{-1}$  ของข้อจำกัดที่พบบ่อยๆ รวมถึงข้อต่อแบบต่างๆพร้อมกับที่มาของสูตรได้ในงานวิจัย [13]

โดยทั่วไปแล้วจะทำการวนพิจารณาทุกๆข้อจำกัด และลดไปหลายๆ รอบจนกว่าการลดที่แก้ได้น้อยพอ หรือจนกว่าจะถึงจำนวนรอบที่กำหนดก็เป็นอันจบขั้นตอนที่สองของการลดตามลำดับ

ขั้นตอนที่สามของการลดตามลำดับคือการปรับตำแหน่ง และการวางแผนของวัตถุด้วยการใช้วิธีการออยเลอร์แบบไปข้างหน้า พร้อมทั้งทำการปรับให้เมทริกซ์ของการวางแผนเป็นเมทริกซ์หมุน หรือทำการปรับให้ควอเตอร์เนียนที่แทนการวางแผนเป็นควอเตอร์เนียนหน่วย

### การปรับปรุงการลดตามลำดับ

ถ้าทำตามเฉพาะขั้นตอนที่กล่าวถึงในขั้นต้น จะพบว่าข้อจำกัดจะค่อยๆ ผิดอย่างช้าๆ กล่าวคือ  $C$  จะไม่เป็น 0 ทั้งนี้เนื่องมาจากว่า แม้จะพยายามทำให้  $\dot{C}$  เป็น 0 แต่เนื่องจากการมีอยู่ของข้อผิดพลาด (Error) ต่างๆ เช่น ข้อผิดพลาดทางตัวเลข (Numerical Error) การที่ไม่ได้แก้ข้อจำกัดทั้งหมดพร้อมๆกัน ข้อผิดพลาดจากการอินทิเกรตในเชิงเวลาด้วยวิธีทางตัวเลข และอื่นๆ จึงทำให้  $\dot{C}$  อาจไม่เป็น 0 พอดีจึงทำให้  $C$  ค่อยๆเปลี่ยนไปช้าๆ ซึ่งปรากฏการณ์นี้เรียกว่าการเลื่อน (Drifting)

วิธีที่นิยมใช้ในการจัดการกับการเลือนนี้คือวิธีการทำให้มันคงของบวมการ์ด (Baumgarte Stabilization) [14]

ซึ่งคือการปรับ  $\dot{c}$  ที่อยู่ในรูป  $\dot{c} = jv$  (คือพจน์  $b(t) = 0$ ) ให้เป็น

$$\dot{c} = jv - \frac{\beta}{\Delta t} c$$

โดย  $0 \leq \beta \leq 1$  ซึ่งเรียกว่าสัมประสิทธิ์อคติ (Bias Coefficient) แทน โดยวิธีนี้จะดึงให้ค่า  $c$  กลับมาเป็น 0

ถ้า  $c$  ห่างออกจาก 0 เนื่องจากว่า คำตอบของสมการอนุพันธ์  $\dot{c} = -\frac{\beta}{\Delta t} c$  นั่นคือ  $c(t) = c_0 e^{-\frac{\beta}{\Delta t} t}$  เมื่อ

$c_0$  คือค่าเริ่มต้นและ  $e$  คือฐานของลอการิทึมธรรมชาติ  $\approx 2.71828 \dots$  ซึ่งเป็นเอกซ์โพเนนเชียลที่มีค่า

ลดลงเรื่อยๆ (Exponential Decay) ในทางปฏิบัติแล้ววิธีในการกำหนดค่าของ  $\beta$  ทำได้โดยเริ่มจาก 0 แล้วค่อยๆ

เพิ่ม  $\beta$  เรื่อยๆจนการจำลองไม่เสถียรจากนั้นก็ใช้ค่า  $\beta$  เป็นครึ่งหนึ่งของตอนที่เริ่มไม่เสถียร [12]

ปัญหาต่อไปที่ต้องจัดการคือการจำกัดค่าของ  $\lambda$  ซึ่งต้องใช้ในข้อจำกัดการชน และข้อจำกัดแรงเสียดทานซึ่งสิ่งที่ยากทำคือให้

$$\lambda_{\min} \leq \lambda \leq \lambda_{\max}$$

ด้วยการยึด  $\lambda = \text{clamp}(\lambda, \lambda_{\min}, \lambda_{\max})$  แต่จะไม่สามารถทำในแต่ละขั้นตอนของการติดตามลำดับได้

เนื่องจาก  $\Delta t \lambda$  ที่ห่อออกมาในแต่ละขั้นนั้นจริงๆ แล้วเป็นการดลที่เพิ่มขึ้นมา (Incremental Impulse) ที่แก้ไข

ความเร็วชั่วคราว ณ ปัจจุบันให้ถูกต้องตามข้อจำกัดมากขึ้น แต่ไม่ใช้การดลทั้งหมด (Total Impulse) ซึ่งสิ่งที่จะต้อง

ยึดคือการดลทั้งหมด ดังนั้นสิ่งที่ต้องทำคือให้แต่ละข้อจำกัดในตอนเริ่มมีการดลทั้งหมด  $\lambda_{\text{total}} = 0$  แล้วใน

การติดตามลำดับให้ทำดังนี้ [12]

1. คำนวณ  $\lambda$  แต่ยังไม่ทำการแก้ไขความเร็ว
2. ให้  $\lambda_{\text{old}} = \lambda_{\text{total}}$
3. ให้  $\lambda_{\text{total}} = \text{clamp}(\lambda_{\text{total}} + \lambda, \lambda_{\min}, \lambda_{\max})$
4. ให้  $\lambda_{\text{corrected}} = \lambda_{\text{total}} - \lambda_{\text{old}}$
5. ใช้  $\lambda_{\text{corrected}}$  ในการปรับความเร็วแทน

วิธีนี้จะทำให้การยึดการดลทำงานได้ถูกต้อง

อีกเรื่องที่ต้องจัดการก็คือข้อจำกัดประเภททอสมการซึ่งวิธีที่สามารถทำได้ก็คือ ในระหว่างที่พิจารณาข้อจำกัดนี้ในขั้นตอนการลดตามลำดับให้ตรวจสอบว่าข้อจำกัดนี้ถูกต้องอยู่แล้วไหม ถ้าถูกต้องอยู่แล้วก็ข้ามข้อจำกัดนี้ไป แต่ถ้าไม่ก็คำนวณตามปกติแล้วทำการยึด  $\lambda_{total}$  ให้อยู่ในช่วง  $0 \leq \lambda_{total} \leq \infty$  และในทางปฏิบัติแล้ว สำหรับข้อจำกัดของการสัมผัส (คือการชนที่ความเร็วสัมพัทธ์ในทิศตั้งฉากกับพื้นผิวตัว) ถ้าพยายามจะทำให้ไม่มีการทับลอยอาจจะทำให้เกิดการสั่น (Jittering) ของวัตถุแกร่งได้ เนื่องจากข้อจำกัดของการสัมผัสอาจจะถูกข้ามสลับกับไม่ข้ามในทุกๆ ลำดับเวลา ซึ่งวิธีที่จะเลี่ยงปัญหานี้คือการยอมให้วัตถุมีการทับกักันเล็กน้อย โดยให้  $d$  ไม่ต่ำกว่า  $d_{slop} < 0$  โดยปรับข้อจำกัดดังนี้

$$\text{ถ้า } d < d_{slop} \text{ ใช้ } \dot{c} = jv + \frac{\beta}{\Delta t} (d - d_{slop}) \text{ มิฉะนั้นใช้ } \dot{c} = jv \quad [12]$$

ซึ่งจะทำให้การสั่นนั้นน้อยลงเพราะแรงที่ผลักวัตถุนั้นจะค่อยๆ เพิ่มขึ้นเมื่อวัตถุทับกักันมากขึ้นเรื่อยๆ (ถ้าทับลจน  $d < d_{slop}$ ) โดยที่  $\beta$  เป็นค่าที่จะต้องปรับด้วยวิธีเดียวกับการทำให้มันคงของบวมมาร์ต

ประเด็นสุดท้ายคือการใช้ค่า  $\lambda_{total}$  จากขั้นตอนเวลาที่แล้วเพื่อประมาณค่าของการลดที่ควรจะเป็นในขั้นตอนเวลานี้ ซึ่งจะสามารถเพิ่มความเร็วในการคำนวณการลดที่ถูกต้องเป็นอย่างมาก โดยเฉพาะในกรณีที่กำลังล่องกองซ้อนของวัตถุแกร่งเนื่องจากค่าของการลดในขั้นตอนเวลาที่แล้วนั้นใกล้เคียง หรือเหมือนกับการลดในขั้นตอนนี้ซึ่งถ้านำมาใช้ ก็จะมีประเด็นว่ามีการทำการลดตามลำดับที่มีจำนวนครั้งเพิ่มขึ้นเรื่อยๆ ซึ่งก็คือแม่นยำขึ้นเรื่อยๆ นั่นเอง โดยวิธีที่จะต้องทำหลังจากคำนวณความเร็วชั่วคราว แต่ก่อนเริ่มทำการลดตามลำดับก็คือ

1. นำค่าของ  $\lambda_{total}$  จากขั้นตอนเวลาที่แล้วมาทำการปรับความเร็วชั่วคราวของวัตถุแกร่ง
2. ใช้ค่า  $\lambda_{total}$  จากขั้นตอนเวลาที่แล้วเป็นค่าเริ่มต้นของ  $\lambda_{total}$  ของขั้นตอนเวลานี้

## ประเด็นอื่นๆเกี่ยวกับการจำลองวัตถุแกร่ง

ในการจำลองวัตถุนั้นจริงๆแล้วยังมีประเด็นต่างๆอีกมากมายที่ผู้สนใจสามารถศึกษาเพิ่มเติมได้ เช่นการตรวจสอบว่าวัตถุแกร่งคู่ใดบ้างที่มีโอกาสชนกันนั้นมีหลากหลายวิธีที่ทำได้ ทั้งการตรวจแบบคร่าว (Broad Phase Collision Detection) ซึ่งจะพิจารณาจากปริมาตรที่ครอบวัตถุ (Bounding Volume) เช่นกล่อง หรือทรงกลม และการตรวจแบบแม่นยำ (Narrow Phase Collision Detection) ซึ่งจะต้องคำนวณการชนจากรูปร่างจริงๆ ของวัตถุ

เช่นทรงหลายเหลี่ยม (Polytope) ทรงหลายเหลี่ยมแบบป่อง (Convex Polytope) ทรงกระบอก ทรงแคบซูล เป็นต้น [15]

วิธีการอื่นๆ ที่ใช้ในการจำลองวัตถุแกร่งรวมถึงการแทนวัตถุแกร่งที่เชื่อมต่อกันด้วยข้อต่อให้มีจำนวนพิกัดน้อยลง (Reduced Coordinates) มีกล่าวถึงในงานวิจัย [16] และวิธีในการใช้ข้อจำกัดเพื่อการจำลองฟิสิกส์นั้นสามารถนำไปปรับเพื่อใช้กับระบบที่ไม่ได้มีแต่เพียงวัตถุแกร่งได้เช่น ในงานวิจัย [17] ซึ่งสามารถจำลองระบบที่มีทั้งวัตถุแกร่งและวัตถุนิ่มที่มีการชนและแรงเสียดทานระหว่างกันโดยที่สามารถจำลองแรงเสียดทานเป็นแบบที่เป็นโคเนได้

## บทสรุป

บทนี้กล่าวถึงการจำลองวัตถุแกร่งซึ่งมีคุณลักษณะทางฟิสิกส์ดังนี้ ตำแหน่ง การวางแนว ความเร็ว ความเร็วเชิงมุม มวล โมเมนต์ความเฉื่อย โดยมวลและโมเมนต์ความเฉื่อยสามารถคำนวณได้จากการปฏิยานุพันธ์บนปริมาตรของวัตถุโดยจะต้องนำความหนาแน่นมาใช้ในการคำนวณด้วย จากนั้นกล่าวถึงระบบสมการเชิงอนุพันธ์สามัญของการเคลื่อนไหวของวัตถุแกร่ง ตามด้วยแนวคิดการใช้ข้อจำกัดเพื่อจัดการกับการชน การสัมผัส และ แรงเสียดทาน ด้วยการคำนวณหาการดลจากนั้นจึงกล่าวถึงการใช้วิธีการดลตามลำดับในการหาการดลที่จะทำให้ข้อจำกัดเหล่านี้เป็นจริง จากนั้นจึงนำเสนอวิธีในการปรับปรุงการดลตามลำดับให้ทำงานได้อย่างมีประสิทธิภาพมากขึ้น และสุดท้ายกล่าวถึงเอกสารอ้างอิงต่างๆ ในการที่จะทำให้จำลองวัตถุแกร่งได้มีประสิทธิภาพมากขึ้น และสมจริงขึ้น

## บทที่ 4: การจำลองทางฟิสิกส์ด้วยวิธีไดนามิกเชิงตำแหน่ง

บทที่แล้วได้กล่าวถึงการจำลองวัตถุแกร่งด้วยการกำหนดข้อจำกัดต่างๆ แล้วใช้การลดตามลำดับในการทำให้ข้อจำกัดต่างๆนั้นเป็นจริง ในบทนี้จะกล่าวถึงวิธีการจำลองทางฟิสิกส์ของวัตถุหลากหลายชนิดที่ไม่ใช่วัตถุแกร่งเช่น วัตถุอ่อนนุ่ม (Soft Body) และของเหลวด้วยวิธีไดนามิกเชิงตำแหน่ง (Position Based Dynamics) ที่คิดค้นโดย [18] ซึ่งก็จะอาศัยข้อจำกัดต่างๆ ด้วยเช่นกันแต่ความแตกต่างหลักก็คือว่าการปรับต่างๆ จะกระทำกับตำแหน่งโดยตรง และความเร็วจะถูกปรับให้ถูกกัน (Consistent) กับตำแหน่ง โดยข้อจำกัดที่ใช้ในการจำลองวิธีไดนามิกเชิงตำแหน่งนั้นมีหลากหลายประเภท เช่นข้อจำกัดของระยะทางระหว่างอนุภาค (Distance Constraint) ข้อจำกัดการงอระหว่าง 4 อนุภาค (Bending Constraint) ข้อจำกัดของปริมาตรของรูปร่าง (Volume Constraint) ข้อจำกัดของความหนาแน่นของอนุภาค (Density Constraint) เป็นต้น

### พื้นฐานของการจำลองวิธีไดนามิกเชิงตำแหน่ง

ขั้นตอนหลักของวิธีไดนามิกเชิงตำแหน่งในแต่ละขั้นตอนเวลาของระบบที่ประกอบด้วยอนุภาค  $N$  อนุภาคที่มี

ตำแหน่ง  $x_i$ , ความเร็ว  $v_i$ , มวล  $m_i$  และ  $w_i = \frac{1}{m_i}$  โดยที่  $1 \leq i \leq N$  เป็นดังนี้

1. สำหรับทุกๆอนุภาค  $v_i = v_i + \Delta t w_i f_i^{\text{ext}}$
2. สำหรับทุกๆอนุภาค  $x_i^{\text{old}} = x_i$  และ  $x_i += \Delta t v_i$
3. สร้างข้อจำกัด  $C_1, \dots, C_M$  ขึ้นมาโดยพิจารณาจาก  $x_i$  โดย  $M$  คือจำนวนข้อจำกัด
4. วนไป  $K$  รอบ, ทำการปรับให้  $x_1, \dots, x_N$  ให้ข้อจำกัด  $C_1, \dots, C_M$  เป็นจริงด้วยวิธีการโปรเจกต์ลงบนเนื้อที่ของข้อจำกัด (Project Onto Constraint Manifold)
5. สำหรับทุกๆอนุภาค  $v_i = \frac{(x_i - x_i^{\text{old}})}{\Delta t}$
6. ทำการปรับความเร็ว  $v_1, \dots, v_N$  (ถ้าต้องการ)

วิธีการโปรเจกต์ลงบนเนื้อที่ของข้อจำกัดทำได้โดยการพิจารณาที่ละข้อจำกัด  $c(x)$  ถ้าเป็นข้อจำกัดประเภท

$c(x) = 0$  หรือ ถ้าเป็นข้อจำกัดประเภทมากกว่า  $c(x) \geq 0$  และ ณ ปัจจุบันนั้น  $c(x) < 0$  จะทำการปรับ  $x$  เพื่อให้  $c(x)$  เป็น 0 ซึ่งทำได้โดยการประมาณการ

$$c(x + \Delta x) \approx c(x) + \nabla c(x) \Delta x = 0$$

$$\text{โดยที่ } \nabla c(x) = \left[ \frac{\partial c}{\partial x_1} \quad \frac{\partial c}{\partial x_2} \quad \dots \quad \frac{\partial c}{\partial x_N} \right]$$

เพื่อความง่ายในการอธิบายในเบื้องต้นจะสมมุติว่า มวลของอนุภาคเท่ากันหมดก่อน กล่าวคือ  $m_i =$

$$w_i = 1$$

ถ้าข้อจำกัด  $c(x)$  เป็นข้อจำกัดภายใน (Internal Constraint) คือ  $c(x)$  ไม่เปลี่ยนถ้า  $x$  ทั้งระบบเกิดการเคลื่อนที่แบบวัตถุแกร่งกล่าวคือการเลื่อนหรือการหมุนแล้วถ้ากำหนดให้  $\Delta x$  อยู่ในทิศ  $(\nabla c(x))^T$  (และ  $|\Delta x| \rightarrow 0$ ) แล้วก็จะไม่ทำให้โมเมนตัมและโมเมนตัมเชิงมุมเปลี่ยนเพราะ  $(\nabla c(x))^T$  ตั้งฉากกับทิศที่  $c(x)$  ไม่เปลี่ยนซึ่งคือทิศที่เป็นการเคลื่อนที่แบบวัตถุแกร่ง ดังนั้นจึงให้

$$\Delta x = -\lambda (\nabla c(x))^T$$

ถ้ามวลของอนุภาคไม่เท่ากันทุกอันแล้วจะสามารถปรับปรุงให้  $\Delta x$  ไม่เปลี่ยนโมเมนตัมและโมเมนตัมเชิงมุม (เมื่อ  $|\Delta x| \rightarrow 0$ ) ได้โดยให้

$$\Delta x = -\lambda M^{-1} (\nabla c(x))^T$$

เมื่อ  $M$  คือเมทริกซ์แยงของมวลของอนุภาคต่างๆ ซึ่งสามารถนำไปใช้แก้สมการหา  $\lambda$  ได้โดย

$$c(x) - \lambda \nabla c(x) M^{-1} (\nabla c(x))^T = 0$$

$$\lambda = \frac{c(x)}{\nabla c(x) M^{-1} (\nabla c(x))^T}$$

หรือเขียนในรูปของ  $w_i$  ได้เป็น

$$\lambda = \frac{c(x)}{\sum_j w_j |\nabla_{x_j} c(x)|}$$

และ

$$\Delta x_i = -w_i \lambda_i \nabla_{x_i} c(x)$$

โดยยังสามารถกำหนดความแข็งทื่อ (Stiffness) ของข้อจำกัดได้โดยการระบุค่า  $0 \leq k \leq 1$  ของแต่ละข้อจำกัดและให้

$$\Delta x_i = -k w_i \lambda_i \nabla_{x_i} c(x)$$

## การจำลองวัตถุอ่อนนุ่มด้วยไดนามิกเชิงตำแหน่ง

ในการจำลองวัตถุอ่อนนุ่มด้วยไดนามิกเชิงตำแหน่งนั้นจะต้องสร้างข้อจำกัดที่ระหว่างอนุภาคเพื่อให้วัตถุคืนสภาพเดิมหลังจากแรงภายนอกหายไปข้อจำกัดที่พบบ่อยๆมีดังนี้

ข้อจำกัดการยืด

$$c(x_1, x_2) = |x_2 - x_1| - d = 0$$

โดยที่  $d$  คือระยะห่างระหว่างอนุภาค 1 และ 2 ที่ต้องการซึ่งมี

$$\nabla_{x_1} c(x_1, x_2) = n$$

$$\nabla_{x_2} c(x_1, x_2) = -n$$

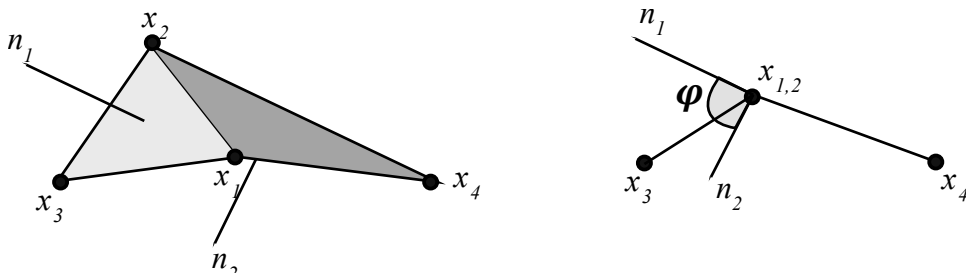
โดยที่  $n = \frac{x_2 - x_1}{|x_2 - x_1|}$  ซึ่งถ้าความแข็งที่คือ  $k_{\text{stretch}}$  การโปรเจกต์ลงบนเนื้อที่ของข้อจำกัดจะทำโดย

$$x_1^- = k_{\text{stretch}} \frac{w_1}{w_1 + w_2} (|x_2 - x_1| - d)n$$

$$x_2^+ = k_{\text{stretch}} \frac{w_1}{w_1 + w_2} (|x_2 - x_1| - d)n$$

ข้อจำกัดการงอ

เป็นข้อจำกัดระหว่าง 4 อนุภาคสำหรับการจำลองผ้า หรือวัตถุที่เป็นเปลือกบางๆ (Thin Shell) โดยอนุภาคนั้นเชื่อมต่อกันเป็นเมซสามเหลี่ยม (Triangle Mesh) ดังแสดงในภาพประกอบ 21



ภาพประกอบ 21: ข้อจำกัดการงอระหว่าง 4 อนุภาค ดัดแปลงจาก [19]

โดยถ้าต้องการให้มุมระหว่างสามเหลี่ยม (Dihedral Angle)  $(x_1, x_2, x_3)$  และ  $(x_2, x_1, x_4)$  นั้นเป็น  $\varphi_0$  ซึ่งสามารถใช้ข้อจำกัด

$$c(x_1, x_2, x_3, x_4) = \cos^{-1} \left( \frac{(x_2 - x_1) \times (x_3 - x_1)}{|(x_2 - x_1) \times (x_3 - x_1)|} \cdot \frac{(x_2 - x_1) \times (x_4 - x_1)}{|(x_2 - x_1) \times (x_4 - x_1)|} \right) - \varphi_0 = 0$$

ซึ่งถ้าความแข็งที่คือ  $k_{\text{bend}}$  แล้วการโปรเจกต์ลงบนเนื้อที่ของข้อจำกัดจะทำโดย

$$x_i = k_{\text{bend}} \frac{w_i \sqrt{1 - d^2} (\cos^{-1}(d) - \varphi_0)}{\sum_j w_j |q_j|^2} q_j$$

โดยที่

$$\begin{aligned} n_1 &= \frac{(x_2 - x_1) \times (x_3 - x_1)}{|(x_2 - x_1) \times (x_3 - x_1)|} \\ n_2 &= \frac{(x_2 - x_1) \times (x_4 - x_1)}{|(x_2 - x_1) \times (x_4 - x_1)|} \\ d &= n_1 \cdot n_2 \\ q_2 &= - \frac{(x_3 - x_1) \times n_2 + d(n_1 \times (x_3 - x_1))}{|(x_2 - x_1) \times (x_3 - x_1)|} \\ &\quad - \frac{(x_4 - x_1) \times n_1 + d(n_2 \times (x_4 - x_1))}{|(x_2 - x_1) \times (x_4 - x_1)|} \\ q_3 &= \frac{(x_2 - x_1) \times n_2 + d(n_1 \times (x_2 - x_1))}{|(x_2 - x_1) \times (x_3 - x_1)|} \\ q_4 &= \frac{(x_2 - x_1) \times n_1 + d(n_2 \times (x_2 - x_1))}{|(x_2 - x_1) \times (x_4 - x_1)|} \\ q_1 &= -q_2 - q_3 - q_4 \end{aligned}$$

ข้อจำกัดพื้นที่ของสามเหลี่ยม

จะสามารถทำให้สามเหลี่ยมรักษาพื้นที่ได้ด้วยการใช้ข้อจำกัด

$$c(x_1, x_2, x_3) = \frac{1}{2} |(x_2 - x_1) \times (x_3 - x_1)| - a_0 = 0$$

โดยที่  $a_0$  คือพื้นที่เริ่มต้นของสามเหลี่ยม

ข้อจำกัดปริมาตรของจตุรมุข

ถ้าอนุภาคเชื่อมต่อกันเป็นเมชของจตุรมุข (Tetrahedral Mesh) แล้วจะสามารถสร้างข้อจำกัดให้จตุรมุขนั้น พยายามรักษาปริมาตรของมันไว้ได้ โดยการใส่ข้อจำกัด

$$c(x_1, x_2, x_3, x_4) = \frac{1}{6} ((x_2 - x_1) \times (x_3 - x_1)) \cdot (x_2 - x_1) - v_0 = 0$$

เมื่อ  $v_0$  คือปริมาตรเริ่มต้นของ จตุรมุข

ข้อจำกัดของการชนระหว่างอนุภาคกับสามเหลี่ยม

ถ้าระบบอนุภาคแทนวัตถุที่มีพื้นผิวแทนด้วยเมชสามเหลี่ยมแล้ว จะสามารถทำให้สามเหลี่ยมเหล่านั้นไม่ซ้อนทับกันด้วยการเพิ่มข้อจำกัดระหว่างอนุภาค  $q$  ที่อาจจะชนกับสามเหลี่ยม  $(x_1, x_2, x_3)$

$$c(q, x_1, x_2, x_3) = (q - x_1) \cdot \frac{(x_3 - x_1) \times (x_2 - x_1)}{|(x_3 - x_1) \times (x_2 - x_1)|} - h \geq 0$$

โดยที่  $h$  คือความหนาของสามเหลี่ยม

ข้อจำกัดของการชนระหว่างอนุภาคกับสิ่งแวดล้อม

ถ้าสิ่งแวดล้อมที่อนุภาคสามารถชนได้ประกอบด้วยสามเหลี่ยมหรือระนาบต่างๆแล้ว จะสามารถบังคับให้อนุภาคไม่ซ้อนทับสิ่งแวดล้อมได้ด้วยการเพิ่มข้อจำกัดสำหรับระหว่างทุกๆ คู่ของอนุภาค  $x$  และระนาบที่มีเวกเตอร์ฉาก  $n$  ที่ใกล้กันดังนี้

$$c(x) = n \cdot x - d_{\text{rest}} \geq 0$$

โดยที่  $d_{\text{rest}}$  คือระยะห่างอย่างน้อยที่อนุภาคจะต้องห่างจากระนาบนี้

ข้อจำกัดของการชนระหว่างอนุภาคกับอนุภาค

การชนระหว่างอนุภาค  $x_i$  และ  $x_j$  สามารถจำลองได้โดยเพิ่มข้อจำกัด

$$c(x_i, x_j) = |x_i - x_j| - (r_i + r_j) \geq 0$$

โดย  $r_i$  และ  $r_j$  คือขนาดของอนุภาค

ข้อจำกัดของแรงเสียดทานระหว่างอนุภาคกับอนุภาค

แรงเสียดทานสามารถจำลองได้โดยใช้วิธีที่นำเสนอใน [20]

สำหรับอนุภาค  $i$  และ  $j$  ที่ชนกันสามารถคำนวณ

$$\Delta x^{\text{rel}} = (x_i - x_i^{\text{old}}) - (x_j - x_j^{\text{old}})$$

$$n = \frac{x_j - x_i}{|x_j - x_i|}$$

$$\Delta x_{\perp}^{\text{rel}} = \Delta x^{\text{rel}} - (\Delta x^{\text{rel}} \cdot n)n$$

$$d = |x_i - x_j| - (r_i + r_j)$$

$$\Delta_i = \frac{w_i}{(w_i + w_j)} \begin{cases} \Delta x_{\perp}^{\text{rel}}, & \text{ถ้า } |\Delta x_{\perp}^{\text{rel}}| < \mu_s d \\ \min\left(\frac{\mu_k d}{|\Delta x_{\perp}^{\text{rel}}|}, 1\right) \Delta x_{\perp}^{\text{rel}}, & \text{นอกจากนั้นแล้ว} \end{cases}$$

โดยที่  $\mu_s$  และ  $\mu_k$  คือสัมประสิทธิ์แรงเสียดทานหยุดนิ่ง (Static Friction Coefficient) และ สัมประสิทธิ์แรงเสียดทานเคลื่อนไหวย (Dynamic Friction Coefficient)

แล้วทำการปรับ

$$x_i += \Delta_i$$

$$x_j -= \frac{w_j}{(w_i + w_j)} \Delta_i$$

แรงเสียดทานระหว่างอนุภาคกับสิ่งแวดล้อมก็ทำได้ด้วยวิธีคล้ายๆกันโดยให้  $n$  เป็นเวกเตอร์ตั้งฉากกับผิวของสิ่งแวดล้อมและ  $W$  ของสิ่งแวดล้อมเป็น 0

### การจำลองของเหลวด้วยไดนามิกเชิงตำแหน่ง

วิธีไดนามิกเชิงตำแหน่งสามารถใช้ในการจำลองของเหลวได้ด้วยการสร้างข้อจำกัดระหว่างกลุ่มของอนุภาคให้มีความหนาแน่นประมาณเป็นค่าคงตัว [21] โดยใช้ข้อจำกัด

$$c_i(x_1, \dots, x_n) = \frac{\rho_i}{\rho_0} - 1$$

โดย  $\rho_0$  คือความหนาแน่นเป้าหมายและ  $\rho_i$  คือความหนาแน่นในปัจจุบันซึ่งคำนวณโดย

$$\rho_i = \sum_j m_j w(x_i - x_j, h)$$

โดยที่  $w$  คือฟังก์ชันแกน (Kernel) ชนิด Poly6 [22] ซึ่งมีสูตรการคำนวณดังนี้

$$w(r, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - |r|^2)^3 & \text{ถ้า } 0 \leq |r| \leq h \\ 0 & \text{นอกจากนั้นแล้ว} \end{cases}$$

โดย  $h$  คือระยะที่อนุภาคสามารถออกแรงต่อกันได้ซึ่งจะสามารถหาเกรเดียนต์ (Gradient) ของข้อจำกัดได้เป็น

$$\nabla_{x_k} c_i = \begin{cases} \sum_j \nabla w(x_i - x_j, h) & \text{ถ้า } k = i \\ -\nabla w(x_i - x_j, h) & \text{ถ้า } k = j \end{cases}$$

ซึ่งสำหรับเกรเดียนต์  $\nabla w$  ในทางปฏิบัติแล้วมักจะใช้เกรเดียนต์ของฟังก์ชันแกนชนิด Spiky เพื่อลดการเกาะกลุ่มของอนุภาค [22] ซึ่งคำนวณได้โดย

$$\nabla w(r, h) = -\frac{45}{\pi h^6} (h - |r|)^2 \frac{r}{|r|}$$

จากนั้นทำการคำนวณ

$$\lambda_i = -\frac{c_i(x_1, \dots, x_n)}{\sum_k |\nabla_{x_k} c_i|^2 + \varepsilon}$$

ซึ่ง  $\varepsilon$  เป็นค่าคงที่ที่ทำให้เลี่ยงการหารด้วยค่าใกล้เคียง 0

และเพื่อลดการเกาะกลุ่มของอนุภาคเมื่อตอนที่อนุภาคมีจำนวนอนุภาคข้างเคียงน้อย จะสามารถเพิ่มแรงผลักระหว่างอนุภาคได้โดยคำนวณ

$$s_{\text{corr}} = -k \left( \frac{w(p_i - p_j, h)}{w(\Delta q, h)} \right)^n$$

โดยที่  $\Delta q$  คือเวกเตอร์อะไรก็ได้ที่ขนาดน้อยกว่า  $h$  และ  $k, n$  เป็นค่าคงที่ โดยงานวิจัย [21] ใช้

$$0.1h \leq |\Delta q| \leq 0.3h, k=0.1, n=4$$

จากนั้นก็จะสามารถคำนวณการปรับตำแหน่งได้ด้วย

$$\Delta x_i = \frac{1}{\rho_0} \sum_j (\lambda_i + \lambda_j + s_{\text{corr}}) \nabla w(x_i - x_j, h)$$

หลังจากที่คำนวณ  $\Delta x_i$  ของทุกๆอนุภาคแล้วจะสามารถทำการปรับปรุงตำแหน่งด้วย

$$x_i += \Delta x_i$$

## บทสรุป

บทนี้กล่าวถึงการจำลองทางฟิสิกส์ด้วยวิธีไดนามิกเชิงตำแหน่งซึ่งใช้ตำแหน่งในการเก็บสถานะของวัตถุและทำการปรับแก้ข้อจำกัดต่างๆ ด้วยการแก้ไขตำแหน่งโดยตรง ส่วนความเร็วนั้นจะถูกนิยามโดยเป็นผลต่างของตำแหน่งซึ่งข้อดีคือจะไม่มี การเลื่อน (Drift) ของข้อจำกัดต่างๆ และสามารถจำลองวัตถุได้หลากหลายชนิดเช่น วัตถุอ่อนนุ่มของเหลว ด้วยการกำหนดข้อจำกัดที่เหมาะสมจะสังเกตได้ว่าวิธีไดนามิกเชิงตำแหน่งนั้นมีการใช้การคำนวณ และการประมาณต่างๆที่ไม่ถูกต้องตามหลักฟิสิกส์นัก เช่นการปรับแก้  $x$  ในทิศ  $(\nabla c(x))^T$  จะไม่ทำให้โมเมนตัมและโมเมนตัมเชิงมุมเปลี่ยนแปลงเฉพาะเมื่อการปรับมีขนาดเข้าใกล้ 0 เท่านั้นซึ่งไม่จริงในทางปฏิบัติรวมถึงข้อจำกัดต่างๆที่ใช้ และวิธีในการแก้ข้อจำกัดก็ไม่ได้มีที่มาจากกลศาสตร์ของแข็ง (Solid Mechanics) และ กลศาสตร์ของเหลว (Fluid Mechanics) แต่ไว้ในทางปฏิบัติแล้ววิธีไดนามิกเชิงตำแหน่งนั้นให้ผลการจำลองที่ดูแล้วคล้ายของจริง (Physically Plausible) และ เสถียร (Stable) แม้ในกรณีที่  $\Delta t$  มีค่ามากรวมถึงยังทำงานได้รวดเร็ว จึงเป็นที่นิยมใช้อย่างแพร่หลายทั้งในเกม และภาพยนตร์

## บทที่ 5: การจำลองทางฟิสิกส์ด้วยตาราง

เนื้อหาทั้งหมดในหนังสือก่อนหน้านี้เป็นการจำลองทางฟิสิกส์โดยใช้อนุภาค หรือสิ่งที่คล้ายอนุภาคซึ่งเก็บตำแหน่ง ความเร็ว และ คุณสมบัติทางฟิสิกส์อื่นๆ ที่เคลื่อนที่ไปด้วยกันกับอนุภาค หรือสิ่งที่คล้ายอนุภาค ซึ่งเรียกรวมๆว่าเป็นการคำนวณแบบลากรางเจียน (Lagrangian) แต่อีกแนวทางหนึ่งที่สามารถใช้ในการจำลองทางฟิสิกส์ได้ คือการการคำนวณหาการเปลี่ยนแปลงของคุณสมบัติทางฟิสิกส์ ณ ตำแหน่งคงที่โดยไม่เคลื่อนที่ไปกับอนุภาค ซึ่งเรียกรวมๆว่าเป็นการคำนวณแบบออยเลอร์เรียน (Eulerian)

ความแตกต่างของการคำนวณแบบลากรางเจียน และ แบบออยเลอร์เรียนนั้นสามารถเปรียบเทียบได้เช่นเดียวกับการวัดอุณหภูมิของผิวน้ำของแม่น้ำทั้งสาย การวัดแบบลากรางเจียนนั้นเปรียบเทียบได้กับการนำเทอร์โมมิเตอร์ติดกับกระถางจำนวนมากแล้วปล่อยกระถางให้ลอยไปตามแม่น้ำแล้วรายงานอุณหภูมิ ณ ตำแหน่งที่กระถางแต่ละอันอยู่ซึ่งเปลี่ยนไปเรื่อยๆ ส่วนการวัดแบบออยเลอร์เรียนนั้นเปรียบเทียบได้กับการเอาเทอร์โมมิเตอร์ติดกับทุ่นลอยที่มีสมอถ่วงให้อยู่กับที่แล้วคอยรายงานอุณหภูมิ ณ ตำแหน่งที่ทุ่นอยู่ซึ่งคงที่

### พื้นฐานของการจำลองด้วยตาราง

การคำนวณแบบออยเลอร์เรียนที่ใช้อย่างแพร่หลายที่สุดในการจำลองแอนิเมชันเชิงฟิสิกส์คือการใช้ตาราง (Grid) โดยเป็นการแบ่งปริมาตร หรือ พื้นที่ที่ต้องการคำนวณออกเป็นช่องๆ ซึ่งการจำลองด้วยตารางแบบพื้นฐานนั้นจะให้ทุกช่อง (Cell) มีขนาดเท่ากัน โดยคุณสมบัติทางฟิสิกส์ที่จะจำลองนั้นจัดเก็บอยู่ที่กลางช่อง (Cell Center) หรือตรงขอบ (Edge) หรือตรงมุม (Node) หรือถ้าเป็นตารางสามมิติก็อาจจะเก็บตรงหน้าได้ (Face)

การจำลองแอนิเมชันเชิงฟิสิกส์ที่มักใช้ตารางในการคำนวณก็คือการจำลองของไหล (Fluid) เนื่องจากการคำนวณทำได้ไม่ยุ่งยากและได้ผลลัพธ์ที่ดี โดยก่อนที่จะทำการจำลองของไหลได้นั้น จะต้องทำความรู้จักกับสมการที่เกี่ยวข้องกับของไหลก่อน

## สมการของของไหล

สมการที่ใช้อธิบายของไหลที่ใช้อย่างแพร่หลายคือสมการเชิงอนุพันธ์ย่อยนาเวียร์-สโตกส์ของของไหลที่มีความหนืดคงที่ และไม่สามารถบีบอัดได้ (Incompressible Navier-Stokes, Constant Viscosity Equations) ซึ่งกล่าวถึงอัตราการเปลี่ยนแปลงของความเร็วของของไหล ณ แต่ละตำแหน่งดังนี้

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p = \mathbf{a}^{\text{ext}} + \mu \nabla \cdot \nabla \mathbf{u}$$
$$\nabla \cdot \mathbf{u} = 0$$

โดย  $\mathbf{u}$  คือความเร็วของของไหล ณ แต่ละตำแหน่งหน่วยเป็น  $\text{m/s}$ ,  $\rho$  คือความหนาแน่นหน่วยเป็น

$\text{kg/m}^3$ ,  $p$  คือความดันหน่วยเป็น  $\text{N/m}^2$ ,  $\mathbf{a}^{\text{ext}}$  คือความเร่งจากแรงภายนอกหน่วยเป็น  $\text{m/s}^2$ ,  $\mu$  คือความหนืด (Kinematic Viscosity) หน่วยเป็น  $\text{Nm}^{-1}\text{s}^{-1}$  และ

$$\mathbf{u} \cdot \nabla \mathbf{u} = \begin{bmatrix} u_x \frac{\partial u_x}{\partial x} + u_y \frac{\partial u_x}{\partial y} + u_z \frac{\partial u_x}{\partial z} \\ u_x \frac{\partial u_y}{\partial x} + u_y \frac{\partial u_y}{\partial y} + u_z \frac{\partial u_y}{\partial z} \\ u_x \frac{\partial u_z}{\partial x} + u_y \frac{\partial u_z}{\partial y} + u_z \frac{\partial u_z}{\partial z} \end{bmatrix},$$

$$\nabla \cdot \nabla \mathbf{u} = \begin{bmatrix} \frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2} \\ \frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_y}{\partial z^2} \\ \frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial y^2} + \frac{\partial^2 u_z}{\partial z^2} \end{bmatrix}, \quad \nabla p = \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} \end{bmatrix}$$

$$\nabla \cdot \mathbf{u} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}$$

เงื่อนไข  $\nabla \cdot \mathbf{u} = 0$  นั้นคือไดเวอร์เจนซ์ (Divergence) ของ  $\mathbf{u}$  จะต้องเป็น 0 ซึ่งคือเงื่อนไขที่บอกวาปริมาตรของของไหลจะต้องไม่เปลี่ยนนั่นเอง เนื่องมาจากทฤษฎีบทไดเวอร์เจนซ์ (Divergence Theorem) ซึ่งกล่าวว่า

$$\iiint_V \nabla \cdot \mathbf{u} \, dV = \oiint_S \mathbf{u} \cdot \mathbf{n} \, dS$$

โดยที่  $V$  คือปริมาตร  $S$  คือผิวของปริมาตร  $\mathbf{n}$  คือเวกเตอร์ฉากของผิว ซึ่ง  $\mathbf{u} \cdot \mathbf{n}$  นั้นคือความเร็วในการไหลออกของของไหลต่อหน่วยพื้นที่  $\oiint_S \mathbf{u} \cdot \mathbf{n} \, dS$  นั้นคือปริมาตรของของไหลที่ไหลออกจากปริมาตร  $V$  นั้นเองซึ่ง

ถ้า  $\iiint_V \nabla \cdot \mathbf{u} \, dV = \iiint_V 0 \, dV = 0$  แล้วนั่นก็คือว่าสำหรับปริมาตรใดๆ ของไหลที่ไหลเข้ามาเท่าไรก็ต้องไหลออกไปเท่านั้นเสมอ ก็คือแปลว่าปริมาตรของของไหลนั้นไม่เปลี่ยนนั่นเอง

สมการของของไหลนั้นดูมีความซับซ้อนมาก แต่จริงๆแล้วมีวิธีที่จะใช้ในการจำลองของไหลแบบเบื้องต้นได้ไม่ยากนัก ในทางปฏิบัติแล้วการจำลองแอนิเมชันเชิงฟิสิกส์นั้น มักก่อให้เกิดความหนืดเทียม (Artificial Viscosity) มากอยู่แล้วจึงมักจะกำหนดให้  $\mu$  เป็น 0 ซึ่งจะทำให้สมการเหลือเพียง

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p = \mathbf{a}^{\text{ext}}$$

$$\nabla \cdot \mathbf{u} = 0$$

ซึ่งเรียกว่าสมการเชิงอนุพันธ์ย่อยออยเลอร์ที่ไม่สามารถบีบอัดได้ (Incompressible Euler Equations)

### การจำลองน้ำด้วยสมการคลื่น

วิธีพื้นฐานที่ใช้ในการจำลองน้ำในสระว่ายน้ำก็คือการจำลองน้ำด้วยสนามความสูง (Height Field) [23] ซึ่งมีการตั้งสมมุติฐาน (Assumption) หลายอย่างเพื่อลดรูปสมการเชิงอนุพันธ์ย่อยออยเลอร์ที่ไม่สามารถบีบอัดได้ ดังนี้

1. ฝิวน้ำจะต้องเป็นสนามความสูงเท่านั้น นั่นคือสำหรับแต่ละพิกัด  $(\mathbf{x}, z)$  แล้ว ฝิวน้ำจะมีความสูงในแกน  $y$  ได้เพียงค่าเดียว ซึ่งจะทำให้สามารถจำลองคลื่นบนฝิวน้ำที่ไม่ม้วนทับตัวเองและไม่มีละอองหรือมวลของน้ำที่ลอยแยกออกมาจากฝิวน้ำ
2. ไม่สนใจความเร็วของน้ำทางแกน  $y$  การเคลื่อนที่ของมวลน้ำทางแกน  $y$  มาจากแรงดันที่เกิดจากแรงภายนอกที่มีเพียงแรงโน้มถ่วงเท่านั้น

3. ความเร็วของน้ำทางแกน  $X$  และ  $Z$  ไม่เปลี่ยนแปลงตามความลึกของน้ำ

จากสมมติฐานทั้ง 3 ข้อจะสามารถสรุปได้ว่าความดันนั้นแปรตามความลึกของน้ำ และสามารถลดรูปสมการออยเลอร์ที่ไม่สามารถบีบอัดได้ให้เป็นสมการของน้ำตื้น (Shallow Water Equation) [24] หน้า 170-173 เหลือเป็น

$$\frac{\partial d}{\partial t} + \mathbf{u} \cdot \nabla d = -d(\nabla \cdot \mathbf{u})$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -g\nabla h$$

ซึ่ง  $\mathbf{u} = \begin{bmatrix} u_x \\ u_z \end{bmatrix}$ ,  $d$  คือความลึกของน้ำ,  $H$  คือความสูงของพื้นใต้น้ำ,  $h = d + H$ ,  $g$  คือขนาดของความเร่งที่เกิดจากแรงโน้มถ่วง ( $\sim 9.81 \text{ m/s}^2$ ) สมการนี้สามารถใช้ในการจำลองน้ำได้หลากหลาย [25] [26]

นอกจากนี้ยังสามารถลดรูปอีกได้ด้วยการตั้งข้อสมมติเพิ่มเติมอีก กล่าวคือ

1. สมมติว่า  $d$  เปลี่ยนแปลงช้า นั่นคือ  $\nabla d \approx 0$
2. สมมติว่า  $\mathbf{u} \cdot \nabla \mathbf{u} \approx 0$  ซึ่งคือไม่สนใจการที่ความเร็วนำพาความเร็วให้ไหลไปตามความเร็ว (Self Advection)

จากสองข้อนี้จะทำให้สามารถลดรูปสมการของน้ำตื้นให้เหลือเป็น

$$\frac{\partial d}{\partial t} = -d(\nabla \cdot \mathbf{u})$$

$$\frac{\partial \mathbf{u}}{\partial t} = -g\nabla h$$

ซึ่งสามารถกำจัด  $\mathbf{u}$  ที่ได้จากเหลือเพียง

$$\frac{\partial^2 h}{\partial t^2} = gd\nabla \cdot \nabla h$$

และเนื่องจาก  $d$  เปลี่ยนแปลงช้าจึงสมมติได้ว่า  $gd$  เป็นค่าคงที่  $c = \sqrt{gd}$  ซึ่งจะสามารถเขียนสมการได้เป็น

$$\frac{\partial^2 h}{\partial t^2} = c^2 \nabla \cdot \nabla h$$

และเรียกสมการนี้เรียกว่าสมการคลื่น (Wave Equation) โดยความเร็วของคลื่นจะเป็นค่าคงที่เท่ากับ  $c \text{ m/s}^2$

โดยที่

$$\nabla \cdot \nabla h = \frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial z^2}$$

จะสามารถจำลองสมการคลื่นได้โดยการทำให้เป็นดิสครีต (Discretization) ซึ่งคือเก็บค่าของฟังก์ชัน (ในที่นี้คือ  $h$ ) ณ จุดต่างๆ ด้วยตารางสองมิติในระนาบ  $x-z$  แล้วประมาณค่าของอนุพันธ์ต่างๆ ด้วยวิธีผลต่างจำกัด (Finite Difference Method) ซึ่งก็คือการนำค่าของ  $h$  ในช่องที่ติดกันมาลบกันแล้วหารด้วยขนาดช่อง เพื่อประมาณค่าของอนุพันธ์นั่นเอง โดยจะใช้ตารางที่เก็บ  $h$  และ  $r = \frac{\partial h}{\partial t}$  ดังภาพประกอบ 22

•	$h_{i,k+1}$	•
$h_{i-1,k}$	$h_{i,k}$	$h_{i+1,k}$
•	$h_{i,k-1}$	•

ภาพประกอบ 22: การเก็บค่าความสูงของน้ำในตารางเพื่อการจำลองสมการคลื่น

โดยให้แต่ละช่องของตารางกว้างและสูง  $\Delta x$  และเก็บความสูงที่ตรงกลางช่องด้วย  $h_{i,k}$  และอนุพันธ์เชิงเวลาของความสูงด้วย  $r_{i,k}$  โดยที่  $0 \leq i \leq N - 1$ ,  $0 \leq j \leq N - 1$  โดยที่  $N$  คือจำนวนช่องทางแกน  $x$  และ  $z$

และจะสามารถแปลงสมการคลื่นเป็นดิสครีตได้ดังนี้

$$\begin{aligned} \frac{\partial h_{i+\frac{1}{2},k}}{\partial x} &\approx \frac{h_{i+1,k} - h_{i,k}}{\Delta x} \\ \frac{\partial h_{i,k+\frac{1}{2}}}{\partial z} &\approx \frac{h_{i,k+1} - h_{i,k}}{\Delta z} \\ \frac{\partial^2 h_{i,k}}{\partial x^2} &\approx \frac{\frac{\partial h_{i+\frac{1}{2},k}}{\partial z} - \frac{\partial h_{i-\frac{1}{2},k}}{\partial z}}{\Delta x} \end{aligned}$$

$$\frac{\partial^2 h_{i,k}}{\partial z^2} \approx \frac{\frac{\partial h_{i,k+1}}{\partial z} - \frac{\partial h_{i,k-1}}{\partial z}}{\Delta x}$$

$$\frac{\partial r_{i,k}}{\partial t} \approx c^2 \left( \frac{\partial^2 h_{i,k}}{\partial x^2} + \frac{\partial^2 h_{i,k}}{\partial z^2} \right)$$

$$\frac{\partial h_{i,k}}{\partial t} \approx \frac{\partial r_{i,k}}{\partial t}$$

ซึ่งถ้าทำการปฏิยานุพันธ์ไปตามเวลาด้วยวิธีการออยเลอร์แบบกึ่งนัย และแทนค่าของประมาณการของอนุพันธ์ต่างๆ แล้วก็จะสามารถปรับค่าของ  $r_{i,k}$  และ  $h_{i,k}$  ได้โดย

$$r_{i,k} += \frac{\Delta t c}{\Delta x^2} (h_{i+1,k} + h_{i,k+1} + h_{i,k-1} + h_{i-1,k} - 4h_{i,k})$$

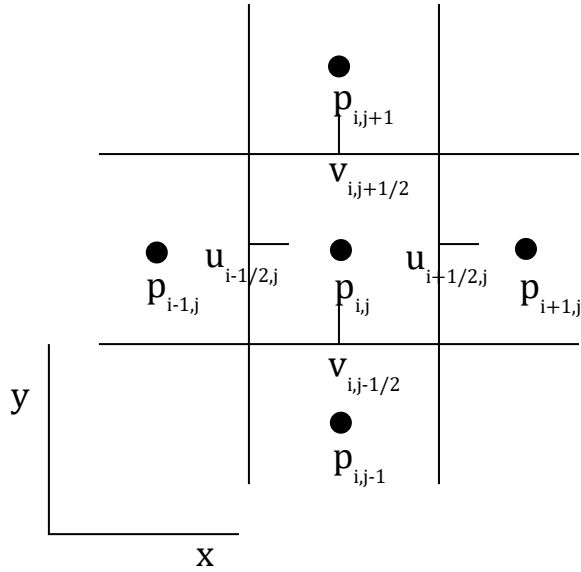
$$h_{i,k} += \Delta t r_{i,k}$$

จะสังเกตได้ว่าในการคำนวณของช่องใกล้เคียง ขอบจะมีการใช้ค่าที่อาจอยู่นอกตารางซึ่งถ้าต้องการให้คลื่นสะท้อนกับขอบก็สามารถทำได้โดยใช้ค่าในตารางที่ตำแหน่งใกล้เคียงที่สุดแทนเช่น ใช้ค่าของ  $h_{N-1,k}$  แทนค่าของ  $h_{N,k}$  เป็นต้นซึ่งเรียกขอบแบบนี้ว่าขอบปิด (Close Boundary) และถ้าอยากให้คลื่นสามารถเคลื่อนที่จากขวาสุด  $\leftrightarrow$  ซ้ายสุด และจากบนสุด  $\leftrightarrow$  ล่างสุด จะสามารถทำได้โดยใช้ค่าของอีกฟากของตารางมาแทน เช่นใช้  $h_{0,k}$  แทนค่าของ  $h_{N,k}$  เป็นต้นซึ่งสามารถเรียกขอบแบบนี้ว่าขอบซ้ำเป็นช่วงๆ (Periodic Boundary)

ส่วนการแสดงผลนั้นจะสามารถนำ  $h_{i,k}$  มาใช้ในการกำหนดความสูงของเมฆสามมิติเพื่อนำไปวาดได้

## การจำลองควัน

ควันเป็นของไหลประเภทหนึ่งที่มีความน่าสนใจและเป็นส่วนประกอบที่พบได้มากในสเปเชียลเอฟเฟค โดยในเนื้อหาดูต่อไปนี้จะอธิบายวิธีการจำลองควันในสองมิติ ซึ่งจะสามารถปรับให้เป็นสามมิติที่หลังได้อย่างตรงไปตรงมา โดยการจำลองควันจะใช้ตาราง 2 มิติในระนาบ  $xy$  เพื่อเก็บคุณสมบัติทางฟิสิกส์ดังภาพประกอบ 23



ภาพประกอบ 23: ตาราง 2 มิติเก็บคุณสมบัติทางฟิสิกส์สำหรับจำลองควัน

$p_{i,j}$  คือความดัน,  $u_{i+\frac{1}{2},j}$  คือความเร็วทางแกน  $x$ ,  $v_{i,j+\frac{1}{2}}$  คือความเร็วทางแกน  $y$  โดยตารางจะเป็นแบบส่าย (Staggered Grid) เพื่อให้การประมาณการอนุพันธ์นั้นให้ผลที่แม่นยำขึ้นคล้ายกับการประมาณค่าอนุพันธ์ของสมการคลื่น แต่ต่างกันตรงที่จะต้องปรับค่าของความเร็วในทุกๆ ขั้นตอนเวลาด้วย จึงต้องเก็บข้อมูลไว้ในตาราง โดยจะถือว่าแต่ละช่องมีความกว้าง/ความสูง  $\Delta x$  ซึ่ง  $p_{i,j}$  เก็บ  $p$  ที่ตำแหน่ง  $(i\Delta x, j\Delta x)$ ,  $u_{i+\frac{1}{2},j}$  เก็บ  $u$  ที่ตำแหน่ง  $(i + \frac{1}{2})\Delta x, j\Delta x$  เป็นต้น

โดยหลักการของการจำลองควันนั้นทำการหาความเร็ว ณ ตำแหน่งต่างๆ ในแต่ละขั้นตอนเวลา จากนั้นจะสามารถนำความเร็วนี้ไปขยายอนุภาค แล้วนำอนุภาคมาแสดงผลเป็นกลุ่มควัน หรือสามารถเพิ่มความเข้มของควันเข้าไปเป็นคุณสมบัติหนึ่งที่จะต้องถูกปรับในแต่ละขั้นตอนเวลาด้วย แล้วนำความเข้มของควันนี้มาแสดงผล โดยในขั้นตอนนี้จะนำเสนอวิธีในการปรับความเร็วในแต่ละขั้นตอนเวลาเป็นหลัก ซึ่งจะใช้วิธีของไหลเสถียร (Stable Fluid) [27]

สมการเชิงอนุพันธ์ย่อยของลอว์ที่ไม่สามารถบีบอัดได้

$$\frac{\partial u}{\partial t} + u \cdot \nabla u + \frac{1}{\rho} \nabla p = a^{\text{ext}}$$

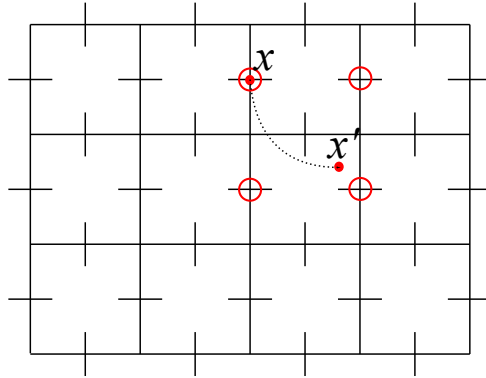
$$\nabla \cdot u = 0$$

มีความซับซ้อนและไม่เป็นสมการเชิงเส้น (Non-Linear) ซึ่งถ้าแก้ด้วยการทำให้เป็นดิสครีต แล้วประมาณค่าของอนุพันธ์ต่างๆ ด้วยวิธีผลต่างจำกัดเช่นเดียวกับที่ทำกับสมการคลื่นวิธีการที่ได้จะต้องใช้  $\Delta t$  ขนาดเล็กมากเพื่อความเสถียรจนไม่สามารถใช้งานได้ทางปฏิบัติ โดยพจน์หลักที่ก่อให้เกิดปัญหาคือพจน์  $u \cdot \nabla u$  งานวิจัย [27] จึงนำเสนอวิธีที่จะแก้สมการอนุพันธ์ของไหลด้วยการแบ่งโอเปอเรเตอร์ (Operator Splitting) ด้วยขั้นตอนดังนี้

1. เริ่มจาก  $u$  แก้พจน์  $u \cdot \nabla u$  ด้วยวิธีกึ่งลากранжеียน (Semi-Lagrangian Method) ให้ได้ความเร็วชั่วคราว  $u^*$
2. เริ่มจาก  $u^*$  แก้พจน์  $a^{\text{ext}}$  ด้วยการปรับความเร็วด้วยความเร่งจากภายนอกให้ได้  $u^{**}$
3. เริ่มจาก  $u^{**}$  แก้พจน์  $\frac{1}{\rho} \nabla p$  ด้วยการโปรเจคความดัน (Pressure Projection) ให้ได้ความเร็ว  $u$  ที่ไม่มีไดเวอร์เจนซ์ (Divergence Free) ของลำดับเวลาถัดไป

โดยรายละเอียดของแต่ละขั้นตอนเป็นดังนี้

ในขั้นที่หนึ่งพจน์ที่จะแก้คือ  $u \cdot \nabla u$  ซึ่งคือการที่ความเร็วนำพาความเร็วให้ไหลไปตามความเร็ว วิธีที่งานวิจัย [27] นำเสนอให้ใช้ในการแก้พจน์นี้คือการใช้วิธีกึ่งลากранжеียน โดยมีหลักการว่าความเร็ว ณ ตำแหน่ง  $x$  ณ ขั้นตอนเวลาในลำดับถัดไปนั้น คือความเร็ว ณ ตำแหน่ง  $x'$  ณ ขั้นตอนเวลาปัจจุบัน โดยที่  $x'$  คือตำแหน่งที่สมมุติว่ามีอนุภาคที่อยู่ที่นั่นในเวลาปัจจุบัน แล้วให้อนุภาคไหลไปตามความเร็ว โดยอนุภาคจะไปอยู่ที่ตำแหน่ง  $x$  ในขั้นตอนเวลาในลำดับถัดไปดังแสดงในภาพประกอบ 24



ภาพประกอบ 24: วิธีกึ่งลากรางเจียนโดยค่า  $u$  ที่  $x$  ในขั้นตอนเวลาถัดไปจะได้มาจากการประมาณค่าในช่วง ณ ตำแหน่ง  $x'$  จาก ค่า  $u$  ในขั้นตอนเวลาปัจจุบันที่เก็บอยู่ที่ ตำแหน่งที่โดยวงไว้

ซึ่งวิธีที่สามารถใช้หาตำแหน่ง  $x'$  นั้นทำได้โดยการปฏิยานุพันธ์เริ่มต้นจากตำแหน่ง  $x$  โดยใช้ผลติดลบของความเร็วปัจจุบันด้วยวิธีการออยเลอร์แบบไปข้างหน้า หรือ กฎของจุดกลางพอได้  $x'$  แล้วก็ทำการประมาณค่าในช่วง (Interpolate) โดยใช้ค่าที่เก็บอยู่ ณ ตำแหน่งสี่ตำแหน่งที่อยู่รอบ  $x'$  มาทำ Bilinear Interpolation ซึ่ง มีรหัสเทียมดังนี้

BilinearInterpolation\_u(x, y):

$$i_r = \frac{x}{\Delta x} - \frac{1}{2}$$

$$j_r = \frac{y}{\Delta y}$$

$$i = \lfloor i_r \rfloor$$

$$j = \lfloor j_r \rfloor$$

$$f_x = i_r - i$$

$$f_y = j_r - j$$

$$\text{return } (1 - f_y)((1 - f_x)u_{i+\frac{1}{2};j} + (f_x)u_{i+1+\frac{1}{2};j}) + f_y((1 - f_x)u_{i+\frac{1}{2};j+1} + (f_x)u_{i+1+\frac{1}{2};j+1})$$

BilinearInterpolation\_v(x,y):

$$i_r = \frac{x}{\Delta x}$$

$$j_r = \frac{y}{\Delta y} - \frac{1}{2}$$

$$i = \lfloor i_r \rfloor$$

$$j = \lfloor j_r \rfloor$$

$$f_x = i_r - i$$

$$f_y = j_r - j$$

$$\text{return } (1 - f_y)((1 - f_x)v_{i,j+\frac{1}{2}} + (f_x)v_{i+1,j+\frac{1}{2}}) + f_y((1 - f_x)v_{i,j+\frac{1}{2}+1} + (f_x)v_{i+1,j+\frac{1}{2}+1})$$

รหัสเทียมของขั้นตอนของวิธีกึ่งลากรางเจียน(แบบที่ใช้วิธีการออยเลอร์แบบไปข้างหน้าในการหา  $x'$  สามารถเขียนได้ดังนี้

SemiLagrangian\_u:

for all  $i, j$ :

$$v_x = u_{i+\frac{1}{2},j}$$

$$v_y = \frac{1}{4}(v_{i,j-1+\frac{1}{2}} + v_{i,j+\frac{1}{2}} + v_{i+1,j-1+\frac{1}{2}} + v_{i+1,j+\frac{1}{2}})$$

$$x' = \begin{bmatrix} (i + \frac{1}{2})\Delta x - v_x\Delta t \\ j\Delta x - v_y\Delta t \end{bmatrix}$$

$$u^*_{i+\frac{1}{2},j} = \text{BilinearInterpolation}_u(x'_x, x'_y):$$

SemiLagrangian\_v:

for all  $i, j$ :

$$v_x = \frac{1}{4}(u_{i-1+\frac{1}{2},j} + u_{i+\frac{1}{2},j} + u_{i-1+\frac{1}{2},j+1} + u_{i+\frac{1}{2},j+1})$$

$$v_y = v_{i,j+\frac{1}{2}}$$

$$x' = \begin{bmatrix} i\Delta x - v_x\Delta t \\ (j + \frac{1}{2})\Delta x - v_y\Delta t \end{bmatrix}$$

$$v^*_{i,j+\frac{1}{2}} = \text{BilinearInterpolation}_v(x'_x, x'_y):$$

ในขั้นที่สองจะนำความเร่งจากแรงภายนอกมาปรับความเร็วซึ่งสามารถใช้วิธีการออยเลอร์แบบไปข้างหน้าได้ดังนี้

$$\begin{aligned}u_{i+\frac{1}{2},j}^{**} &= u_{i+\frac{1}{2},j}^* + \Delta t a_x^{\text{ext}}{}_{i+\frac{1}{2},j} \\v_{i,j+\frac{1}{2}}^{**} &= v_{i,j+\frac{1}{2}}^* + \Delta t a_y^{\text{ext}}{}_{i,j+\frac{1}{2}}\end{aligned}$$

ในขั้นที่สามจะทำการโปรเจกต์ความดันโดยจะต้องหา  $p$  ที่เมื่อนำมาปรับสนามความเร็วแล้วสนามความเร็วจะมีไดเวอร์เจนซ์เป็น 0 หรือทางคณิตศาสตร์คือ

$$\begin{aligned}\nabla \cdot \left( u^{**} - \frac{\Delta t}{\rho} \nabla p \right) &= 0 \\ \frac{\Delta t}{\rho} \nabla^2 p &= \nabla \cdot u^{**} \\ \nabla^2 p &= \frac{\rho}{\Delta t} \nabla \cdot u^{**}\end{aligned}$$

ซึ่งสามารถทำการแปลงเป็นดิคริตด้วยวิธีผลต่างจำกัดได้ดังนี้

$$\begin{aligned}\frac{(p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1} - 4p_{i,j})}{\Delta x^2} &= \frac{\rho}{\Delta t} \frac{(u_{i+\frac{1}{2},j}^{**} - u_{i-1+\frac{1}{2},j}^{**} + v_{i,j+\frac{1}{2}}^{**} - v_{i,j-1+\frac{1}{2}}^{**})}{\Delta x} \\ (p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1} - 4p_{i,j}) &= \frac{\rho \Delta x}{\Delta t} (u_{i+\frac{1}{2},j}^{**} - u_{i-1+\frac{1}{2},j}^{**} + v_{i,j+\frac{1}{2}}^{**} - v_{i,j-1+\frac{1}{2}}^{**}) \\ 4p_{i,j} - p_{i+1,j} - p_{i-1,j} - p_{i,j+1} - p_{i,j-1} &= b_{i,j}\end{aligned}$$

โดยที่  $b_{i,j} = -\frac{\rho \Delta x}{\Delta t} (u_{i+\frac{1}{2},j}^{**} - u_{i-1+\frac{1}{2},j}^{**} + v_{i,j+\frac{1}{2}}^{**} - v_{i,j-1+\frac{1}{2}}^{**})$  และสามารถเขียนเป็นระบบสมการเชิงเส้นของ  $p_{i,j}$  ได้ซึ่งถ้าทำการแก้แล้วก็จะสามารถนำมาใช้ในการหา  $u$  ในขั้นตอนเวลาลัดไปได้ดังนี้

$$\begin{aligned}u_{i+\frac{1}{2},j} &= u_{i+\frac{1}{2},j}^{**} - \frac{\Delta t}{\rho \Delta x} (p_{i+1,j} - p_{i,j}) \\ v_{i,j+\frac{1}{2}} &= v_{i,j+\frac{1}{2}}^{**} - \frac{\Delta t}{\rho \Delta x} (p_{i,j+1} - p_{i,j})\end{aligned}$$

เป็นอันเสร็จสิ้นขั้นตอนในการจำลองของไหล

ระบบสมการเชิงเส้นเชิงเส้นของ  $p_{i,j}$  นั้นมีรายละเอียดปลีกย่อยในการจัดการกับขอบคือเมื่อต้องอ้างอิงถึง  $(i, j)$  ที่อยู่นอกตารางนั้นโดยมากจะสามารถทำได้สองแบบ แบบแรกคือถ้าขอบนั้นเป็นสิ่งกีดขวาง (Closed Boundary) ก็จะใช้ความดันของช่องที่อยู่ในตารางที่ใกล้ที่สุดมาแทนที่เช่นใช้  $p_{N-1,j}$  แทน  $p_{N,j}$  แบบที่สองคือถ้าขอบนั้นเป็นที่เปิด (Open Boundary) จะใช้ค่า 0 แทนเช่นใช้ 0 แทน  $p_{N,j}$

วิธีในการแก้ระบบสมการเชิงเส้นของ  $p_{i,j}$  นั้นมีหลากหลายวิธี วิธีที่ได้รับความนิยมและมีประสิทธิภาพในทางปฏิบัติคือวิธี Pre-Conditioned Conjugated Gradient [24] ซึ่งสามารถแก้ระบบที่มีจำนวนช่องมากๆ ได้ แต่วิธีที่เขียนโปรแกรมได้ง่ายมากและทำงานได้ดีพอสมควรในกรณีของการจำลองควันคือวิธี Jacobi Iteration ซึ่งมีขั้นตอนดังนี้

Loop K times:

$$\text{for all } i, j: p'_{i,j} = \frac{b_{i,j} + p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1}}{4}$$

$$\text{for all } i, j: p_{i,j} = p'_{i,j}$$

ซึ่งคือการเดาค่าของ  $p_{i,j}$  จากค่าของช่องรอบๆ แล้ววนไป K รอบเพื่อหวังว่า  $p_{i,j}$  จะเข้าใกล้ค่าที่ถูกต้องมากขึ้นเรื่อยๆ ซึ่งในกรณีของระบบสมการเชิงเส้นของความดันนี้ ถ้า  $K \rightarrow \infty$  ก็จะได้คำตอบที่ถูกต้อง แต่ในทางปฏิบัติสำหรับการจำลองควันอาจใช้เพียง K ประมาณ 40 - 100 รอบก็มักได้ผลลัพธ์ที่ยอมรับได้

## บทสรุป

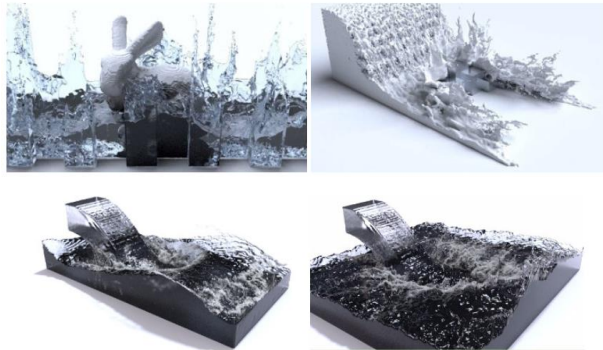
บทนี้กล่าวถึงการจำลองทางฟิสิกส์ด้วยตาราง ซึ่งมักใช้กับของไหลโดยเริ่มจากสมการเชิงอนุพันธ์ย่อยนาเวียร์-สโตกส์ของของไหลที่มีความหนืดคงที่ และไม่สามารถบีบอัดได้ แล้วใช้การทำให้เป็นดิสครีตและเก็บสถานะของระบบไว้ด้วยตาราง จากนั้นจึงทำการแก้พจน์ต่างๆ ของสมการเชิงอนุพันธ์ย่อย ด้วยการใช่วิธีผลต่างจำกัดเพื่อประมาณค่าของอนุพันธ์ หรือวิธีอื่นเช่นวิธีกึ่งลากรางเจียน โดยแสดงการจำลองน้ำด้วยสมการคลื่นซึ่งลดรูปจากสมการเชิงอนุพันธ์ย่อยนาเวียร์-สโตกส์ จากนั้นจึงกล่าวถึงการจำลองควัน ซึ่งจะต้องใช้การแก้สมการเชิงเส้นในการหาความดัน การจำลองโดยใช้ตารางนั้นเหมาะใช้กับการจำลองในพื้นที่ที่มีขอบเขตตายตัว ในแง่การเขียนโปรแกรมนั้นการเก็บข้อมูลในตารางมีรูปแบบคงที่จึงทำให้การอ่านเขียนหน่วยความจำสามารถทำได้อย่างมีประสิทธิภาพ

## บทที่ 6: การจำลองทางฟิสิกส์แบบผสมตารางและอนุภาค

บทที่แล้วกล่าวถึงวิธีการจำลองของไหลด้วยตาราง ซึ่งสำหรับการจำลองน้ำนั้นจะพิจารณาเฉพาะน้ำที่เป็นสภาวะความสูงเท่านั้นซึ่งไม่สามารถจำลองน้ำที่กระเซ็นขึ้นมาหรือม้วนตัวได้ ในบทนี้จะกล่าวถึงวิธีการจำลองน้ำโดยกรนำข้อดีของตารางและอนุภาคมาใช้เพื่อให้ได้ผลลัพธ์ที่ดีขึ้น

### วิธี Fluid-Implicit Particle (FLIP)

วิธีการจำลองน้ำแบบผสมตารางและอนุภาคที่ได้รับความนิยม และมีความสมจริงเป็นอย่างมากคือวิธี FLIP (Fluid-Implicit-Particle) [28] ซึ่งถูกนำมาใช้ในการจำลองน้ำและทรายในคอมพิวเตอร์กราฟฟิกในงานวิจัย [29] หลักการสำคัญของวิธี FLIP คือการใช้ตารางในการโปรเจกต์ความดัน ซึ่งได้ผลดีมากในการทำให้สภาวะความเร็วไม่มีไดเวอร์เจนท์ และใช้อนุภาคในการเก็บความเร็วแทนตารางเพื่อแก้พจน์  $\mathbf{u} \cdot \nabla \mathbf{u}$  แทนวิธีกึ่งตารางเจียนเพื่อลดความหนืดเทียมและใช้อนุภาคเป็นตัวบอกว่าในบริเวณไหนมีน้ำอยู่บ้าง เพื่อนำไปใช้สร้างระบบสมการเชิงเส้นของความดัน และใช้ในการแสดงผลด้วยดั่งตัวอย่างที่แสดงใน ภาพประกอบ 25



ภาพประกอบ 25: ตัวอย่างการจำลองน้ำด้วยวิธี FLIP เครดิต Copyright (c) 2016 Ryan L. Guy

<https://github.com/rlguy/GridFluidSim3D>

วิธี FLIP นั้นจะใช้อนุภาคเป็นหลักในการเก็บสถานะของระบบ โดยแต่ละอนุภาคจะเก็บตำแหน่งและความเร็วเอาไว้ และจะใช้ตารางแบบสายชั่วคราวในแต่ละขั้นตอนเวลา เพื่อช่วยในการโปรเจกต์ความดัน โดยจะต้องใช้ความดัน  $(p_{i,j})$  ความเร็วทางแกน  $x$  ( $u_{i+\frac{1}{2},j}$ ) ความเร็วทางแกน  $y$  ( $v_{i,j+\frac{1}{2}}$ )

ขั้นตอนของวิธี FLIP ในการจำลองน้ำเป็นดังนี้

1. กำหนดตำแหน่งเริ่มต้น และความเร็วเริ่มต้นของอนุภาคต่างๆ
2. สำหรับแต่ละขั้นตอนเวลา
  - a) ใช้การประมาณค่าในช่วงเพื่อหา  $u_{i+\frac{1}{2},j}$  และ  $v_{i,j+\frac{1}{2}}$  จากความเร็วที่เก็บบนอนุภาค
  - b) แก้พจน์  $a^{ext}$  ด้วยการปรับความเร็วด้วยความเร่งจากภายนอกให้ได้  $u^*_{i+\frac{1}{2},j}$ ,  $v^*_{i,j+\frac{1}{2}}$
  - c) แก้พจน์  $\frac{1}{\rho} \nabla p$  ด้วยการโปรเจกต์ความดัน (Pressure Projection) ให้ได้ความเร็ว  $u^{**}_{i+\frac{1}{2},j}$ ,  $v^{**}_{i,j+\frac{1}{2}}$  ที่ไม่มีไดเวอร์เจนซ์ (Divergence Free) ของลำดับเวลาถัดไป
  - d) สำหรับแต่ละอนุภาค ใช้การประมาณค่าในช่วงเพื่อหาผลต่างของความเร็วจาก  $u^{**}_{i+\frac{1}{2},j} - u_{i+\frac{1}{2},j}$
  - e) และ  $v^{**}_{i,j+\frac{1}{2}} - v_{i,j+\frac{1}{2}}$  แล้วนำมาบวกกับความเร็วปัจจุบันของอนุภาค
  - f) เคลื่อนที่อนุภาคด้วยการปฏิยานุพันธ์เชิงเวลา โดยใช้กฎจุดกลางหรือวิธีอื่นโดยใช้สนามความเร็ว  $u^{**}_{i+\frac{1}{2},j}$ ,  $v^{**}_{i,j+\frac{1}{2}}$

โดยรายละเอียดแต่ละขั้นตอนเป็นดังนี้

ในขั้นตอนที่ 1 ใช้การสุ่มตำแหน่งของอนุภาคภายในแต่ละช่องที่เป็นของเหลวโดยแบ่งแต่ละช่องเป็นตารางย่อย (Sub Grid) ขนาด  $2 \times 2$  (ใน 2 มิติ) หรือ  $2 \times 2 \times 2$  (ใน 3 มิติ) โดยสุ่มตำแหน่งของอนุภาค 1 อนุภาคภายในแต่ละช่องของตารางย่อย วิธีนี้จะทำให้มีจำนวนอนุภาคไม่น้อยไปและไม่มากเกินไป

ในขั้นตอน 2a ใช้การประมาณค่าในช่วงเพื่อหาสนามความเร็วจากอนุภาคโดยใช้น้ำหนักจาก Bilinear Interpolation ดังนี้

Splat\_u(x, y, v\_x):

$$i_r = \frac{x}{\Delta x} - \frac{1}{2}$$

$$j_r = \frac{y}{\Delta x}$$

$$i = [i_r]$$

$$\begin{aligned}
j &= \lfloor j_r \rfloor \\
f_x &= i_r - i \\
f_y &= j_r - j \\
u_{i+\frac{1}{2},j}^w &+= (1 - f_x)(1 - f_y) \\
u_{i+\frac{1}{2},j} &+= (1 - f_x)(1 - f_y)v_x \\
u_{i+1+\frac{1}{2},j}^w &+= f_x(1 - f_y) \\
u_{i+1+\frac{1}{2},j} &+= f_x(1 - f_y)v_x \\
u_{i+\frac{1}{2},j+1}^w &+= (1 - f_x)f_y \\
u_{i+\frac{1}{2},j+1} &+= (1 - f_x)f_yv_x \\
u_{i+1+\frac{1}{2},j+1}^w &+= f_xf_y \\
u_{i+1+\frac{1}{2},j+1} &+= f_xf_yv_x
\end{aligned}$$

Splat\_v(x, y, v\_y):

$$\begin{aligned}
i_r &= \frac{x}{\Delta x} \\
j_r &= \frac{y}{\Delta y} - \frac{1}{2} \\
i &= \lfloor i_r \rfloor \\
j &= \lfloor j_r \rfloor \\
f_x &= i_r - i \\
f_y &= j_r - j \\
v_{i,j+\frac{1}{2}}^w &+= (1 - f_x)(1 - f_y) \\
v_{i,j+\frac{1}{2}} &+= (1 - f_x)(1 - f_y)v_y \\
v_{i+1,j+\frac{1}{2}}^w &+= f_x(1 - f_y) \\
v_{i+1,j+\frac{1}{2}} &+= f_x(1 - f_y)v_y
\end{aligned}$$

$$v_{i,j+1+\frac{1}{2}}^w = (1 - f_x)f_y$$

$$v_{i,j+1+\frac{1}{2}} = (1 - f_x)f_y v_y$$

$$v_{i+1,j+1+\frac{1}{2}}^w = f_x f_y$$

$$v_{i+1,j+1+\frac{1}{2}} = f_x f_y v_y$$

กำหนดค่าทุกๆ  $u_{i+\frac{1}{2},j}, u_{i+\frac{1}{2},j}^w, v_{i,j+\frac{1}{2}}, v_{i,j+\frac{1}{2}}^w$  เป็น 0

สำหรับแต่ละอนุภาค i:

$$\text{Splat}_u(x_{ix}, x_{iy}, v_{ix})$$

$$\text{Splat}_v(x_{ix}, x_{iy}, v_{iy})$$

กำหนดค่าทุกๆ  $u_{i+\frac{1}{2},j}/= u_{i+\frac{1}{2},j}^w$  และ  $v_{i,j+\frac{1}{2}}/= v_{i,j+\frac{1}{2}}^w$

ขั้นตอน 2b นั้นทำเหมือนควีนในบทที่แล้วกล่าวคือ

$$u_{i+\frac{1}{2},j}^* = u_{i+\frac{1}{2},j} + \Delta t a_x^{\text{ext}}_{i+\frac{1}{2},j}$$

$$v_{i,j+\frac{1}{2}}^* = v_{i,j+\frac{1}{2}} + \Delta t a_y^{\text{ext}}_{i,j+\frac{1}{2}}$$

ขั้นตอน 2c จะถือว่าช่องที่มีอนุภาคน้อย 1 อนุภาคเป็นบริเวณน้ำ และจะต้องคำนวณหาความดันโดยจะต้องทำการขยาย (Extrapolate) สนามความเร็วเข้าไปในบริเวณที่เป็นอากาศ 1 ช่องด้วยเนื่องจากจะต้องใช้ในการคำนวณได้เวอร์เจนส์ ซึ่งทำได้โดยดูแต่ละ  $u_{i+\frac{1}{2},j}^*$  และ  $v_{i,j+\frac{1}{2}}^*$  ที่เป็นอากาศว่ามีความเร็วที่ติดกันอันไหนบ้างที่เป็นบริเวณน้ำ แล้วนำค่าเฉลี่ยมาใช้

สำหรับบริเวณที่เป็นของแข็งก็นำความเร็วของแข็งมาใช้ แล้วก็ทำการโปรเจกความดันเหมือนกับในบทที่แล้ว (แต่อาจจะต้องแก้ระบบสมการเชิงเส้นให้มีความแม่นยำสูงโดยการใช้จำนวนรอบของ Jacobi Iteration มากๆ หรือวิธี Conjugated Gradient เพื่อให้หน้าไม่สูญเสียปริมาตร) ให้ได้ความเร็ว  $u_{i+\frac{1}{2},j}^{**}$  และ  $v_{i,j+\frac{1}{2}}^{**}$  จากนั้นก็ทำการขยายความเร็วเข้าไปในบริเวณอากาศ 4-5 ช่องด้วยวิธีที่อธิบายไปก่อนหน้านี้ 4-5 รอบโดยแต่ละรอบหา

ค่า  $u^{**}_{i+\frac{1}{2},j}$  และ  $v^{**}_{i,j+\frac{1}{2}}$  ของช่องที่เป็นอากาศโดยเฉลี่ยจากเฉพาะช่องที่เป็นน้ำ หรือช่องที่เป็นอากาศที่หาความเร็วได้จากรอบที่แล้ว

ขั้นตอน 2d นั้นก็ใช้ Bilinear Interpolation ในการหา  $u^{**}_{i+\frac{1}{2},j} - u_{i+\frac{1}{2},j}$  และ  $v^{**}_{i,j+\frac{1}{2}} - v_{i,j+\frac{1}{2}}$  ณ แต่ละตำแหน่งของอนุภาคแล้วนำมาบวกเข้าไปที่  $v_i$

ขั้นตอน 2e ก็คือการเคลื่อนที่อนุภาคด้วยการปฏิยานุพันธ์เชิงเวลาโดยใช้กฎจุดกลาง หรือวิธีอื่นโดยการหาความเร็วที่ต้องใช้ด้วยการทำ Bilinear Interpolation จากสนามความเร็ว  $u^{**}_{i+\frac{1}{2},j}$ ,  $v^{**}_{i,j+\frac{1}{2}}$  เป็นอันจบการทำงานในขั้นตอนเวลานี้

## บทสรุป

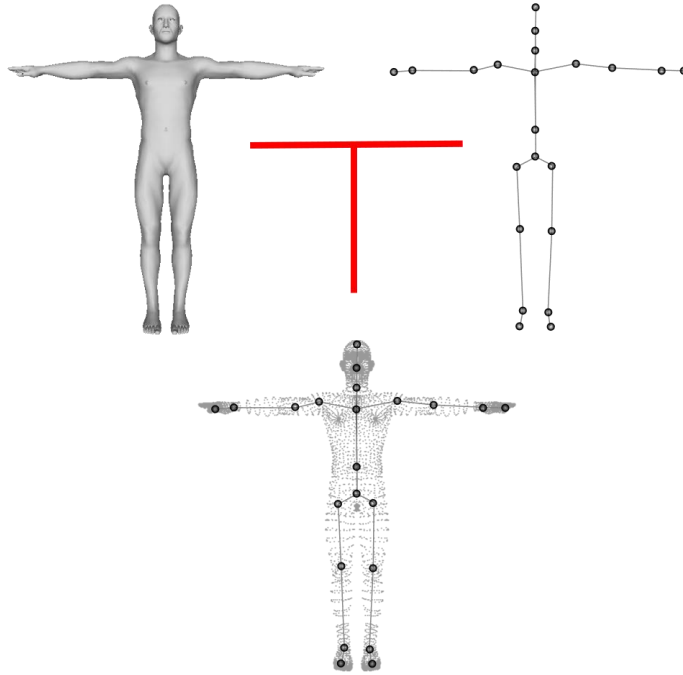
บทนี้กล่าวถึงการจำลองฟิสิกส์ด้วยตารางผสมอนุภาคซึ่งนำข้อดีของตารางซึ่งคือการทำให้องค์ของเหลวไม่บีบอัดได้ดี และข้อดีของอนุภาคซึ่งคือที่สามารถหาค่าความหนืดเทียบของการนำพาความเร็วให้ไหลไปตามความเร็วและการที่สามารถกำหนดบริเวณที่มีของเหลวอยู่ได้อย่างแม่นยำ มาประสานกันเพื่อจำลองของเหลวให้ได้ผลที่สมจริงอย่างมีประสิทธิภาพ โดยผู้ที่สนใจสามารถนำวิธีปรับแก้สมการความดันให้นำตำแหน่งของผิวของของเหลว [30] และรูปร่างของของแข็ง [31] มาใช้ในการกำหนดสมการเพื่อให้ได้ผลที่มีความสมจริงขึ้นโดยไม่ต้องใช้เวลาในการคำนวณมากขึ้นเท่าไรนัก

## บทที่ 7: การจำลองตัวละครด้วยฟิลิกส์

การสร้างแอนิเมชันของตัวละครนั้น มีความสำคัญอย่างยิ่งทั้งในภาพยนตร์และเกม เนื่องจากตัวละครเป็นตัวละครเป็น ตัวเดินเรื่องและสามารถสื่อสารกับผู้ชมได้เป็นอย่างดี ซึ่งการสร้างแอนิเมชันของตัวละครที่มีคุณภาพสูงนั้นมักมีความ ซับซ้อนมาก ด้วยเพราะมนุษย์นั้นเห็นการเคลื่อนไหวต่างๆของคนและสัตว์ในชีวิตประจำวันบ่อยๆ จึงทำให้การ เคลื่อนไหวที่ดูไม่เป็นธรรมชาตินั้นจะสะดุดตาผู้ชมได้ง่าย โดยมากแล้วการสร้างแอนิเมชันของตัวละครจะทำโดย นักสร้างภาพเคลื่อนไหว (Animator) โดยอาจจะใช้การวาดลงกระดาษเป็น 2 มิติ หรือใช้โปรแกรมคอมพิวเตอร์ เพื่อสร้างตัวละครและกำหนดท่าทางต่างๆ เป็น 2 มิติหรือ 3 มิติก็ได้โดยในปัจจุบันการสร้างตัวละครเป็น 3 มิตินั้น เป็นที่นิยมเนื่องจากขนาด สัดส่วน และท่าทางต่างๆ จะสามารถอ้างอิงจากโลกความจริงได้ง่าย สามารถเปลี่ยน มุมมองได้ สามารถที่จะปรับแก้ไขได้ง่าย และยังสามารถแสดงผลให้ดูเหมือนวาดเป็น 2 มิติได้อีกด้วย ดังนั้น หนังสือเล่มนี้จึงจะกล่าวถึงพื้นฐานของการสร้างแอนิเมชันของตัวละคร 3 มิติเท่านั้น โดยเริ่มจากเนื้อหาที่เกี่ยวข้อง กับแอนิเมชันของตัวละครแบบไม่ใช้ฟิลิกส์ จากนั้นจะเป็นพื้นฐานของโมชันแคปเจอร์ จากนั้นจึงเป็นการจำลองตัว ละครด้วยฟิลิกส์ต่อไป

### แอนิเมชันของตัวละครแบบพื้นฐานโดยไม่ใช้ฟิลิกส์

ในการสร้างตัวละครแบบไม่ใช้ฟิลิกส์นั้น โดยทั่วไปแล้วศิลปินจะสามารถใช้ซอฟต์แวร์ต่างๆ เช่น Autodesk 3D Studio Max, Autodesk Maya หรือ Blender หรือภายในสตูดิโอชั้นนำเช่น Pixar, Dreamworks ก็จะมี ซอฟต์แวร์เฉพาะเป็นของตัวเอง แต่ในเบื้องต้น หลักการในการสร้างแอนิเมชันของตัวละครนั้นมีความคล้ายคลึงกัน โดยตัวละครมักจะถูกเก็บด้วยเมชโพลีกอน (Polygon Mesh) หรือตัวแทนของรูปทรงสามมิติอื่นๆ (3D Geometry Representation) เช่น Subdivision Surface, NURBS Surface (Non-Uniform Rational B-Spline Surface) และมักมีพื้นผิว (Texture) ที่ถูกนำมาใช้ในการแสดงผล โดยเมชโพลีกอน หรือตัวแทนของ รูปทรงสามมิติอื่นๆ นั้นจะมีข้อมูลตำแหน่งอยู่ เช่นในกรณีของเมชโพลีกอนนั้นจะเป็นตำแหน่งของ จุดยอด (Vertex) และตำแหน่งของจุดยอดนั้นจะถูกควบคุมด้วยกระดูก (Bone) ซึ่งกระดูกแต่ละท่อนจะเก็บตำแหน่งและ การวางแนวดังแสดงในภาพประกอบ 26



ภาพประกอบ 26: การ Skinning ซึ่งคือการที่ตำแหน่งของจุดยอดของเมชโพลีกอนของตัวละครถูกคำนวณจากตำแหน่งและการวางแนวของกระดูก ดัดแปลงจาก [32] โมเดลคนจาก Turbosquid

ซึ่งการคำนวณตำแหน่งของจุดยอดจากกระดูกนั้นเรียกว่าการสกินนิง (Skinning) ซึ่งมักจะทำได้ด้วยวิธี Linear Blend Skinning (LBS) โดยถ้า ณ เวลา  $t$  กระดูก  $i$  มีตำแหน่ง  $p_i(t)$  และการวางแนว  $R_i(t)$  แล้ว และจุดยอด  $j$  มีพิกัดในเนื้อที่ร่างกายของกระดูก  $i$  เป็น  $q_{ij}$  และกระดูก  $i$  มีความสำคัญกับจุดยอด  $j$  เป็นค่า  $w_{ij}$  (ศิลปินจะต้องระบุค่าของ  $w_{ij}$  ด้วยการระบายน หรือไม่ก็ใช้การคำนวณทางคณิตศาสตร์ เช่นตามระยะห่างระหว่างกระดูกกับจุดยอด) โดยที่  $\sum_i w_{ij} = 1$  แล้วจะสามารถคำนวณตำแหน่งของจุดยอด  $j$  ได้โดย

$$\sum_i w_{ij} (p_i(t) + R_i(t)q_{ij})$$

ในความเป็นจริงแล้ว การสกินนิงนั้นสามารถใช้วิธีซับซ้อนกว่านี้ได้เช่น Dual Quaternion [33] ซึ่งอาจได้ผลที่มีความต่อเนื่องมากขึ้น แต่ว่าการสกินนิงด้วย Linear Blend Skinning นั้นก็ยังคงถูกใช้อย่างแพร่หลาย

การสร้างภาพเคลื่อนไหวของตัวละครในเบื้องต้นนั้นก็คือการระบุค่าของ  $p_i(t)$  และ  $R_i(t)$  ณ เวลาต่างๆ นั้นเอง ซึ่งในทางปฏิบัติแล้วศิลปินจะใช้โปรแกรม 3 มิติเพื่อระบุค่าของ  $p_i(t)$  และ  $R_i(t)$  ผ่านทางส่วนต่อประสานกับผู้ใช้ (User Interface) และมักจะระบุเพียงค่าที่ ณ จุดเวลาที่สำคัญๆ เท่านั้นซึ่งเรียกว่า คีย์เฟรม (Key Frame) สมมติว่าจุดเวลาที่ศิลปินกำหนดค่าคือ  $t_0 < t_1 < \dots < t_{m-1}$  แล้วค่าของ  $p_i(t)$  และ  $R_i(t)$  ณ เวลา  $t_0 < t < t_{m-1}$  ใดๆจะหาได้จากการประมาณค่าในช่วงจากค่าของ  $p_i(t_k)$  และ  $R_i(t_k)$  ด้วยวิธีการประมาณค่าต่างๆ เช่น การประมาณค่าในช่วงเชิงเส้น (Linear Interpolation) สำหรับ  $p_i$  และการประมาณค่าในช่วงเชิงเส้นทรงกลม (Spherical Linear Interpolation, SLERP) สำหรับ  $R_i$  ดังนี้ สำหรับค่า  $t_k < t < t_{k+1}$  นั่นคือ

$$p_i(t) \approx (1 - f) p_i(t_k) + f p_i(t_{k+1})$$

$$q_i(t) \approx \frac{\sin((1-f)\omega)}{\sin(\omega)} q_i(t_k) + \frac{\sin(f\omega)}{\sin(\omega)} q_i(t_{k+1})$$

เมื่อ  $f = \frac{t - t_k}{t_{k+1} - t_k}$  และ  $q_i(t_k)$  คือควอดเทอร์เนียนของ  $R_i(t_k)$  และ  $\omega = \text{acos}(q_i(t_k) \cdot q_i(t_{k+1}))$

โดยถ้า  $q_i(t_k) \cdot q_i(t_{k+1}) < 0$  จะต้องปรับ  $q_i(t_{k+1}) \leftarrow -q_i(t_{k+1})$  ก่อน หลังจากนั้นก็คำนวณค่า  $R_i(t)$  ซึ่งเป็นเมทริกซ์หมุนของ  $q_i(t)$  จากนั้นก็การใช้การสกินนิง เพื่อคำนวณหาตำแหน่งของจุดยอดต่างๆ ของเมชโพลีกอน ณ เวลา  $t$  ได้และนำไปแสดงผล ซึ่งขั้นตอนการประมาณค่าในช่วงนี้จะทำหน้าที่เดียวกับการวาดภาพระหว่างคีย์เฟรม (Inbetweening) ซึ่งเป็นงานที่ต้องทำซ้ำ และมีปริมาณมากของภาพเคลื่อนไหวที่สร้างโดยไม่ใช้คอมพิวเตอร์

การกำหนดค่าของ  $p_i(t)$  และ  $R_i(t)$  ณ เวลาต่างๆ เพื่อสร้างภาพเคลื่อนไหวนั้นอาจจะไม่จำเป็นต้องอาศัยศิลปินเสมอไป แต่สามารถที่จะให้คนแสดงแล้วทำการถ่ายทำเพื่อนำมาใช้ในคอมพิวเตอร์ได้ ซึ่งเรียกว่า โมชันแคปเจอร์ (Motion Capture)

## พื้นฐานของโมชันแคปเจอร์

โมชันแคปเจอร์คือกระบวนการในการบันทึกการเคลื่อนไหวของคน สัตว์ หรือสิ่งของเพื่อนำเข้ามาในคอมพิวเตอร์ ซึ่งโดยมากแล้วจะใช้เพื่อการนำการเคลื่อนไหวของนักแสดงมาใช้กับตัวละครในคอมพิวเตอร์ โดยมีวิธีที่ใช้ในการบันทึกหลากหลายแบบ กล่าวคือ

### โมชันแคปเจอร์แบบแม่เหล็ก (Magnetic Motion Capture)

ใช้เซ็นเซอร์สนามแม่เหล็ก (Magnetic Sensor) ในการวัดความเข้มสัมพัทธ์ของสนามแม่เหล็กที่ตั้งฉากกันในสามแกน ซึ่งจะทำให้ได้ข้อมูลทั้งตำแหน่ง และการวางแนวของแต่ละกระดูกที่แม่นยำใน ณ แต่ละจุดเวลา โดยนักแสดงจะสวมชุดที่มีเซ็นเซอร์อยู่ใกล้ข้อต่อต่างๆ โดยข้อดีของโมชันแคปเจอร์แบบแม่เหล็กคือ สามารถบันทึกข้อมูลได้ในเวลาจริง ไม่จำเป็นต้องใช้ที่มาก ได้ข้อมูลทั้งตำแหน่งและการวางแนว และราคาไม่สูงมาก ข้อเสียคือ มีน้ำหนักมาก มีสายระโยงระยาง และอาจถูกรบกวนจากโลหะได้ (เช่น คานเหล็ก หรือ เส้า) จึงอาจต้องใช้งานบนเวทีที่ไม่มีโลหะเป็นส่วนประกอบ

### โมชันแคปเจอร์แบบกล (Mechanics Motion Capture)

ในการใช้โมชันแคปเจอร์แบบกลนั้น นักแสดงจะสวมอุปกรณ์ที่เป็นเหมือนโครงกระดูกที่มีข้อต่ออยู่ให้แนบกับหลังและแขนขา โดยแต่ละข้อต่อจะวัดการหมุนสัมพัทธ์ โดยก่อนใช้งานจะต้องทำการปรับเทียบ (Calibrate) ก่อนเพื่อให้ได้ค่าเฉพาะตัวของนักแสดงแต่ละคน ข้อดีของโมชันแคปเจอร์แบบกล คือจะไม่มีการรบกวนจากแสง หรือสนามแม่เหล็ก และไม่จำเป็นต้องอยู่ในบริเวณจำกัด แต่ข้อเสียคือข้อมูลที่ได้เป็นเพียงมุมของแต่ละข้อต่อเท่านั้น และไม่มีตำแหน่งสัมบูรณ์ (Absolute Position) การเคลื่อนไหวของนักแสดงอาจไม่เป็นธรรมชาติ และข้อมูลที่ได้อาจมีข้อผิดพลาดสะสม (Accumulated Error) เมื่อใช้อุปกรณ์เป็นเวลานานจึงอาจต้องทำการปรับเทียบบ่อย

### โมชันแคปเจอร์แบบแสง (Optical Motion Capture)

โมชันแคปเจอร์แบบแสงนั้นเป็นวิธีที่ใช้แพร่หลายที่สุดโดยหลักการคือจะติดมาร์คเกอร์ (Marker) ซึ่งมีลักษณะเป็นทรงกลมซึ่งอาจมีแสงในตัวหรือไม่ก็ได้ที่ตำแหน่งต่างๆ บนตัวนักแสดง (โดยมากมักจะติดบนชุดบอดี้สูท) และใช้กล้องสองตัวหรือมากกว่าในการถ่ายภาพการเคลื่อนไหวของนักแสดง ซึ่งจะสามารถคำนวณตำแหน่งใน 3 มิติของมาร์คเกอร์ได้จากตำแหน่ง 2 มิติของมาร์คเกอร์ในวิธีที่ค้นที่ถ่ายจากกล้องต่างๆ ได้ โดยข้อดีของโมชันแคปเจอร์

แบบแสงก็คือ ชุดบอดี้สูทและมาร์คเกอร์นั้นมึ้น้ำหนักเบาไม่เทอะทะ นักแสดงสามารถแสดงได้เต็มที่ในพื้นที่กว้าง และข้อมูลมีความถูกต้องสูง ส่วนข้อเสียก็คือ ราคาที่ค่อนข้างแพง ในบางมุกกล้องมาร์คเกอร์อาจโดนบังโดยนักแสดงหรือสิ่งของต่างได้ และอาจถูกรบกวนจากแสงภายนอกได้

โดยข้อมูลที่ได้จากโมชันแคปเจอร์ในรูปแบบต่างๆ จะถูกนำมาคำนวณหา  $p_i(t)$  และ  $R_i(t)$  ณ เวลาต่างๆ แล้วนำไปใช้สกินนิ่งเพื่อให้ได้ตำแหน่งจุดยอดเพื่อนำไปแสดงผลต่อไป

ในเกมคอมพิวเตอร์นั้นไม่ว่าจะเป็น ภาพเคลื่อนไหวที่กำหนดโดยศิลปินก็ดี หรือการใช้โมชันแคปเจอร์ก็ดี ก็จะมีจำนวนจำกัดและอาจไม่สามารถครอบคลุมทุกการใช้งานได้ ซึ่งวิธีที่ใช้แก้ปัญหานี้อย่างแพร่หลายคือการทำโมชันแมชชิง (Motion Matching) ซึ่งก็คือการเลือกว่าจะทำการเปลี่ยนผ่าน (Transition) จากภาพเคลื่อนไหวปัจจุบัน ณ เฟรมใดไปยังภาพเคลื่อนไหวใด ณ ที่เฟรมไหน ถึงจะมีความต่อเนื่องเหมาะสม ซึ่งจะพิจารณาด้วยคะแนนความเข้ากันได้ (Matching Score) โดยจะต้องนำ ตำแหน่ง การวางแนว ความเร็ว และความเร็วเชิงมุมของกระดูกต่างๆ มาใช้ในการพิจารณา จากนั้นจึงทำการประมาณค่าในช่วงเพื่อผสม (Blend) ส่วนของภาพเคลื่อนไหวสองอันนี้เข้าด้วยกัน ตัวอย่างของงานวิจัยที่กล่าวถึงเรื่องนี้คือ Motion Graphs [34] ซึ่งทำการแบ่งภาพเคลื่อนไหวเป็นช่วงสั้นๆ แล้วทำการคำนวณหาว่าภาพเคลื่อนไหวส่วนไหนเหมาะสมที่จะเปลี่ยนผ่านไปเป็นภาพเคลื่อนไหวไหนบ้าง ณ ช่วงเวลาไหนบ้าง ซึ่งสามารถที่จะสร้างภาพเคลื่อนไหวที่มีความยาวมากๆ ได้

อย่างไรก็ดีการสร้างภาพเคลื่อนไหวด้วยโมชันแคปเจอร์ หรือที่ศิลปินสร้างก็มีข้อจำกัดว่าการเคลื่อนไหวของตัวละครจะไม่สามารถตอบสนองกับสิ่งแวดล้อมได้ เช่นถ้ามีหินตกใส่ตัวละครก็จะทะลุตัวละครไป หรือตัวละครเดินออกจากกระเป๋าก็จะลอยบนอากาศ หรือตัวละครเดินชนหินก็จะทะลุไม่สะดุด นอกเสียจากได้มีการสร้างภาพเคลื่อนไหวที่เหมาะสมเอาไว้ล่วงหน้าเท่านั้น ซึ่งกล่าวรวมๆ คือตัวละครจะไม่เคลื่อนที่ตามกฎฟิสิกส์นั่นเอง ในปัจจุบันจึงได้มีการนำการจำลองทางฟิสิกส์มาใช้ในการคำนวณการเคลื่อนไหวของตัวละครด้วย เพื่อเพิ่มความสมจริง

## พื้นฐานของการจำลองตัวละครด้วยฟิสิกส์

การใช้ฟิสิกส์ในการจำลองตัวละครที่พบบ่อยที่สุดนั่นคือ การจำลองตุ๊กตาผ้า (Ragdoll Simulation) ซึ่งคือการใช้วัตถุแก่แทนอวัยวะต่างๆ เช่น แขน ขา หัว ลำตัว แล้วใช้ข้อต่อเชื่อมอวัยวะส่วนต่างๆ เข้าด้วยกันดังในภาพประกอบ 27 จากนั้นจึงจำลองด้วยการจำลองวัตถุแก่แบบมีข้อต่อให้หล่นลงมาชนกับสิ่งกีดขวางต่างๆ ซึ่งการจำลองตุ๊กตาผ้านี้มักใช้ในการสร้างการเคลื่อนไหวของตัวละครขณะตกบันได หรือถูกแรงกระแทกมากๆ เช่น รถชน เป็นต้น



ภาพประกอบ 27 ตัวอย่างการจำลองตุ๊กตาผ้า (Ragdoll Simulation) เครดิต: <http://www.animats.com/>

การเคลื่อนที่ขณะตกจากที่สูง ขณะล้มลง หรือขณะกระเด็นนั้นเกิดขึ้นในเวลาอันรวดเร็วจนเกือบจะไม่มีเวลาในการใช้เจตจำนงในการเคลื่อนไหว ซึ่งการที่ประมาณว่าเป็นการเคลื่อนที่ตามกฎฟิสิกส์ของวัตถุแก่ที่เชื่อมด้วยข้อต่อ นั้น เป็นการประมาณการที่ได้ผลค่อนข้างจะสมจริง

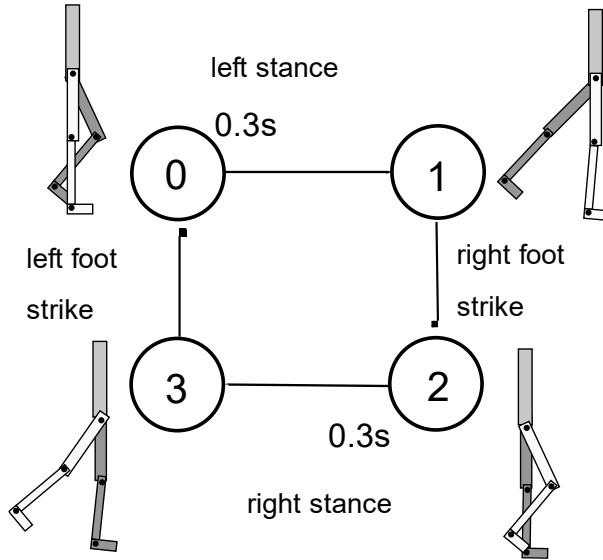
ส่วนการจำลองตัวละครด้วยฟิสิกส์ที่ตัวละครเคลื่อนไหวตามเจตจำนงเช่น เดิน วิ่ง คลานที่สามารถตอบสนองกับสิ่งแวดล้อมภายนอกโดยใช้ฟิสิกส์นั้นมีความซับซ้อนค่อนข้างมาก และยังเป็นหัวข้อที่ยังคงมีผู้วิจัยอยู่ในปัจจุบัน เนื่องจากเกี่ยวข้องกับการใช้ปัญญาประดิษฐ์ค่อนข้างมาก ในบทความนี้จะกล่าวถึงการควบคุมการเคลื่อนที่ของตัวละครอย่างง่ายเท่านั้น

สมมุติว่าตัวละครของแทนด้วยวัตถุแกร่งที่เชื่อมกันด้วยข้อต่อ การที่ตัวละครจะเคลื่อนไหวตามกฎของฟิสิกส์นั้น โมเมนตัมและโมเมนตัมเชิงมุมที่เกิดจากแรงภายในของตัวละครจะต้องเป็นศูนย์เสมอ เนื่องจากแรงที่เกิดจากภายในระบบไม่สามารถทำให้เกิดงานได้ ดังนั้นแรงบิดที่กระทำกับวัตถุแกร่งสองอันที่เชื่อมด้วยข้อต่อใดๆ จะต้องหักล้างกันพอดี ดังนั้นที่ตัวละครจะเคลื่อนที่ได้มันจะต้องเกิดจากการถ่ายเปลี่ยนโมเมนตัมกับวัตถุภายนอกเสมอ ตัวอย่างเช่นตอนมนุษย์เดินไปข้างหน้า เขาก็ไปผลักพื้นให้ไปข้างหลังผ่านแรงเสียดทาน แต่เนื่องจากมวลของโลกเยอะกว่ามนุษย์มากผู้สังเกตภายนอกจึงเพียงเห็นว่ามนุษย์เคลื่อนไปข้างหน้าเท่านั้น และอีกข้อสังเกตคือมนุษย์จะเดินบนพื้นมีสัมประสิทธิ์แรงเสียดทานต่ำมากๆ เช่นพื้นน้ำแข็ง ได้ด้วยความยากลำบาก เนื่องจากการถ่ายเปลี่ยนโมเมนตัมผ่านแรงเสียดทานทำได้ไม่สะดวกนั่นเอง

โดยทั่วไปแล้วการควบคุมตัวละครในเชิงฟิสิกส์นั้นจะใช้ตัวควบคุมอนุพันธ์ตามสัดส่วน (Proportional Derivative (PD) Controller) ซึ่งจะสร้างแรงบิดที่มีขนาดเท่ากันแต่กลับทิศกับวัตถุแกร่งที่เชื่อมด้วยข้อต่อแบบบานพับเท่ากับ

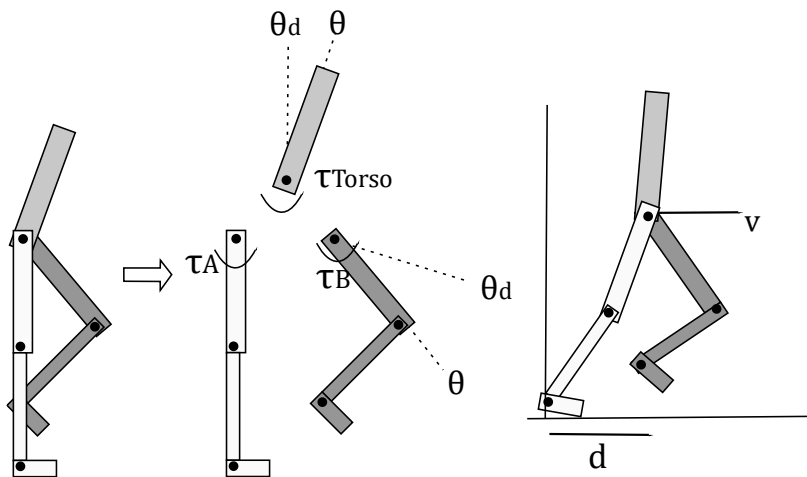
$$\tau = k_p(\theta_d - \theta) - k_d\dot{\theta}$$

ซึ่งเปรียบเสมือนมีสปริง และแดมเปอร์เชิงมุมอยู่ที่ข้อต่อนั้น โดย  $\theta$  คือมุมสัมพัทธ์ของวัตถุแกร่งที่เชื่อมด้วยข้อต่อ,  $\dot{\theta}$  คือความเร็วเชิงมุมสัมพัทธ์,  $\theta_d$  คือมุมสัมพัทธ์ที่สปริงพยายามจะดึงให้เป็น,  $k_p$  คือค่าคงที่ความแข็งของสปริง และ  $k_d$  คือค่าคงที่ความหนืดของแดมเปอร์ โดยการกำหนดค่าของ  $\theta_d$  นั้นจะทำคล้ายๆ กับการกำหนดคีย์เฟรมเพื่อให้ตัวละครอยู่ในท่าทางที่เหมาะสมนั่นเอง และอาจมีเวลาที่กำหนดหรือเงื่อนไขที่จะเปลี่ยน  $\theta_d$  ตัวอย่างเช่นในภาพประกอบ 28 จากงานวิจัย [35] นั้นจะเปลี่ยน  $\theta_d$  ตามก้าวของการเดินเช่น จากจังหวะที่เท้าซ้ายแตะพื้นจะเริ่มก้าวไป 0.3 วินาที แล้วเปลี่ยนท่า และเมื่อเท้าขวาแตะพื้นจึงเปลี่ยนท่าอีกเป็นต้น



ภาพประกอบ 28: การควบคุมตัวละครด้วยวิธี SIMBICON ดัดแปลงจาก [35]

ซึ่งในงานวิจัย [35] จะจัดการกับข้อต่อที่เชื่อมระหว่างเอวกับขาที่เท้าเหยียบพื้น (Stance Hip) และข้อต่อที่เชื่อมระหว่างเอวกับขาที่แกว่ง (Swing Hip) เป็นพิเศษ โดยจะคำนวณแรงบิดที่เกิดขึ้นโดยใช้มุมที่นิยามในพิกัดโลก เพื่อให้ทรงตัวได้ไม่ล้ม โดยแรงบิดที่เอว  $\tau_{\text{Torso}}$  และแรงบิดของขาที่แกว่ง  $\tau_B$  จะคำนวณดังในภาพประกอบ 29



ภาพประกอบ 29: การคำนวณแรงบิดของข้อต่อระหว่างขากับเอว (a) และการคำนวณมุมเป้าหมายของขาที่แกว่ง โดยใช้ผลกระทบบ (Feedback) จากระยะทางระหว่างข้อเท้าที่วางไปยังจุดศูนย์กลางมวลดัดแปลงจาก [35]

โดยคำนวณ  $\theta_d$  ของขาที่แกว่งด้วยวิธีพิเศษดังนี้

$$\theta_d = \theta_{d0} + c_d d + c_v v$$

โดยที่  $\theta_{d0}$  คือมุมเป้าหมายเริ่มต้น  $d$  คือระหว่างข้อเท้าที่วางไปยังจุดศูนย์กลางมวล  $v$  คือความเร็วของจุดศูนย์กลางมวลทางทิศที่เคลื่อนที่ไป  $c_d$  และ  $c_v$  เป็นค่าคงที่ จากนั้น แรงบิดของขาที่เท้าเหยียบพื้น,  $\tau_A$ , จะคำนวณโดย  $\tau_A = -\tau_{\text{Torso}} - \tau_B$

## บทสรุป

บทนี้กล่าวถึงการสร้างแอนิเมชันของตัวละครโดยเริ่มจากการสร้างแอนิเมชันของตัวละครแบบพื้นฐานโดยไม่ใช้ฟิสิกส์ ด้วยการกำหนดคีย์เฟรมซึ่งเป็นวิธีที่ใช้อย่างแพร่หลาย จากนั้นจึงกล่าวถึงพื้นฐานของโมชันแคปเจอร์ซึ่งเป็นการนำการแสดงของนักแสดงเข้าไปใช้ในแอนิเมชัน และ พื้นฐานของการจำลองตัวละครด้วยฟิสิกส์โดยใช้การจำลองวัตถุแกร่งแบบมีข้อต่อ ซึ่งทำให้สามารถสร้างการเคลื่อนที่ของตัวละครที่ตอบสนองกับวัตถุอื่นที่มาชนได้

## บทที่ 8: คอมพิวเตอร์แอนิเมชันเชิงฟิสิกส์ขั้นสูง

บทนี้จะกล่าวถึงงานวิจัยเกี่ยวกับแอนิเมชันเชิงฟิสิกส์ที่เกี่ยวข้องกับเนื้อหาในบทก่อนๆ เพื่อให้ผู้ที่มีความสนใจเป็นพิเศษในหัวข้อต่างๆ สามารถใช้เป็นจุดเริ่มต้นในการศึกษาเพิ่มเติมได้

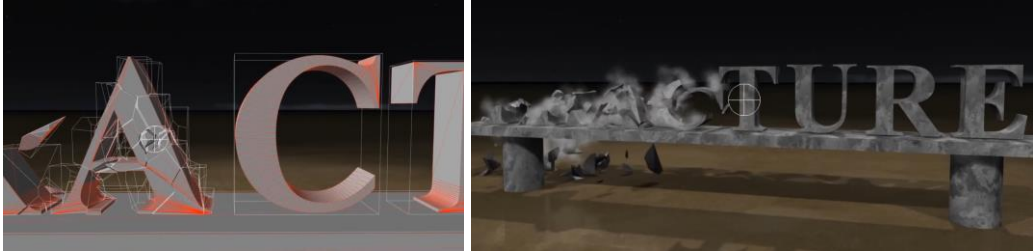
### ระบบอนุภาคขั้นสูง

ในการสร้างแอนิเมชันของระบบอนุภาคนั้นการที่จะได้ผลลัพธ์ที่สมจริงจะต้องใช้อนุภาคเป็นจำนวนมาก ซึ่งในปัจจุบันมักใช้ Compute Shader หรือ CUDA ในการคำนวณบน GPU ซึ่งประเด็นต่างๆ เช่นการเรียงลำดับอนุภาคจากหลังมาหน้า เพื่อให้การแสดงผลมีความโปร่งแสงถูกต้อง การชนกับวัตถุในฉาก การวาดอย่างมีประสิทธิภาพ [36] ส่วนการเขียนโปรแกรมสำหรับระบบอนุภาคด้วย CUDA ให้มีประสิทธิภาพนั้นมีรายละเอียดระบุใน บทความ [37]

### การจำลองวัตถุแกร่งขั้นสูง

งานวิจัยที่สำคัญเกี่ยวกับการจำลองวัตถุแกร่งจนถึงปี 2012 นั้นมีการรวบรวมใน [38] ซึ่งมีการกล่าวถึงแนวทางการประยุกต์ใช้การจำลองวัตถุแกร่งในด้านต่างๆ โมเดลของแรงเสียดทาน การสร้างและแก้ระบบสมการ และข้อจำกัดแบบต่างๆ การตรวจสอบการชน และการแก้ไขการชน รวมถึงกระบวนการในการเปรียบเทียบคุณภาพของอัลกอริทึมต่างๆ ที่ใช้ในการจำลองวัตถุแกร่ง ในส่วนของผู้เขียนเองได้มีส่วนร่วมในงานวิจัยที่เกี่ยวข้องกับการจำลองวัตถุแกร่งขั้นสูงดังนี้

งานวิจัย Real time dynamic fracture with volumetric approximate convex decompositions [39] แบ่งวัตถุแกร่งเป็นทรงหลายเหลี่ยมแบบปองหลายๆ อันที่เชื่อมกัน โดยสามารถจำลองการแตกหักของวัตถุได้โดย ทำลายการเชื่อมต่อเหล่านั้นตั้งในภาพประกอบ 30



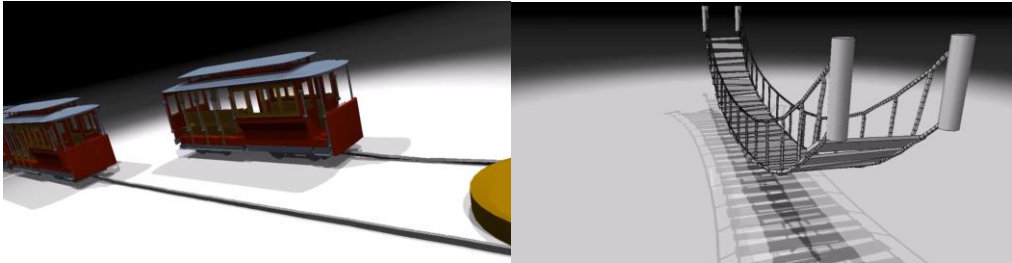
ภาพประกอบ 30: การแตกหักของวัตถุแกร่ง เครดิต: [39]

งานวิจัย Simulating Visual Geometry [40] นำเสนอการใช้ไดนามิกเชิงตำแหน่งในการจำลองวัตถุแกร่ง โดยแต่ละวัตถุแกร่งจะแทนด้วยตำแหน่งและการวางแนว และรูปร่างของวัตถุจะเป็นรูปหลายเหลี่ยมแบบปองเท่านั้นเพื่อความรวดเร็วในการคำนวณ โดยงานวิจัยนี้นำเสนอวิธีในการสร้างวัตถุแกร่งเหล่านี้จากเมชหลายเหลี่ยมใดๆ โดยสามารถจำลองข้อต่อ การบวม และการแตกหักของวัตถุได้ด้วยดังตัวอย่างในภาพประกอบ 31



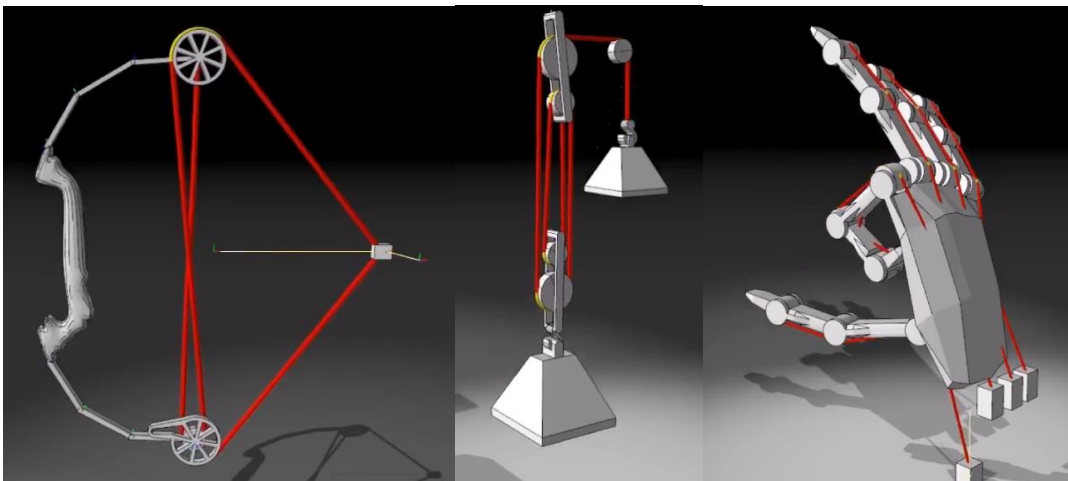
ภาพประกอบ 31 การจำลองเมชหลายเหลี่ยมใดๆด้วยการแทนด้วยวัตถุแกร่งหลายชิ้นโดยอัทโนมัติ เครดิต: [40]

งานวิจัย Long range constraints for rigid body simulations [41] นำเสนอการใช้ข้อจำกัดระยะไกลในการลดจำนวนรอบที่ต้องใช้ในการแก้ข้อจำกัดไปได้อย่างมาก เพื่อให้ได้ผลที่สมจริงของการจำลองวัตถุแข็งที่มีการเชื่อมต่อระหว่างวัตถุแข็งที่มีมวลต่างกันมากๆ หรือมีการเชื่อมต่อหลายๆ ท่อน เช่น รถกระเช้าไฟฟ้าที่ขึ้นลงภูเขา หรือสะพานไม้ที่ตั้งแสดงในภาพประกอบ 32



ภาพประกอบ 32: รถกระเช้าไฟฟ้าที่ขึ้นลงภูเขาและสะพานไม้ จำลองอย่างมีประสิทธิภาพด้วยข้อจำกัดระยะไกล เครดิต: [41]

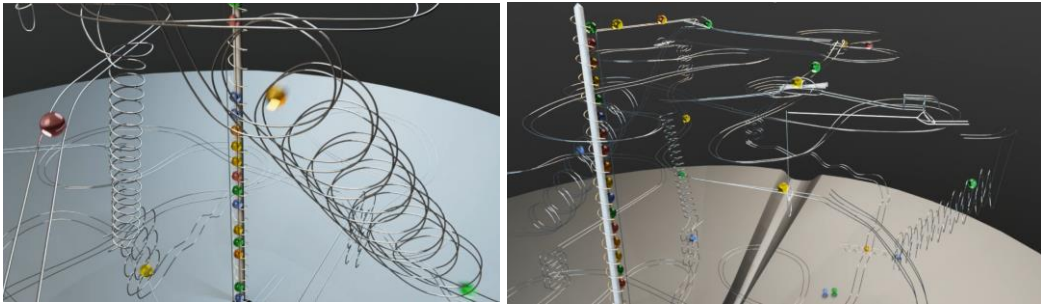
งานวิจัย Cable Joints [42] นำเสนอข้อต่อระหว่างวัตถุแข็งแบบใหม่ ที่ทำหน้าที่เป็นรอกระหว่างคู่ของวัตถุแข็งหรือวัตถุแข็งที่ถูกร้อยเข้าด้วยกันก็ได้ ซึ่งทำให้การจำลองระบบที่มีสายเคเบิลสามารถทำได้เร็ว โดยสามารถใช้จำลองปรากฏการณ์เช่น สายสลิงในเครื่องออกกำลังกาย คันธนู รอกต่างๆ รวมถึงเส้นเอ็นที่ใช้ควบคุมมือหุ่นยนต์ ได้ดังในภาพประกอบ 33



ภาพประกอบ 33: คันธนู, รอก, มือหุ่นยนต์ที่คุมด้วยเอ็นจำลองด้วยข้อต่อประเภทสายเคเบิล เครดิต: [42]

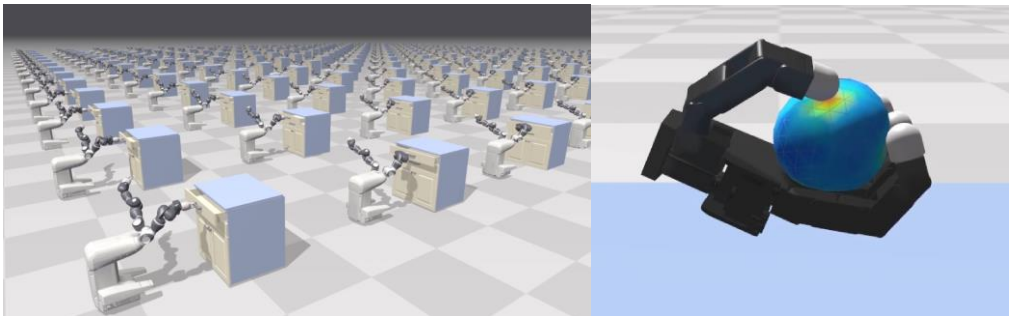
งานวิจัย Detailed Rigid Body Simulation with Extended Position Based Dynamics [43] นำเสนอการจำลองวัตถุแข็งด้วยการอินทิเกรตเชิงเวลาแบบใหม่ และใช้ทิศทางล่าสุดของเกรเดียนต์ของข้อจำกัดในการ

คำนวณเสมอในทุกๆ รอบจึงทำให้สามารถจำลองการเคลื่อนที่ของวัตถุแกร่งที่เคลื่อนที่เร็วและชนกับรางโค้งได้เหมือนจริง ดังในภาพประกอบ 34



ภาพประกอบ 34: ลูกแก้วเคลื่อนบนรางเหล็กโค้ง เครดิต: [43]

งานวิจัย Non-smooth Newton Methods for Deformable Multi-body Dynamics [44] นำเสนอวิธีการจำลองวัตถุแกร่ง และวัตถุที่ยืดหยุ่นได้ที่สามารถชนและสัมผัสกันได้ โดยมีแรงเสียดทานระหว่างกัน โดยสามารถทำการคำนวณได้อย่างมีประสิทธิภาพด้วย GPU และการคำนวณมีความแม่นยำสมจริงตามหลักฟิสิกส์ รวมถึงยังสามารถนำมาใช้จำลองหุ่นยนต์เพื่อหาวิธีการบังคับเพื่อให้บรรลุวัตถุประสงค์ที่ต้องการได้ตามตัวอย่างที่แสดงในภาพประกอบ 35



ภาพประกอบ 35: แขนและมือหุ่นยนต์จำลองด้วยวิธี Non-Smooth Newton เครดิต: [44]

งานวิจัย Primal/Dual Descent Methods for Dynamics [45] ใช้การตั้งปัญหาแบบ Primal ซึ่งตัวแปรที่จะต้องคำนวณจะเป็นตำแหน่ง ความเร็ว แรง ซึ่งต่างจากวิธี Dual ที่พบบ่อยกว่าที่ตัวแปรที่แก้คือ Lagrange Multiplier ของแต่ละข้อจำกัด (เช่นวิธีในบทของการจำลองวัตถุแกร่งในหนังสือเล่มนี้) โดยวิธีที่นำเสนอสามารถ

จัดการกับกรณีที่มีวัตถุวางทับกันได้มาก และสามารถคำนวณได้อย่างมีประสิทธิภาพบน GPU โดยวิธี Primal นั้นสามารถคำนวณหาค่าของอนุพันธ์ต่างๆ ได้ง่ายด้วย โดยตัวอย่างของฉากที่จำลองด้วยวิธีที่นำเสนอในงานวิจัยนี้แสดงอยู่ในภาพประกอบ 36



ภาพประกอบ 36: ซามจำนวนมากซ้อนกัน และ กรวดจำนวนมากที่โดนทับด้วยเหล็กทรงกระบอกจำลองด้วยวิธี Primal Descent *เครดิต: [45]*

### การจำลองทางฟิสิกส์ด้วยวิธีไดนามิกเชิงตำแหน่งขั้นสูง

การจำลองทางฟิสิกส์ด้วยวิธีไดนามิกเชิงตำแหน่งนั้นได้รับความนิยมในวงกว้างในการจำลองฟิสิกส์ในเกม และแอปพลิเคชัน อินเทอร์แอคทีฟต่างๆ ที่ ต้องการแสดงผลด้วยความเร็วสูงเนื่องจากสามารถมั่นใจได้ว่าผลลัพธ์ที่ได้จะยังคงเสถียรอยู่ โดยถูกใช้งานในทั้ง Unity และ Unreal Engine ซึ่งเป็นเกมเอนจินสองอันดับแรก ที่ได้รับความนิยมสูงสุด ณ ปี 2019 จากการสำรวจของเว็บไซต์เกี่ยวกับการสร้างเกมชื่อดัง Gamasutra [46] และยังได้มีการใช้งานในการสร้างวิช่วลเอฟเฟคในภาพยนตร์หลากหลายเรื่องอีกด้วยโดยใช้งานได้ในเฟรมเวิร์ค Vellum ของโปรแกรม Houdini จากบริษัท SideFX [47]

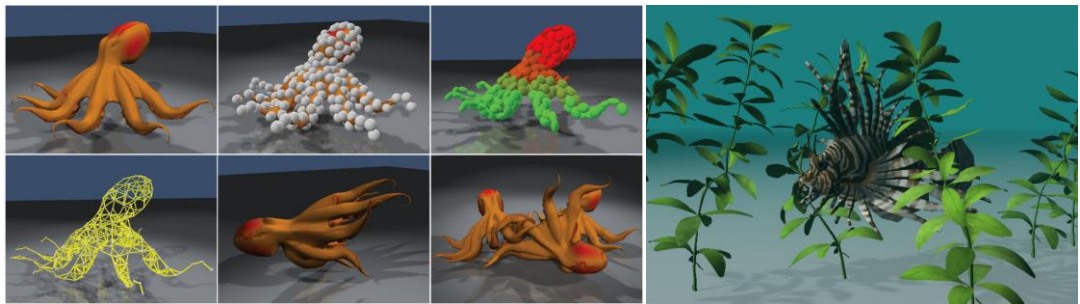
งานวิจัยสำคัญๆที่เกี่ยวข้องกับการจำลองทางฟิสิกส์ด้วยวิธีไดนามิกเชิงตำแหน่งได้ถูกรวบรวมไว้ที่ [19] พร้อมทั้งรหัสต้นฉบับของหลากหลายอัลกอริทึมเพื่อให้ผู้สนใจได้ศึกษาเพิ่มเติมได้ ในส่วนของผู้เขียนได้มีส่วนร่วมในงานวิจัยทางด้าน การจำลองทางฟิสิกส์ด้วยวิธีไดนามิกเชิงตำแหน่งดังนี้

งานวิจัย Wrinkle Meshes [48] ได้ใช้วิธีไดนามิกเชิงตำแหน่งในการจำลองผ้าที่ความละเอียดต่ำ จากนั้นใช้วิธีการไดนามิกเชิงตำแหน่งที่ลดรูป และไม่สนใจความเร็วและการชนกับวัตถุอื่นๆ เพื่อคำนวณตำแหน่งของผ้าที่ความละเอียดสูงที่มีรอยพับรอยย่นอย่างรวดเร็วบน GPU โดยตัวอย่างของผลลัพธ์ที่ได้แสดงในภาพประกอบ 37



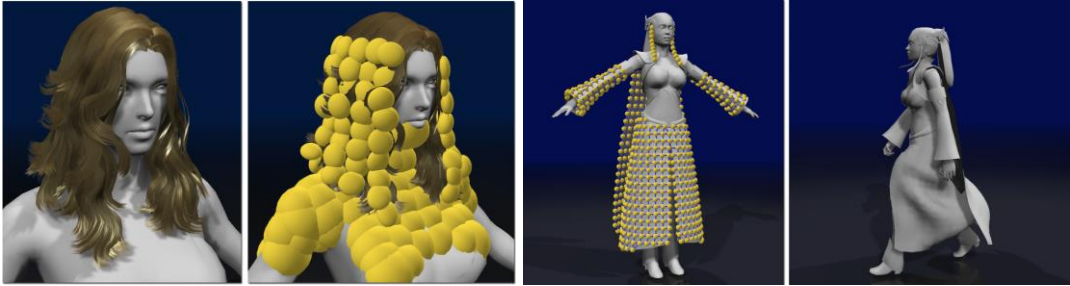
ภาพประกอบ 37: ผ้า, ชุดเลื้อยกระโปรง, เสื้อแขนยาว จำลองด้วย Wrinkle Mesh เครดิต: [48]

งานวิจัย Solid simulation with oriented particles [49] นำเสนอให้เพิ่มการวางแนวในวิธีไดนามิกเชิงตำแหน่ง และแทนรูปร่างของอนุภาคด้วยรูปทรงรี ซึ่งทำให้สามารถแทนที่รูปทรงที่ซับซ้อนด้วยอนุภาคจำนวนน้อยได้ พร้อมทั้งนำเสนอวิธีในการสร้างอนุภาคและข้อจำกัดจากเมชสามเหลี่ยมใดๆ ให้ได้โดยง่าย โดยสามารถจำลองฉากที่มีความซับซ้อนได้ด้วยความเร็วสูงดังตัวอย่างในภาพประกอบ 38



ภาพประกอบ 38: การจำลองวัตถุยืดหยุ่นได้ด้วยอนุภาคที่มีการวางแนว เครดิต: [49]

งานวิจัย Adding Physics to Animated Characters with Oriented Particles [50] เป็นการนำอนุภาคที่มีการวางแนวมาใช้ในการจำลองเสื้อผ้า และกลุ่มของผมให้กับตัวละครที่เคลื่อนไหวอย่างซับซ้อนได้ โดยสามารถประมวลผลได้ด้วยความเร็วสูงดังตัวอย่างในภาพประกอบ 39



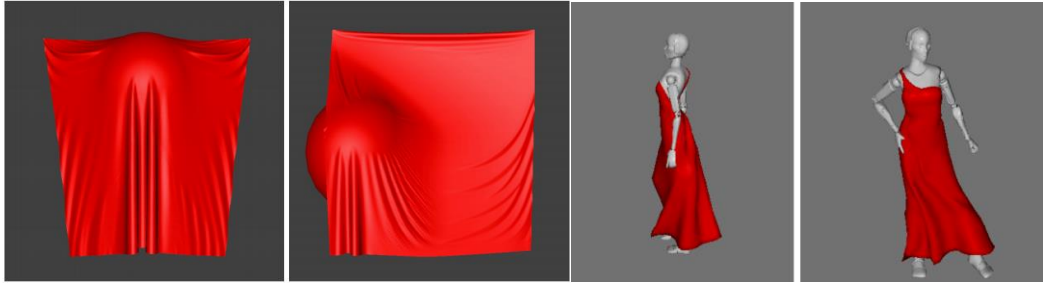
ภาพประกอบ 39: การใช้อนุภาคที่มีการวางแนวในการจำลองผมและเสื้อผ้าให้กับตัวละคร เครดิต: [50]

งานวิจัย Fast Simulation of Inextensible Hair and Fur [51] นำเสนอวิธี “ตามผู้นำ” ในการจำลองเส้นผมที่สามารถรับประกันว่าเส้นผมจะไม่มีรอยย่นหรือหดรัด แม้ว่าจะใช้จำนวนรอบในการแก้ข้อจำกัดเพียงครั้งเดียวก็ตาม โดยแลกกับการที่โมเมนตัมของระบบอาจลดลงได้มาก แต่เนื่องจากผมมักจะเชื่อมต่อกับตัวละครที่ขยับไปมาอยู่แล้วจึงทำให้ไม่เป็นปัญหามากนักในทางปฏิบัติ โดยตัวอย่างของผลลัพธ์แสดงอยู่ในภาพประกอบ 40



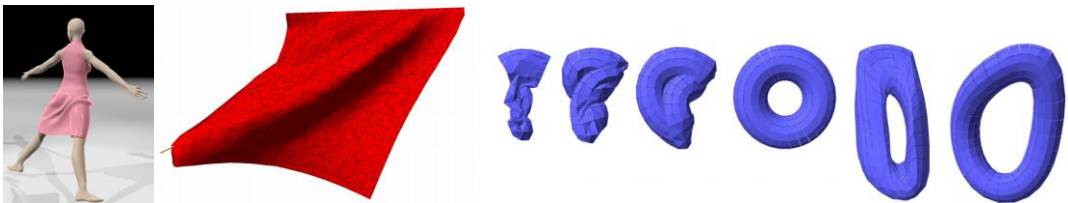
ภาพประกอบ 40: การจำลองผมด้วยวิธี “ตามผู้นำ” ใช้จำลองผมทรงต่างๆ เครดิต: [51]

งานวิจัย Long Range Attachments - A Method to Simulate Inextensible Clothing in Computer Games [52] นำเสนอข้อจำกัดแบบใหม่สำหรับวิธีไดนามิกเชิงตำแหน่ง โดยระบุว่าจะระยะทางระหว่างอนุภาคใดๆ จะต้องไม่มากกว่าระยะทางเริ่มต้นของอนุภาคนั้นไปยังอนุภาคที่ยึดอยู่กับวัตถุแกร่ง เช่นถ้าเป็นเส้นผมก็คือระยะทางตรงไปยังรากผม ถ้าเป็นผ้าคลุมก็เป็นระยะทางตรงไปยังหมุดยึดผ้า ซึ่งข้อจำกัดนี้ทำให้สามารถจำลองผม และผมที่ไม่ยืดได้ แม้ว่าจะใช้จำนวนรอบในการแก้ข้อจำกัดน้อยมากก็ตาม โดยตัวอย่างฉากที่ใช้วิธีนี้แสดงในภาพประกอบ 41



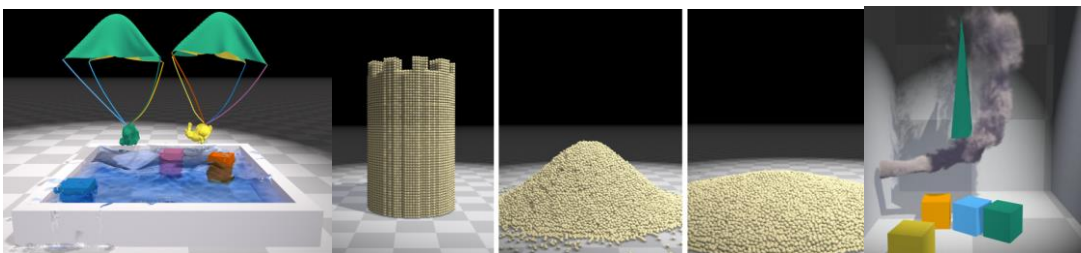
ภาพประกอบ 41: ผ้าและชุดเดรสจำลองได้อย่างสมจริงแม้ใช้จำนวนรอบในการแก้ข้อจำกัดน้อยมากด้วยการใช้ข้อจำกัดระยะไกล เครดิต: [52]

งานวิจัย Strain Based Dynamics [53] นำเสนอการสร้างข้อจำกัดของวิธีไดนามิกเชิงตำแหน่งโดยใช้ Strain Tensor ซึ่งมีรากฐานจากฟิสิกส์ได้โดยตรง และสามารถกำหนดความยืดหยุ่นและสัมประสิทธิ์ของแรงหน่วงในแต่ละแกนในสามมิติได้อิสระต่อกัน และไม่ขึ้นกับเมทริกซ์ที่ใช้ในการจำลอง โดยตัวอย่างของการใช้งานแสดงในภาพประกอบ 42



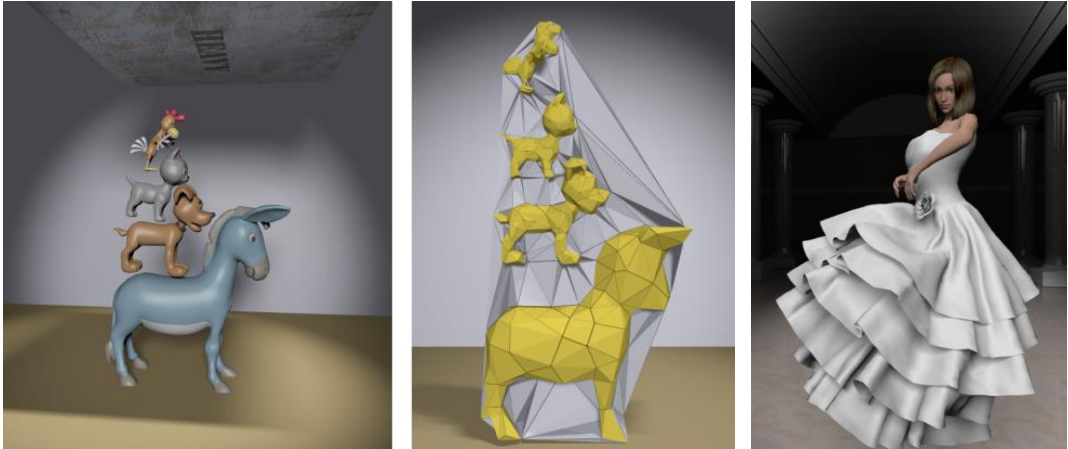
ภาพประกอบ 42: ชุดเดรส, ผ้า และรูปทรงโดนัทจำลองด้วย Strain Based Dynamics เครดิต: [53]

งานวิจัย Unified particle physics for real-time applications [20] นำเสนอข้อจำกัดหลากหลายแบบเพื่อให้การจำลองทางฟิสิกส์ด้วยวิธีไดนามิกเชิงตำแหน่งสามารถจำลอง ของแข็ง ของเหลว และแก๊สในระบบเดียวกันได้ พร้อมทั้งนำเสนอวิธีการจัดการกับแรงเสียดทาน และอัลกอริทึมที่สามารถคำนวณบน GPU ได้อย่างมีประสิทธิภาพดังตัวอย่างในภาพประกอบ 43



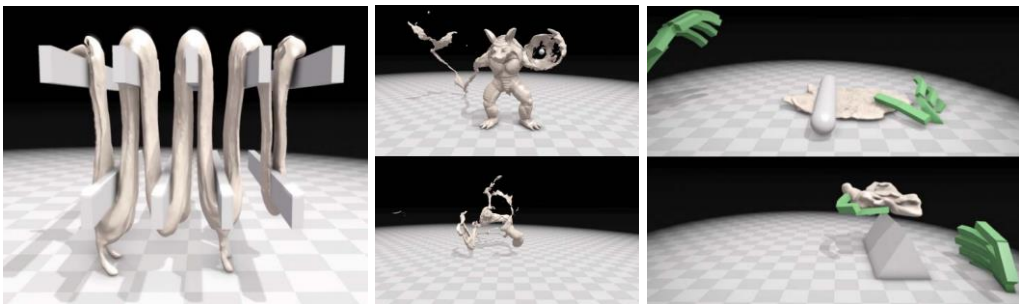
ภาพประกอบ 43: การจำลองของแข็ง ของเหลว และ แก๊สในระบบเดียวกัน เครดิต: [20]

งานวิจัย Air meshes for robust collision handling [54] นำเสนอวิธีการสร้างเมฆของอากาศรอบของแข็งเพื่อใช้ในการจัดการกับการชนได้อย่างมีประสิทธิภาพ และสามารถคืนรูปให้กับวัตถุหลากหลายชั้นที่ซ้อนกันอยู่ได้ ถูกต้องตามลำดับบนล่างได้แม้ว่าวัตถุทั้งหลายจะโดนบีบจนแบนเป็นแผ่น และสามารถใช้ในการจำลองชุดที่มีหลายชั้นซ้อนกันให้ไม่ทะลุกันได้อย่างมีประสิทธิภาพก็ตามโดยตัวอย่างแสดงในภาพประกอบ 44



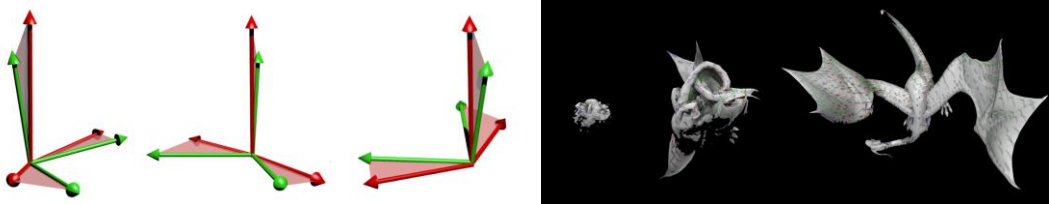
ภาพประกอบ 44: วัตถุรูปสัตว์ซ้อนกันหลายๆชั้น และชุดแต่งงานที่มีหลายชั้น จำลองด้วยวิธีการ Air Mesh  
เครดิต: [54]

งานวิจัย Real-time simulation of large elasto-plastic deformation with shape matching [55] นำเสนอการจำลองวัตถุที่เปลี่ยนรูปร่างอย่างถาวร ซึ่งวิธีที่นำเสนอจะทำการเก็บและปรับปรุงค่าส่วนของเมทริกซ์ที่เก็บการเปลี่ยนแปลงรูปร่างในทุกๆ ขั้นตอนเวลา พร้อมทั้งทำการเพิ่มและลบอนุภาค เพื่อให้เหมาะสมกับรูปร่างที่อาจเปลี่ยนได้อย่างมากระหว่างการจำลอง ตัวอย่างของฉากที่ใช้วิธีนี้จำลองแสดงอยู่ในภาพประกอบ 45



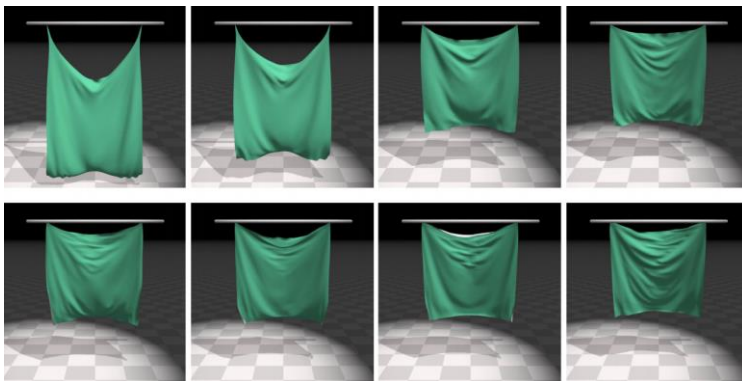
ภาพประกอบ 45: การจำลองวัตถุที่เปลี่ยนแปลงรูปร่างอย่างถาวรได้ เครดิต: [55]

งานวิจัย A robust method to extract the rotational part of deformations [11] นำเสนอวิธีการหาเมทริกซ์หมุนที่ใกล้เคียงกับเมทริกซ์  $3 \times 3$  ใดๆ ก็ตาม ซึ่งการคำนวณนี้ใช้ในการจำลองทางฟิสิกส์หลากหลายชนิด รวมถึงข้อจำกัดบางชนิดของวิธีไดนามิกเชิงตำแหน่งด้วย ซึ่งวิธีที่ใช้อย่างแพร่หลายอาจจะไม่สามารถหาเมทริกซ์หมุนที่เหมาะสมได้เสมอไป ซึ่งทำให้ในบางกรณีผลลัพธ์ที่ได้จะไม่ถูกต้อง โดยวิธีที่งานวิจัยนี้นำเสนอสามารถคำนวณเมทริกซ์หมุนที่เหมาะสมได้ในทุกกรณี ตัวอย่างของเมทริกซ์หมุน (แทนโดยลูกศรเขียว) ที่คำนวณจากเมทริกซ์ขาเข้า (แทนด้วยลูกศรแดง) และตัวอย่างการใช้งานจำลองมังกรที่สุ่มตำแหน่งของจุดยึดเริ่มต้นแต่ก็ยังสามารถคืนรูปจนกลับสภาพเดิมได้แสดงในภาพประกอบ 46



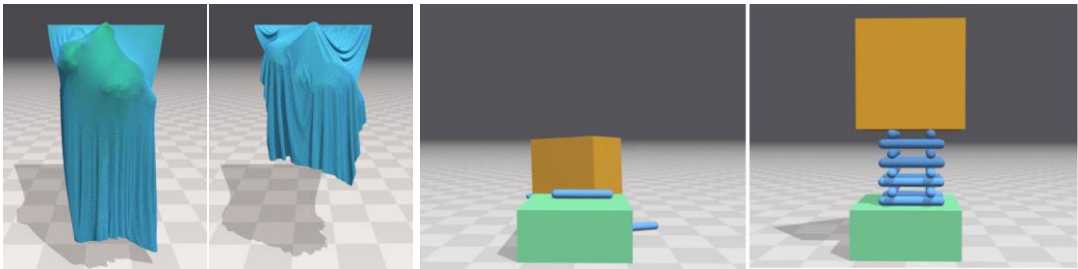
ภาพประกอบ 46: ซ้าย เมทริกซ์หมุน (แทนโดยลูกศรเขียว) ที่คำนวณจากเมทริกซ์ขาเข้า (แทนด้วยลูกศรแดงขวา) การจำลองมังกรโดยสุ่มตำแหน่งของจุดยึดเริ่มต้นแต่ก็ยังสามารถคืนรูปจนกลับสภาพเดิมได้ เครดิต: [11]

งานวิจัย XPBD: position-based simulation of compliant constrained dynamics [56] นำเสนอวิธีการแก้ไขปัญหาสำคัญของวิธีไดนามิกเชิงตำแหน่งที่ว่าผลลัพธ์ที่ได้ขึ้นกับจำนวนรอบในการแก้ข้อจำกัดที่ใช้ โดยยิ่งใช้รอบมากวัตถุก็จะยิ่งยืดหยุ่นน้อยลงเรื่อยๆ โดยงานวิจัยนี้นำเสนอวิธีแก้ปัญหานี้อย่างได้ผลและใช้เวลาในการคำนวณเพิ่มขึ้นเพียงเล็กน้อยเท่านั้นโดยผลของ PBD แสดงในแถวบน และ ผลของ XPBD แสดงในแถวล่างของภาพประกอบ 47



ภาพประกอบ 47: บน PBD, ล่าง XPBD ที่ใช้ 20, 40, 80, 160 รอบในการแก้ข้อจำกัด จะเห็นได้ว่า XPBD ได้ผลที่ใกล้เคียงกันมาก เครดิต: [56]

งานวิจัย Small steps in physics simulation [57] พิจารณา และทดลองเกี่ยวกับปัญหาในการจัดสรรทรัพยากรในการคำนวณของวิธีไดนามิกเชิงตำแหน่ง ว่าควรจะใช้แบ่งขั้นตอนเวลาเป็นช่วงจำนวนน้อยแต่ใช้จำนวนครั้งในการแก้ข้อจำกัดมาก หรือ แบ่งขั้นตอนเวลาเป็นช่วงจำนวนมากแต่ใช้จำนวนครั้งในการแก้ข้อจำกัดน้อย ซึ่งใช้เวลาในการคำนวณรวมกันแล้วเท่ากัน ซึ่งงานวิจัยชิ้นนี้ก็พบว่าควรจะใช้แบ่งขั้นตอนเวลาเป็นช่วงจำนวนมากที่สุด แต่ใช้จำนวนครั้งในการแก้ข้อจำกัดน้อยที่สุด (1 ครั้ง) ซึ่งจะได้ผลดีกว่า ดังแสดงในด้านขวาของแต่ละภาพย่อยในภาพประกอบ 48



ภาพประกอบ 48: ภาพย่อยซ้าย 1 ขั้นตอนเวลาย่อย, 100 ครั้งในการแก้ข้อจำกัด เทียบกับภาพย่อยขวา 100 ขั้นตอนเวลาย่อย, 1 ครั้งในการแก้ข้อจำกัด ซึ่งอย่างหลังได้ผลดีกว่าอย่างเห็นได้ชัด (ผ้าไม่ยืต, กองซ้อนไม่ทลาย)  
 เครดิต: [57]

## การจำลองทางฟิสิกส์ด้วยตารางขั้นสูง และ การจำลองทางฟิสิกส์แบบผสมตารางและอนุภาคขั้นสูง

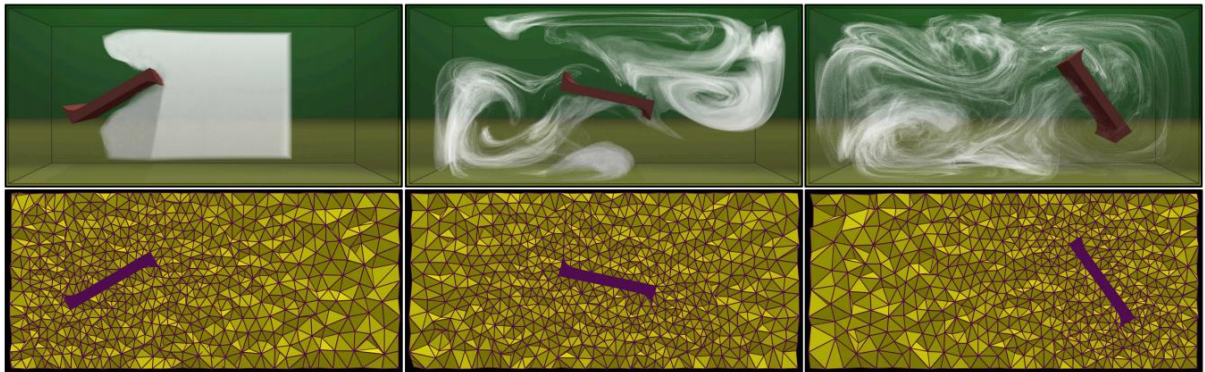
การจำลองฟิสิกส์ด้วยตารางและการจำลองทางฟิสิกส์แบบผสมตารางและอนุภาคนั้นโดยมากจะใช้กับการจำลองของไหล (ก๊าซ และ ของเหลว) มีการใช้งานอย่างกว้างขวางในวงการภาพยนตร์เพื่อสร้างวิช่วเอฟเฟคของ ควีนไฟ และ น้ำ โดยหนังสือ

[58] กล่าวถึงอัลกอริทึมและงานวิจัยที่มีการใช้งานมากในทางปฏิบัติ และ เร็วๆ นี้มีงานวิจัยสำรวจเกี่ยวกับการจำลองของไหลที่มีการปฏิสัมพันธ์กันซึ่งส่วนใหญ่ใช้ตารางใน [59]

ไม่กี่ปีที่ผ่านมาได้มีการนำวิธี Material Point Method (MPM) [60] ซึ่งเป็นวิธีแบบผสมตารางและอนุภาคขั้นสูงมาใช้ในคอมพิวเตอร์กราฟิกเพื่อการจำลองหิมะในงาน [61] หลังจากนั้นได้มีการนำ MPM มาใช้ในการจำลองวัตถุที่มีคุณสมบัติของทั้งของเหลวและของแข็งอื่นๆ เช่น ทราย โคลน ลาวา ในงานวิจัย [62] [63] [64] และ [65]

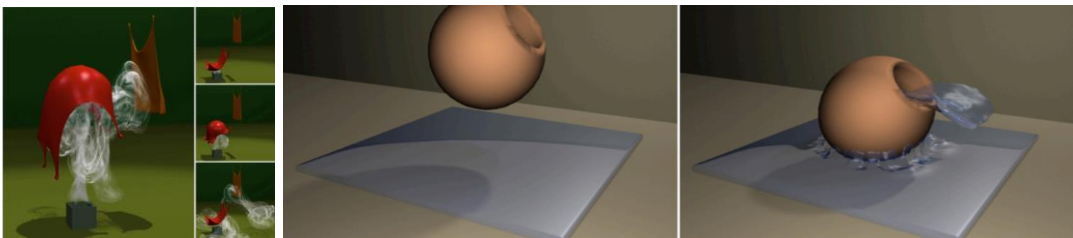
โดยในส่วนของผู้เขียนเองได้มีส่วนร่วมในงานวิจัยที่เกี่ยวข้องกับการจำลองทางฟิสิกส์ด้วยตารางและการจำลองทางฟิสิกส์แบบผสมตารางและอนุภาคขั้นสูงดังนี้

งานวิจัย Fluid animation with dynamic meshes [66] นำเสนอการใช้เมชจตุรมุข (Tetrahedral Mesh) ซึ่งเป็นตารางประเภทรูปร่างไม่คงตัวและสามารถปรับให้เหมาะสมกับรูปร่างของปริมาตรที่ต้องการจำลองได้ พร้อมทั้งนำเสนอวิธีในการจำลองของไหลร่วมกับวัตถุแข็ง โดยของไหลและวัตถุแข็งสามารถออกแรงกระทำต่อกันและกันได้ ดังตัวอย่างใน



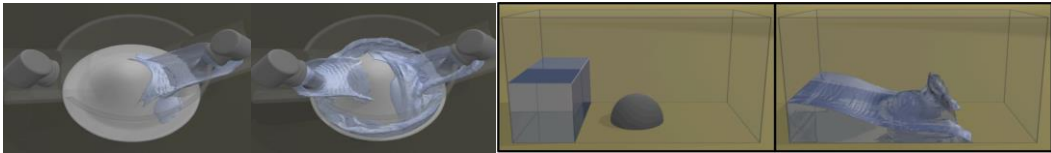
ภาพประกอบ 49: การจำลองควันด้วยเมชจตุรมุขที่เปลี่ยนรูปร่างได้ในระหว่างการทำงาน เครดิต: [66]

งานวิจัย Simultaneous coupling of fluids and deformable bodies [67] นำเสนอการจำลองควัน หรือของเหลวพร้อมๆ กับของแข็งที่ยืดหยุ่นได้ เช่นผ้า หรือวัตถุที่ทำจากยางเป็นต้น โดยควัน/ของเหลว และของแข็งสามารถออกแรงกระทำซึ่งกันและกันได้ โดยสามารถใช้ขั้นตอนเวลาที่ค่อนข้างใหญ่ โดยระบบยังคงความเสถียรอยู่ ดังตัวอย่างในภาพประกอบ 50



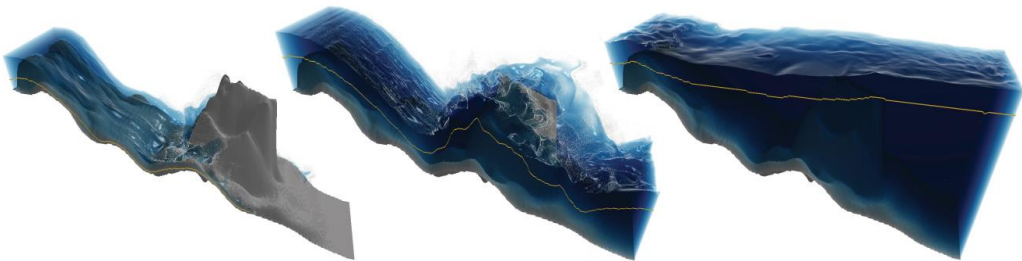
ภาพประกอบ 50: การจำลองแรงที่กระทำระหว่างของไหลและของแข็งโดยการแก้ระบบสมการเดียว เครดิต: [67]

งานวิจัย Liquid simulation on lattice-based tetrahedral meshes [68] นำเสนอวิธีการจำลองของเหลว โดยใช้เมชจัตุรมุขซึ่งจะถูกสร้างใหม่ในทุกๆ ขั้นตอนเวลาจากปริมาตรของของเหลว ณ ปัจจุบัน ด้วยการย้ายปริมาตรทางฟิสิกส์ต่างๆ เช่นความเร็วจากเมชในขั้นตอนเวลาปัจจุบันไปยังขั้นตอนเวลาลัดไป โดยไม่ต้องใช้ขั้นตอนการประมาณค่าในช่วงนอกเหนือจากที่ใช้ในวิธีกึ่งลากรางเจียน โดยสามารถจำลองของเหลวในภาชนะที่มีรูปร่างซับซ้อนได้ ดังตัวอย่างในภาพประกอบ 51



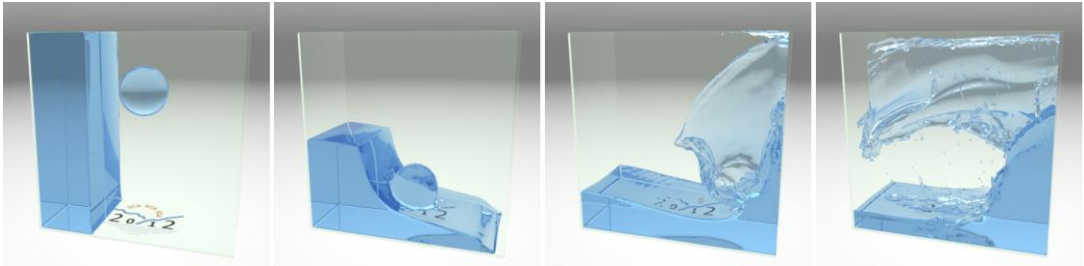
ภาพประกอบ 51: การจำลองของเหลวโดยใช้เมชจัตุรมุขที่เปลี่ยนรูปร่างตามของเหลว [68]

งานวิจัย Real-time Eulerian water simulation using a restricted tall cell grid [69] นำเสนอการใช้ตารางแบบที่ช่องด้านล่างสุดสามารถยืดหดให้สูงหรือเตี้ยได้และสามารถกำหนดความสูงของพื้นใต้น้ำได้ ซึ่งทำให้สามารถจำลองน้ำที่ลึกได้โดยใช้หน่วยความจำและเวลาในการคำนวณไม่มาก พร้อมทั้งสามารถจำลองบน GPU ได้อย่างมีประสิทธิภาพด้วยการแก้สมการของความดันด้วยวิธีพหุตาราง (Multigrid) ซึ่งใช้เวลา  $O(n)$  เมื่อ  $n$  คือจำนวนช่องในตารางโดยตัวอย่างแสดงในภาพประกอบ 52



ภาพประกอบ 52: การจำลองน้ำด้วยการใช้ตารางที่มีช่องสูง เครดิต: [69]

งานวิจัย Mass-Conserving Eulerian Liquid Simulation [70] นำเสนอการใช้ตารางในการจำลองของเหลวโดยไม่อ่าคัยอนุภาค โดยวิธีการที่นำเสนอจะรับประกันว่าของเหลวจะมีมวลคงที่เสมอและปริมาตรเกือบจะคงที่ โดยใช้การเก็บความหนาแน่นของของเหลวในแต่ละช่อง พร้อมกับการทำให้บริเวณที่เปลี่ยนระหว่างของเหลวไปยังอากาศแคบที่สุดเท่าที่ทำได้ โดยที่สามารถคำนวณได้อย่างมีประสิทธิภาพด้วย GPU ดังตัวอย่างในภาพประกอบ 53



ภาพประกอบ 53: การจำลองน้ำโดยที่มวลในระบบคงที่และปริมาตรเกือบคงที่เสมอ เครดิต: [70]

งานวิจัย A Multigrid Fluid Pressure Solver Handling Separating Solid Boundary Conditions [71] นำเสนอวิธีพหุตารางในการแก้สมการของความดันที่อยู่ในรูป Linear Complementarity Problem ที่เกิดขึ้นจากการที่กำหนดให้ของเหลวสามารถแยกตัวออกจากของแข็งได้โดยอิสระดังรูปขวาสุดในภาพประกอบ 54



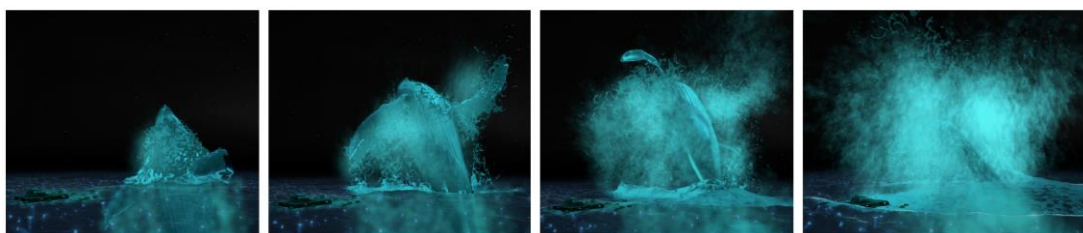
ภาพประกอบ 54: ซ้าย) สถานะเริ่มต้น กลาง) สถานะหลังจากที่น้ำไหลลงมาและติดกับภาชนะไม่สามารถแยกออกได้โดยอิสระ ขวา) สถานะหลังจากที่น้ำไหลลงมาและแยกออกจากภาชนะได้อย่างอิสระ เครดิต: [71]

งานวิจัย Real-time Simulation of Large Bodies of Water with Small Scale Details [72] นำเสนอการใช้ตารางสองมิติในการจำลองระบบสมการน้ำตื้น (Shallow Water Equations) โดยมีการนำเสนอกระบวนการหลายอย่างเพื่อให้การจำลองเสถียรแม้ในสถานการณ์ที่น้ำเคลื่อนที่เร็ว และความสูงของพื้นใต้น้ำมีความชันมาก พร้อมนำเสนอวิธีที่แปลงระหว่างอนุภาค และตารางในการจำลองน้ำตก และนำเสนอวิธีในการสร้างอนุภาคที่แทนละอองน้ำ และโฟมบนผิวน้ำ ดังในภาพประกอบ 55



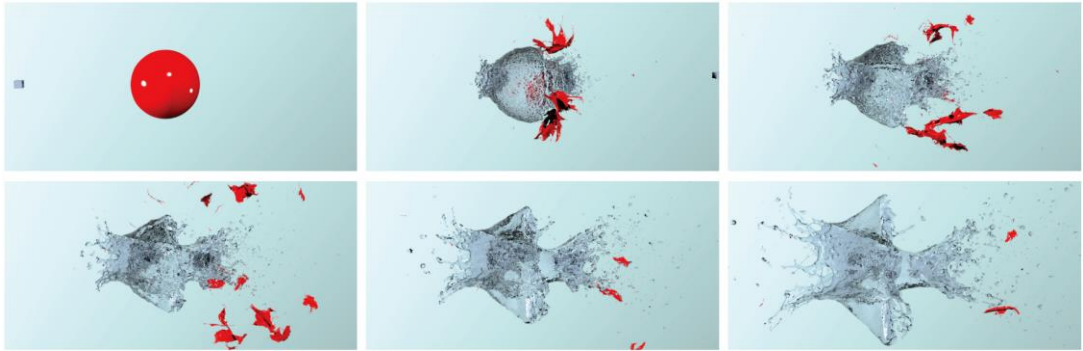
ภาพประกอบ 55: การจำลองน้ำโดยใช้ระบบสมการน้ำตื้นและใช้อนุภาคประเภทต่างๆในการเสริมความสมจริง  
เครดิต: [72]

งานวิจัย Coupling 3D Eulerian, Heightfield and Particle Methods for Interactive Simulation of Large Scale Liquid Phenomena [73] นำเสนอการใช้ตารางสามมิติในการจำลองน้ำที่เชื่อมกับอนุภาคในการจำลองบริเวณผิวน้ำที่ใกล้กล้อง และก้อนน้ำที่กระเด็นลอยขึ้นมาในอากาศ ร่วมกับการใช้ตารางสองมิติในการจำลองน้ำลึกที่อยู่ห่างกล้อง โดยสามารถใช้จำลองฉากขนาดใหญ่เช่น ปลาวาฬกระโจนขึ้นมาจากน้ำ โดยที่ผิวน้ำแยกออกจากกัน และบางส่วนค้างบนตัวปลาวาฬก่อนที่จะไหลลงผิวน้ำในเวลาจริง (Real-time) โดยการคำนวณใช้ GPU ดังในภาพประกอบ 56



ภาพประกอบ 56: ปลาวาฬกระโจนจากน้ำแล้วหล่นกระแทกผิวน้ำ เครดิต: [73]

งานวิจัย Fast grid-free surface tracking [74] นำเสนอการติดตามผิว (Surface Tracking) ของน้ำโดยใช้เมซสามเหลี่ยมโดยทำการขยับจุดยอดของเมซตามความเร็วของน้ำแล้วทำการแก้ไขให้เมซไม่มีการตัดกัน และทำการปรับปรุงคุณภาพของเมซให้มีสามเหลี่ยมที่ไม่มีมุมแคบหรือกว้างเกินไป และมีขนาดไม่เล็กไม่ใหญ่เกินไป ซึ่งทำให้สามารถติดตามผิวน้ำที่มีรายละเอียดมากๆ ได้ โดยที่ใช้หน่วยความจำแปรผันตรงกับพื้นที่ผิว (แทนที่จะแปรผันตรงตามปริมาตรเหมือนเทคนิคอื่นๆ) ดังตัวอย่างในภาพประกอบ 57



ภาพประกอบ 57: ลูกโป่งน้ำถูกกระสุนยิงแล้วระเบิดออกมาทำให้เกิดการกระจายของน้ำที่สวยงามและมีความซับซ้อน เครดิต: [74]

งานวิจัย Grid-Free Surface Tracking on the GPU [75] ปรับปรุงวิธีติดตามพื้นผิวให้สามารถคำนวณด้วย GPU ได้อย่างมีประสิทธิภาพโดยที่ได้คุณภาพใกล้เคียงเดิมแต่ประมวลผลเร็วกว่าประมาณ 30 เท่าดังในภาพประกอบ 58 (บน ใช้ GPU, ล่างใช้ CPU)



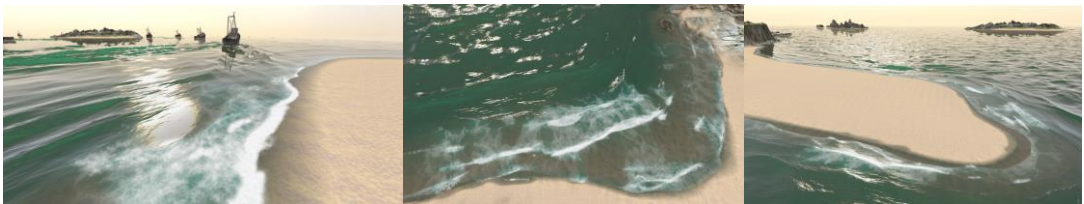
ภาพประกอบ 58: น้ำพุจำลอง บน) ติดตามผิวน้ำด้วย GPU ล่าง) ติดตามผิวน้ำด้วย CPU ซึ่งได้ผลใกล้เคียงกัน เครดิต: [75]

งานวิจัย Water surface wavelets [76] นำเสนอการใช้การแปลงเวฟเล็ต (Wavelet Transform) แบบใหม่ ที่แทนคลื่นน้ำโดยอธิบายด้วยฟังก์ชันที่แปลตามตำแหน่ง ความถี่ และทิศทางของคลื่น ซึ่งทำให้สามารถแทนคลื่นน้ำที่มีรายละเอียดสูงมากโดยใช้ตารางที่ความละเอียดไม่สูงได้ ตัวอย่างเช่นคลื่นน้ำในทะเลที่มีคลื่นที่เกิดจากเรือหลายๆ ลำ ดังตัวอย่างในภาพประกอบ 59



ภาพประกอบ 59: คลื่นน้ำที่มีรายละเอียดมาก ๆ ซึ่งแทนที่ด้วยการแปลงเวฟเล็ต เครดิต: [76]

งานวิจัย Making Procedural Water Waves Boundary-aware [77] นำเสนอการปรับปรุงวิธีการสร้างภาพเคลื่อนไหวของคลื่นน้ำแบบวิธีการขั้นตอน (Procedural Method) ซึ่งโดยปกติแล้วใช้กับการจำลองคลื่นกลางทะเลเปิดได้เสมือนจริงมาก แต่ไม่สามารถจำลองผิวน้ำใกล้ๆ ชายฝั่งได้เนื่องจากไม่ได้อยู่ในแบบจำลอง งานวิจัยนี้แนะนำวิธีการใส่ชายฝั่งเข้าไปในแบบจำลองและนำเสนอเงื่อนไขในการสร้างอนุภาคที่แทนคลื่นที่แตกกระจาย (Breaking Wave) เป็นละอองน้ำและโฟมที่ลอยอยู่บนผิวน้ำใกล้ๆ ชายฝั่งด้วยทำให้สามารถจำลองฉากที่เป็นชายฝั่งขนาดใหญ่ได้สมจริงและแสดงผลได้ในเวลาจริง ดังตัวอย่างในภาพประกอบ 60



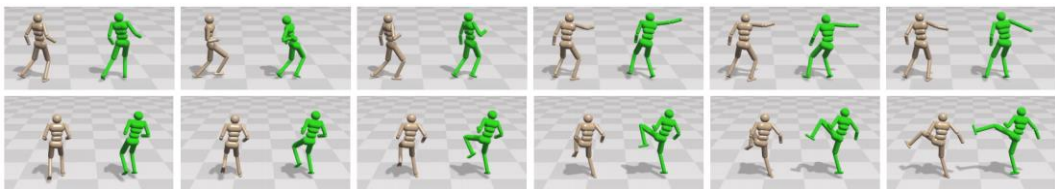
ภาพประกอบ 60) ชายฝั่งทะเลจำลองด้วยการปรับวิธีการขั้นตอนให้พิจารณาชายฝั่ง เครดิต: [77]

## การจำลองตัวละครด้วยฟิสิกส์ขั้นสูง

การทบทวนวรรณกรรมที่มีเนื้อหาครอบคลุมเกี่ยวกับการจำลองตัวละครด้วยฟิสิกส์สำหรับตัวละครที่มีสองขา (Biped) คือ [78] และสำหรับตัวละครทั่วๆ ไปคือ [79] โดยทั้งสองงานนี้มีกล่าวถึงวิธีการตั้งแต่พื้นฐานจนถึงวิธีที่มีความซับซ้อนมาก แต่จะยังไม่รวมถึงงานวิจัยใหม่ๆ ที่ใช้การเรียนรู้จากการลองผิดลองถูกเชิงลึก (Deep Reinforcement Learning) ซึ่งได้รับความนิยมและได้ผลลัพธ์ที่ดีอย่างมากตั้งแต่ปี 2017 เป็นต้นมาซึ่งงานวิจัยเหล่านี้ถูกรวบรวมไว้ที่ [80] ในส่วนของผู้เขียนได้มีส่วนร่วมในงานวิจัยที่เกี่ยวข้องกับการจำลองตัวละครด้วยฟิสิกส์ขั้นสูงดังนี้

งานวิจัย Physics-based motion capture imitation with deep reinforcement learning

[81] นำเสนอวิธีการใช้การลองผิดลองถูกเชิงลึกในการเทรนโครงข่ายประสาทเทียม (Neural Network) ที่ควบคุมตัวละครให้เคลื่อนที่เลียนแบบโมชันแคปเจอร์ โดยที่ยังมีความถูกต้องทางฟิสิกส์อยู่ เช่น แรงและแรงบิดภายในที่กระทำจะต้องมีผลรวมเป็นศูนย์ และแรงและแรงบิดที่ใช้สำหรับแต่ละข้อต่อจะต้องไม่เกินค่าที่กำหนด (ซึ่งเป็นการประมาณการแรงที่กล้ามเนื้อแต่ละส่วนของตัวละครออกแรงได้) ซึ่งโมชันแคปเจอร์อาจจะมีข้อผิดพลาดหรือมีสัญญาณรบกวน (noise) ได้และอาจจะไม่ได้มาจากตัวละครเดียวกันกับที่ใช้จำลองในคอมพิวเตอร์ ทำให้การเคลื่อนไหวซึ่งมาจากโมชันแคปเจอร์อาจจะไม่ถูกต้องทางฟิสิกส์เสมอไป ซึ่งวิธีที่นำเสนอในงานวิจัยนี้ต้องเรียนรู้ที่จะไม่สนใจข้อมูลจากโมชันแคปเจอร์บางส่วน เพื่อที่จะยังคงเคลื่อนไหวต่อไปโดยไม่ล้มลง ในขณะที่ในภาพรวมแล้วการเคลื่อนที่ยังคงคล้ายกับโมชันแคปเจอร์อยู่ ดังตัวอย่างในภาพประกอบ 61



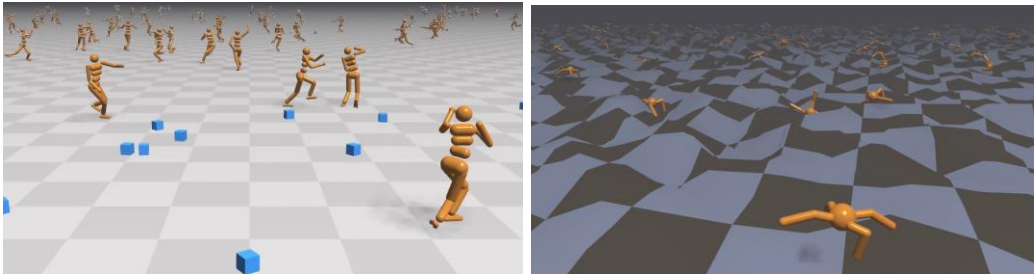
ภาพประกอบ 61: การควบคุมให้ตัวละครเคลื่อนที่อยากถูกต้องทางฟิสิกส์ (สีน้ำตาล) ที่เลียนแบบโมชันแคปเจอร์ (สีเขียว) เครดิต:

[81]

งานวิจัย GPU-Accelerated Robotic Simulation for Distributed Reinforcement Learning

[82] นำเสนอการวิจัยการจำลองฟิสิกส์ของตัวละครพร้อมทั้งอัลกอริทึมสำหรับการลองผิดลองถูกเชิงลึกที่ใช้ในการเทรนโครงข่ายประสาทเทียม เพื่อควบคุมการเคลื่อนที่ของตัวละครให้ทำตามวัตถุประสงค์ที่กำหนดเช่น เดินไปในทิศทางที่กำหนดด้วยความเร็วที่กำหนด โดยเริ่มจากสถานะไหนก็ได้ โดยที่ในฉากอาจจะมีพื้นขรุขระหรือสิ่งกีดขวาง และขั้นตอนทั้งหมดคำนวณใน GPU โดยสามารถประมวลผลได้เร็วกว่าใช้ CPU มาก ดังตัวอย่างในภาพประกอบ

62



ภาพประกอบ 62) ตัวละครที่คล้ายมนุษย์และตัวละครที่คล้ายแมลงเคลื่อนที่ตามหลักฟิสิกส์และควบคุมด้วยโครงข่ายประสาทเทียม เพื่อเคลื่อนที่ไปยังทิศและความเร็วที่กำหนด เครดิต: [82]

## การใช้ปัญญาประดิษฐ์ในคอมพิวเตอร์แอนิเมชันเชิงฟิสิกส์ขั้นสูง

ไม่กี่ปีที่ผ่านมาได้มีการนำปัญญาประดิษฐ์ (Artificial Intelligence) มาใช้ในการสร้างภาพเคลื่อนไหวเชิงฟิสิกส์มากขึ้นเรื่อยๆ โดยเฉพาะการใช้การเรียนรู้เชิงลึก (Deep Learning) อาทิเช่นการจำลองการแพร่กระจายของไฟ [83] การจำลองควัน [84], [85] การจำลองน้ำกระเซ็น [86] จำลองวัตถุที่ยืดหยุ่นได้ [87] เป็นต้น รวมถึงการนำการจำลองฟิสิกส์ที่สามารถคำนวณหาอนุพันธ์ได้ (Differentiable Physics Simulation) มาใช้เป็นส่วนหนึ่งของการเรียนรู้เชิงลึกเพื่อควบคุมการจำลองฟิสิกส์ให้เป็นไปตามเป้าหมายที่ต้องการได้เช่น ให้แกว่งเชือกให้ปลายมาชนกับเป้าหมายในงาน [88] ให้ควันท่อตัวเป็นรูปที่ต้องการและคลื่นน้ำเป็นดังภาพที่กำหนดในงาน [89] เป็นต้น รวมถึงการใช้การเรียนรู้เชิงลึกเพื่อเพิ่มรายละเอียดของการจำลองทางฟิสิกส์ให้ได้ผลใกล้เคียงกับโลกความจริงมากขึ้นเช่น เพิ่มการไหลแบบปั่นป่วน (Turbulent Flow) ในการจำลองของไหลในงาน [90] เป็นต้น โดยผู้เขียนมีส่วนร่วมในงานวิจัยที่เกี่ยวข้องกับการใช้ปัญญาประดิษฐ์ในคอมพิวเตอร์แอนิเมชันเชิงฟิสิกส์ขั้นสูงดังนี้

งานวิจัย Cloth and Skin Deformation with a Triangle Mesh Based Convolutional Neural Network [91] นำเสนอโครงข่ายประสาทเทียมรูปแบบใหม่ที่ทำงานบนเมชสามเหลี่ยมที่การเชื่อมต่อคงที่ (แต่ตำแหน่งของจุดยอดเปลี่ยนแปลง) ซึ่งสามารถใช้ในการแทนที่ผ้า เสื้อผ้า ผิวหนังได้ และนำเสนอตัวอย่างการใช้แก้ปัญหาที่เกี่ยวข้องกับการจำลองฟิสิกส์แบบต่างๆ เช่น การเพิ่มความละเอียด (Upsampling) ของเสื้อผ้าที่จำลองด้วยฟิสิกส์ที่ความละเอียดต่ำให้มีรอยยับรอยย่นเหมือนกับเสื้อผ้าที่จำลองด้วยความละเอียดสูง การหาฟังก์ชันที่แปลงตำแหน่งและการวางแนวของส่วนต่างๆ ของร่างกายของตัวละครไปเป็นการขยับและย่นของผ้าเพื่อใช้แทนการจำลองทางฟิสิกส์ การหาฟังก์ชันที่แปลงมุมของข้อต่อต่างๆ ของมือไปยังการขยับของผิวหนังของมือ โดยทั้งหมดนี้เรียนรู้จากผลของการจำลองทางฟิสิกส์ที่ความละเอียดสูงที่ใช้เวลาการคำนวณมาก โดยโครงข่ายประสาทเทียมสามารถคำนวณได้รวดเร็วกว่าหลายสิบถึงหลายร้อยเท่า ดังตัวอย่างในภาพประกอบ 63



ภาพประกอบ 63) การใช้โครงข่ายประสาทเทียมที่ทำงานบนเมชสามเหลี่ยมในการ ซ้าย) เพิ่มความละเอียดของผ้า กลาง) แทนที่การจำลองผ้าด้วยฟิสิกส์ ขวา) แทนที่การจำลองผิวหนังของมือที่ขยับด้วยเนื้อที่ขยับเนื่องจากกระดูก ขยับ เครดิต: [91]

งานวิจัย Tranquil Clouds: Neural Networks for Learning Temporally Coherent Features in Point Clouds [92] นำเสนอโครงข่ายประสาทเทียมที่ทำงานบนกลุ่มจุดโดยนำข้อมูลทั้งตำแหน่งและเวลามาใช้ในการเพิ่มจำนวนจุดให้มีมากขึ้นเพื่อเพิ่มรายละเอียดโดยมีการคำนึงถึงความต่อเนื่องเชิงเวลา (Temporal Coherence) ด้วยซึ่งสามารถนำมาใช้เพิ่มความละเอียดให้กับภาพเคลื่อนไหวของกลุ่มจุด และภาพเคลื่อนไหวของของเหลวที่แทนด้วยอนุภาคด้วย ดังตัวอย่างในภาพประกอบ 64



ภาพประกอบ 64) การเพิ่มจำนวนจุดด้วยการใช้โครงข่ายประสาทเทียมโดยคำนึงถึงความต่อเนื่องเชิงเวลา เครดิต:

[92]

## บทสรุป

บทนี้กล่าวถึงงานวิจัยและเอกสารอ้างอิงอื่นๆ สำหรับให้ผู้ที่มีความสนใจในการสร้างแอนิเมชันเชิงฟิสิกส์ขั้นสูงสามารถศึกษาต่อยอดจากเนื้อหาในแต่ละบทของหนังสือเล่มนี้ได้ โดยมีการอ้างอิงถึงงานวิจัยที่ผู้เขียนมีส่วนร่วมในการวิจัยด้วย นอกเหนือจากงานวิจัยที่กล่าวถึงในบทนี้แล้ว ผู้ที่สนใจยังสามารถติดตามงานวิจัยชั้นนำที่เกี่ยวกับการสร้างแอนิเมชันเชิงฟิสิกส์ได้จากวารสารชั้นนำระดับนานาชาติดังต่อไปนี้

1. ACM Transactions on Graphics (TOG)
2. IEEE Transactions on Visualization and Computer Graphics (TVCG)
3. Computer Graphics Forum (CGF)
4. An International Journal of Systems & Applications in Computer Graphics
5. Computer Animation and Virtual Worlds

และการประชุมนานาชาติดังต่อไปนี้

1. ACM SIGGRAPH
2. ACM SIGGRAPH / Eurographics Symposium on Computer Animation
3. Pacific Conference on Computer Graphics and Applications
4. Eurographics
5. Motion, Interaction, and Games (MIG)
6. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games
7. VRIPHYS: Virtual Reality Interactions and Physical Simulations

## บรรณานุกรม

- [1] J. R. Chasnov, "Matrix Algebra for Engineer," 2018. [Online]. Available: <https://www.math.ust.hk/~machas/matrix-algebra-for-engineers.pdf>.
- [2] M. J. Baker, "Maths - Conversion Matrix to Quaternion," 1998-2017. [Online]. Available: <http://www.euclideanspace.com/maths/geometry/rotations/conversions/matrixToQuaternion/>.
- [3] G. Strang, "Calculus Online Textbook," 1991. [Online]. Available: <https://ocw.mit.edu/resources/res-18-001-calculus-online-textbook-spring-2005/textbook/>.
- [4] U. o. B. Stephen Wiggins, "Open Text Book Library," 2017. [Online]. Available: <https://open.umn.edu/opentextbooks/textbooks/ordinary-differential-equations>.
- [5] E. Miersemann, "Partial Differential Equations," 2012. [Online]. Available: <https://www.math.uni-leipzig.de/~miersemann/pdebook.pdf>.
- [6] W. T. Reeves, "Particle Systems A Technique for Modeling a Class of Fuzzy Objects," *Computer Graphics, Volume 17, Number 3*, pp. 359-375, 1983.
- [7] "กฎของนิวตัน (Newton's laws)," 2003-2010. [Online]. Available: [http://www.pw.ac.th/emedia/media/science/lesa/2/law\\_orbit/newton/newton.html](http://www.pw.ac.th/emedia/media/science/lesa/2/law_orbit/newton/newton.html).
- [8] I. Newton, *Philosophiae Naturalis Principia Mathematica*, London, 1687.
- [9] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, Third Edition, John Wiley & Sons, Ltd, 2016.
- [10] A. Witkin and D. Baraff, "Physically Based Modeling: Principles and Practice (SIGGRAPH 97 Course Note)," 1997. [Online]. Available: <https://www.cs.cmu.edu/~baraff/sigcourse/notesd1.pdf>.

- [11] M. Müller, J. Bender, N. Chentanez and M. Macklin, "A Robust Method to Extract the Rotational Part of Deformations," in *MIG '16: Proceedings of the 9th International Conference on Motion in Games*, 2016.
- [12] E. Catto, "Fast and simple physics using sequential impulse," in *Game Developer Conference*, 2006.
- [13] D. Chappuis, "Constraints Derivation for Rigid Body Simulation in 3D," 2013. [Online]. Available: <https://danielchappuis.ch/download/ConstraintsDerivationRigidBody3D.pdf>.
- [14] P. Flores, R. Pereira, M. Machado and E. Seabra, "Investigation on the Baumgarte Stabilization Method for Dynamic Analysis of Constrained Multibody Systems," in *Proceedings of EUCOMES*, 2008.
- [15] C. Fares and Y. Hamam, "Collision Detection for Rigid Bodies: A State of the Art Review," in *GraphiCon 2005 - International Conference on Computer Graphics and Vision, Proceedings*, 2005.
- [16] J. Bender, K. Erleben, J. Trinkle and E. Coumans, "Interactive Simulation of Rigid Body Dynamics in Computer Graphics," in *EUROGRAPHICS 2012 State of the Art Reports*, Cagliari, Sardinia, Italy, Eurographics Association, 2012.
- [17] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke and V. Makoviychuk, "Non-Smooth Newton Methods for Deformable Multibody Dynamics," *ACM Transactions on Graphics*, 2019.
- [18] M. Müller, B. Heidelberger, M. Hennix and J. Ratcliff, "Position Based Dynamics," in *Virtual Reality Interactions and Physical Simulation "VRIPHYS"*, 2006.
- [19] J. Bender, M. Müller and M. Macklin, "A Survey on Position Based Dynamics," *EuroGraphics*, 2017.
- [20] M. Macklin, M. Muller, N. Chentanez and T.-Y. Kim, "Unified Particle Physics for Real-Time Applications," *ACM Transactions on Graphics*, 2014.

- [21] M. Macklin and M. Müller, "Position Based Fluids," *Transaction on Graphics*, 2013.
- [22] M. Müller, D. Charypar and M. Gross, "Particle-Based Fluid Simulation for Interactive Applications," in *Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003.
- [23] M. Kass and G. Miller, "Rapid, Stable Fluid Dynamics for Computer Graphics," *Computer Graphics*, 1990.
- [24] R. Bridson, "Fluid Simulation for Computer Graphics," in *Fluid Simulation for Computer Graphics*, A K Peters/CRC Press, 2008.
- [25] N. Thurey, M. Müller-Fischer, S. Schirm and M. Gross, "Real-time Breaking Waves for Shallow Water Simulations," in *Pacific Conference on Computer Graphics and Applications*, 2007.
- [26] N. Chentanez and M. Mueller., "Real-time simulation of large bodies of water with small scale details," in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2010.
- [27] J. Stam, "Stable Fluids," *ACM Transactions on Graphics*, 1999.
- [28] J. U. Brackbill and H. M. Ruppel, "FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimension," *Journal of Computational Physics*, pp. 314-343, 1986.
- [29] Y. Zhu and R. Bridson, "Animating Sand as a Fluid," *ACM Transactions on Graphics*, 2005.
- [30] D. Enright and R. Fedkiw, "Robust treatment of interfaces for fluid flows and computer graphics," in *Hyperbolic Problems: Theory, Numerics, Applications*, Springer, 2003, p. 153-164.
- [31] C. Batty, F. Bertails and R. Bridson, "A Fast Variational Framework for Accurate Solid-Fluid Coupling," in *ACM SIGGRAPH 2007 Papers*, New York, NY, USA, 2007.
- [32] C.-H. Chen, I.-C. Lin, M. Tsai and P.-H. Lu, "Lattice-Based Skinning and Deformation for Real-Time Skeleton-Driven Animation," *2011 12th International Conference on Computer-Aided Design and Computer Graphics*, pp. 306-312, 2011.

- [33] L. Kavan, S. Collins, J. Žára and C. O'Sullivan, "Skinning with Dual Quaternions," in *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, New York, NY, USA, 2007.
- [34] L. Kovar, M. Gleicher and F. Pighin, "Motion Graphs," *ACM Trans. Graph.*, vol. 21, p. 473–482, 7 2002.
- [35] K. Yin, K. Loken and M. van de Panne, "SIMBICON: Simple Biped Locomotion Control," in *ACM SIGGRAPH 2007 Papers*, New York, NY, USA, 2007.
- [36] G. Thomas, "Compute-Based GPU Particle Systems," in *Game Developers Conference*, San, 2014.
- [37] G. Bilotta, V. Zago and A. Héroult, "Design and Implementation of Particle Systems for Meshfree Methods with High Performance," in *High Performance Parallel Computing*, S. Chickerur, Ed., Rijeka, IntechOpen, 2019.
- [38] J. Bender, K. Erleben and J. Trinkle, "Interactive Simulation of Rigid Body Dynamics in Computer Graphics," *Computer Graphics Forum*, vol. 33, pp. 246-270, 2014.
- [39] M. Müller, N. Chentanez and T.-Y. Kim, "Real Time Dynamic Fracture with Volumetric Approximate Convex Decompositions," *ACM Trans. Graph.*, vol. 32, 7 2013.
- [40] M. Müller, N. Chentanez and M. Macklin, "Simulating Visual Geometry," in *Proceedings of the 9th International Conference on Motion in Games*, New York, NY, USA, 2016.
- [41] M. Müller, N. Chentanez, M. Macklin and S. Jeschke, "Long Range Constraints for Rigid Body Simulations," in *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, New York, NY, USA, 2017.
- [42] M. Müller, N. Chentanez, S. Jeschke and M. Macklin, "Cable Joints," *Comput. Graph. Forum*, vol. 37, p. 1–10, 2018.
- [43] M. Müller, M. Macklin, N. Chentanez, S. Jeschke and T.-Y. Kim, "Detailed Rigid Body Simulation with Extended Position Based Dynamics," *Comput. Graph. Forum*, vol. 39, 2020.

- [44] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke and V. Makoviychuk, "Non-Smooth Newton Methods for Deformable Multi-Body Dynamics," *ACM Trans. Graph.*, vol. 38, 10 2019.
- [45] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke and T.-Y. Kim, "Primal/Dual Descent Methods for Dynamics," *Comput. Graph. Forum*, vol. 39, 2020.
- [46] M. Toftedahl, "Which are the most commonly used Game Engines?," 30 9 2019. [Online]. Available:  
[https://www.gamasutra.com/blogs/MarcusToftedahl/20190930/350830/Which\\_are\\_the\\_most\\_commonly\\_used\\_Game\\_Engines.php](https://www.gamasutra.com/blogs/MarcusToftedahl/20190930/350830/Which_are_the_most_commonly_used_Game_Engines.php).
- [47] "Vellum Documentation," 2019. [Online]. Available:  
<https://www.sidefx.com/docs/houdini/vellum/index.html>.
- [48] M. Müller and N. Chentanez, "Wrinkle Meshes," in *Proceedings of the 2010 Eurographics/ACM SIGGRAPH Symposium on Computer Animation, SCA 2010, Madrid, Spain, 2010*, 2010.
- [49] M. Müller and N. Chentanez, "Solid simulation with oriented particles," *ACM Trans. Graph.*, vol. 30, p. 92, 2011.
- [50] M. Müller and N. Chentanez, "Adding Physics to Animated Characters with Oriented Particles," in *Proceedings of the 8th Workshop on Virtual Reality Interactions and Physical Simulations, VRIPHYS 2011, Lyon, France, 2011*, 2011.
- [51] M. Müller, T.-Y. Kim and N. Chentanez, "Fast Simulation of Inextensible Hair and Fur," in *VRIPHYS 12: 9th Workshop on Virtual Reality Interactions and Physical Simulations, Darmstadt, Germany, 2012. Proceedings*, 2012.
- [52] T.-Y. Kim, N. Chentanez and M. Müller-Fischer, "Long Range Attachments - A Method to Simulate Inextensible Clothing in Computer Games," in *Proceedings of the 2012*

*Eurographics/ACM SIGGRAPH Symposium on Computer Animation, SCA 2012, Lausanne, Switzerland, 2012, 2012.*

- [53] M. Müller, N. Chentanez, T.-Y. Kim and M. Macklin, "Strain Based Dynamics," in *The Eurographics / ACM SIGGRAPH Symposium on Computer Animation, SCA 2014, Copenhagen, Denmark, 2014, 2014.*
- [54] M. Müller, N. Chentanez, T.-Y. Kim and M. Macklin, "Air meshes for robust collision handling," *ACM Trans. Graph.*, vol. 34, p. 133:1–133:9, 2015.
- [55] N. Chentanez, M. Müller and M. Macklin, "Real-time simulation of large elasto-plastic deformation with shape matching," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Zurich, Switzerland, July 11-13, 2016, 2016.*
- [56] M. Macklin, M. Müller and N. Chentanez, "XPBD: position-based simulation of compliant constrained dynamics," in *Proceedings of the 9th International Conference on Motion in Games, MIG 2016, Burlingame, California, USA, October 10-12, 2016, 2016.*
- [57] M. Macklin, K. Storey, M. Lu, P. Terdiman, N. Chentanez, S. Jeschke and M. Müller, "Small steps in physics simulation," in *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2019, Los Angeles, CA, USA, July 26-28, 2019, 2019.*
- [58] R. Bridson, *Fluid Simulation for Computer Graphics*, Second Edition, Taylor & Francis, 2015.
- [59] B. Ren, X.-Y. Yang, M. C. Lin, N. Thuerey, M. Teschner and C. Li, "Visual Simulation of Multiple Fluids in Computer Graphics: A State-of-the-Art Report," *Journal of Computer Science and Technology*, vol. 33, p. 431–451, 2018.
- [60] D. Sulsky, Z. Chen and H. L. Schreyer, "A particle method for history-dependent materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 118, pp. 179-196, 1994.
- [61] A. Stomakhin, C. Schroeder, L. Chai, J. Teran and A. Selle, "A Material Point Method for Snow Simulation," *ACM Trans. Graph.*, vol. 32, 7 2013.

- [62] A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran and A. Selle, "Augmented MPM for phase-change and varied materials," *ACM Trans. Graph.*, vol. 33, pp. 138:1-138:11, 2014.
- [63] C. Jiang, C. Schroeder, A. Selle, J. Teran and A. Stomakhin, "The Affine Particle-in-Cell Method," *ACM Trans. Graph.*, vol. 34, 7 2015.
- [64] C. Fu, Q. Guo, T. Gast, C. Jiang and J. Teran, "A Polynomial Particle-in-Cell Method," *ACM Trans. Graph.*, vol. 36, 11 2017.
- [65] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana and C. Jiang, "A Moving Least Squares Material Point Method with Displacement Discontinuity and Two-Way Rigid Body Coupling," *ACM Trans. Graph.*, vol. 37, 7 2018.
- [66] B. M. Klingner, B. E. Feldman, N. Chentanez and J. F. O'Brien, "Fluid animation with dynamic meshes," *ACM Trans. Graph.*, vol. 25, p. 820–825, 2006.
- [67] N. Chentanez, T. Göktekin, B. E. Feldman and J. F. O'Brien, "Simultaneous coupling of fluids and deformable bodies," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2006, Vienna, Austria, September 2-4, 2006*, 2006.
- [68] N. Chentanez, B. E. Feldman, F. Labelle, J. F. O'Brien and J. R. Shewchuk, "Liquid simulation on lattice-based tetrahedral meshes," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2007, San Diego, California, USA, August 2-4, 2007*, 2007.
- [69] N. Chentanez and M. Müller, "Real-time Eulerian water simulation using a restricted tall cell grid," *ACM Trans. Graph.*, vol. 30, p. 82, 2011.
- [70] N. Chentanez and M. Müller, "Mass-Conserving Eulerian Liquid Simulation," in *Proceedings of the 2012 Eurographics/ACM SIGGRAPH Symposium on Computer Animation, SCA 2012, Lausanne, Switzerland, 2012*, 2012.

- [71] N. Chentanez and M. Müller-Fischer, "A Multigrid Fluid Pressure Solver Handling Separating Solid Boundary Conditions," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, p. 1191–1201, 2012.
- [72] N. Chentanez and M. Müller, "Real-time Simulation of Large Bodies of Water with Small Scale Details," in *Proceedings of the 2010 Eurographics/ACM SIGGRAPH Symposium on Computer Animation, SCA 2010, Madrid, Spain, 2010*, 2010.
- [73] N. Chentanez, M. Müller and T.-Y. Kim, "Coupling 3D Eulerian, Heightfield and Particle Methods for Interactive Simulation of Large Scale Liquid Phenomena," in *The Eurographics / ACM SIGGRAPH Symposium on Computer Animation, SCA 2014, Copenhagen, Denmark, 2014*, 2014.
- [74] N. Chentanez, M. Müller, M. Macklin and T.-Y. Kim, "Fast grid-free surface tracking," *ACM Trans. Graph.*, vol. 34, p. 148:1–148:11, 2015.
- [75] N. Chentanez, M. Müller, M. Macklin and T.-Y. Kim, "Grid-Free Surface Tracking on the GPU," in *Workshop on Virtual Reality Interaction and Physical Simulation, VRIPHYS 2015, Lyon, France, November 4-5, 2015*, 2015.
- [76] S. Jeschke, T. Skrivan, M. Müller-Fischer, N. Chentanez, M. Macklin and C. Wojtan, "Water surface wavelets," *ACM Trans. Graph.*, vol. 37, p. 94:1–94:13, 2018.
- [77] S. Jeschke, C. Hafner, N. Chentanez, M. Macklin, M. Müller and C. Wojtan, "Making Procedural Water Waves Boundary-aware," *Comput. Graph. Forum*, vol. 39, 2020.
- [78] K. Yin, S. Coros and M. van de Panne, "Biped Controller for Character Animation," in *Handbook of Human Motion*, B. Müller, S. I. Wolf, G. Brueggemann, Z. Deng, A. McIntosh, F. Miller and W. S. Selbie, Eds., Cham, Springer International Publishing, 2017, p. 1–23.
- [79] K. Yin, L. Liu and M. v. de Panne, "Simulation for Control," in *Humanoid Robotics: A Reference*, A. Goswami and P. Vadakkepat, Eds., Dordrecht, Springer Netherlands, 2018, p. 1–44.
- [80] J. Booth and V. Ivanov, *Realistic Physics Based Character Controller*, 2020.

- [81] N. Chentanez, M. Müller, M. Macklin, V. Makoviychuk and S. Jeschke, "Physics-based motion capture imitation with deep reinforcement learning," in *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games, MIG 2018, Limassol, Cyprus, November 08-10, 2018*, 2018.
- [82] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin and D. Fox, "GPU-Accelerated Robotic Simulation for Distributed Reinforcement Learning," in *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*, 2018.
- [83] B. Y. Lattimer, J. L. Hodges and A. M. Lattimer, "Using machine learning in physics-based simulation of fire," *Fire Safety Journal*, vol. 114, p. 102991, 2020.
- [84] J. Tompson, K. Schlachter, P. Sprechmann and K. Perlin, "Accelerating Eulerian Fluid Simulation With Convolutional Networks," *ArXiv e-prints*, 7 2016.
- [85] X. Xiao, Y. Zhou, H. Wang and X. Yang, "A Novel CNN-based Poisson Solver for Fluid Simulation," *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, pp. 1-1, 10 2018.
- [86] K. Um, X. Hu and N. Thuerey, "Liquid splash modeling with neural networks," in *Computer Graphics Forum*, 2018.
- [87] L. Fulton, V. Modi, D. Duvenaud, D. I. W. Levin and A. Jacobson, "Latent-space Dynamics for Reduced Deformable Simulation," *Computer Graphics Forum*, 2019.
- [88] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum and J. Z. Kolter, "End-to-end differentiable physics for learning and control," in *Advances in Neural Information Processing Systems*, 2018.
- [89] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley and F. Durand, "DiffTaichi: Differentiable programming for physical simulation," *arXiv preprint arXiv:1910.00935*, 2019.
- [90] K. Um, Raymond, Fei, P. Holl, R. Brand and N. Thuerey, *Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers*, 2020.

- [91] N. Chentanez, M. Müller, M. Macklin, S. Jeschke and T.-Y. Kim, "Cloth and Skin Deformation with a Triangle Mesh Based Convolutional Neural Network," *Comput. Graph. Forum*, vol. 39, 2020.
- [92] L. Prantl, N. Chentanez, S. Jeschke and N. Thuerey, "Tranquil Clouds: Neural Networks for Learning Temporally Coherent Features in Point Clouds," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.
- [93] D. Baraff, "Physically based modeling: Rigid body simulation," *SIGGRAPH Course Notes*, *ACM SIGGRAPH*, vol. 2, p. 2-1, 2001.

---

**A**

advection · 61  
angular momentum · 36  
angular velocity · 33  
animation · 1  
artificial intelligence · 102  
artificial viscosity · 60

---

**B**

baumgarte stabilization · 47  
bias coefficient · 47  
bilinear interpolation · 66, 71  
block diagonal · 45  
bone · 75  
broad phase collision detection · 48

---

**C**

calibrate · 78  
chain rules · 20  
clamping · 42  
close boundary · 63  
collider · 29  
collision · 38  
conjugate · 16  
constraint · 38  
constraint manifold · 50  
contact · 38  
convex polytope · 49  
coordinate · 8  
cross product · 10

---

**D**

deep learning · 102

definite integral · 22  
derivative · 19  
diagonal matrix · 12  
differentiable function · 21  
discretization · 62  
divergence · 59  
divergence theorem · 59  
dot product · 10  
drifting · 46  
dynamic friction coefficient · 55

---

**E**

elastic collision · 30  
emitter · 25  
error · 46  
eulerian · 58  
extrapolate · 73

---

**F**

finite difference method · 62  
fluid implicit particle (flip) · 70  
forward euler method · 27

---

**G**

gradient · 56  
grid · 58, 71

---

**H**

height field · 60

---

**I**

identity matrix · 13

implicit function · 40  
inbetweening · 77  
incompressible · 59  
incremental impulse · 47  
inelastic collision · 30  
integral · 21  
interaction · 38  
interpolate · 66  
inverse matrix · 14

---

## *J*

jacobi iteration · 69  
Jacobi Iteration · 73  
**jacobian** · 40  
jittering · 48  
joint · 38

---

## *K*

kernel · 56

---

## *L*

lagrangian · 58  
life time · 24  
linear blend skinning · 76

---

## *M*

magnetic motion capture · 78  
mass · 33  
matrix · 11  
mechanics motion capture · 78  
mid point rule · 28  
moment of inertia · 33  
momentum · 36  
motion blur · 31  
motion capture · 78

---

## *N*

narrow phase collision detection · 48  
navier-stokes equations · 59  
non-differentiable function · 21  
norm · 16  
numerical error · 37, 46

---

## *O*

operator splitting · 65  
optical motion capture) · 78  
ordinary differential equation · 22  
orientation · 33

---

## *P*

partial derivative · 23  
particle · 24  
particle system · 24  
periodic boundary · 63  
physics based computer animation · 2  
polytope · 49  
position · 24, 33  
position based dynamics · 50  
pre-conditioned conjugated gradient · 69  
pressure projection · 65  
principle of virtual work · 40  
projectile · 27  
proportional derivative controller · 81  
pseudo random number generator · 25  
pure quaternion · 17

---

## *Q*

quaternion · 15

---

## *R*

ragdoll simulation · 80  
restitution coefficient · 30  
rigid body · 33, 86  
rodrigues' rotation formula · 15

---

## S

scalar function · 39  
semi-implicit euler method · 27  
semi-lagrangian method · 65  
sequential impulses · 39  
shallow water equation · 61  
skinning · 76  
sprite · 30  
sprite sheet · 30  
stable fluid · 64  
state · 37  
static friction coefficient · 55  
symmetric matrix · 12

---

## T

temporal coherence · 104  
tetrahedral mesh · 54  
torque · 36  
total impulse · 47  
trace · 15  
transition · 79  
transpose · 12  
turbulent flow · 102

---

## U

unit quaternion · 17  
unit vector · 10  
upsampling · 103

---

## V

vector · 6  
vector function · 21  
velocity · 24, 33  
verlet integration · 28  
viscosity · 59

---

## W

wave equation · 61

work · 40  
world space · 33

---

## I

เกรเดียนต์ · 56  
เนื้อที่โลก · 33  
เนื้อที่ของข้อจำกัด · 50  
เมทริกซ์ · 11, 77  
เมทริกซ์เอกลักษณ์ · 13  
เมทริกซ์แนวทแยง · 12  
เมทริกซ์ก่อกองในแนวทแยง · 45  
เมทริกซ์จัตุรัส · 14  
เมทริกซ์ทแยง · 51  
เมทริกซ์ผกผัน · 14, 46  
เมทริกซ์สมมาตร · 12  
เมทริกซ์สลับเปลี่ยน · 12  
เวกเตอร์ · 6, 40  
เวกเตอร์หน่วย · 10, 17  
เวลาชีวิต · 24

---

## II

แผ่นสไปรท์ · 30  
แรงเสียดทาน · 44  
แรงบิด · 36  
แอนิเมชันของตัวละคร · 75

---

## III

โพรเจกไทล์ · 27  
โมเมนต์ความเฉื่อย · 33  
โมเมนต์ · 36, 51  
โมเมนต์เชิงมุม · 36, 51

โมชั่นแคบเจอร์ · 78  
โมชั่นแคบเจอร์แบบแม่เหล็ก · 78  
โมชั่นแคบเจอร์แบบแสง · 78  
โมชั่นแคบเจอร์แบบกล · 78

---

## ๗

ไคเวอร์เจนท์ · 59, 68, 70  
ไม่สามารถบีบอัดได้ · 59

---

## ก

กฎของนิวตัน · 26  
กฎจุดกลาง · 28, 37  
กฎลูกโซ่ · 20  
กระดูก · 75  
การเปลี่ยนผ่าน · 79  
การเพิ่มความละเอียด · 103  
การเรียนรู้เชิงลึก · 102  
การเคลื่อน · 46  
การแบ่งไอเปอร์เตอร์ · 65  
การโปรเจกความดัน · 65  
การไหลแบบปั่นป่วน · 102  
การขยาย · 73  
การคูณเชิงเวกเตอร์ · 10  
การคูณเชิงสเกลาร์ · 10  
การคูณเมทริกซ์ · 13  
การจำลองตัวละคร · 101  
การจำลองศู๊กตาฟ้า · 80  
การชน · 38  
การชนแบบไม่ยืดหยุ่น · 30  
การชนแบบยืดหยุ่น · 30  
การลดตามลำดับ · 39  
การลดทั้งหมด · 47

การลดที่เพิ่มขึ้นมา · 47  
การตรวจแบบแม่นยำ · 48  
การตรวจแบบคร่าว · 48  
การทำให้เป็นดิสครีต · 62  
การทำให้มีน้คงของบวมการ์ด · 47  
การนำพา · 61  
การประมาณค่าในช่วง · 66  
การเปรียบเทียบ · 78  
การมีว่เนื่องจากเคลื่อนที่ · 31  
การยืด · 42  
การวางแนว · 33, 75  
การวาดภาพระหว่างลึฟเฟรม · 77  
การสกินนึ่ง · 76  
การสั้น · 48  
การสัมผัส · 38

---

## ข

ขนาด · 16  
ขนาดของเวกเตอร์ · 9  
ข้อจำกัด · 38, 43, 50  
ข้อจำกัดการงอ · 52  
ข้อจำกัดการยืด · 52  
ข้อจำกัดของแรงเสียดทานระหว่างอนุภาคกับอนุภาค · 55  
ข้อจำกัดของการชนระหว่างอนุภาคกับสามเหลี่ยม · 54  
ข้อจำกัดของการชนระหว่างอนุภาคกับสิ่งแวดล้อม · 54  
ข้อจำกัดของการชนระหว่างอนุภาคกับอนุภาค · 54  
ข้อจำกัดพื้นที่ของสามเหลี่ยม · 53  
ข้อต่อ · 38, 80  
ขอบซ้ำเป็นช่วงๆ · 63  
ขอบปิด · 63  
ข้อผิดพลาด · 46  
ข้อผิดพลาดทางตัวเลข · 46

---

**ก**

คณิตศาสตร์ · 6  
ควอเตอร์เนียน · 15  
ควอเตอร์เนียนบริสุทธ์ · 17  
ควอเตอร์เนียนหน่วย · 17  
ความเร็ว · 24, 33, 50  
ความเร็วเชิงมุม · 33  
ความต่อเนื่องเชิงเวลา · 104  
ความคิดพลาดทางตัวเลข · 37  
ความหนืด · 59  
ความหนืดเทียม · 60  
ค่าผัน · 16

---

**ง**

งาน · 40

---

**จ**

จักรमुख · 54  
จาโคเบียน · 40

---

**ด**

ตัวเลขสุ่มหลอก · 25  
ตัวควบคุมอนุพันธ์ตามสัดส่วน · 81  
ตัวปล่อยอนุภาค · 25  
ตาราง · 58, 71  
ตำแหน่ง · 24, 33, 50, 75

---

**ท**

ทรงหลายเหลี่ยม · 49  
ทรงหลายเหลี่ยมแบบป่อง · 49  
ทฤษฎีบทไดเวอร์เจนท์ · 59

---

**ป**

ปฏิยานุพันธ์ · 21, 66  
ปฏิยานุพันธ์แบบเวอเลอร์ · 28  
ปฏิยานุพันธ์แบบมิชอบเขต · 22  
ปฏิสัมพันธ์ · 38  
ปัญหาประติษฐ์ · 102

---

**พ**

พิกัด · 8

---

**ฟ**

ฟังก์ชันแก่น · 56  
ฟังก์ชันโดยนัย · 40  
ฟังก์ชันที่ไม่สามารถหาอนุพันธ์ได้ · 21  
ฟังก์ชันที่มีค่าเป็นเวกเตอร์ · 21  
ฟังก์ชันที่สามารถหาอนุพันธ์ได้ · 21  
ฟิสิกส์ · 1, 6

---

**ภ**

ภาพเคลื่อนไหว · 1  
ภาพเคลื่อนไหวเชิงฟิสิกส์ · 2

---

**ม**

มวล · 33, 50

---

**ร**

รอย · 15

ระบบอนุภาค · 24, 84

---

**ล**

ลากรางเจียน · 58

---

**ว**

วัตถุแกร่ง · 33, 80, 84

วิธีไดนามิกเชิงตำแหน่ง · 50, 88

วิธีการออยเลอร์แบบไปข้างหน้า · 27, 37

วิธีการออยเลอร์แบบกึ่งนัย · 27, 37, 63

วิธีกึ่งลากรางเจียน · 65

วิธีของไหลเสถียร · 64

วิธีผลต่างจำกัด · 62

---

**ส**

สเกลาร์ · 6

สเกลาร์ฟังก์ชัน · 39

สไปร์ท · 30

สถานะ · 37

สนามความสูง · 60, 70

สมการเชิงอนุพันธ์ย่อยนาเวียร์-สโตกส์ · 59

สมการเชิงอนุพันธ์สามัญ · 22

สมการของน้ำเดิน · 61

สมการคลื่น · 61

สัมประสิทธิ์แรงเสียดทาน · 55

สัมประสิทธิ์แรงเสียดทานเคลื่อนไหว · 55

สัมประสิทธิ์แรงเสียดทานหยุดนิ่ง · 55

สัมประสิทธิ์การกระดอน · 30, 44

สัมประสิทธิ์ออกคิ · 47

สิ่งกีดขวาง · 29

---

**ห**

หลักการงานเสมือน · 40

---

**อ**

อนุพันธ์ · 19

อนุพันธ์ย่อย · 23

อนุภาค · 24

ออยเลอร์เรียน · 58

## ประวัติผู้เขียน



รองศาสตราจารย์ ดร.ณัฐพงศ์ ชินธเนศ  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
จุฬาลงกรณ์มหาวิทยาลัย

### คุณวุฒิ

Ph.D. (Computer Science)	University of California, Berkeley พ.ศ. 2550
M.Sc. (Computer Science)	University of California, Berkeley พ.ศ. 2547
B.S.E. (Computer Science/ Electrical Engineering)	University of Michigan, Ann Arbor พ.ศ. 2545

### กิจกรรมทางวิชาชีพ

#### ผู้อ่านบทความ Peer Review

SIGGRAPH 2008, 2010-2020 SIGGRAPH Asia 2010, 2011-2018, ACM Transaction on Computer Graphics 2012-2017, Symposium of Computer Animation 2012-2018, Journal of Computing And Information Science In Engineering (JCISE) 2010, Eurographics 2010,2012, Pacific Graphics 2009, Source Code for Biology 2013, The Visual Computer Journal 2013, 2015

## คณะกรรมการ

Symposium of Computer Animation 2011, 2013, 2015-2020

## การบรรยายรับเชิญ

ACM SIGGRAPH Motion, Interaction and Games, Keynote, South Carolina, USA, 2020

IEEE SIMPAR Plenary Talk, Brisbane, Australia, 2018

Intel Corporation, Hillsboro, Oregon, November, 2008

Stanford University, Palo Alto, California, May, 2009

## ผลงานทางวิชาการ

ก.บทความวิจัยในวารสาร (Research Article) 2018-2020

1. **Chentanez, N., Macklin, M., Müller, M., Jeschke, S., & Kim, T. Y. (2020).** Cloth and Skin Deformation with a Triangle Mesh Based Convolutional Neural Network [Article]. *Computer Graphics Forum*, 39(8), 123-134. **Scopus**
2. **Müller, M., Macklin, M., Chentanez, N., Jeschke, S., & Kim, T. Y. (2020).** Detailed rigid body simulation with extended position based dynamics. 2020 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2020.
3. **Müller M., Macklin M., Chentanez N., Jeschke S. & Kim T.-Y. (2020).** Detailed Rigid Body Simulation with Extended Position Based Dynamics. *Computer Graphics Forum*. Volume 39, Issue 8 (December 2020): 101-112. **Scopus**
4. **Chentanez N., Macklin M., Müller M., Jeschke S. & Kim T.-Y. (2020).** Cloth and Skin Deformation with a Triangle Mesh Based Convolutional Neural Network. *Computer Graphics Forum*. Volume 39, Issue 8(December 2020):123-134. **Scopus**
5. **Jeschke S., Hafner C., Chentanez N., Macklin M., Müller-Fischer M. & Wojtan C. (2020).** Making Procedural Water Waves Boundary-aware. *Computer Graphics Forum*. Volume 39, Issue 8(December 2020):47-54. **Scopus**
6. **Macklin M., Erleben K., Müller M., Chentanez N., Jeschke S. & Kim T.Y. (2020).** Primal/Dual Descent Methods for Dynamics. *Computer Graphics Forum*. Volume 39, Issue 8(December 2020):89-100. **Scopus**

7. **Macklin M., Erleben K., Müller M., Chentanez N., Jeschke S., Makoviychuk V. (2019).** Non-smooth Newton methods for deformable multi-body dynamics. *ACM Transactions on Graphics*. Volume 38, Issue 5(October 2019):1-10. Article number 140. **Scopus**
8. **Vanakittistien N., Sudsang A.& Chentanez N. (2019).** Game-ready 3D hair model from a small set of images. *Computer Animation and Virtual Worlds*. Volume 30, Issue 2(March-April 2019). Article number e1855. **Scopus**
9. **Müller M., Chentanez N., Jeschke S., Macklin M.** Cable Joints. *Computer Graphics Forum*. Volume 37, Issue 8(December 2018):1-10. **Scopus**
10. **Eaksarayut W., Tunwattanapong B., Sitthi-Amorn P., Chentanez N.** Mesh-based multi-view normal integration with energy minimization using surface reflectance properties. *Journal of WSCG*. Volume 26, Issue 2(2018):85-93. **Scopus**
11. **Jeschke S., Skřivan T., Müller-Fischer M., Chentanez N., Macklin M., Wojtan C.** Water surface wavelets. *ACM Transactions on Graphics*. Volume 37, Issue 4(2018). Article number A55. **Scopus**

**ข. รายงานการประชุมฉบับสมบูรณ์ (ที่มี peer review) 2018-2020**

1. **Macklin M., Storey K., Lu M., Terdiman P., Chentanez N., Jeschke S., Müller M.** Small steps in physics simulation. 18th Annual ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA 2019; Los Angeles; United States. **26 - 28 July 2019**. Article number 2.
2. **Chentanez N., Müller M., Macklin M., Makoviychuk V., Jeschke S.** Physics-based motion capture imitation with deep reinforcement learning. 11th Annual International Conference on Motion, Interaction and Games, MIG 2018; Limassol; Cyprus. **8 -10 November 2018**.