สำนักหอสมุดกลาง พระจอมเกล้าฉลาดกระบัง

การแก้ปัญหาไฟไนต์เอลิเมนต์แบบขนานบนระบบคลัสเตอร์ของพีซี

PARALLEL FINITE ELEMENT PROBLEM SOLVING ON THE CLUSTER OF PCs

ตฤณ แก้วปัญญา
กิติคุณ ล้วนตั้งศรีสุข
ธนวรรษ พานนิล

วน.
ต 198 ก
2550
เลขหมู่.....
เลขทะเบียน 82763
วัน,เดือน,ปี 23 ก.ค. 2551

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

b 11950699
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARALLEL FINITE ELEMENT PROBLEM SOLVING ON THE CLUSTER OF PCs

TRIN KAEWPANYA

KITIKUN LUNGTASESUK

THANAWAT PANNIL

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การแก้ปัญหาไฟไนต์เอลิเมนต์แบบขนานบนระบบคลัสเตอร์ของพีซี PARALLEL FINITE ELEMENT PROBLEM SOLVING ON THE CLUSTER OF PCs
ชื่อนักศึกษา	นายตฤณ แก้วปัญญา 47050329 นายกิติคุณ ล้วนตั้งศรีสุข 47050767 นายธนวรรษ พานนิล 47050783
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.จีรพร วีระพันธุ์ รองศาสตราจารย์ธีรวัฒน์ ประกอบผล

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2550

คณะกรรมการสอบ	ลายมือชื่อ
ผู้ช่วยศาสตราจารย์ กฤษฎา บุศรา ประธานกรรมการ	
อาจารย์วิสันต์ ตั้งวงษ์เจริญ กรรมการ	
ผู้ช่วยศาสตราจารย์ ดร.จีรพร วีระพันธุ์ กรรมการและอาจารย์ที่ปรึกษา	
รองศาสตราจารย์ ธีรวัฒน์ ประกอบผล กรรมการและอาจารย์ที่ปรึกษา	

(รองศาสตราจารย์ ไพรบูลย์ พันธรักษ์พงษ์)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แด่พ่อแม่และผู้เป็นที่รักทุกคน

ตฤณ

แด่พ่อแม่และครอบครัวอันเป็นที่รัก

กิติคุณ

แด่พ่อแม่ ครอบครัว และคนรักที่คอยเป็นกำลังใจ

ธนวรรษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การแก้ปัญหาไฟไนต์เอลิเมนต์แบบขนานบนระบบคลัสเตอร์ของพีซี
ชื่อนักศึกษา	นายตฤณ แก้วปัญญา 47050329
	นายกิติคุณ ล้วนตั้งศรีสุข 47050767
	นายธนวรรษ พานนิล 47050783
ปริญญา	วิทยาศาสตรบัณฑิต
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตรจารย์ ดร.จีรพร วีระพันธุ์
	รองศาสตราจารย์ธีรวัฒน์ ประกอบผล

บทคัดย่อ

ปัจจุบันเทคโนโลยีด้านการติดต่อสื่อสารและเครื่องคอมพิวเตอร์ส่วนบุคคลนั้น ได้มีการพัฒนาไปอย่างรวดเร็ว ดังนั้นจึงเป็นโอกาสที่ดีในการนำเครื่องคอมพิวเตอร์เหล่านี้มาใช้ประโยชน์แทนที่ซุปเปอร์คอมพิวเตอร์หรือเครื่องคอมพิวเตอร์แบบเมนเฟรม ที่มีต้นทุนสูงและยากแก่การบำรุงรักษา จึงได้มีการพัฒนาระบบคลัสเตอร์ ขึ้นมาใช้งานกันอย่างแพร่หลายในปัจจุบันสำหรับการประมวลผลทั้งแบบลำดับขั้นและแบบขนาน ปัญหาพิเศษนี้มีจุดประสงค์เพื่อนำการประมวลผลแบบขนานบนระบบคลัสเตอร์ของพีซีมาประยุกต์ใช้ในการแก้ปัญหาสมการปัวส์ซองด้วยระเบียบวิธีไฟไนต์เอลิเมนต์ ในการวัดประสิทธิภาพของการประมวลผลแบบขนานจะทำการเปรียบเทียบกับการประมวลผลแบบลำดับขั้นว่าแนวโน้มของเวลาที่ใช้ในการประมวลผลลดลงอย่างไรเมื่อเพิ่มหน่วยประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	PARALLEL FINITE ELEMENT PROBLEM SOLVING ON THE CLUSTER OF PCs
Students	Mr.Trin Kaewpanya 47050329
	Mr.Kitikun Lungtasesuk 47050767
	Mr.Thanawat Pannil 47050783
Degree	Bachelor of Science
Department	Mathematics and Computer Science , Faculty of Science
Programme	Computer Science
Academic Year	2007
Special Project Advisor	Assistant Professor Dr.Jeeraporn Werapun
	Associate Professor Teerawat Prakobphon

ABSTRACT

Nowadays, the technologies of communication and personal computers (PCs) have a rapid development. This provides the great opportunity to use a number of PCs connected as a cluster instead of a super computer or a mainframe computer, which is very expensive and difficult to maintenance. The cluster of PCs can execute both sequential processing and parallel processing. The propose of this special project is using this system for solving the Poisson's equation by the finite element method. In the performance evaluation, the response time of the sequential program and parallel program were recorded and compared when a number of processors were increased in terms of speed up and efficiency.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่อง การแก้ปัญหาไฟไนต์เอลิเมนต์แบบขนานบนระบบคลัสเตอร์ของพีซี (Parallel Finite Element Problem Solving on the Cluster of PCs) นี้สำเร็จลุล่วงได้เป็นอย่างดี ด้วยคำแนะนำ คำปรึกษาและคอยดูแลจากหลายๆฝ่ายด้วยกัน โดยเฉพาะอาจารย์ที่ปรึกษาที่ให้โอกาสให้ข้าพเจ้าได้ทำปัญหาพิเศษนี้ คอยให้ความเอาใจใส่ แนะนำ และให้ความช่วยเหลือ รวมถึงเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้ คือ ผศ.ดร.จีรพร วีระพันธุ์ และ รศ.ธีรวัฒน์ ประกอบผล ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ที่ได้จัดเตรียมสิ่งอำนวยความสะดวก เพื่อให้การศึกษาและพัฒนาระบบเป็นไปด้วยความสะดวกและรวดเร็ว รวมทั้งยังมีอินเทอร์เน็ตความเร็วสูงให้บริการสำหรับการค้นคว้าข้อมูลต่างๆ ซึ่งท้ายที่สุดแล้วจึงประกอบเป็นส่วนหนึ่งของปัญหาพิเศษนี้

ขอขอบพระคุณ รศ.ดร.มนัส สังวรศิลป์ ผู้อำนวยการสำนักวิจัยและบริการคอมพิวเตอร์ ที่กรุณาอนุญาตให้พวกเราได้ไปทำการศึกษาและทดลองใช้งานระบบคลัสเตอร์ที่สำนักวิจัยและบริการคอมพิวเตอร์ ขอบคุณพี่สมโชค กิ้มปาน และ พี่ๆฝ่ายพัฒนาระบบและโปรแกรมที่สำนักวิจัยและบริการคอมพิวเตอร์สำหรับคำแนะนำ และคำปรึกษาที่ดีให้กับพวกเรา

ขอขอบคุณพี่ๆเพื่อนๆ สาขาวิทยาการคอมพิวเตอร์โดยเฉพาะนายปิติพงษ์ อักโขที่คอยให้คำปรึกษาส่วนหนึ่งและคอยเป็นกำลังใจให้ความครื้นเครงโดยตลอด

และสุดท้ายนี้ต้องขอขอบคุณ บิดา มารดาและบุคคลในครอบครัวอันเป็นที่เคารพรัก คอยสั่งสอนพวกเรามาเป็นอย่างดี พร้อมให้โอกาสในการศึกษาอย่างเต็มที่และยังให้กำลังใจเสมอมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย i
บทคัดย่อภาษาอังกฤษ ii
กิตติกรรมประกาศ iii
สารบัญ iv
สารบัญภาพ vii
สารบัญตาราง xi
คำอธิบายสัญลักษณ์และคำย่อ xii
บทที่ 1 บทนำ 1
1.1 ความสำคัญและที่มาของปัญหา 1
1.2 วัตถุประสงค์ของการทำวิจัย 2
1.3 ขอบเขตของปัญหา 2
1.4 ประโยชน์คาดว่าจะได้รับ 3
1.5 ขั้นตอนในการดำเนินงาน 4
1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ 4
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 5
2.1 ทฤษฎีและหลักการของระเบียบวิธีไฟไนต์เอลิเมนต์ 5
2.1.1 วิธีการไฟไนต์เอลิเมนต์ 7
2.1.2 ข้อแตกต่างระหว่างวิธีการผลต่างสืบเนื่องและวิธีการไฟไนต์เอลิเมนต์ 8
2.1.3 ขั้นตอนทั่วไปของระเบียบวิธีไฟไนต์เอลิเมนต์ 8
2.2 การสร้างสมการโดยวิธีการแปรผัน 11
2.2.1 วิธีการแปรผัน 11
2.2.2 ฟังก์ชันการประมาณภายในเอลิเมนต์สองมิติ 12
2.2.3 การสร้างสมการไฟไนต์เอลิเมนต์แบบสองมิติโดยวิธีการแปรผัน 16
2.3 การสร้างสมการไฟไนต์เอลิเมนต์โดยวิธีการถ่วงน้ำหนักเศษตกค้าง 19
2.3.1 วิธีกาเลอร์คิน (Galerkin) 19
2.3.2 ขั้นตอนการถ่วงน้ำหนักเศษตกค้างสำหรับไฟไนต์เอลิเมนต์ 20
2.4 ฟังก์ชันการประมาณภายในเอลิเมนต์และการอินทิเกรตเอลิเมนต์เมตริกซ์เชิงตัวเลข ..23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.5 วิธีของจาโคบี 26
2.6 การประมวลผลแบบขนาน (Parallel processing) 27
2.6.1 การเขียนโปรแกรมแบบขนาน (Parallel programming) 28
2.6.2 Parallel Computer Architectures 28
2.6.3 Parallel Programming Paradigms 30
2.6.4 วิธีการโปรแกรมแบบขนาน 32
2.6.5 ขั้นตอนการสร้างโปรแกรมแบบขนาน 32
2.7 มาตรฐานที่ใช้ในการส่งเมสเซส 34
2.7.1 PVM (Parallel Virtual Machine) 34
2.7.2 MPI (Message Passing Interface) 34
2.8 ระบบคลัสเตอร์ (Cluster system) 38
2.8.1 จุดประสงค์ของการนำเทคโนโลยีคลัสเตอร์มาใช้งาน 40
2.8.2 ชนิดของคลัสเตอร์ 42
2.9 การวัดผลของประสิทธิภาพของการประมวลผลแบบขนาน 43
2.10 Sun Grid Engine (SGE) 44

บทที่ 3 ขั้นตอนการดำเนินงานและหลักการทำงานของโปรแกรม 45
3.1 ขั้นตอนในการดำเนินงาน 45
3.2 ลักษณะการทำงานของโปรแกรม 48
3.3 ขั้นตอนการทำงานของโปรแกรม 49
3.4 ส่วนของการรับส่งข้อมูลสำหรับโปรแกรมแบบขนาน 68
3.4.1 ส่วนที่ทำการรวมเมตริกซ์ K ของแต่ละเอลิเมนต์เข้าด้วยกันเป็นเมตริกซ์ K ของระบบ 68
3.4.2 ส่วนที่ทำการหาผลเฉลยด้วยวิธีการของจาโคบี 69

บทที่ 4 การทดลองและประเมินผล 70
4.1 การทดลองเพื่อทดสอบความถูกต้องของผลลัพธ์ 70
4.2 การทดลองเพื่อวัดเวลาที่ใช้ในการประมวลผล 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 5 สรุปผล ปัญหาและอุปสรรค ข้อเสนอแนะ 80

5.1 สรุปผลการดำเนินงาน 80

5.2 ปัญหาและอุปสรรค 80

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ 81

รายการอ้างอิง 82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

2.1 ลักษณะของแผ่นอะลูมิเนียมที่ใช้เสริมความแข็งแรงของโครงสร้างภายในปีกเครื่องบิน 6
2.2 ตารางสี่เหลี่ยมที่ใช้วิธีผลต่างสืบเนื่อง 6
2.3 การวิเคราะห์หาผลเฉลยบนแผ่นอะลูมิเนียมด้วยการใช้วิธีการไฟไนต์เอลิเมนต์ 7
2.4 แสดงถึงการแบ่งรูปร่างลักษณะของปัญหาออกเป็นเอลิเมนต์แบบต่าง ๆ 8
2.5 เอลิเมนต์สามเหลี่ยมแบบอย่าง ประกอบด้วยสามจุดต่อ โดยมีตัวไม่รู้ค่าอยู่
ณ ตำแหน่งที่จุดต่อ.....9
2.6 การแบ่งขอบเขตออกเป็นไฟไนต์เอลิเมนต์ย่อย 12
2.7 เอลิเมนต์สามเหลี่ยม 12
2.8 ตัวอย่างเอลิเมนต์สามเหลี่ยม 14
2.9 ลักษณะการกระจายของฟังก์ชันการประมาณภายในเอลิเมนต์ 15
2.10 ลักษณะการกระจายของผลเฉลยโดยประมาณภายในเอลิเมนต์ 15
2.11 การแบ่งลักษณะรูปร่างของปัญหาออกเป็นเอลิเมนต์.....21
2.12 ฟังก์ชันการประมาณภายในเอลิเมนต์รูปสามเหลี่ยม 24
2.13 เอลิเมนต์สามเหลี่ยมอันดับสูงที่ประกอบด้วยหลายจุดต่อ 25
2.14 ลักษณะการแบ่งงานของการประมวลผลแบบขนาน 27
2.15 ประสิทธิภาพและข้อจำกัดของการประมวลผลแบบขนาน 27
2.16 Shared Memory.....28
2.17 Distributed Memory 29
2.18 Hybrid Distributed-Shared Memory 29
2.19 ตัวอย่างของ Timing Threads Model 30
2.20 ตัวอย่างการรับส่งข้อมูลในData Parallel Model 31
2.21 ตัวอย่างการแบ่งข้อมูลในData Parallel Model 32
2.22 ตัวอย่างการ Decompose Data Set 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

2.23 ระบบ CPlant ที่ Sandia National Laboratory 39
2.24 โครงสร้างระบบแบวูฟล์คลัสเตอร์ 40
2.25 พิรุณคลัสเตอร์ที่มหาวิทยาลัยเกษตรศาสตร์ 40
2.26 แนวคิดของการประมวลผลแบบขนาน 41
2.27 ระบบ High Availability 41
2.28 โครงสร้างระบบคลัสเตอร์แบบปิด 42
2.29 โครงสร้างระบบคลัสเตอร์แบบเปิด 42
3.1 ผลลัพธ์ของโปรแกรม SerialMatrix.c (1) 46
3.2 ผลลัพธ์ของโปรแกรม SerialMatrix.c (2) 46
3.3 ผลลัพธ์ของโปรแกรม ParallelMatrix.c 47
3.4 แสดงลำดับขั้นตอนการทำงานของโปรแกรม 49
3.5 ภาพตัวอย่างของไฟล์ problem.txt 50
3.6 ลักษณะของปัญหาหลังจากรับอินพุตและแบ่งเอลิเมนต์ 51
3.7 ลักษณะของปัญหาหลังจากกำหนดหมายเลขให้แก่จุดต่อทุกจุดแล้ว 52
3.8 ระยะห่างระหว่างจุดต่อในแนวแกน x และ y 53
3.9 ลักษณะของเอลิเมนต์ที่ 1 54
3.10 kMatrix[][] ของเอลิเมนต์ที่ 1 55
3.11 ลักษณะของเอลิเมนต์ที่ 2 55
3.12 kMatrix[][] ของเอลิเมนต์ที่ 2 55
3.13 ลักษณะของเอลิเมนต์ที่ 3 56
3.14 kMatrix[][] ของเอลิเมนต์ที่ 3 56
3.15 ความสัมพันธ์ระหว่างหมายเลขของจุดต่อกับตำแหน่งใน kMatrix[][] 57
3.16 kSys[][] หลังจากนำค่าใน kMatrix[][] ของเอลิเมนต์ที่ 1 มาใส่แล้ว 57
3.17 kSys[][] หลังจากนำค่าใน kMatrix[][] ของเอลิเมนต์ที่ 2 มารวมแล้ว 58
3.18 kSys[][] หลังจากนำค่าใน kMatrix[][] ของเอลิเมนต์ที่ 3 มารวมแล้ว 58
3.19 kSys[][]หลังจากนำค่าใน kMatrix[][] ของแต่ละเอลิเมนต์มารวมกันแล้ว 59
3.20 fMatrix[] ของเอลิเมนต์ที่ 1 59
3.21 fSys[] หลังจากนำค่าใน fMatrix[] ของเอลิเมนต์ที่ 1 มาใส่แล้ว 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

3.22 fSys[] หลังจากนำค่าใน fMatrix[] ของเอลิเมนต์ที่ 2 มารวมแล้ว 60
3.23 fSys[] หลังจากนำค่าใน fMatrix[] ของทุกเอลิเมนต์มารวมกันแล้ว 61
3.24 fSys[] หลังจากนำค่า x1, x2 และx3 ของเอลิเมนต์ที่ 1 มาใส่แล้ว 61
3.25 fSys[] หลังจากนำค่า x1, x2 และx3 ของเอลิเมนต์ที่ 2 มารวมแล้ว 62
3.26 fSys[] หลังจากนำค่า x1, x2 และx3 ของเอลิเมนต์ที่เหลือมารวมแล้ว 62
3.27 fSys[] หลังจากนำค่า y1, y2 และy3ของเอลิเมนต์ที่ 1 มาใส่แล้ว 63
3.28 fSys[] หลังจากนำค่า y1, y2 และy3ของเอลิเมนต์ที่ 2 มารวมแล้ว 63
3.29 fSys[] หลังจากนำค่า y1, y2 และy3ของเอลิเมนต์ที่เหลือมารวมแล้ว 64
3.30 fSys[] หลังจากนำค่า x1*y1, x2*y2 และx3*y3ของเอลิเมนต์ที่ 1 มาใส่แล้ว 64
3.31 fSys[] หลังจากนำค่า x1*y1, x2*y2 และx3*y3ของเอลิเมนต์ที่ 2 มารวมแล้ว 65
3.32 fSys[] หลังจากนำค่า x1*y1, x2*y2 และx3*y3ของเอลิเมนต์ที่เหลือมารวมแล้ว 65
3.33 kSys[][] หลังจากกำหนดค่าขอบเขตแล้ว 66
3.34 fSys[] หลังจากปรับค่าตามค่าขอบเขตที่รับมาแล้ว 67
3.35 ลักษณะของระบบสมการที่จะทำการหาผลเฉลย 67
4.1 ตัวอย่างผลลัพธ์ของโปรแกรมแบบลำดับขั้น 70
4.2 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับเวลาที่ใช้ในการประมวลผลของ
โปรแกรมสำหรับปัญหาขนาด 50x50 73
4.3 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับเวลาที่ใช้ในการประมวลผลของ
โปรแกรมสำหรับปัญหาขนาด 100x100 73
4.4 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับเวลาที่ใช้ในการประมวลผลของ
โปรแกรมสำหรับปัญหาขนาด 150x150 74
4.5 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่า speed up สำหรับปัญหา
ขนาด 50x50 75
4.6 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่า speed up สำหรับปัญหา
ขนาด 100x100 76
4.7 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่า speed up สำหรับปัญหา
ขนาด 150x150 76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

4.8 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่าประสิทธิภาพ สำหรับปัญหา ขนาด 50x50 77
4.9 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่าประสิทธิภาพ สำหรับปัญหา ขนาด 100x100 78
4.10 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่าประสิทธิภาพ สำหรับปัญหา ขนาด 150x150 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แสดง MPI Operation Name 38
4.1 ผลการทดลองวัดเวลาที่ใช้ในการประมวลผลของโปรแกรม 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายสัญลักษณ์และคำย่อ

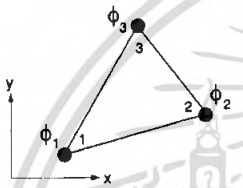
1. ∂ เป็นสัญลักษณ์แทนการอนุพันธ์บางส่วน

   เช่น $\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}$

2. ∇ เป็นสัญลักษณ์แทนความหมาย $\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}$

   เช่น $\nabla u = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}$

3. Φ เป็นสัญลักษณ์แทนตัวไม่ทราบค่าที่ต้องการหาคำตอบ

   เช่น 

   ต้องการหาค่า Φ_1, Φ_2 และ Φ_3 ตามจุดที่กำหนด

4. $\sum$ เป็นสัญลักษณ์แทนการบวกสะสม

5. $\int$ เป็นสัญลักษณ์แทนการหาผลรวมของค่าภายในพื้นที่ที่กำหนด

6. Ω เป็นสัญลักษณ์แทนพื้นที่ภายในที่ถูกปิดล้อม

7. Γ เป็นสัญลักษณ์แทนขอบเขตของพื้นที่ปิดล้อม

8. const เป็นคำย่อมาจาก constant หมายถึง ค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

เมื่อพิจารณาปัญหาทางคณิตศาสตร์ และวิทยาศาสตร์ จะพบว่าบ่อยครั้งที่ปัญหาเหล่านั้นเป็นปัญหาที่อยู่ในรูปของสมการคลื่น สมการความร้อน สมการของไหล เป็นต้น ซึ่งปัญหาเหล่านี้เป็นปัญหาที่มีความยุ่งยากซับซ้อน และมักต้องการคำตอบที่มีความละเอียดมาก ดังนั้นนักคณิตศาสตร์จึงได้ทำการแปลงปัญหาเหล่านั้นให้อยู่ในรูปของระบบสมการเชิงเส้น เพื่อทำให้การหาคำตอบมีความยุ่งยากและซับซ้อนน้อยลง ซึ่งส่วนใหญ่แล้วระบบสมการเชิงเส้นที่ได้จากการแปลงมักมีขนาดใหญ่มาก ถ้าหากจะทำการคำนวณโดยใช้เครื่องคิดเลขอาจต้องใช้เวลานาน และคำตอบอาจจะผิดพลาดได้

เนื่องจากในปัจจุบันเทคโนโลยีทางด้านคอมพิวเตอร์มีการพัฒนาอย่างรวดเร็ว จึงได้มีการนำคอมพิวเตอร์มาใช้ในการคำนวณเพื่อแก้ปัญหาทางคณิตศาสตร์มากขึ้น แต่การใช้คอมพิวเตอร์แบบลำดับขั้นที่มีหน่วยประมวลผลเดียวในการคำนวณ เพื่อหาคำตอบของระบบสมการเชิงเส้นที่มีขนาดใหญ่มาก จะต้องใช้เวลาในการคำนวณมากเช่นกัน จึงได้มีการศึกษาเทคโนโลยีคอมพิวเตอร์แบบขนานที่มีหลายหน่วยประมวลผล และนำมาใช้ในการแก้ปัญหาระบบสมการเชิงเส้นที่มีขนาดใหญ่ เพื่อลดเวลาที่ใช้ในการคำนวณให้น้อยกว่าการคำนวณด้วยคอมพิวเตอร์แบบลำดับขั้น ซึ่งในงานนี้จะนำปัญหาเกี่ยวกับการแก้ปัญหาด้วยวิธีไฟไนต์เอลิเมนต์สำหรับสมการปัวส์ซองมาประยุกต์สร้างเป็นโปรแกรมคอมพิวเตอร์แบบขนานเนื่องจากรูปแบบของสมการปัวส์ซองนั้นสามารถทำความเข้าใจได้ง่าย อีกทั้งยังสามารถนำไปแก้ปัญหาต่างๆ ทางด้านคณิตศาตร์และวิทยาศาตร์ได้อย่างหลากหลายอีกด้วย

สมการปัวส์ซองเป็นสมการทางคณิตศาสตร์ที่นำไปใช้แก้ปัญหาในเชิงวิทยาศาสตร์ ซึ่งตัวสมการปัวส์ซองจะอยู่ในรูปของสมการเชิงอนุพันธ์ย่อย (Partial Differential Equation: PDE) ดังนี้

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$$

หรือ $$\nabla^2 u = f(x, y)$$

และการแก้สมการในรูปของ PDE นี้จะใช้ระเบียบวิธีไฟไนต์เอลิเมนต์ในการแก้สมการ ส่วนในเรื่องของการแก้ปัญหาระบบสมการเชิงเส้นนั้นมีอยู่หลายวิธี ทั้งวิธีทางตรง (Direct Method) และวิธีการทำซ้ำ (Iteration Method) โดยในที่นี้ได้นำเสนอการแก้ปัญหาโดยวิธีการทำซ้ำแบบวิธีของจาโคบี (Jacobi's Method) ซึ่งเป็นวิธีที่สามารถทำความเข้าใจได้ง่ายเมื่อนำมาศึกษาควบคู่กับการทำงานแบบขนาน พร้อมทั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการออกแบบแนวคิดในการเขียนโปรแกรมแบบขนาน (Parallel Algorithm) เพื่อนำมาใช้กับคอมพิวเตอร์แบบขนานที่มีอยู่ในปัจจุบันให้เกิดประสิทธิผลสูงสุดในการคำนวณ

1.2 วัตถุประสงค์ของการทำวิจัย

1. เพื่อศึกษาวิธีการแก้ระบบสมการเชิงเส้นขนาดใหญ่ โดยวิธีการทำซ้ำแบบวิธีของจาโคบีและพัฒนาโปรแกรมเพื่อหาผลเฉลยของสมการปัวส์ซอง 2 มิติโดยระเบียบวิธีไฟไนต์เอลิเมนต์ในรูปแบบของการประมวลผลแบบขนาน (Parallel Processing)
2. เพื่อศึกษาแนวคิดและกระบวนการของเทคโนโลยีคอมพิวเตอร์ทางด้านการประมวลผลแบบขนาน (Parallel Processing) บนระบบคลัสเตอร์ของพีซี (Cluster of PCs)
3. เพื่อสามารถนำความรู้ด้านการประมวลผลแบบขนาน (Parallel Processing) ไปประยุกต์ใช้ในการแก้ปัญหาระบบสมการเชิงเส้นขนาดใหญ่ในรูปแบบอื่นๆนอกเหนือจากสมการปัวส์ซอง 2 มิติได้
4. ทำการทดลอง และศึกษาถึงประสิทธิภาพการทำงาน วัดผล และเปรียบเทียบระหว่างการประมวลผลบนเครื่องเดียว กับการประมวลผลแบบขนาน(Parallel Processing) เพื่อหาผลเฉลยของสมการปัวส์ซอง 2 มิติโดยระเบียบวิธีไฟไนต์เอลิเมนต์

1.3 ขอบเขตของปัญหา

สภาพปัญหาที่ใช้ในการศึกษาเพื่อพัฒนาโปรแกรม เป็นการศึกษาการถ่ายเทความร้อนบนเนื้อวัตถุแผ่นเรียบที่มีลักษณะเป็นสี่เหลี่ยมผืนผ้า โดยวัตถุแผ่นเรียบนี้จะกำหนดพิกัดอยู่ในจตุภาคที่ 1 เท่านั้น เพราะปัญหาทางกายภาพเมื่อกล่าวถึงพิกัด จะต้องเป็นพิกัดที่มีอยู่จริง ซึ่งในความหมายทางวิทยาศาสตร์พิกัดที่มีอยู่จริงจะต้องมีค่าเป็นบวก ดังเช่นระยะทางหรือระยะห่าง จะไม่บอกค่าออกมาเป็นลบ เพราะไม่สามารถสื่อความหมายออกมาได้ โดยรูปปัญหาจะกำหนดให้ขอบสัมผัสไปกับแนวแกน x และแนวแกน y ซึ่งจะให้มุมเริ่มต้นอยู่ที่จุดกำเนิด และสมการปัวส์ซองที่ใช้ในการคำนวณจะมีเพียง 4 รูปแบบเท่านั้น

โดยรูปแบบของสมการปัวส์ซอง เป็นดังนี้

$$\nabla^2 \phi = f(x, y)$$

โดยที่

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งขึ้นอยู่กับฟังก์ชันที่กำหนดให้ดังต่อไปนี้

1. f(x,y) = ค่าคงที่
2. f(x,y) = x
3. f(x,y) = y
4. f(x,y) = xy

ภายใต้เงื่อนไขขอบเขต

Dirichlet เป็นการกำหนดค่าที่ขอบเป็นค่าคงที่ โดยมีลักษณะเหมือนว่าเมื่อมีการถ่ายเทความร้อนมาที่ขอบหรือจากขอบเข้าไป จะต้องมีค่าขอบเท่าที่กำหนดให้เสมอ

ดังนั้น วัตถุที่จะทำการคำนวณจะเป็นสี่เหลี่ยมผืนผ้าซึ่งอยู่ในจตุภาคที่ 1 เท่านั้น และสมการที่ใช้ในการคำนวณเป็นสมการปัวส์ซอง 4 แบบตามที่กล่าวไว้ข้างต้น

จากนั้นศึกษาวิธีการแก้ระบบสมการเชิงเส้นขนาดใหญ่โดยวิธีการทำซ้ำแบบขนาน โดยเลือกศึกษาวิธีของจาโคบี ซึ่งวิธีการศึกษาการทำงานในแบบขนานนั้น ต้องทำการแบ่งงานออกเป็นงานย่อยๆ เพื่อศึกษากระบวนการทำงานแบบขนาน หลังจากนั้นทำการออกแบบแนวคิด และเขียนโปรแกรมการแก้ระบบสมการโดยวิธีของจาโคบี ทั้งแบบลำดับขั้น และแบบขนาน แล้วนำโปรแกรมที่ได้มาทดลองใช้งาน และเปรียบเทียบประสิทธิภาพโดยการหาค่า Speed up ทั้งทางทฤษฎี และทางการทดลอง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ทราบถึงวิธีการแก้ระบบสมการเชิงเส้นขนาดใหญ่โดยวิธีการทำซ้ำแบบขนาน
2. ได้ทราบเกี่ยวกับแนวคิดและหลักการของเทคโนโลยีด้านคอมพิวเตอร์ในเรื่องของการประมวลผลแบบขนาน (Parallel Processing)
3. สามารถเขียนโปรแกรมคอมพิวเตอร์ในรูปแบบของการประมวลผลแบบขนานบนระบบคลัสเตอร์ของพีซีได้
4. สามารถนำความรู้ด้านการประมวลผลแบบขนาน (Parallel Processing) ไปประยุกต์ใช้ในการแก้ปัญหาระบบสมการเชิงเส้นขนาดใหญ่ในรูปแบบอื่นๆนอกเหนือจากสมการปัวส์ซอง 2มิติได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขั้นตอนในการดำเนินงาน

1. ศึกษาวิธีการแก้ระบบสมการเชิงเส้นขนาดใหญ่โดยวิธีการทำซ้ำแบบวิธีของจาโคบี
2. ศึกษาแนวคิดและกระบวนการของเทคโนโลยีคอมพิวเตอร์ทางด้านการประมวลผลแบบขนาน (Parallel Processing)
3. ศึกษาหลักและวิธีการเขียนโปรแกรมสำหรับใช้ในการประมวลผลแบบขนาน (Parallel Processing)
4. ศึกษาแนวคิดและกระบวนการของเทคโนโลยีคอมพิวเตอร์ทางด้านการประมวลผลแบบขนาน (Parallel processing) บนระบบคลัสเตอร์ของพีซี (Cluster of PCs)
5. ศึกษาการทำงานของ MPI (Message Passing Interface)
6. เขียนโปรแกรม
7. ทดสอบคุณภาพของโปรแกรมและทำการแก้ไขหากพบข้อผิดพลาด
8. ทำการทดลองเพื่อเปรียบเทียบความแตกต่างระหว่างประมวลผลทั่วไปกับระบบประมวลผลแบบขนาน จากนั้นทำการสรุปผลการทดลองและอภิปรายผลการทดลองที่ได้
9. จัดทำเอกสาร

1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

1. ระบบปฏิบัติการ Windows XP
2. ระบบปฏิบัติการ Unix
3. โปรแกรม Xshell
4. โปรแกรม Edit Plus 3
5. โปรแกรม Microsoft Visual C++
6. MPI (Message passing interface)
7. Computer PC ที่สามารถใช้งานบนระบบคลัสเตอร์ (Cluster) ได้ โดยมีสเปคเครื่อง ดังนี้

เครื่อง Front end

หน่วยประมวลผลกลาง(CPU) 2.8 GHz

แรม(RAM) 2 GB

ฮาร์ดดิกส์(Hard disk) 120 GB

เครื่อง Compute node จำนวน 9 เครื่อง

หน่วยประมวลผลกลาง(CPU) 2.8 GHz

แรม(RAM) 2 GB

ฮาร์ดิกส์(Hard disk) 80 GB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ทฤษฎีและหลักการของระเบียบวิธีไฟไนต์เอลิเมนต์

ความสามารถในการแก้ปัญหาทางด้านวิศวกรรมศาสตร์และวิทยาศาสตร์ มีส่วนอย่างมากในการสร้างเสริมปรับปรุงความเป็นอยู่ของมนุษย์ให้ดียิ่งขึ้น ปรากฏการณ์ส่วนใหญ่ที่เกิดขึ้นรอบตัวสามารถอธิบายได้โดยกฎเกณฑ์ทางฟิสิกส์และทำการประดิษฐ์ขึ้นมาในลักษณะของสมการต่างๆได้ ซึ่งจะอยู่ในรูปแบบของสมการเชิงอนุพันธ์ (differential equation) หรือในรูปแบบของสมการอินทิกรัล (integral equation) ตัวอย่างเช่น การคำนวณหาการกระจายอุณหภูมิบนเครื่องยนต์ของรถยนต์อาจเริ่มมาจากสมการเชิงอนุพันธ์ที่อธิบายสถานะของความสมดุลในการถ่ายเทความร้อน เป็นต้น

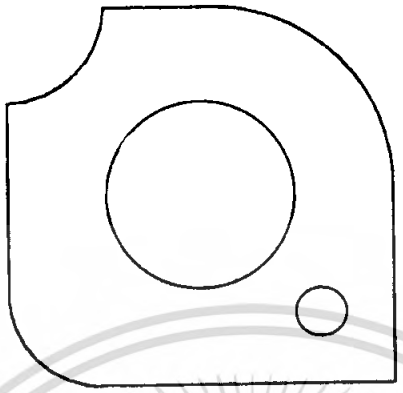
สมการเชิงอนุพันธ์ที่สอดคล้องกับปัญหาต่าง ๆ นั้น ปกติสามารถสร้างขึ้นมาได้โดยไม่ยากเลย หากแต่ว่า ผลเฉลยแม่นตรง (exact solution) ที่ต้องการและจำเป็นต้องทำการหาโดยวิธีวิเคราะห์ (analysis method) นั้นทำได้ยากมากหรืออาจจะหาไม่ได้เลยก็ได้ เหตุผลดังกล่าวก่อให้เกิดวิธีการหาผลเฉลยโดยประมาณ (approximate solution) ขึ้น วิธีการหาผลเฉลยโดยประมาณนั้นมีหลาย ๆ วิธี ส่วนวิธีที่นิยมกันอย่างกว้างขวาง คือ วิธีการผลต่างสืบเนื่อง (finite different method)

หลักการที่สำคัญของวิธีผลต่างสืบเนื่องก็คือการหาผลเฉลยโดยประมาณโดยเริ่มจากการเขียนสมการเชิงอนุพันธ์ให้อยู่ในรูปแบบของระบบสมการผลต่างสืบเนื่อง ข้อดีของวิธีการผลต่างสืบเนื่องนี้ก็คือ เป็นวิธีที่ง่ายแก่การศึกษาและทำความเข้าใจ รวมไปถึงความสะดวกในการเขียนโปรแกรมคอมพิวเตอร์เพื่อใช้ในการคำนวณหาผลเฉลยของปัญหานั้นๆ และที่สำคัญที่สุด ได้กลายเป็นสิ่งที่จำเป็นเพื่อการพัฒนาปรับปรุงคุณภาพสิ่งของนั้นๆ ให้ดียิ่งขึ้น ส่วนข้อเสียของวิธีการหาผลต่างสืบเนื่องก็มีหลายประการ เช่น ความไม่สะดวกในการกำหนดเงื่อนไขขอบเขต

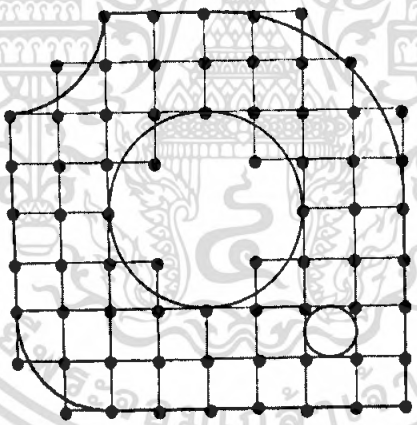
สมมติว่ามีแผ่นอะลูมิเนียมที่ใช้เสริมความแข็งแรงของโครงสร้างภายในปีกเครื่องบิน ซึ่งมีลักษณะดังภาพที่ 2.1 การวิเคราะห์หาการกระจายของความเค้น บนแผ่นอะลูมิเนียมนี้ภายใต้แรงกระทำที่กำหนดให้โดยการใช้วิธีผลต่างสืบเนื่อง (Finite different method) จะเริ่มจากการแบ่งแผ่นอะลูมิเนียมนี้ออกเป็นช่องตารางสี่เหลี่ยม ดังภาพที่ 2.2 โดยที่ตารางสี่เหลี่ยมเหล่านี้ต่อกันที่จุดต่อ(grid points) ตามหัวมุมของสี่เหลี่ยมต่าง ๆ และขนาดของปัญหาหรือจำนวนตัวไม่รู้ค่า จะขึ้นอยู่กับจำนวนของจุดต่อนี้เอง จากภาพที่ 2.2 จะเห็นได้ว่าตารางสี่เหลี่ยมที่ใช้วิธีผลต่างสืบเนื่องไม่สามารถจำลองรูปร่างลักษณะดั้งเดิมที่แท้จริงของแผ่นอะลูมิเนียมดังกล่าวได้โดยเที่ยงตรง หากใช้ตารางสี่เหลี่ยมให้มีขนาดเล็กลงซึ่งหมายถึงจะต้องเพิ่มจำนวนตารางสี่เหลี่ยมให้มากขึ้นก็จะสามารถจำลองรูปร่างลักษณะดั้งเดิมที่แท้จริงได้ใกล้เคียงมากยิ่งขึ้นแต่ในขณะเดียวกันจำนวนจุดต่อที่เพิ่มมากขึ้นทำให้จำนวนสมการผลต่างสืบเนื่องมากขึ้นด้วย และกระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แก้ปัญหาจำเป็นต้องการหน่วยความจำบนเครื่องคอมพิวเตอร์เพิ่มขึ้นรวมถึงเวลาที่ใช้ในการคำนวณจะสูงมากขึ้นตามไปด้วย



ภาพที่ 2.1 ลักษณะของแผ่นอะลูมิเนียมที่ใช้เสริมความแข็งแรงของโครงสร้างภายในปีกเครื่องบิน

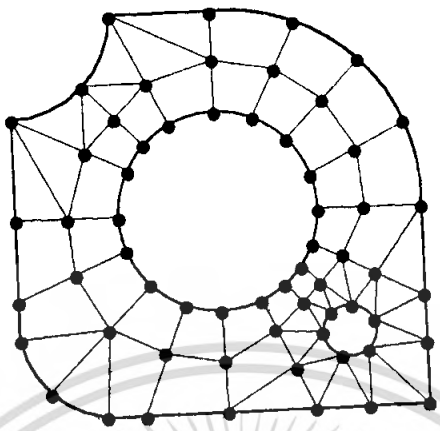


ภาพที่ 2.2 ตารางสี่เหลี่ยมที่ใช้วิธีผลต่างสืบเนื่อง

จากรูปข้างต้น แสดงให้เห็นถึงความยากในการนำวิธีการนี้ไปประยุกต์ใช้กับปัญหาที่มีรูปร่างลักษณะซับซ้อนในปัจจุบัน สาเหตุของความยากดังกล่าวมีส่วนก่อให้เกิดวิธีการหาผลเฉลยโดยประมาณวิธีใหม่ที่เรียกว่า วิธีไฟไนต์เอลิเมนต์ (Finite Element Method: FEM) ซึ่งวิธีการนี้สามารถนำมาใช้กับปัญหาที่มีรูปร่างลักษณะซับซ้อนเช่นใดก็ได้ โดยสามารถจำลองรูปร่างลักษณะดั้งเดิมที่แท้จริง ได้ใกล้เคียงเที่ยงตรงกว่าหลักการของวิธีการผลต่างสืบเนื่อง กล่าวคือ เริ่มจากการแบ่งรูปร่างของปัญหาออกเป็นเนื้อที่หลาย ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชิ้นที่เรียกว่า “เอลิเมนต์” โดยเอลิเมนต์ต่าง ๆ อาจจะอยู่ในรูปลักษณะของสามเหลี่ยมและสี่เหลี่ยมด้านไม่เท่าก็ได้ ดังภาพที่ 2.3



ภาพที่ 2.3 การวิเคราะห์หาผลเฉลยบนแผ่นอะลูมิเนียมด้วยการใช้วิธีการไฟไนต์เอลิเมนต์

ผลประโยชน์แรกที่เห็นได้ชัดเมื่อเปรียบเทียบภาพที่ 2.2 โดยวิธีผลต่างสืบเนื่อง (finite different method) กับภาพที่ 2.3 โดยวิธีการไฟไนต์เอลิเมนต์ (finite element method) ซึ่งประกอบด้วยเอลิเมนต์ขนาดต่าง ๆ กัน ที่สามารถจำลองรูปร่างลักษณะดั้งเดิมของแผ่นอะลูมิเนียมได้เป็นอย่างดี ซึ่งหมายถึงปัญหานี้จะถูกแก้เพื่อหาผลเฉลยโดยประมาณจากแบบจำลองทางคณิตศาสตร์ ที่มีรูปร่างลักษณะใกล้เคียงกับของจริงดั้งเดิมมากที่สุด ดังนั้นค่าผลเฉลยโดยประมาณที่จะคำนวณได้จะมีความแม่นยำมากขึ้นตามไปด้วย

2.1.1 วิธีการไฟไนต์เอลิเมนต์

ในการแก้ปัญหาใดปัญหาหนึ่ง ปัญหานั้นจะประกอบด้วยสมการเชิงอนุพันธ์และเงื่อนไขขอบเขตที่กำหนดมาให้ ค่าผลเฉลยแม่นตรง (exact solution) ดังกล่าวจะประกอบด้วยค่าของตัวแปรต่าง ๆ กันตามตำแหน่งต่าง ๆ บนรูปร่างลักษณะของปัญหานั้น หรือกล่าวอีกนัยหนึ่งก็คือ ค่าผลเฉลยแม่นตรงประกอบด้วยค่าต่าง ๆ ทั้งหมดนับเป็นจำนวนอนันต์ค่า แทนที่จะทำการหาค่าแม่นตรงที่ประกอบด้วยค่าต่าง ๆ จำนวนมากมายเช่นนี้ ซึ่งสำหรับปัญหาในทางปฏิบัติจะทำไม่ได้ หลักการก็คือ ทำการเปลี่ยนค่าทั้งหมดที่มีจำนวนอนันต์ค่านั้นมาเป็นค่าโดยประมาณที่มีจำนวนนับได้ (finite) ด้วยการแทนรูปร่างลักษณะของปัญหาด้วยเอลิเมนต์ (element) ซึ่งมีขนาดต่าง ๆ กัน

โดยหลักการของวิธีไฟไนต์เอลิเมนต์จะเริ่มต้นจากการพิจารณาเอลิเมนต์ทีละเอลิเมนต์ โดยทำการสร้างสมการสำหรับแต่ละเอลิเมนต์ที่ตั้งอยู่บนรากฐานที่ว่า สมการที่สร้างขึ้นมานั้นจำเป็นต้องสอดคล้องกับสมการเชิงอนุพันธ์ของปัญหาที่ทำอยู่ จากนั้นนำสมการของแต่ละเอลิเมนต์ที่สร้างขึ้นมาได้ มาประกอบกันเข้าก่อให้เกิดระบบสมการชุดใหญ่ ซึ่งในความหมายทางกายภาพ ก็คล้ายกับการนำทุกเอลิเมนต์มาประกอบรวมเข้าด้วยกันก่อให้เกิดรูปร่างลักษณะทั้งหมดของปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่แท้จริง จากนั้นจึงทำการกำหนดเงื่อนไขขอบเขตที่ให้มาลงในระบบสมการชุดใหญ่นี้แล้วจึงทำการแก้สมการดังกล่าว ซึ่งจะก่อให้เกิดผลเฉลยโดยประมาณที่ต้องการ ณ ตำแหน่งต่าง ๆ ของปัญหานั้น เป็นคำอธิบายในลักษณะกว้าง ๆ เพื่อที่จะให้เห็นภาพรวมของวิธีไฟไนต์เอลิเมนต์ ได้อธิบายในขั้นต้นว่าทำไมถึงต้องใช้วิธีไฟไนต์เอลิเมนต์ ซึ่งจะกล่าวถึงข้อแตกต่างของวิธีผลต่างสืบเนื่อง(finite different method) และวิธีไฟไนต์เอลิเมนต์ (finite element method)

2.1.2 ข้อแตกต่างระหว่างวิธีการผลต่างสืบเนื่องและวิธีการไฟไนต์เอลิเมนต์

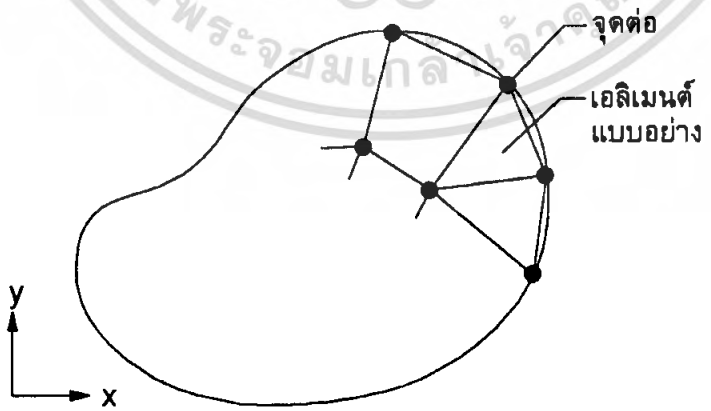
วิธีการผลต่างสืบเนื่อง เป็นการใช้ระเบียบวิธีเชิงตัวเลข (numerical method) เพื่อใช้ในการหาผลเฉลยโดยประมาณของปัญหาที่กำหนดมาให้ หลักการของวิธีนี้คือ การแทนตัวอนุพันธ์ที่ปรากฏอยู่ในสมการเชิงอนุพันธ์ ด้วยสมการทางพีชคณิตโดยประมาณ ซึ่งอยู่ในรูปของตัวแปร ณ จุดต่อต่าง ๆ ในขอบเขตของรูปร่างของลักษณะปัญหานั้น

ส่วนวิธีการไฟไนต์เอลิเมนต์ เป็นการใช้ระเบียบวิธีเชิงตัวเลขเพื่อหาผลลัพธ์โดยประมาณของปัญหาที่กำหนดมาให้เช่นกัน โดยแบ่งรูปร่างลักษณะของปัญหาที่จะใช้ในการคำนวณออกเป็นชิ้นส่วนย่อย ๆ ที่เรียกว่า เอลิเมนต์ เอลิเมนต์เหล่านี้เชื่อมต่อกันที่จุดต่อ (nodes) ซึ่งเป็นตำแหน่งที่จะคำนวณหาค่าตัวแปรตาม (dependent variables) ที่ต้องการ

2.1.3 ขั้นตอนทั่วไปของระเบียบวิธีไฟไนต์เอลิเมนต์

วิธีการไฟไนต์เอลิเมนต์ประกอบด้วยขั้นตอนใหญ่ทั้งหมด 6 ขั้นตอน ดังต่อไปนี้

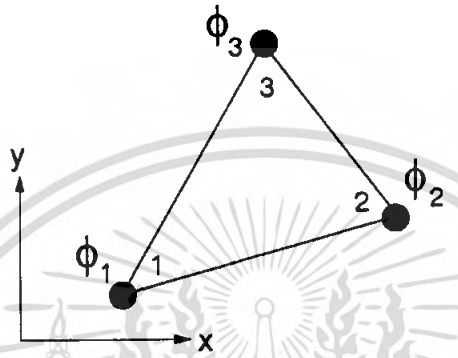
ขั้นตอนที่ 1 การแบ่งขอบเขตรูปร่างลักษณะของปัญหาที่ต้องการที่จะหาผลลัพธ์นั้นออกเป็นเอลิเมนต์ย่อย ๆ **ดังภาพที่ 2.4** ขอบเขตดังกล่าวอาจเป็นขอบเขตของปัญหาชนิดต่างๆกัน



ภาพที่ 2.4 แสดงถึงการแบ่งรูปร่างลักษณะของปัญหาออกเป็นเอลิเมนต์แบบต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ขั้นตอนที่ 2</u> การเลือกประมาณฟังก์ชันภายในเอลิเมนต์ (element interpolation function) เช่น เอลิเมนต์สามเหลี่ยม เอลิเมนต์ ดังกล่าวประกอบด้วย 3 จุดต่อที่มีหมายเลข 1, 2, 3 ดังแสดงในภาพที่ 2.5 โดยที่จุดต่อนี้นั้นเป็นตำแหน่งของตัวที่ไม่รู้ค่า (nodal unknowns) ซึ่งก็คือ ϕ_1, ϕ_2 และ ϕ_3 ตัวที่ไม่รู้ค่าที่จุดต่อเหล่านี้อาจเป็นค่าการยืดหรือหดตัว หากเราทำปัญหาความยืดหยุ่นในของแข็ง หรืออาจเป็นค่าอุณหภูมิ หากเราทำปัญหาเกี่ยวกับ ปัญหาการถ่ายเทความร้อน หรือไม่ก็อาจเป็นปัญหาการถ่ายเทของของเหลว เป็นต้น



ภาพที่ 2.5 เอลิเมนต์สามเหลี่ยมแบบอย่าง ประกอบด้วยสามจุดต่อ โดยมีตัวไม่รู้ค่าอยู่ ณ ตำแหน่งที่จุดต่อ

ลักษณะการกระจายตัวไม่รู้ค่าบนเอลิเมนต์นี้สามารถเขียนให้อยู่ในรูปของฟังก์ชันการประมาณภายในและตัวไม่รู้ค่าที่จุดต่อดังนี้

$$\phi(x,y) = N_1(x,y)\,\phi_1 + N_2(x,y)\,\phi_2 + N_3(x,y)\,\phi_3 \qquad (2.1)$$

โดย Ni(x,y) ,i = 1,2,2 คือฟังก์ชันประมาณภายในเอลิเมนต์
จากสมการที่ (2.1) สามารถเขียนให้อยู่ในรูปเมทริกซ์ได้ดังนี้

$$\phi(x,y) = \begin{bmatrix} N_1 & N_2 & N_3 \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}$$

$$= \underset{(1\times 3)}{[N]} \underset{(3\times 1)}{\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}} \qquad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย $[N]$ คือ เมทริกซ์ของฟังก์ชันการประมาณค่าภายในเอลิเมนต์

$\{\phi\}$ คือ เวกเตอร์เมทริกซ์ที่ประกอบด้วยตัวที่ไม่รู้ค่าที่จุดต่อของเอลิเมนต์นั้น

ซึ่งสัญลักษณ์ [ ] หมายถึงเมทริกซ์ในแนวนอน และ { } หมายถึงเมทริกซ์ในแนวตั้ง

ขั้นตอนที่ 3 การสร้างสมการของเอลิเมนต์ ดังตัวอย่างเช่น สมการเอลิเมนต์สามเหลี่ยมแบบอย่าง จะอยู่ในรูปแบบดังนี้

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}_e \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}_e = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix}_e \qquad (2.3)$$

ซึ่งเขียนย่อได้เป็น

$$[k]_e \{\phi\}_e = \{F\}_e \qquad (2.4)$$

ขั้นตอนที่ 3 นี้ถือว่าเป็นหัวใจสำคัญของวิธีไฟไนต์เอลิเมนต์

ขั้นตอนที่ 4 การนำสมการของแต่ละเอลิเมนต์ที่ได้มาประกอบกัน ก่อให้เกิดระบบสมการพร้อมกันขึ้น ในรูปแบบดังนี้

$$\sum(element equation) \Rightarrow [k]_{sys} \{\phi\}_{sys} = \{F\}_{sys} \qquad (2.5)$$

ขั้นตอนที่ 5 ทำการประยุกต์เงื่อนไขขอบเขต ลงในสมการ (2.5) แล้วจึงทำการแก้สมการนั้นเพื่อหาค่า $\{\phi\}_{sys}$ อันประกอบด้วยตัวไม่รู้ค่าที่จุดต่อ ซึ่งอาจจะเป็นค่าของการเคลื่อนตัว ณ ตำแหน่งต่าง ๆ ของโครงสร้าง หรือค่าของอุณหภูมิที่จุดต่าง ๆ หากเป็นปัญหาที่เกี่ยวกับการถ่ายเทความร้อน เป็นต้น

ขั้นตอนที่ 6 เมื่อคำนวณค่าต่าง ๆ ที่จุดต่อออกมาได้แล้วก็สามารถทำการหาค่าอื่นๆที่ต้องการทราบต่อไปได้ เช่น เมื่อรู้อุณหภูมิที่จุดต่าง ๆ ก็สามารถคำนวณหาปริมาณการถ่ายเทความร้อนได้ หรือเมื่อรู้ความเร็วของของไหลก็สามารถนำไปคำนวณหาปริมาณอัตราการไหลของของไหลทั้งหมดได้ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากขั้นตอนทั้ง 6 ขั้นตอนนี้จะเห็นได้ว่าวิธีการไฟไนต์เอลิเมนต์เป็นวิธีการที่มีระเบียบแบบแผนขั้นตอน โดยมีหัวใจสำคัญคือการสร้างสมการของเอลิเมนต์ จากโครงงานเรื่องนี้ได้กล่าวถึงการแก้ปัญหาในรูปแบบสองมิติซึ่งอยู่ในรูปของสมการปัวส์ซอง คือ

$$u(x,y) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

โดยต้องการที่จะทราบค่าของ u(x,y) จึงจะขอกล่าวถึงวิธีไฟไนต์เอลิเมนต์ที่ใช้ในการแก้ปัญหาสองมิติ

2.2 การสร้างสมการโดยวิธีการแปรผัน

2.2.1 วิธีการแปรผัน (variational method) : ในการแก้ปัญหาภาวะต่อเนื่อง ไม่ว่าจะเป็นทางด้านของแข็ง ของไหล ฯลฯ ปกติเราสามารถทำได้ 2 วิธี คือ

ก. โดยการใช้สมการเชิงอนุพันธ์ (differential equations) ร่วมกับการใช้เงื่อนไขขอบเขต ที่เหมาะสม

ข. โดยการใช้สูตรการแปรผัน (variational formulation)

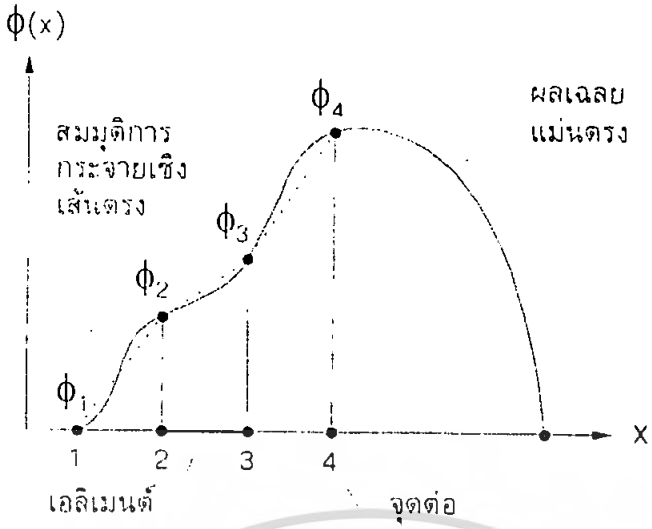
หลักการสำคัญในการใช้สูตรการแปรผันก็คือ เราจำเป็นต้องทำการหาหรือสร้างฟังก์ชัน ซึ่งเมื่อเราทำการหาค่าต่ำสุด (minimization) ของฟังก์ชันนั้นแล้ว จะเป็นผลให้เกิดสมการเชิงอนุพันธ์และเงื่อนไขขอบเขตที่สอดคล้องกับปัญหาที่เราทำการแก้อยู่นั้น ดังตัวอย่างเช่น สมการอนุพันธ์ ต่อไปนี้

$$\frac{d^2 u}{dx^2} = -f(x) \qquad (2.6)$$

โดย u = u(x) และเงื่อนไขขอบเขตที่เหมาะสมนั้นจะมีความสมมูลกัน

การสร้างสมการไฟไนต์เอลิเมนต์ในวิธีการแปรผัน จะเริ่มจากการสมมุติผลเฉลยโดยประมาณขึ้นมาซึ่งครอบคลุมทั้งขอบเขตปัญหาที่กำหนดมาให้ โดยจะมีการแบ่งขอบเขตออกเป็นเอลิเมนต์ย่อย ๆ ดังแสดงใน**ภาพที่ 2.6** ซึ่งต่อกันที่จุดต่อต่าง ๆ โดยค่าที่จุดต่อ Φ_i, i = 1, 2, 3, ... นั้นไม่รู้ค่าและต้องการหาการกระจายของผลเฉลยระหว่างจุดต่ออาจสมมุติให้อยู่ในรูปแบบที่ง่าย เช่น ในลักษณะเชิงเส้นตรงดังแสดงใน**ภาพที่ 2.6** เป็นต้น

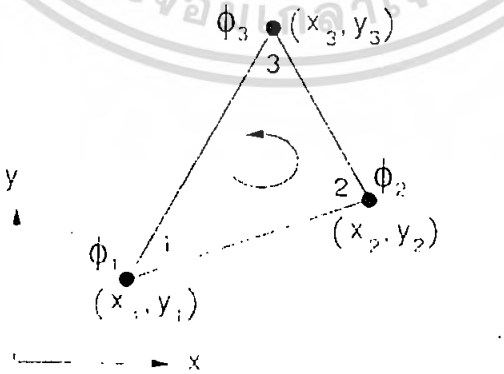
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.6 การแบ่งขอบเขตออกเป็นไฟไนต์เอลิเมนต์ย่อย

2.2.2 ฟังก์ชันการประมาณภายในเอลิเมนต์สองมิติ

ปัญหาส่วนใหญ่ในสองมิติจะใช้เอลิเมนต์ที่มีลักษณะเป็นรูปสามเหลี่ยม ทั้งนี้ก็เพราะว่า ลักษณะขอบเขตของปัญหาแบบสองมิติทั่ว ๆ ไปสามารถแบ่งออกเป็นเอลิเมนต์สามเหลี่ยมย่อย ๆ ได้โดยง่าย อีกทั้งสมการเอลิเมนต์รูปสามเหลี่ยมสามารถสร้างขึ้นได้โดยง่ายและนำไปใช้ในการประดิษฐ์โปรแกรมได้โดยสะดวกด้วย เอลิเมนต์สามเหลี่ยมแบบที่ง่ายที่สุดประกอบด้วยสามจุดต่อหมายเลข 1, 2, 3 ซึ่งวนอยู่ในทิศทวนเข็มนาฬิกา ดังแสดงใน**ภาพที่ 2.7** โดยที่จุดต่อซึ่งอยู่ตำแหน่ง (x_i,y_i), $i = 1, 2, 3$ เป็นตำแหน่งของตัวไม่รู้ค่า ϕ_i การสร้างฟังก์ชันประมาณภายในเอลิเมนต์สามเหลี่ยมก็สามารถทำได้เช่นเดียวกับเอลิเมนต์ในหนึ่งมิติ ซึ่งเอลิเมนต์สามเหลี่ยมเราจะเริ่มโดยสมมุติลักษณะการกระจายของผลลัพธ์โดยประมาณแบบแผ่นเรียบบนเอลิเมนต์ นั่นคือ



ภาพที่ 2.7 เอลิเมนต์สามเหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\phi\ (x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y \qquad (2.7)$$

โดย α_i , i = 1, 2, 3 เป็นค่าคงตัว ซึ่งหาได้จากเงื่อนไขที่จุดต่อทั้งสาม ดังนี้

จุดต่อที่ 1 : $\phi(x_1, y_1) = \phi_1 = \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1$

จุดต่อที่ 2 : $\phi(x_2, y_2) = \phi_2 = \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2$

จุดต่อที่ 3 : $\phi(x_3, y_3) = \phi_3 = \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3$

จากทั้ง 3 สมการนี้ เราสามารถหาค่า α_i , i = 1, 2, 3 ให้อยู่ในรูปของค่าที่จุดต่อ ϕ_i และตำแหน่งของจุดต่อ (x_i, y_i) ซึ่งหลังจากแทนกลับลงไปในสมการ (2.7) แล้วทำการจัดพจน์ต่าง ๆ เราจะได้ลักษณะของการกระจายของผลเฉลยสำหรับเอลิเมนต์ที่อยู่ในรูป

$$\phi(x,y) = \begin{bmatrix} N_1(x,y) & N_2(x,y) & N_3(x,y) \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix} = \underset{(1\times 3)}{[N]} \underset{(3\times 1)}{\{\phi\}} \qquad (2.8)$$

โดย $[N]$ เรียกว่า เมทริกซ์ของการประมาณภายในเอลิเมนต์ (element interpolation matrix)

$\{\phi\}$ คือ เวกเตอร์ของตัวไม่รู้ค่าที่จุดต่อ (vector of nodal unknowns)

และในที่นี้

$$N_i(x,y) = \frac{1}{2A}(a_i + b_i x + c_i y) \quad i = 1, 2, 3 \qquad (2.9)$$

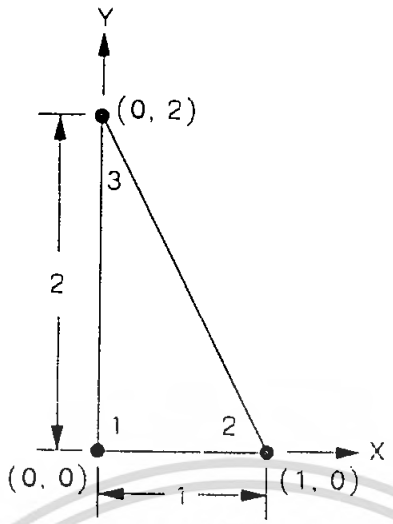
โดย A = พื้นที่ของเอลิเมนต์สามเหลี่ยม

$$A = \frac{1}{2}\left[x_2(y_3 - y_1) + x_1(y_2 - y_3) + x_3(y_1 - y_2)\right] \qquad (2.10)$$

$$\begin{aligned} a_1 &= x_2 y_3 - x_3 y_2 & b_1 &= y_2 - y_3 & c_1 &= x_3 - x_2 \\ a_2 &= x_3 y_1 - x_1 y_3 & b_2 &= y_3 - y_1 & c_2 &= x_1 - x_3 \\ a_3 &= x_1 y_2 - x_2 y_1 & b_3 &= y_1 - y_2 & c_3 &= x_2 - x_1 \end{aligned} \qquad (2.11)$$

เพื่อให้เกิดความเข้าใจในลักษณะของฟังก์ชันการประมาณภายในเอลิเมนต์ เราจะพิจารณาเอลิเมนต์ดังแสดงในภาพที่ 2.8 ซึ่งจะมีขนาดและตำแหน่งของจุดต่อต่าง ๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.8 ตัวอย่างเอลิเมนต์สามเหลี่ยม

$x_1 = 0$ $y_1 = 0$

$x_2 = 1$ $y_2 = 0$

$x_3 = 0$ $y_3 = 2$

จากสมการ(2.10) พื้นที่ของเอลิเมนต์คือ

$$A = \frac{1}{2}\left[1(2-0)+0(0-2)+0(0-0)\right]$$

และจากสมการ(2.11) สัมประสิทธิ์ a_i, b_i, c_i, $i = 1, 2, 3$ คือ

$a_1 = (1)(2)-(0)(0) = 2$ $b_1 = 0-2 = 2$ $c_1 = 0-1 = -1$

$a_2 = (0)(0)-(0)(2) = 0$ $b_2 = 2-0 = 2$ $c_2 = 0-0 = 0$

$a_3 = (0)(0)-(1)(0) = 0$ $b_3 = 0-0 = 0$ $c_3 = 1-0 = 1$

ดังนั้นฟังก์ชันการประมาณการภายในเอลิเมนต์นี้คือ

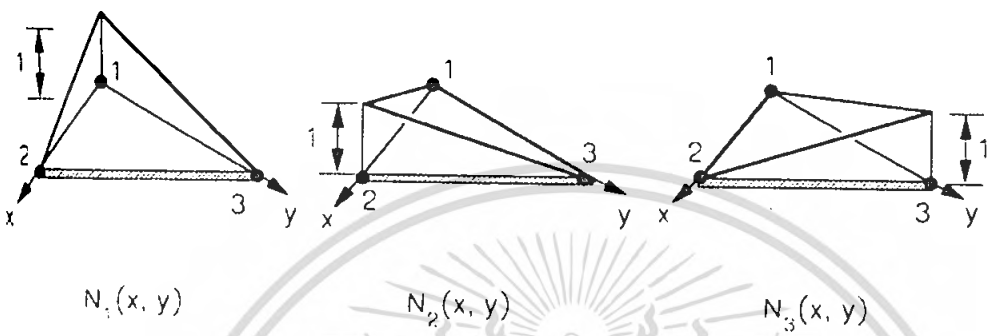
$$N_1 = \frac{1}{2(1)}(2 + (-2)x + (-1)y) = 1 - x - \frac{1}{2}y$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$N_2 = \frac{1}{2(1)}(0 + (2)x + (0)y) = x$$

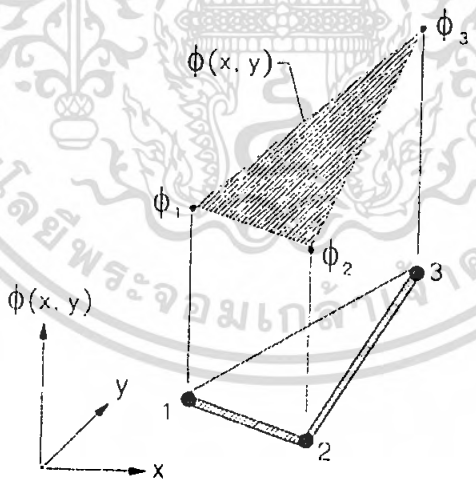
$$N_3 = \frac{1}{2(1)}(0 + (0)x + (1)y) = \frac{1}{2}y$$

ซึ่งต่างมีการกระจายในลักษณะแผ่นเรียบดังแสดงในภาพที่ 2.9



ภาพที่ 2.9 ลักษณะการกระจายของฟังก์ชันการประมาณภายในเอลิเมนต์

ฟังก์ชันการประมาณภายใน N_i เหล่านี้ เมื่อนำไปคูณกับค่าที่จุดต่อ Φ_i ดังแสดงในสมการ (2.8) จะก่อให้เกิดลักษณะของการกระจายของผลเฉลยโดยประมาณภายในเอลิเมนต์ดังแสดงใน **ภาพที่ 2.10**



ภาพที่ 2.10 ลักษณะการกระจายของผลเฉลยโดยประมาณภายในเอลิเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การสร้างสมการไฟไนต์เอลิเมนต์แบบสองมิติโดยวิธีการแปรผัน

เพื่อแสดงลำดับขั้นตอนการสร้างสมการไฟไนต์เอลิเมนต์แบบสองมิติโดยวิธีการแปรผัน เราจะยกตัวอย่างง่าย ๆ เช่น ปัญหาที่เป็นไปตามสมการอนุพันธ์ย่อยเชิงเส้นซึ่งอยู่ในรูปของสมการปัวส์ซองดังนี้

$$\nabla^2\phi = \int_A \frac{1}{2}\left(\frac{\partial^2\phi}{\partial x^2}\right)+\left(\frac{\partial^2\phi}{\partial y^2}\right) = -f(x,y) \qquad (2.12)$$

สมการเชิงอนุพันธ์ในรูปแบบนี้พบกันมากในงานออกแบบ ตัวอย่างเช่น สมการการกระจายของอุณหภูมิในสถานะอยู่ตัว (steady-state) บนแผ่นโลหะซึ่งได้รับความร้อนที่ผิว เป็นต้น หากแผ่นโลหะมีรูปร่างหรืออยู่ภายใต้เงื่อนไขขอบเขตที่ซับซ้อนจะเป็นการยากที่จะหาผลเฉลยแม่นตรงได้ แต่ปัญหาเช่นนี้ สามารถทำได้โดยง่ายหากเรารู้สมการไฟไนต์เอลิเมนต์ ซึ่งสมการไฟไนต์เอลิเมนต์สามารถสร้างได้โดยการหาค่าต่ำสุดของฟังก์ชันการแปรผันที่สอดคล้องกับสมการ(2.12) ก็คือ

$$J(\phi) = \int_A \left[\frac{1}{2}\left(\frac{\partial\phi}{\partial x}\right)^2+\frac{1}{2}\left(\frac{\partial\phi}{\partial y}\right)^2 - f\phi\right] dxdy \qquad (2.13)$$

หากเราใช้เอลิเมนต์สามเหลี่ยมซึ่งมีจุด 3 จุดต่อ โดยตัวไม่รู้ค่าที่จุดต่อ คือ ϕ_1,ϕ_2,ϕ_3 ดังกล่าวที่ได้แสดงใน**ภาพที่ 2.10** เราจะได้สมการเอลิเมนต์ทั้งหมด 3 สมการ จากการหาค่าต่ำสุดของฟังก์ชันการแปรผัน J ที่เกี่ยวข้องกับตัวไม่รู้ค่าที่จุดต่อทั้งสามนั้น

เราเริ่มทำการหาค่าต่ำสุดของฟังก์ชันการแปรผัน J ที่เกี่ยวข้องกับ ϕ_1 จะได้

$$\frac{\partial J}{\partial\phi_1} = \int_A \left[\left(\frac{\partial\phi}{\partial x}\right)\frac{\partial}{\partial\phi_1}\left(\frac{\partial\phi}{\partial x}\right)+\left(\frac{\partial\phi}{\partial y}\right)\frac{\partial}{\partial\phi_1}\left(\frac{\partial\phi}{\partial y}\right)-f\frac{\partial\phi}{\partial\phi_1}\right] dxdy \qquad (2.14)$$

เนื่องจาก $\phi = N_1\phi_1+N_2\phi_2+N_3\phi_3$

ดังนั้น $\frac{\partial\phi}{\partial x} = \frac{\partial N_1}{\partial x}\phi_1+\frac{\partial N_2}{\partial x}\phi_2+\frac{\partial N_3}{\partial x}\phi_3$

และ $\frac{\partial}{\partial\phi_1}\left(\frac{\partial\phi}{\partial x}\right) = \frac{\partial N_1}{\partial x}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ทำนองเดียวกัน

$$\frac{\partial \phi}{\partial y} = \frac{\partial N_1}{\partial y}\phi_1 + \frac{\partial N_2}{\partial y}\phi_2 + \frac{\partial N_3}{\partial y}\phi_3$$

$$\frac{\partial}{\partial \phi_1}\left(\frac{\partial \phi}{\partial y}\right) = \frac{\partial N_1}{\partial y}$$

และ

$$\frac{\partial \phi}{\partial \phi_1} = N_1$$

เมื่อแทนพจน์เหล่านี้ลงในสมการ (2.14) จะได้

$$\iint_A \left[\left(\frac{\partial N_1}{\partial x}\phi_1 + \frac{\partial N_2}{\partial x}\phi_2 + \frac{\partial N_3}{\partial x}\phi_3 \right)\left(\frac{\partial N_1}{\partial x} \right) + \left(\frac{\partial N_1}{\partial y}\phi_1 + \frac{\partial N_2}{\partial y}\phi_2 + \frac{\partial N_3}{\partial y}\phi_3 \right)\left(\frac{\partial N_1}{\partial y} \right) - fN_1 \right] dxdy = 0$$

จากนั้นจึงคูณออกมาและจัดเรียงพจน์ใหม่จะได้

$$\iint_A \left(\frac{\partial N_1}{\partial x}\frac{\partial N_1}{\partial x} + \frac{\partial N_1}{\partial y}\frac{\partial N_1}{\partial y} \right) dxdy\phi_1 + \iint_A \left(\frac{\partial N_1}{\partial x}\frac{\partial N_2}{\partial x} + \frac{\partial N_1}{\partial y}\frac{\partial N_2}{\partial y} \right) dxdy\phi_2 + \iint_A \left(\frac{\partial N_1}{\partial x}\frac{\partial N_3}{\partial x} + \frac{\partial N_1}{\partial y}\frac{\partial N_3}{\partial y} \right) dxdy\phi_3 = \int_A fN_1 dxdy \qquad (2.15)$$

ซึ่งเป็นสมการที่จากการหาค่าต่ำสุดของ J ที่เกี่ยวข้องกับ ϕ_1 ในทำนองเดียวกันเราจะได้สมการที่สองและสามหากเราทำการหาค่าต่ำสุดของ J ที่เกี่ยวข้องกับ ϕ_2 และ ϕ_3 ตามลำดับทั้งสาม สมการที่ได้นี้ คือ สมการไฟไนต์เอลิเมนต์ของรูปสามเหลี่ยม ซึ่งจะเขียนให้อยู่ในรูปแบบสมการของเมทริกซ์ได้ดังนี้

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} \qquad (2.16)$$

หรือ

$$\underset{(3\times3)}{[K]^{(e)}} \underset{(3\times1)}{\{\phi\}^{(e)}} = \underset{(3\times1)}{\{F\}^{(e)}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
82763

โดย $$K_{ij} = \int_A \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \right) dxdy \qquad (2.17)$$

$$F_i = \int_A f N_i dxdy \qquad i,j = 1,2,3 \qquad (2.18)$$

แต่จากสมการ (2.9) ฟังก์ชันการประมาณภายในเอลิเมนต์สามเหลี่ยมที่ได้สร้างมาแล้วคือ

$$N_i = \frac{1}{2A}(a_i + b_i x + c_i y)$$

ดังนั้น $$\frac{\partial N_i}{\partial x} = \frac{b_i}{2A} \quad \text{และ} \quad \frac{\partial N_i}{\partial y} = \frac{c_i}{2A}$$

แทนกลับลงในสมการ (2.17) จะได้

$$K_{ij} = \int_A \left(\frac{b_i}{2A}\frac{b_j}{2A} + \frac{c_i}{2A}\frac{c_j}{2A} \right) dxdy$$

$$= \frac{b_i b_j + c_i c_j}{4A^2} \int_A dxdy$$

$$K_{ij} = \frac{b_i b_j + c_i c_j}{4A} \qquad (2.19)$$

ทำนองเดียวกัน หากค่า f ในสมการ (2.18) คงที่ เราจะได้

$$F_i = f \int_A N_i dxdy$$

$$= f \int_A \frac{1}{2A}(a_i + b_i x + c_i y)\, dxdy$$

$$F_i = \frac{fA}{3} \qquad (2.20)$$

สำหรับสมการ (2.20) นั้นสามารถอินทิเกรตได้ง่าย ๆ โดยใช้สูตรที่ว่า

$$\int_A N_1^{\alpha} N_2^{\beta} N_3^{\chi}\, dxdy = \frac{\alpha!\,\beta!\,\chi!}{(\alpha+\beta+\chi+2)!} \cdot 2A \qquad (2.21)$$

ดังตัวอย่างเช่น หาก $\alpha = 1, \beta = \chi = 0$ เราจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\int_A N_1\,dxdy = \frac{1!\,0!\,0!}{(1+0+0+2)!}\cdot 2A$$

$$= \frac{2A}{3!} = \frac{A}{3} \qquad \text{เป็นต้น}$$

ซึ่งในหัวข้อนี้เราได้เรียนรู้ลำดับขั้นตอนในการสร้างสมการไฟไนต์เอลิเมนต์แบบสองมิติโดยวิธีการแปรผัน เราพบว่าประโยชน์ของวิธีการนี้คือเราสามารถสร้างสมการไฟไนต์เอลิเมนต์ขึ้นมาได้โดยง่าย อย่างไรก็ตามวิธีการนี้ก็มีจุดอ่อนซึ่งก็คือ เราจำเป็นต้องรู้ฟังก์ชันการแปรผันที่สอดคล้องกับสมการเชิงอนุพันธ์ของปัญหานั้น ๆ ซึ่งสำหรับปัญหาหลาย ๆ ชนิดในทางปฏิบัติ เช่น การคำนวณปรากฏการณ์ของของไหลบางอย่าง เราสามารถสร้างสมการเชิงอนุพันธ์ขึ้นมาได้ แต่ไม่สามารถหาฟังก์ชันการแปรผันที่สอดคล้องกันนั้นได้ สาเหตุเช่นนี้เอง ทำให้การสร้างสมการไฟไนต์เอลิเมนต์ในช่วงหลังพัฒนาวิธีการไฟไนต์เอลิเมนต์นี้โดยการใช้สมการอนุพันธ์โดยตรง ซึ่งจะกล่าวในหัวข้อถัดไป

2.3 การสร้างสมการไฟไนต์เอลิเมนต์โดยวิธีการถ่วงน้ำหนักเศษตกค้าง

เราได้ทราบหลักการสร้างสมการไฟไนต์เอลิเมนต์โดยวิธีการแปรผัน สมการเหล่านี้สามารถสร้างขึ้นได้โดยง่ายหากเรารู้ฟังก์ชันการแปรผันที่สอดคล้องกับปัญหาที่เราจะแก้นั้น แต่สำหรับปัญหาทั่ว ๆ ไปในทางปฏิบัติ โดยปกติเราจะรู้เพียงแต่สมการเชิงอนุพันธ์ และไม่สามารถหาฟังก์ชันแปรผันที่สอดคล้องได้ จึงต้องเรียนรู้อีกวิธีหนึ่งที่เรียกว่าวิธีถ่วงน้ำหนักเศษตกค้าง (method of weighted residuals) คือหลักการสร้างสมการไฟไนต์เอลิเมนต์จากการใช้สมการอนุพันธ์โดยตรง โดยไม่จำเป็นต้องรู้ฟังก์ชันแปรผันที่สอดคล้องกัน โดยวิธีการถ่วงน้ำหนักเศษตกค้าง สามารถจำแนกออกได้หลายวิธีการย่อย แต่วิธีที่สะดวกในการนำไปใช้ในการสร้างสมการไฟไนต์เอลิเมนต์ ที่จะกล่าวถึง ก็คือ วิธีการกาเลอร์คิน (Galerkin)

2.3.1 วิธีกาเลอร์คิน (Galerkin)

การถ่วงน้ำหนักเศษตกค้างโดยวิธีกาเลอร์คิน ทำโดยการคูณฟังก์ชันเศษตกค้าง R ด้วยฟังก์ชันน้ำหนัก (weighting function) W จากนั้นทำการอินทิเกรตตลอดทั้งโดเมนแล้วกำหนดผลที่ได้ให้เท่ากับศูนย์ นั่นคือ

$$\int_0^1 R(t)\,W_i(t)\,dt = 0 \qquad (2.22)$$

และเนื่องจากเราต้องการ 2 สมการดังนั้นค่า i = 1, 2 การเลือกฟังก์ชันน้ำหนัก $W_i(t)$ นั้นโดยปกติจะเลือกจากฟังก์ชันที่เกี่ยวข้องกับตัวไม่รู้ค่า C_1 และ C_2 ที่ปรากฏอยู่ในสมการของผลเฉลยโดยประมาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X(t) = 1 + C_1 \underbrace{t}_{W_1} + C_2 \underbrace{t^2}_{W_2} \qquad (2.23)$$

แทน $W_1 = t$ และ $W_2 = t^2$ ลงในสมการ (2.22) จะได้

$$\int R(t)\, t\, dt = 0 \; ; \qquad \frac{1}{2} + \frac{5}{6} C_1 + \frac{11}{12} C_2 = 0$$

$$\int R(t)\, t^2 dt = 0 \; ; \qquad \frac{1}{3} + \frac{7}{12} C_1 + \frac{7}{10} C_2 = 0$$

และจาก 2 สมการนี้ เราจะได้

$C_1 = -0.9143$ และ $C_2 = 0.2857$

ดังนั้น

$$x(t) = 1 - 0.9143\, t + 0.2857\, t^2 \qquad (2.24)$$

อนึ่ง ด้วยเหตุผลหลาย ๆ ประการดังที่ได้อธิบายในตอนท้ายของหัวข้อย่อยนี้ เราจะใช้วิธีการ กาเลอร์คินนี้ในการสร้างสมการไฟไนต์เอลิเมนต์ในเอลิเมนต์ต่าง ๆ เช่น ลักษณะการกระจายของผลเฉลย โดยประมาณบนเอลิเมนต์หนึ่งมิติ ดังเช่นสมการดังนี้

$$\phi = \underbrace{N_1}_{W_1} \phi_1 + \underbrace{N_2}_{W_2} \phi_2 \qquad (2.25)$$

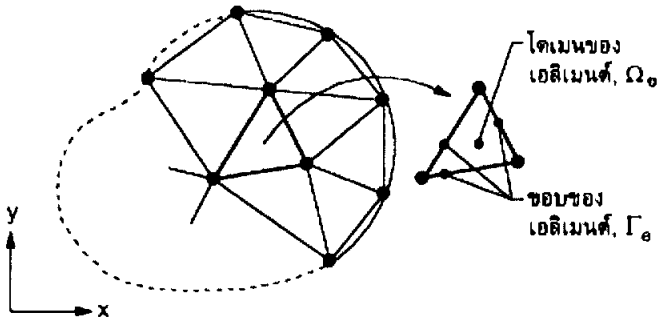
ในที่นี้ ϕ_1 และ ϕ_2 คือตัวไม่รู้ค่าเช่นเดียวกับ C_1 และ C_2 ในสมการ (2.23) ดังนั้น N_1 และ N_2 สามารถนำไปใช้เป็นฟังก์ชันน้ำหนัก W_1 และ W_2 ในการสร้างสมการไฟไนต์เอลิเมนต์ได้

2.3.2 ขั้นตอนการถ่วงน้ำหนักเศษตกค้างสำหรับไฟไนต์เอลิเมนต์

การแก้ปัญหาไฟไนต์เอลิเมนต์โดยวิธีการถ่วงน้ำหนักเศษตกค้างประกอบด้วยลำดับ ขั้นตอนที่สำคัญ 6 ขั้นตอนดังต่อไปนี้

<u>ขั้นที่ 1</u> แบ่งรูปร่างลักษณะของปัญหาที่กำหนดมาให้ออกเป็นเอลิเมนต์ย่อย เช่น แบ่งออกเป็นเอลิเมนต์สามเหลี่ยมย่อย ๆ สำหรับปัญหาในสองมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.11 การแบ่งลักษณะรูปร่างของปัญหาออกเป็นเอลิเมนต์

จากนั้นให้ทำการหาสมการเชิงอนุพันธ์ที่สอดคล้องกับปัญหาที่ต้องการแก้นั้น สมการเชิงอนุพันธ์โดยทั่วไปสามารถเขียนให้อยู่ในรูป

$$L(\bar{\phi}) = 0 \qquad (2.26)$$

โดย L คือ ตัวดำเนินการเชิงอนุพันธ์ (differential operator)

$\bar{\phi}$ คือ ตัวแปรตามแม่นตรง

ขั้นที่ 2 สมมุติลักษณะการกระจายของผลเฉลยโดยประมาณบนเอลิเมนต์ให้อยู่ในรูป

$$\phi = \phi(x,y)\sum_{i=1}^{m} N_i\phi_i = \underset{1\times m}{[N]}\underset{m\times 1}{\{\phi\}} \qquad (2.27)$$

โดย m คือ จำนวนจุดต่อของเอลิเมนต์นั้น

N_i คือ ฟังก์ชันการประมาณภายในเอลิเมนต์

ϕ_i คือ ตัวไม่รู้ค่าที่จุดต่อ

ขั้นที่ 3 สร้างสมการไฟไนต์เอลิเมนต์โดยวิธีการถ่วงน้ำหนักของเศษตกค้างหากเราแทนผลเฉลยโดยประมาณดังแสดงในสมการ (2.27) ลงในสมการเชิงอนุพันธ์ในสมการ (2.26) เราจะพบว่า

$L(\phi)$ จะ $\neq 0$ แต่จะ $= R$

ซึ่ง R คือเศษตกค้าง (Residual) นั่นหมายถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R = L(\phi) = L([N]\{\phi\}) = L\left(\sum_{i=1}^{m} N_i \phi_i\right) \qquad (2.28)$$

และจากวิธีกาเลอร์คิน (Galerkin) ดังที่ได้อธิบายในหัวข้อย่อย 2.3.1 ซึ่งมีขั้นตอนโดยเริ่มจากการคูณเศษตกค้าง R ด้วยฟังก์ชันน้ำหนัก (weighting function) W จากนั้นจึงอินทิเกรตตลอดทั้งโดเมนของเอลิเมนต์นั้น แล้วกำหนดผลที่ได้ให้เท่ากับศูนย์ นั่นคือ

$$\int W_i R \, d\Omega = 0 \qquad i = 1, 2, \ldots, m \qquad (2.29)$$

และโดยปกติเราจะเลือก $W_i = N_i$ ซึ่งเรียกว่าบับโนฟ-กาเลอร์คิน (Bubnov-Galerkin) แต่ หากเราเลือก $W_i \neq N_i$ ซึ่งใช้กับปัญหาบางชนิดจะเรียกว่าเพทรอฟ-กาเลอร์คิน (Petrov-Galerkin)

<u>ขั้นที่ 4</u> อินทิเกรตทีละส่วน (integrate by parts) ซึ่งหากเราแทนสมการ (2.28) ลงในสมการ (2.29) แล้วอินทิเกรตทีละส่วนจะได้

$$\int W_i R \, d\Omega = \int_{\Omega^{(e)}} W_i L\left(\sum_{i=1}^{m} N_i \phi_i\right) d\Omega$$

$$= \underbrace{\int_{\Omega^{(e)}} (W_i, N_i, \phi_i) \, d\Omega}_{\text{พจน์ที่เกี่ยวข้องกับโดเมน ของเอลิเมนต์ } \Omega^{(e)}} + \underbrace{\int_{\Gamma^{(e)}} (W_i, N_i, \phi_i) \, d\Gamma}_{\text{พจน์ที่เกี่ยวข้องกับขอบเขต ของเอลิเมนต์ } \Gamma^{(e)}} = 0$$

<u>ขั้นที่ 5</u> แทนพจน์ที่เกี่ยวข้องกับขอบเขตของเอลิเมนต์, $\Gamma^{(e)}$, ด้วยภาวะขอบเขตอื่นๆที่เกี่ยวข้อง ซึ่งจะก่อให้เกิดสมการของเอลิเมนต์ที่สมบูรณ์สำหรับปัญหานั้น

<u>ขั้นที่ 6</u> จากนั้นจึงเขียนสมการของเอลิเมนต์ ซึ่งมีทั้งหมด m สมการให้อยู่ในรูปของเมทริกซ์ นั่นคือ

$$\underset{(m\times m)(m\times 1)}{[K]\{\phi\}} = \underset{(m\times 1)}{\{F\}} \qquad (2.30)$$

โดย [K] คือ เอลิเมนต์เมทริกซ์ของความแข็งเกร็ง (element stiffness matrix)

{ϕ} คือ เวกเตอร์ซึ่งประกอบด้วยตัวไม่รู้ค่าที่ จุดต่อต่าง ๆ ของเอลิเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

{F} คือ โหลดเวกเตอร์ของเอลิเมนต์นั้น

เมื่อได้สมการไฟไนต์เอลิเมนต์ดังเช่นแสดงในสมการ (2.30) แล้ว ลำดับขั้นตอนต่อไปก็ทำเช่นเดียวกันกับที่ได้อธิบายข้างต้น กล่าวคือ รวมสมการของเอลิเมนต์ย่อยเข้าด้วยกันก่อให้เกิดสมการระบบรวม จากนั้นก็กำหนดกฎเกณฑ์ขอบเขต แล้วจึงแก้สมการระบบรวมเพื่อหาค่าผลลัพธ์ที่จุดต่อต่าง ๆ

2.4 ฟังก์ชันการประมาณภายในเอลิเมนต์และการอินทิเกรตเอลิเมนต์เมตริกซ์เชิงตัวเลข

เราได้ศึกษาขั้นตอนในการสร้างสมการไฟไนต์เอลิเมนต์ โดยวิธีการแปรผันและโดยวิธีการถ่วงน้ำหนักเศษตกค้างตามลำดับ เราได้พบว่า ขั้นตอนที่สำคัญขั้นตอนหนึ่งในวิธีการไฟไนต์เอลิเมนต์ก็คือ การเลือกลักษณะของฟังก์ชันการประมาณภายในเอลิเมนต์ (element interpolation functions) การเลือกฟังก์ชันการประมาณภายในเอลิเมนต์ที่เหมาะสมสามารถเพิ่มความแม่นยำของผลลัพธ์ที่คำนวณได้ และหากผู้ใช้มีความเข้าใจในลักษณะของปัญหานั้น ๆ ได้โดยถ่องแท้ก็สามารถประดิษฐ์ฟังก์ชันการประมาณภายในเอลิเมนต์เพื่อก่อให้เกิดผลลัพธ์เป็นผลเฉลยแม่นตรงได้สำหรับปัญหาในบางกรณี

พิจารณาฟังก์ชันการประมาณภายในเอลิเมนต์สองมิติซึ่งใช้เอลิเมนต์รูปสามเหลี่ยม

เอลิเมนต์รูปสามเหลี่ยมเป็นเอลิเมนต์ที่นิยมใช้กันมากเนื่องจากแบบจำลองไฟไนต์เอลิเมนต์ที่ประกอบด้วยเอลิเมนต์สามเหลี่ยมจำนวนมากสามารถสร้างขึ้นมาได้โดยง่ายและโดยอัตโนมัติ เราได้ศึกษาเอลิเมนต์รูปสามเหลี่ยมที่ประกอบด้วย 3 จุดต่อดังแสดงในภาพที่ 2.7 และได้ทำการสร้างฟังก์ชันการประมาณภายในเอลิเมนต์รูปสามเหลี่ยมนั้นโดยเริ่มจากการสมมุติลักษณะการกระจายของผลเฉลยโดยประมาณให้อยู่ในลักษณะการกระจายแบบแผ่นเรียบ (flat plane) กล่าวคือ

$$\phi(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y \qquad (2.31)$$

โดย α_1 = 1 , 2, 3 เป็นค่าคงตัวที่หาได้จากเงื่อนไขของค่าที่ตำแหน่งจุดต่อทั้งสาม หลังจากนั้นเราสามารถเขียนลักษณะการกระจายของผลเฉลยโดยประมาณให้อยู่ในรูปแบบของค่าที่จุดต่อ ϕ_1, ϕ_2, ϕ_3 ได้ดังนี้

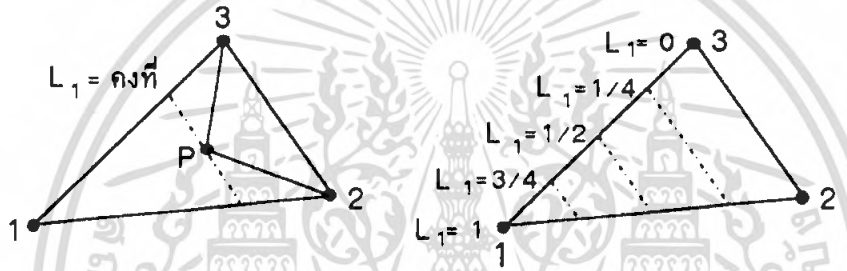
$$\phi = N_1\phi_1 + N_2\phi_2 + N_3\phi_3 \qquad (2.32)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย N_i, i = 1, 2, 3 คือฟังก์ชันการประมาณภายในเอลิเมนต์ ซึ่งอยู่ในรูปแบบของพิกัด x, y ดังแสดงในสมการ (2.9) และต่างมีลักษณะการกระจายในรูปแบบของแผ่นเรียบ ดังแสดงในภาพที่ 2.9 เช่นเดียวกันกับเอลิเมนต์ในหนึ่งมิติ การกระจายของผลเฉลยโดยประมาณของ เอลิเมนต์รูปสามเหลี่ยมสามารถเขียนให้อยู่ในรูปแบบของพิกัดธรรมชาติ ดังนี้

$$\phi = L_1\phi_1 + L_2\phi_2 + L_3\phi_3 \qquad (2.33)$$

โดย L_1 , i = 1, 2, 3 สามารถอธิบายได้โดยการพิจารณาภาพที่ 2.12



ภาพที่ 2.12 ฟังก์ชันการประมาณภายในเอลิเมนต์รูปสามเหลี่ยม

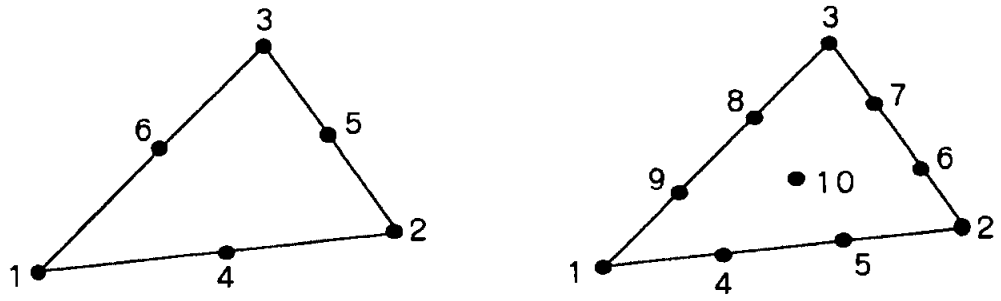
เช่น L_1 = (พื้นที่สามเหลี่ยม P23) / (พื้นที่สามเหลี่ยม 123) (2.34)

เป็นต้น นั่นคือ ตลอดแนวเส้นตรงระหว่างจุดต่อ 2 และ 3 ค่า L_1 นี้จะเท่ากับศูนย์ และบนเส้นประซึ่งขนานกับเส้นตรงระหว่างจุดต่อ 2 และ 3 นี้ดังแสดงในรูปเล็กซ้ายมือ ค่า L_1 จะมีค่าคงตัวไม่ว่าจุด P จะอยู่ ณ ตำแหน่งใดบนเส้นประนี้ และค่า L_1 นี้จะแปรผันในลักษณะเชิงเส้นตรงจากศูนย์จนถึงหนึ่งดังแสดงในรูปเล็กขวามือ ในทำนองเดียวกัน L_2 และ L_3 คืออัตราส่วนระหว่างพื้นที่สามเหลี่ยม P31 และ P12 ต่อพื้นที่สามเหลี่ยม 123 ตามลำดับ เนื่องจากฟังก์ชันการประมาณภายในเอลิเมนต์ L_1 , L_2และ L_3 ต่างอยู่ในรูปของอัตราส่วนของพื้นที่ ดังนั้นจึงถูกเรียกกันโดยทั่วไปว่าเป็นฟังก์ชันการประมาณภายใน ในรูปแบบพิกัดของพื้นที่ (area coordinates) และสามารถเขียนให้อยู่ในรูปแบบของพิกัด x, y ได้เช่นเดียวกันกับสมการ (2.9) นั่นคือ

$$L_i = N_i \qquad i = 1, 2, 3 \qquad (2.35)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอลิเมนต์รูปสามเหลี่ยมอาจประกอบด้วยจุดต่อมากกว่า 3 จุดต่อ ดังแสดงในภาพที่ 2.13 ซึ่งประกอบด้วยเอลิเมนต์อันดับสอง (quadratic) และอันดับสาม (cubic)



(ก) เอลิเมนต์อันดับสอง (ข) เอลิเมนต์อันดับสาม

ภาพที่ 2.13 เอลิเมนต์สามเหลี่ยมอันดับสูงที่ประกอบด้วยหลายจุดต่อ

ฟังก์ชันการประมาณภายในของเอลิเมนต์เหล่านี้สามารถสร้างขึ้นได้โดยง่ายให้อยู่ในรูปของพิกัดของพื้นที่ L_i , i = 1, 2, 3 ดังแสดงในสมการ (2.35) และ (2.9) ดังนี้

<u>เอลิเมนต์อันดับสอง :</u>

จุดต่อที่มุม $N_i = L_i(2L_i - 1)$ i = 1, 2, 3 (2.36)

จุดต่อที่ด้าน $N_4 = 4L_1L_2$ (2.37)

<u>เอลิเมนต์อันดับสาม :</u>

จุดต่อที่มุม $N_i = \frac{1}{2}(3L_i - 1)(3L_i - 2)$ (2.38)

จุดต่อที่ด้าน $N_4 = \frac{9}{2}L_1L_2(3L_1 - 1)$

$N_5 = \frac{9}{2}L_1L_2(3L_1 - 2)$ (2.39)

จุดเซนทรอยด์ $N_{10} = 27L_1L_2L_3$ (2.40)

ในหัวข้อนี้เราจะมาทำการศึกษาลักษณะและขั้นตอนการสร้างฟังก์ชันการประมาณภายในเอลิเมนต์ของเอลิเมนต์ในรูปแบบสองมิติ จากนั้นเราจะได้ศึกษาวิธีการคำนวณเอลิเมนต์เมทริกซ์ต่าง ๆ โดยการใช้ฟังก์ชันการประมาณภายในที่ได้สร้างขึ้นมานี้ ซึ่งในบางกรณีเราจำเป็นต้องใช้การอินทิเกรตเชิงตัวเลข (numerical integration) เพื่อการคำนวณหาเอลิเมนต์เมทริกซ์เหล่านี้ ขั้นตอนดังกล่าวมีความสำคัญที่จำเป็นต้องทำความเข้าใจเพื่อช่วยในการประดิษฐ์ไฟไนต์เอลิเมนต์โปรแกรมคอมพิวเตอร์หรือการใช้โปรแกรมไฟไนต์เอลิเมนต์ทั่ว ๆ ไปได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 วิธีของจาโคบี

วิธีของจาโคบี เป็นระเบียบวิธีการทำซ้ำในรูปแบบที่ง่ายที่สุด เพื่อให้ง่ายแก่การทำความเข้าใจ จะพิจารณาระบบสมการที่ประกอบด้วย 3 สมการย่อยดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix} \qquad (2.41)$$

หลักการของวิธีนี้ ก็คือ ทำการเขียนแต่ละสมการย่อยนี้ใหม่ให้อยู่ในรูปแบบที่สามารถคำนวณ x_1 จากสมการที่ 1 คำนวณ x_2 ได้โดยตรงจากสมการที่ 2 และเช่นนี้เรื่อยไปดังนี้

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3}{a_{11}} \qquad (2.42a)$$

$$x_2 = \frac{b_2 - a_{21}x_1 - a_{23}x_3}{a_{22}} \qquad (2.42b)$$

$$x_3 = \frac{b_3 - a_{31}x_1 - a_{32}x_2}{a_{33}} \qquad (2.42c)$$

จากนั้นเริ่มทำการคำนวณโดยเริ่มจากการเดาค่า x_1, x_2, x_3 และแทนลงในสมการ (2.42a)-(2.42c) เพื่อคำนวณหา x_1, x_2, x_3 ใหม่ แล้วทำเช่นนี้เรื่อยไปจนผลลัพธ์ของ x_1, x_2, x_3 ต่างลู่เข้าสู่ผลลัพธ์ที่มีความผิดพลาด ε ที่ยอมให้ได้ นั่นคือ

$$\left| \frac{x_i^{k+1} - x_i^k}{x_i^{k+1}} \right| \text{ X } 100\% < \quad \varepsilon$$

โดยตัวห้อยล่าง i = 1,2,3 แทนหมายเลขสมการสำหรับระบบสมการในที่นี้และตัวห้อยบน k แทนการทำซ้ำครั้งที่ k

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพของการประมวลผลก็จะเพิ่มขึ้นตามไปด้วยจนถึงจุดๆหนึ่ง ประสิทธิภาพนั้นก็จะค่อยๆลดลง กล่าวคือในการประมวลผลแบบขนานนั้นเราจะต้องทำการทดลองประมวลผลโปรแกรมจนกว่าจะทราบว่าสำหรับปัญหาหนึ่งๆนั้นเหมาะกับจำนวนหน่วยประมวลผลกี่เครื่อง

2.6.1 การเขียนโปรแกรมแบบขนาน (Parallel Programming)

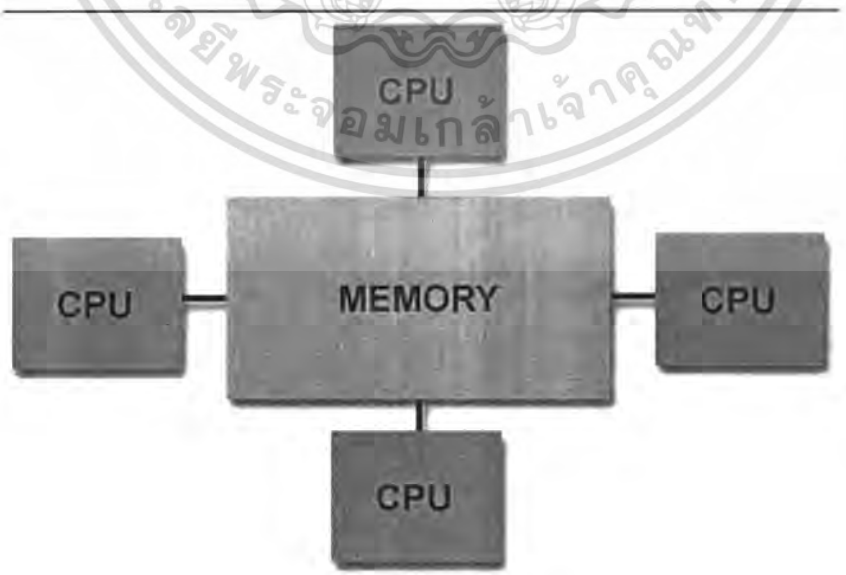
การเขียนโปรแกรมแบบขนาน เป็นเทคนิคการเขียนโปรแกรมที่ใช้สำหรับการประมวลผลแบบขนานพร้อมๆกัน ทั้งภายในเครื่องๆเดียว หรือหลายๆเครื่อง ในการเขียนโปรแกรมแบบขนาน งานเดี่ยวหลายๆงานจะถูกแตกออกเป็นงานย่อยจำนวนหนึ่งที่สามารถถูกประมวลผลแยกจากกันอย่างเป็นอิสระ และถูกนำกลับมารวมกันเพื่อสร้างผลลัพธ์ที่ถูกต้อง การเขียนโปรแกรมแบบขนานทำให้สามารถทำงานที่มีขนาดใหญ่และซับซ้อนให้เร็วขึ้นได้ จากการที่มีงาน (task) ขนาดใหญ่เราสามารถแบ่งย่อยเป็นงานที่มีขนาดเล็กกว่าและทำงานไปพร้อมๆกันได้สิ่งที่จำเป็นในการทำงานแบบขนานคือ

- การแบ่งงาน (task) ออกเป็นงานย่อยๆ
- การมอบหมายงานไปยังเครื่องที่ทำงานเพื่อทำงานพร้อมกัน
- การติดต่อสื่อสารเพื่อทำงานร่วมกันระหว่างเครื่องที่ทำงาน

การเขียนโปรแกรมแบบขนานสามารถนำไปใช้แก้ปัญหา (solving) ต่างๆที่มีขนาดใหญ่หรือใช้เวลานาน เช่น การสร้างตึก การทำงานในองค์กรขนาดใหญ่ การสร้างเครื่องจักรผลิตรถยนต์

2.6.2 Parallel Computer Architectures

2.6.2.1 Shared Memory

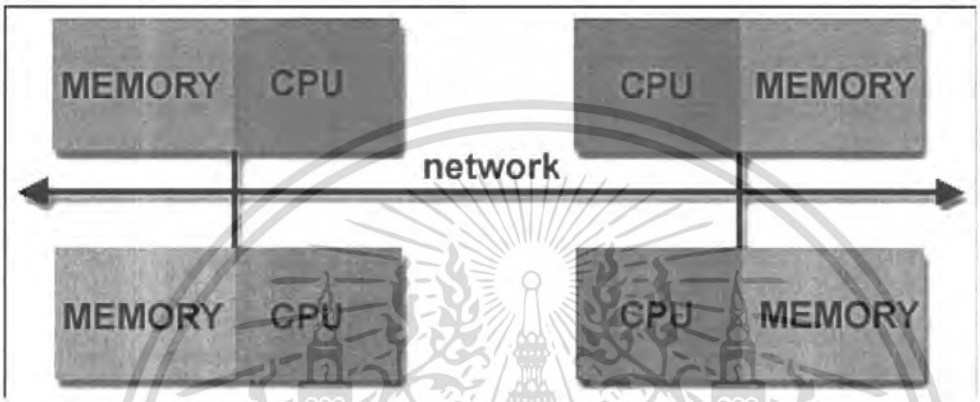


ภาพที่ 2.16 Shared Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรเซสเซอร์แต่ละตัวจะสามารถใช้หน่วยความจำร่วมกัน และอ้างถึงหน่วยความจำโดยใช้ global address space ข้อมูลที่ถูกแก้ไขโดยโปรเซสเซอร์ตัวหนึ่งจะมีผลต่อโปรเซสเซอร์ตัวอื่นๆ แบบนี้จะง่ายต่อการพัฒนาโปรแกรม และมีความเร็วในการอ้างถึงข้อมูลสูง

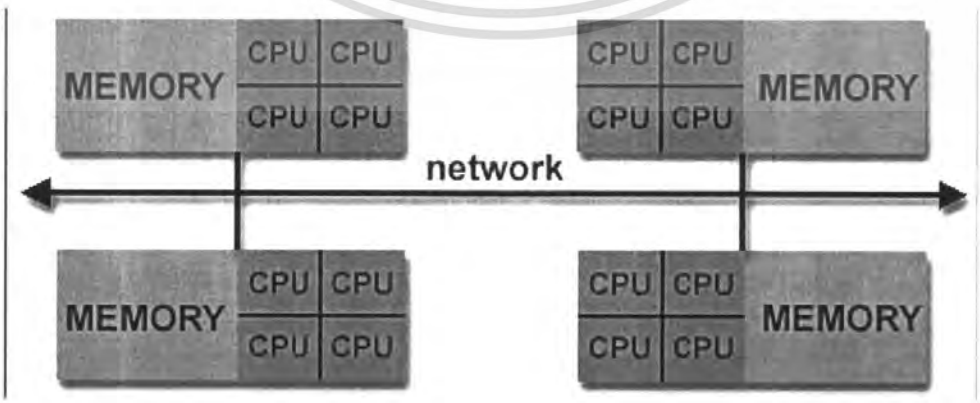
2.6.2.2 Distributed Memory



ภาพที่ 2.17 Distributed Memory

โปรเซสเซอร์แต่ละตัวจะทำงานแบบอิสระต่อกันและมีหน่วยความจำเป็นของตนเอง การแชร์ข้อมูลจะกระทำโดยผ่านการติดต่อทางเครือข่ายและใช้การทำ message passing ซึ่งรูปแบบนี้จะไม่มี overhead ในการดูแล cache coherency

2.6.2.3 Hybrid Distributed-Shared Memory

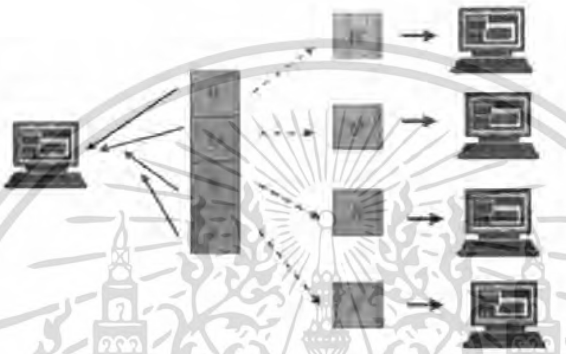


ภาพที่ 2.18 Hybrid Distributed-Shared Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การประมวลผลแบบขนาน (Parallel processing)

การประมวลผลแบบขนาน คือ การแบ่งงานออกเป็นชิ้นเล็ก ๆ แล้วให้แต่ละงานแก่หน่วยประมวลผลหลายๆ หน่วยในเวลาพร้อมกัน ซึ่งประโยชน์ของการใช้วิธีการนี้ คือ แก้ปัญหาขนาดใหญ่ได้ในเวลาที่เร็วขึ้น และลดค่าใช้จ่าย ซึ่งสามารถใช้เครื่องพีซี โดยเชื่อมต่อกันเป็นระบบคลัสเตอร์แทนการใช้เครื่องเมนเฟรมหรือ ซุปเปอร์คอมพิวเตอร์



ภาพที่ 2.14 ลักษณะการแบ่งงานของการประมวลผลแบบขนานแบบหนึ่งบนระบบคลัสเตอร์

ภาพที่ 2.15 ประสิทธิภาพและข้อจำกัดของการประมวลผลแบบขนาน

จากภาพที่ 2.15 สามารถอธิบายได้ว่าเมื่อเราทำการเพิ่มจำนวนของหน่วยประมวลผลให้มากขึ้นแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการนำหลักการของสถาปัตยกรรมแบบ Shared Memory และ แบบ Distributed Memory มาประยุกต์ใช้งานร่วมกัน

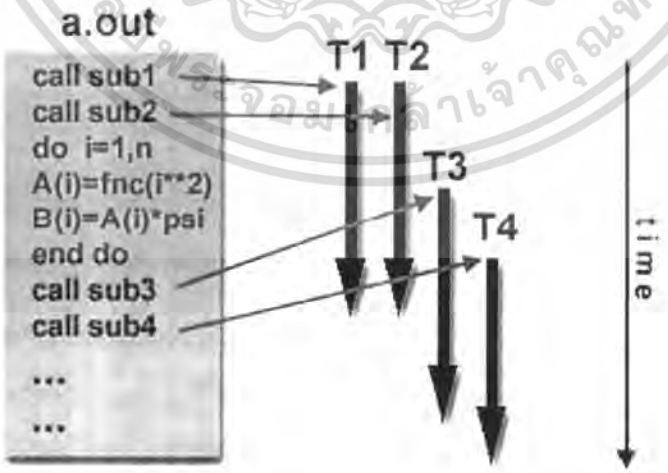
2.6.3 Parallel Programming Paradigms

2.6.3.1 Shared Memory Model

ในโมเดลของการเขียนโปรแกรมแบบแชร์นั้นงานจะถูกแชร์ address ที่ใช้ในการอ่านและ เขียน โดยจะมีกระบวนการต่างๆในการควบคุมการเข้าถึง shared memory เช่น locks และ semaphores ข้อได้เปรียบของโมเดลนี้จากมุมมองของโปรแกรมเมอร์คือไม่ต้องทำการติดต่อข้อมูลระหว่างงาน (task) ทำให้ง่ายในการพัฒนาโปรแกรม แต่จะมีปัญหาในการทำความเข้าใจและจัดการเรื่องของ data locality ในการ implement โมเดลนี้จะใช้ compiler เปลี่ยนตัวแปรของ user program ไปเป็นค่า actual memory addresses ที่เป็นแบบ global

2.6.3.2 Threads Model

ในโมเดลแบบ thread โปรเซสหนึ่งโปรเซสจะสามารถมี path ที่ทำการ execution พร้อมกันได้หลาย path ยกตัวอย่างเช่นโปรแกรม a.out ถูกกำหนดให้ run โดยระบบปฏิบัติการณ์ a.out จะถูกโหลดและจะถือครอง system และ user resources ที่จำเป็นในการ run ไว้ จากนั้น a.out จะสร้าง thread ที่สามารถรันโดยระบบปฏิบัติการณ์พร้อมๆกัน ในแต่ละ thread จะมีข้อมูลของตัวเอง thread จะติดต่อสื่อสารกันผ่านทาง global memory ในการ implement โมเดลนี้จะใช้ POSIX Threads และ OpenMP



ภาพที่ 2.19 ตัวอย่างของ Timing Threads Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3.3 Message Passing Model

โมเดลแบบนี้แต่ละงานจะมีหน่วยความจำของตัวเองในระหว่างการคำนวณ งานหลายๆงานสามารถอยู่บนเครื่องเดียวกันหรือต่างเครื่องกันก็ได้ งานจะแลกเปลี่ยนข้อมูลโดยการรับส่ง message ซึ่งอาศัยการทำงานร่วมกันของทั้งสองโปรเซสยกตัวอย่างเช่น โปรเซสหนึ่งกำลังส่งโปรเซสอีกโปรเซสก็ต้องทำการรับข้อมูล

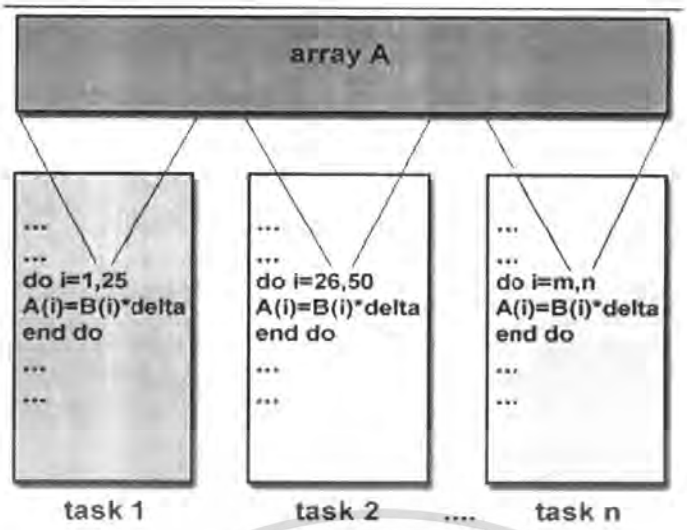
ภาพที่ 2.20 ตัวอย่างการรับ-ส่ง ข้อมูลใน Data Parallel Model

ในการ implement โมเดลนี้เราจะใช้ MPI (Message Passing Interface) ซึ่งเป็นมาตรฐานสำหรับการติดต่อแบบส่งข้อความ (Message Passing) นิยมใช้มากในการคำนวณแบบขนาน (Parallel Computing) และผู้ขาย (Vender) ทั่วไป MPI เป็นมาตรฐานที่ไม่ขึ้นกับแพลตฟอร์ม (Platform-Independent) และไม่ขึ้นกับภาษาที่ใช้เขียน (Language-Independent) สำหรับ library ของการส่ง ข้อความ (Message-Passing Library) นั้นได้รับการพัฒนาโดย MPI Forum ซึ่งเป็นการร่วมมือกัน ระหว่างผู้ขายคอมพิวเตอร์แบบขนาน, ผู้เขียน library และผู้เชี่ยวชาญด้านแอพพลิเคชัน (Application Specialist) และ library ของ MPI นี้จะใช้ภาษา Fortran และ C เป็นหลัก ส่วนในเรื่องของการนำ MPI ไปใช้งานนั้นจะขอธิบายอย่างละเอียดในหัวข้อต่อไป

2.6.3.4 Data Parallel Model

โมเดลนี้จะมีคุณสมบัติดังนี้คือ งานแบบขนานส่วนใหญ่จะทำงานบนเซตของข้อมูลขนาดใหญ่ โดยข้อมูลจะถูกจัดโครงสร้างธรรมดาๆ เช่น array เซตของงานเก็บบนโครงสร้างข้อมูลเดียวกัน อย่างไรก็ตามแต่ละงานจะทำงานบน partition ที่แตกต่างกันบนโครงสร้างข้อมูลเดียวกัน (data structure)ดังภาพที่ 2.21 ในสถาปัตยกรรมแบบ shared memory ทุกๆงานจะเข้าถึงข้อมูลโดยผ่าน global memory ส่วนสถาปัตยกรรมแบบ distributed memory ข้อมูลจะถูกแบ่งเป็น chunks ใน local memory ของแต่ละงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.21 ตัวอย่างการแบ่งข้อมูลในData Parallel Model

2.6.4 วิธีการโปรแกรมแบบขนาน

มีวิธีในการโปรแกรม 4 แบบ คือ

2.6.4.1 **Parallelization compiler** วิธีการนี้ง่าย แค่เพียงศึกษาการทำงานของ compiler ให้รู้แนวทาง แล้วนำมาพัฒนาเพื่อใช้งาน ข้อเสีย คือ มีประสิทธิภาพต่ำ

2.6.4.2 **Programming using Message Passing** คือ การเขียน **MPI** ขึ้นมาใช้งาน **MPI (Message Passing Interface)** วิธีนี้ยาก แต่มีประสิทธิภาพสูง เนื่องจากเป็นการเขียนโปรแกรมที่ทำการแบ่งการทำงานให้กับตัวประมวลผลเอง ซึ่งจะต้องรู้ความสามารถของโปรเซสเซอร์ เพื่อทำการแบ่งงานให้เหมาะสม เช่น ความเร็วของโปรเซสเซอร์ที่ไม่เท่ากัน

2.6.4.3 **Programming using parallel language** คือ การเขียนโปรแกรมด้วยภาษาที่มีคุณสมบัติเป็น parallel programming

2.6.4.4 **Programming using parallel math library** คือ การนำงานทำไว้แล้วมาใช้งาน ซึ่งเป็นงานวิจัยของมหาวิทยาลัยต่าง ๆ ที่ให้ใช้ ข้อเสียคือ เราไม่ทราบการทำงานอะไรเลย

2.6.5 ขั้นตอนการสร้างโปรแกรมแบบขนาน

การเขียนโปรแกรมแบบขนานมีขั้นตอนวิธีการพัฒนาดังต่อไปนี้

2.6.5.1 ถ้าสร้างโปรแกรมขนาน (parallel program) จากโปรแกรมแบบลำดับขั้น (sequencial program) ควรจะ Debug โปรแกรมแบบลำดับขั้นให้เรียบร้อยก่อน จากนั้นควรศึกษา algorithm ด้วยว่าสามารถแก้ปัญหาโดยวิธี parallel programming ได้หรือเปล่า ยกตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างโจทย์ที่แก้ได้โดย parallel programming:

คำนวณพลังงานศักย์ของ conformation หลายพันอันที่เป็นอิสระจากกันของแต่ละโมเลกุล เมื่อเสร็จแล้วหาค่า conformation ของพลังงานที่น้อยที่สุด

จากโจทย์ข้อมูลแต่ละอันเป็นอิสระต่อกันดังนั้นเราจึงสามารถคำนวณไปพร้อมๆกันได้จึงแก้ปัญหาโดยใช้ parallel programming ได้

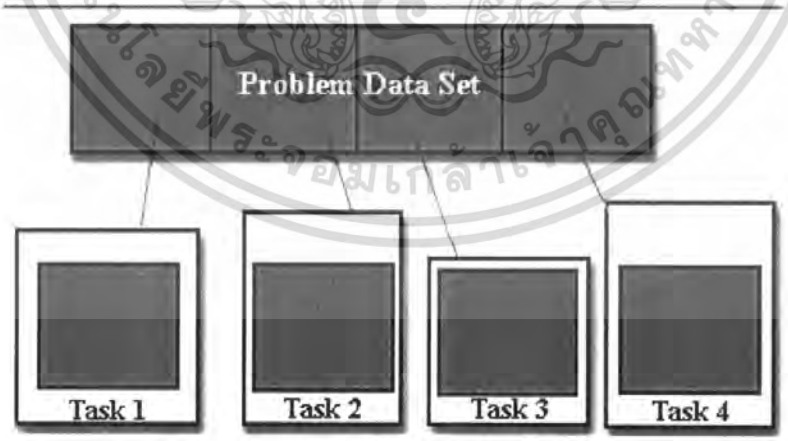
ตัวอย่างโจทย์ที่แก้ไม่ได้ด้วย parallel programming:

การคำนวณ Fibonacci series (1,1,2,3,5,8,13,21,...) โดยใช้สูตรต่อไปนี้ $F(k + 2) = F(k + 1) + F(k)$

โจทย์นี้เราจะเห็นได้ว่าการที่เราจะบวกค่าได้ เราต้องอาศัยข้อมูลก่อนหน้านี้ ดังนั้นเราไม่สามารถจะทำการบวกพร้อมๆกันได้ จึงไม่สามารถใช้ parallel programming ในการแก้ปัญหาได้

2.6.5.2 ทำการระบุว่าส่วนไหนของโปรแกรมสามารถที่จะทำงานพร้อมกันได้

2.6.5.3 ทำการ Decompose ออกตามฟังก์ชันหรือข้อมูล



ภาพที่ 2.22 ตัวอย่างการ Decompose Data Set

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.5.3.1 Function Decomposition (Functional Parallelism)

แบ่งปัญหาออกเป็นหลายๆส่วนเพื่อกระจายไปยังหลายๆ processor เพื่อทำการ execution พร้อมกัน ซึ่งแบบนี้จะเหมาะกับปัญหาที่มีโครงสร้างไม่คงที่และมีจำนวนการคำนวณที่แน่นอน

2.6.5.3.2 Domain Decomposition (Data Parallelism)

แบ่งขอบเขตข้อมูลของปัญหาและกระจายส่วนต่างๆไปยังหลายๆ processor เพื่อทำการ execution พร้อมกัน เหมาะแก่การใช้กับปัญหาซึ่ง ข้อมูลมีค่าคงที่ (การ factor และแก้ไขปัญหา matrix ที่มีขนาดใหญ่หรือจำนวนการคำนวณที่จำกัด)

โครงสร้างข้อมูลที่เป็นแบบ dynamic ถูกรวมเป็น entity เดียวซึ่ง entity สามารถถูกแบ่งเป็น subset ย่อยๆได้ (ปัญหาแบบ multi-body ที่มีขนาดใหญ่)

ขอบเขตมีค่าจำกัดแต่การคำนวณในหลายๆส่วนของขอบเขตจะเป็นแบบdynamic (fluid vortices models)

2.6.5.4 พัฒนาโค้ด (Code development)

2.6.5.5 compile, ทดสอบโปรแกรม, และ Debug

2.6.5.6 Optimization คือการวัดประสิทธิภาพ, กำหนดขอบเขตของปัญหา และปรับปรุงให้ดีขึ้น

2.7 มาตรฐานที่ใช้ในการส่งแมสเซส

เมื่อการประมวลผลแบบขนานได้รับความนิยมมากและความแตกต่างทางสถาปัตยกรรมของเครื่องคอมพิวเตอร์จึงทำให้เกิดมาตรฐานขึ้นเพื่อให้การส่งแมสเซสมีประสิทธิภาพมากขึ้น คือ

2.7.1 PVM (Parallel Virtual Machine)

PVM เป็นซอฟแวร์แพคเกจที่ทำให้กลุ่มของเครื่องคอมพิวเตอร์ที่ต่างกัน ซึ่งเชื่อมต่อกันด้วยเครือข่ายที่ประกอบด้วยสถาปัตยกรรมที่ต่างกัน PVM มีจุดประสงค์เพื่อให้สภาพแวดล้อมที่ต่างกันสามารถติดต่อกันได้ สถาปัตยกรรมแบบหนึ่งสามารถถูกคัดลอกไปยังเครื่องที่มีสถาปัตยกรรมอีกแบบแล้วทำการคอมไพล์(compile)และเอกซีคิวท์(execute)ได้ โดยไม่ต้องทำการดัดแปลงเพื่อให้เครื่องหลายๆเครื่องติดต่อกันเหมือนเป็นคอมพิวเตอร์แบบขนาน PVM ได้กำหนดระบบปฏิบัติการแบบกระจาย โดยจะมีเดมอน (daemon)โพรเซสทำงานบนเครื่องคอมพิวเตอร์ทั้งหมดเพื่อให้เกิด virtual machine และ PVM นี้ master จะถูกสร้างใน local machine และจะสร้างเดมอน (daemon) ในเครื่องอื่นๆโดยอัตโนมัติ แต่ใน MPI นั้น master กับ slave จะถูกสร้างขึ้นพร้อมกัน

2.7.2 MPI (Message Passing Interface)

MPI ถูกสร้างขึ้นเพื่อพัฒนามาตรฐานในการเขียนโปรแกรมส่งแมสเซสโดย MPI Forum MPI มีคุณสมบัติดังนี้

- MPI เป็น library เป็นมาตรฐานในการส่งแมสเซส
- ไม่ต้องทำการดัดแปลงซอสโค้ด(source code) เมื่อจะนำแอพพลิเคชั่นไปใช้กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แพลตฟอร์ม(platform) อื่นที่สนับสนุน MPI

• การอิมพลีเมนต์(implementation)ของผู้ผลิต(vendor)จะสามารถใช้ลักษณะเด่นของฮาร์ดแวร์ที่มีอยู่เพื่อให้เกิดประสิทธิภาพได้อย่างเต็มที่

• มีจำนวนฟังก์ชั่นมากกว่า115 รูทีน(routine) และจะเพิ่มขึ้นได้โดยขึ้นอยู่กับประสบการณ์จากระบบการส่งข้อมูลต่างๆที่ใช้งาน MPI

2.7.2.1 ชุดคำสั่งของ MPI (Message Passing Interface)

The Message Passing Interface (MPI) หรือ ส่วนต่อประสานการส่งผ่านข้อความ เป็น library ที่ใช้ในการสร้างโปรแกรมแบบขนาน ซึ่งมีการติดต่อกันระหว่างโปรเซสโดยการส่งข้อความ

มาตรฐานของ MPI ออกแบบมาให้สนับสนุนกับเครื่องหลายรูปแบบ เช่น คอมพิวเตอร์แบบขนาน ซึ่งรวมไปถึงซูเปอร์คอมพิวเตอร์ และระบบคอมพิวเตอร์แบบกลุ่ม ในการประมวลผลแบบขนานใน MPI จะมีการแตกโปรแกรมออกเป็นโปรเซสย่อยๆ สมมติ n โปรเซส จะเรียกเซตของ n โปรเซสนี้ว่า “กลุ่ม” แต่ละโปรเซสในกลุ่มจะมีหมายเลขกำกับซึ่งเป็นเอกลักษณ์ หมายความว่าในกลุ่มเดียวกันจะไม่มีหมายเลขกำกับใดที่กำกับโปรเซสมากกว่า 1 โปรเซส เราเรียกหมายเลขกำกับนี้ว่า “rank” ในกลุ่มที่มี n โปรเซส จะมีค่า rank ตั้งแต่ 0 ถึง n-1

Communicator คือ กลุ่มของโปรเซสที่สามารถส่งข้อความไปหาโปรเซสอื่นได้ โดยทั่วไปแล้วกลุ่มจะเกี่ยวข้องกับ communicator ที่ใช้ในการดำเนินการของกลุ่ม ในโปรแกรมพื้นฐานทั่วไปจะมีเพียง communicator เดียวที่ต้องการ คือ MPI_COMM_WORLD ซึ่งเป็น communicator ที่ประกอบด้วยทุกโปรเซสที่ทำงานตอนที่โปรแกรมเริ่มขึ้น

ในปัจจุบัน MPI มี ฟังก์ชันถึง 129 แล้ว แต่ในผู้เริ่มต้นนั้น 6 ฟังก์ชันก็เพียงพอที่จะสามารถสามารถสร้าง algorithm อย่างง่ายได้แล้วมีดังต่อไปนี้

MPI_INIT(ierr)

MPI_FINALIZE(ierr)

MPI_COMM_RANK(comm, rank, ierr)

MPI_COMM_SIZE(comm, p, ierr)

MPI_SEND(message, count, data type, dest, tag, comm, ierr)

MPI_RECV(message, count, data type, source, tag, comm, status, ierr)

โดยที่ count,dest,ierr,p,rank,source,status,tag เป็นตัวแปรหรือค่าคงที่ชนิดจำนวนเต็ม

ierr ใช้เก็บค่า error จากการเรียกฟังก์ชันนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

comm คือ communicator ที่ใช้

rank ใช้เก็บค่า rank ของโปรเซสใน communicator

p ใช้เก็บจำนวนของโปรเซสใน communicator

message ใช้เก็บข้อความที่จะส่งหรือรับ

count ใช้เก็บจำนวนของ message ที่จะส่งหรือรับ

data type เป็นชนิดของข้อมูลใน message ที่จะส่งหรือรับ ซึ่งเป็นชนิดข้อมูลของ MPI

dest ใช้เก็บค่า rank ของโปรเซสปลายทาง

source ใช้เก็บค่า rank ของโปรเซสต้นทาง

tag ใช้กำกับ message ซึ่งจะช่วยให้โปรเซสที่รับ message นี้ ระบุได้ว่าเป็น message ใดจากโปรเซสต้นทาง

status ใช้เก็บสถานะหรือข้อมูลของ message ที่จะรับ

การเรียกใช้ฟังก์ชันของ MPI จะใช้คำสั่ง CALL ตามด้วยฟังก์ชันของ MPI ที่ต้องการใช้ เช่น CALL MPI_INIT(IERR) เป็นต้น

คำอธิบายในแต่ละฟังก์ชัน

MPI_INIT - ใช้ในการเริ่ม library และ environment ที่ใช้ในการรันโปรแกรม MPI คำสั่งนี้ต้องใช้ก่อนการใช้คำสั่ง MPI ทุกครั้ง

MPI_FINALIZE - ใช้เพื่อส่งสัญญาณให้ MPI runtime library รู้ว่า MPI program ได้จบการทำงานแล้ว คำสั่งนี้ใช้เพื่อจบการใช้งาน MPI อย่างสมบูรณ์แบบ

MPI_Comm_Rank – ใช้เพื่อให้ได้รับค่า rank ของ process ซึ่งมีความสำคัญในการตัดสินใจว่า จะกระจายงานใหญ่ส่วนไหนให้กับ process นี้

MPI_Comm_Size - ใช้เพื่อให้ได้รับค่าจำนวนของ process ที่อยู่ใน communicator ของโปรแกรมนี้

MPI_SEND ใช้ในการส่ง message ไปยังโปรเซสเป้าหมายที่มี rank ใน communicator ตรงกับค่าในตัวแปร dest ตัวแปร message จะทำหน้าที่เป็น buffer ของ message ที่จะส่งไปยังโปรเซสเป้าหมาย เมื่อข้อมูลใน buffer ได้ถูกส่งมอบไปยังระบบแล้ว ฟังก์ชันนี้จะย้อนกลับมายังโปรแกรม และ buffer จะสามารถใช้ใหม่ได้

MPI_RECV ใช้ในการรับ message จากโปรเซสต้นทางที่มี rank ใน communicator ตรงกับค่าในตัวแปร source และมีค่า tag ที่แนบมากับ message ตรงกับค่าในตัวแปร tag โดยฟังก์ชันนี้จะทำการรอจนกระทั่งพบ message ที่ต้องการรับแล้ว จึงรับ message นั้นผ่านทาง buffer (ตัวแปร message) เราอาจกำหนดค่า MPI_ANY_SOURCE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนตัวแปร source เพื่อกำหนดว่าให้รับข้อมูลจากทุกๆ โปรเซสต้นทางที่มีค่า tag ที่แนบมากับ message ตรงกับค่าในตัวแปร tag และเราอาจกำหนดค่า MPI_ANY_TAG แทนตัวแปร tag เพื่อกำหนดให้รับ message ที่มี tag ใดก็ได้จากโปรเซสต้นทางที่มี rank ใน communicator ตรงกับค่าในตัวแปร source

อัลกอริทึมแบบขนานนั้นมักจะมีการดำเนินการติดต่อประสานงานกันที่เกี่ยวข้องกับโปรเซสหลายโปรเซส เช่น การหาผลรวม การหาค่าสูงสุดต่ำสุดของข้อมูลต่างๆ ที่อยู่กระจัดกระจายในแต่ละโปรเซส อย่างไรก็ตาม MPI มีตัวดำเนินการที่รองรับหน้าที่ดังต่อไปนี้

1. การใช้ Barrier ในการประสานจังหวะโปรเซสทุกโปรเซส

ใช้ฟังก์ชัน MPI_BARRIER(comm,ierr) เพื่อทำการประสานจังหวะทุกๆ โปรเซส หลักการทำงานคือ โปรเซสที่เรียกฟังก์ชันนี้จะถูกบล็อกไว้จนกว่าทุกๆ โปรเซสใน communicator จะเรียกฟังก์ชันนี้

2. การกระจายข้อมูลออกจากหนึ่งโปรเซสไปยังทุกๆ โปรเซส

ใช้ฟังก์ชัน MPI_BCAST(buffer, count, data type, root, comm, ierr) เพื่อทำการกระจายข้อมูลใน buffer จากโปรเซสหนึ่งไปยังทุกๆ โปรเซส

3. การรวบรวมข้อมูลจากทุกๆ โปรเซสไปยังโปรเซสหนึ่งๆ

ใช้ฟังก์ชัน MPI_GATHER(send_buf, send_count, sent_type, recv_buf, recv_count, recv type, root, comm, ierr) เพื่อทำการรวบรวมข้อมูลจากทุกๆ โปรเซสไปยังโปรเซสหนึ่งๆ โดยข้อมูลจากโปรเซสที่ n จะถูกวางใน buffer ก่อนโปรเซสที่ n+1 ขนาดของตัวแปร recv_buf จะต้องเป็น n เท่าของตัวแปร send_buf

4.การแยกย้ายข้อมูลจากโปรเซสหนึ่งไปยังทุกโปรเซส (ตรงข้ามกับการรวบรวมข้อมูล)

ใช้ฟังก์ชัน MPI_SCATTER (send_buf, sent_count, sent_type, recv_buf, recv_count, recv_type, root, comm, ierr) เพื่อแยกย้ายข้อมูลจากโปรเซสหนึ่งไปยังทุกโปรเซสที่เกี่ยวข้อง ซึ่งการดำเนินการจะตรงข้ามกับการรวบรวมข้อมูล

5. การลดรูปข้อมูลที่กระจัดกระจาย เช่น การผลรวม, การหาค่าสูงสุดต่ำสุด ฯลฯ

ใช้ฟังก์ชัน MPI_REDUCE(operand, result, count, datatype, op, root, comm, ierr) เพื่อใช้ตัวดำเนินการ (op) ไปดำเนินการกับแต่ละค่าในตัวแปร operand ของแต่ละโปรเซสในกลุ่มและคืนค่าผลลัพธ์ไปยังโปรเซส root ใช้ฟังก์ชัน MPI_ALLREDUCE(operand, result, count, datatype, op, comm, ierr) ซึ่งจะมีการทำงานเหมือนกับ MPI_REDUCE แต่จะคืนค่าไปยังทุกๆ โปรเซส สำหรับตัวดำเนินการ op นั้นจะเป็นตัวใดตัวหนึ่งในตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operation Name	Meaning
MPI_MAX	Maximum
MPI_MIN	Minimum
MPI_SUM	Sum
MPI_PROD	Product
MPI_LAND	Logical And
MPI_LOR	Logical Or
MPI_LXOR	Logical Exclusive Or
MPI_BAND	Bitwise And
MPI_BOR	Bitwise Or
MPI_BXOR	Bitwise Exclusive Or
MPI_MAXLOC	Maximum and Location of Maximum
MPI_MINLOC	Minimum and Location of Minimum

ตารางที่ 2.1 แสดง MPI Operation Name

2.8 ระบบคลัสเตอร์ (Cluster system)

ระบบคลัสเตอร์ คือ การนำคอมพิวเตอร์หลาย ๆ ตัวมาเชื่อมต่อกันเพื่อให้ทำงานเหมือนเครื่องใหญ่เครื่องเดียว ความแตกต่างระหว่างระบบคลัสเตอร์กับระบบLAN คือ ระบบLAN เป็นการเชื่อมต่อคอมพิวเตอร์จำนวนมากเพื่อใช้ทรัพยากรร่วมกันโดยแต่ละเครื่องยังเป็นเครื่องอิสระ แต่ระบบคลัสเตอร์เป็นการรวมเครื่องหลายๆเครื่องที่ประสานงานกันอย่างแน่นแฟ้นจนเปรียบเสมือนเป็นเครื่องเดียวกัน ระบบคลัสเตอร์สามารถนำมาประยุกต์ใช้งานแทนระบบเซิร์ฟเวอร์ขนาดใหญ่ได้เป็นอย่างดี

โครงสร้างของระบบคลัสเตอร์ทั่วไปจะประกอบด้วยเครื่องพีซีสมรรถนะสูงจำนวนหนึ่งที่นำมาเชื่อมต่อกันด้วยเครือข่ายความเร็วสูง โครงสร้างพื้นฐานของระบบคลัสเตอร์และระบบ LAN ไม่แตกต่างกัน แต่ระบบคลัสเตอร์จะพึ่งพาซอฟต์แวร์พิเศษที่กระจายงานไปในกลุ่มของคอมพิวเตอร์ที่มารวมกันทำงานเป็นระบบคลัสเตอร์ดังนั้นคุณสมบัติสำคัญของเทคโนโลยี คลัสเตอร์มีอยู่สามประการ คือ เครือข่ายความเร็วสูง ระบบซอฟต์แวร์ที่สนับสนุนการทำงานแบบคลัสเตอร์และโปรแกรมประยุกต์ที่ใช้

แนวความคิดในการใช้งานเทคโนโลยีคลัสเตอร์นั้นเริ่มมาจากโครงการแบวูฟล์ของนาซ่า เกิดขึ้นมาในปี 1994 โดยนักวิจัยสองคนคือ ดร.โทมัส สเตอร์ลิ่ง และ โดนัล เบคเกอร์ ซึ่งในขณะนั้นทำงานอยู่ที่ CESDIS (Center of Excellence in Space Data and Information Science) ซึ่งเป็นศูนย์วิจัยอยู่ที่ Goddart Space Flight Center ในรัฐแมรี่แลนด์ ประเทศสหรัฐอเมริกา โครงการนี้ได้รับการสนับสนุนจากโครงการ ESS (Earth and Space Science Project) โดยมีจุดมุ่งหมายที่จะพัฒนาเทคนิคในการนำคอมพิวเตอร์แบบ

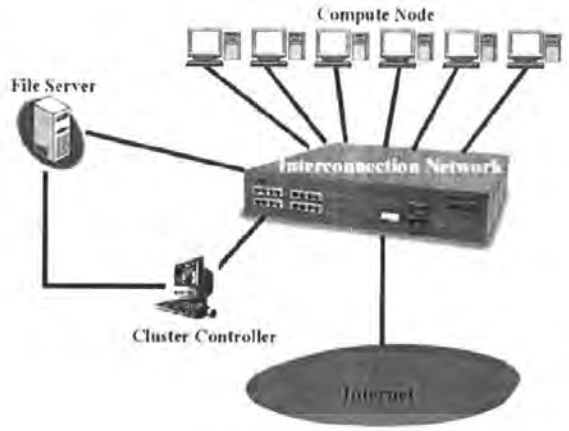
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนานขนาดใหญ่ เพื่อมาใช้แก้ปัญหาหรือจัดการกับข้อมูลขนาดใหญ่ที่ได้รับมาจากการสำรวจอวกาศ แบวูฟล์คลัสเตอร์เครื่องแรกนั้นสร้างขึ้นมาจากเครื่อง PC 486DX100 16 เครื่องที่เชื่อมต่อกันผ่านอีเธอเน็ต โครงการนี้เริ่มเป็นที่รู้จักมากขึ้นเมื่อมีการสาธิตในงาน Supercomputing 97 ด้วยการนำเครื่อง PC PentiumPro 200 จำนวน 16 เครื่องมาต่อเข้าด้วยกันและพิสูจน์พลังให้เห็นโดยการคำนวณได้กว่า 1 พันล้านครั้งต่อวินาที ซึ่งเป็นสมรรถนะระดับซูเปอร์คอมพิวเตอร์อย่างแท้จริง ในขณะที่เครื่องนาซ่าได้สร้างระบบคลัสเตอร์ขนาดใหญ่ไว้ใช้งานหลายระบบ เช่น Chiba city ซึ่งมีพีซี 256 เครื่องมาต่อกัน แต่ระบบที่มีประสิทธิภาพสูงสุดได้แก่ CPlant ซึ่งเกิดจากการนำเครื่อง Alpha มาลงลีนุกซ์แล้วต่อกันทั้งหมด 592 หน่วยประมวลผล ทำให้คำนวณได้กว่า 54.24 จิกะฟลอบหรือกว่า 5 หมื่นล้านคำสั่งต่อวินาที และติดอันดับที่ 62 ของคอมพิวเตอร์ที่เร็วที่สุดในโลก

ภาพที่ 2.23 ระบบ CPlant ที่ Sandia National Laboratory

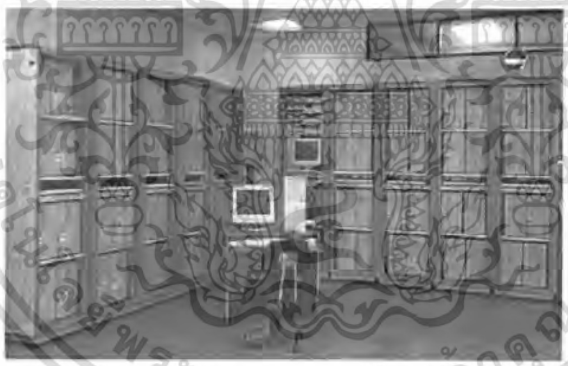
แบวูฟล์คลัสเตอร์(beowulf cluster)เป็นระบบที่มีสถาปัตยกรรมประกอบด้วยเครื่องพีซีคอมพิวเตอร์หลาย ๆ ตัวซึ่งเชื่อมต่อเข้าด้วยกันโดยใช้เครือข่ายความเร็วสูง สถาปัตยกรรมแบวูฟล์จะต้องประกอบด้วยคุณสมบัติต่าง ๆ ดังนี้ อันดับแรก อุปกรณ์ที่นำมาต่อเชื่อมในระบบแบวูฟล์คลัสเตอร์นี้ต้องเป็นของที่หาซื้อได้ง่ายตามท้องตลาดทั่วไป ประการที่สองคือ ซอฟต์แวร์ทั้งหมดที่ใช้ต้องเป็นแบบ Open Source เช่น ใช้ระบบปฏิบัติการลีนุกซ์ ประการถัดมา คือ คอมพิวเตอร์แต่ละตัวต้องเชื่อมต่อกันด้วยเครือข่ายความเร็วสูงภายใน ซึ่งเป็นประการหลักที่ทำให้ระบบมีความแตกต่างจากระบบ LAN ทั่วไป ประการสุดท้ายคือ ภายในระบบต้องมีซอฟต์แวร์สำหรับใช้ในการประมวลผลแบบขนานหรือกระจายที่ทำให้สามารถดึงกำลังของเครื่องมาใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.24 โครงสร้างระบบแบวูฟล์คลัสเตอร์

สำหรับในประเทศไทยก็มีระบบแบวูฟล์คลัสเตอร์ขนาดใหญ่ชื่อว่า พิรุณคลัสเตอร์ ตั้งอยู่ที่สำนักบริการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์ ซึ่งขณะนี้มีอยู่ทั้งหมด 75 โหนดที่เชื่อมต่อกันอยู่ ทำให้ระบบคำนวณได้เร็วถึง 10 จิกะฟลอบ (Gflop) ระบบนี้มีความเร็วติดอันดับที่ 43 ของระบบแบวูฟล์ทั่วโลกในขณะนี้



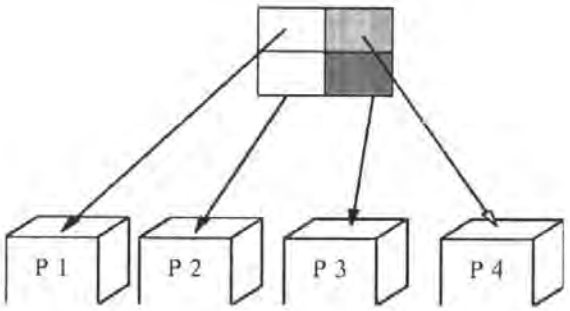
ภาพที่ 2.25 พิรุณคลัสเตอร์ที่มหาวิทยาลัยเกษตรศาสตร์

2.8.1 จุดประสงค์ของการนำเทคโนโลยีคลัสเตอร์มาใช้งาน

การนำเทคโนโลยีระบบคลัสเตอร์มาใช้งานจะมีด้วยกันสองจุดประสงค์หลัก ๆ คือ

2.8.1.1 ช่วยเร่งความเร็วของงานหรือทำให้ระบบรองรับผู้ใช้งานได้มากขึ้น สำหรับประการนี้เป็นไปได้โดยใช้เทคโนโลยีของการประมวลผลแบบขนานหรือแบบกระจาย ซึ่งเป็นการแบ่งงานใหญ่ ๆ ออกเป็นงานเล็กๆจำนวนมากแล้วให้เครื่องหลายเครื่องช่วยกันทำงาน เช่น ถ้าเราต้องการทำระบบเว็บเซิร์ฟเวอร์ที่ขยายขนาดได้มาก ทำได้โดยการกระจายการเรียกค้นของผู้ใช้ไปหลาย ๆ เครื่อง ดังนั้นผู้ใช้รู้สึกได้ทันทีว่าระบบเร็วขึ้น เนื่องจากแต่ละเครื่องรับงานน้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.26 แนวคิดของการประมวลผลแบบขนาน

2.8.1.2 เพื่อทำให้ระบบมีความน่าเชื่อถือได้สูงขึ้นสามารถรองรับงานที่เป็น Mission Critical Operation งานเหล่านี้เช่น การบริการฐานข้อมูล ต้องการระบบที่ทำงานได้ตลอด 24 ชั่วโมง 7 วัน และตลอดปี การประยุกต์หลักการของคลัสเตอร์อาศัยการทำงานประสานกันของระบบจำนวนมากเมื่อเครื่องใดเครื่องหนึ่งหยุดทำงาน งานต่าง ๆ จะถูกโอนย้ายหรือทดแทนโดยอัตโนมัติไปยังเครื่องอื่น ทำให้ผู้ใช้รู้สึกว่าเครื่องทำงานได้ตลอดเวลา

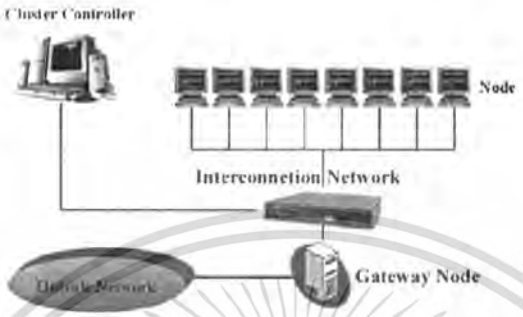
ภาพที่ 2.27 ระบบ High Availability

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 ชนิดของคลัสเตอร์

โครงสร้างของคลัสเตอร์แบ่งออกได้เป็นสองแบบดังนี้

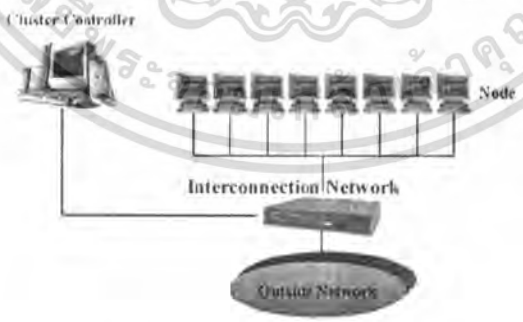
2.8.2.1 คลัสเตอร์แบบปิด



ภาพที่ 2.28 โครงสร้างระบบคลัสเตอร์แบบปิด

ระบบนี้จะต่อคลัสเตอร์ผ่านเกตเวย์ที่ซ่อนทั้งระบบจากโลกภายนอก ข้อดีก็คือ มีความปลอดภัยสูง และจะใช้อินเตอร์เน็ตแอดเดรสเพียงแอดเดรสเดียวเท่านั้น ข้อเสียคือ แต่ละโหนดในระบบไม่สามารถช่วยกันบริการข้อมูลกับโลกภายนอกได้ จึงเหมาะกับการใช้ทำงานคำนวณขนาดใหญ่ภายในเท่านั้น

2.8.2.2 คลัสเตอร์แบบเปิด



ภาพที่ 2.29 โครงสร้างระบบคลัสเตอร์แบบเปิด

ระบบนี้จะต่อกับเน็ตเวิร์คภายนอกโดยตรง ทำให้ผู้ใช้เข้าถึงทุกโหนดในระบบคลัสเตอร์ได้โดยตรง ข้อเสียคือ ความปลอดภัยจะต่ำลงมากเพราะต้องคอยดูแลทุกเครื่องในระบบ และยังต้องการหมายเลขอินเตอร์เน็ตแอดเดรสจำนวนมาก แต่ข้อดีก็คือระบบสามารถช่วยกันบริการข้อมูลได้ จึงเหมาะกับงานบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข่าวสารจำนวนมากเช่นใช้เป็นระบบเซิร์ฟเวอร์สำหรับ WWW หรือ FTP ที่ขยายตัวได้ ระบบนี้จะเหมาะกับการทำเป็นฐานข้อมูลขนาดใหญ่ด้วย

2.9 การวัดผลของประสิทธิภาพของการประมวลผลแบบขนาน

สิ่งที่วัดมีอยู่ 3 อย่าง คือ เวลาที่ใช้ในการรัน (Runtime) ความเร็วที่เพิ่มขึ้น (Speed up) และประสิทธิภาพ (Efficiency)

2.9.1 เวลาที่ใช้ในการรัน แบ่งเป็น

2.9.1.1 เวลาที่ใช้ในการทำงานในระบบแบบลำดับขั้น (T_S) นับตั้งแต่เวลาที่เริ่มทำงานจนจบการทำงานของโปรแกรม (งานถูกทำในเครื่องเดียว)

2.9.1.2 เวลาที่ใช้ในการทำงานในระบบแบบขนาน (T_p) นับตั้งแต่เวลาที่เครื่องแรกในระบบเริ่มการทำงานจนกระทั่งถึงเวลาที่เครื่องสุดท้ายในระบบทำงานเสร็จ (งานถูกแจกจ่ายให้ทุกเครื่องทำ)

2.9.2 ความเร็วที่เพิ่มขึ้น คือ อัตราส่วนระหว่าง T_S และ T_p เป็นบ่งบอกว่าการนำระบบการประมวลผลแบบขนานมาใช้นั้น ช่วยเพิ่มความเร็วได้มากน้อยเท่าใดเทียบกับเวลาที่ใช้ระบบแบบลำดับขั้น ซึ่งแสดงได้ดังนี้

$$S = \frac{Ts}{Tp} \leq P$$

โดยที่ S คือ ความเร็วที่เพิ่มขึ้น ถ้า T_p มีค่าน้อยกว่า T_S เท่าใด ค่า S จะมากขึ้นเท่านั้น และยิ่งค่า S มาก ก็เรียกได้ว่ามีความเร็วที่เพิ่มขึ้นมาก กล่าวคือการนำระบบการประมวลผลแบบขนานมาใช้ได้ช่วยให้ทำงานเร็วขึ้นมากเมื่อเทียบกับเวลาที่ใช้ระบบแบบลำดับขั้น

2.9.3 ประสิทธิภาพ คือ อัตราส่วนระหว่าง S และ P เป็นบ่งบอกว่าคอมพิวเตอร์แต่ละเครื่องช่วยเพิ่มความเร็วได้มากน้อยเท่าใด ซึ่งแสดงได้ดังนี้

$$E = \frac{S}{P} \leq 1$$

โดยที่ P คือ จำนวนเครื่องทั้งหมด ถ้า S มีค่ามากกว่า P เท่าใด ค่า E จะมากขึ้นเท่านั้น และยิ่งค่า E มาก ก็เรียกได้ว่ามีประสิทธิภาพมาก กล่าวคือคอมพิวเตอร์ 1 เครื่องได้ช่วยให้ทำงานแบบขนานเร็วขึ้นมากนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 Sun Grid Engine (SGE)

Sun Grid Engine (SGE) คือ ระบบการจัดการงานและควบคุมจัดสรรทรัพยากรของระบบซึ่งพัฒนาโดยซันไมโครซิสเตมส์ รองรับการทำงานบนระบบปฏิบัติการยูนิกส์และลีนุกซ์ โดยการทำงานของซันกริดเอ็นจิ้น แบ่งการทำงานของเครื่องคอมพิวเตอร์ออกเป็น 4 ส่วนการทำงาน คือ

1. **มาสเตอร์โฮส (Master host)**

เครื่องคอมพิวเตอร์ที่เป็นศูนย์กลางการทำงานของกิจกรรมทั้งหมดบนระบบคลัสเตอร์ เช่น การจัดลำดับงาน การสั่งงาน หรือการปรับปรุงข้อมูลการเข้าใช้งานของผู้ใช้แต่ละคน

2. **เอ็กซคิวชั่นโฮส (Execution host)**

เครื่องคอมพิวเตอร์ลูกข่ายในระบบคลัสเตอร์ที่อนุญาตให้ซันกริดเอ็นจิ้นสามารถส่งงานมาประมวลผลในเครื่องเอ็กซคิวชั่นโฮส ภายใต้คำสั่งของมาสเตอร์โฮส

3. **แอดมินิสเตรเตอร์โฮส (Administration host)**

เครื่องคอมพิวเตอร์ในระบบคลัสเตอร์ที่อนุญาตให้ปรับแต่งค่าการทำงานและแก้ไขไฟล์ที่ใช้ควบคุมการทำงานทั้งหมดในระบบ

4. **ซับมิดโฮส (Submit host)**

เครื่องคอมพิวเตอร์ในระบบคลัสเตอร์ที่อนุญาตให่ส่งงานเข้ามาประมวลผลในระบบ

คุณสมบัติของซันกริดเอ็นจิ้น

1. สนับสนุนระบบคอมพิวเตอร์ที่ประกอบขึ้นจากคอมพิวเตอร์ต่างชนิด
2. สนับสนุนการใช้งานระบบคลัสเตอร์แบบหลายโดเมน
3. สนับสนุนการจัดเก็บข้อมูลการใช้ทรัพยากรของงานแต่ละงาน
4. สนับสนุนการทำจ๊อบอาร์เรย์ (Job array)
5. สนับสนุนงานทั้งรูปแบบลำดับขั้นและแบบขนาน
6. สามารถทำ Load Balancing ได้
7. สามารถทำ Check Point ได้
8. มีระบบสำหรับการตรวจสอบสถานะของระบบ
9. ผู้ใช้ระบบสามารถกำหนดลำดับความสำคัญของงานของตนเองได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขั้นตอนการดำเนินงานและหลักการทำงานของโปรแกรม

3.1 ขั้นตอนในการดำเนินงาน

1. วางหัวข้อและขอบเขตของโครงงาน โดยจะทำการศึกษาหลักการทางคณิตศาสตร์ของสมการปัวส์ซองในรูปแบบ

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$$

หรือ $$\nabla^2 u = f(x, y)$$

โดยศึกษาสมการปัวส์ซองของวิธีผลต่างสืบเนื่องและวิธีไฟไนต์เอลิเมนต์ ซึ่งทำการศึกษาวิธีผลต่างสืบเนื่องก่อน เพื่อที่จะทำให้เข้าใจระเบียบวิธีไฟไนต์เอลิเมนต์ได้ง่ายขึ้น

2. ทำการศึกษาวิธีการเขียนโปรแกรมแบบลำดับขั้นและแบบขนาน โดยได้ทดลองเขียนโปรแกรมพื้นฐานเพื่อความเข้าใจเบื้องต้นทั้งในรูปแบบของโปรแกรมแบบลำดับขั้นและแบบขนาน ซึ่งการทดลองเขียนโปรแกรมแบบลำดับขั้นก่อนนั้นจะทำให้มีความเข้าใจในขั้นตอนวิธีของการทำงานของโปรแกรม ซึ่งจะทำให้การเขียนโปรแกรมนั้นให้อยู่ในรูปโปรแกรมแบบขนานมีความง่ายมากขึ้น โปรแกรมที่ได้ทำการทดลองเขียนมีดังนี้

2.1 โปรแกรมคูณเมตริกซ์แบบลำดับขั้นชื่อ SerialMatrix.c

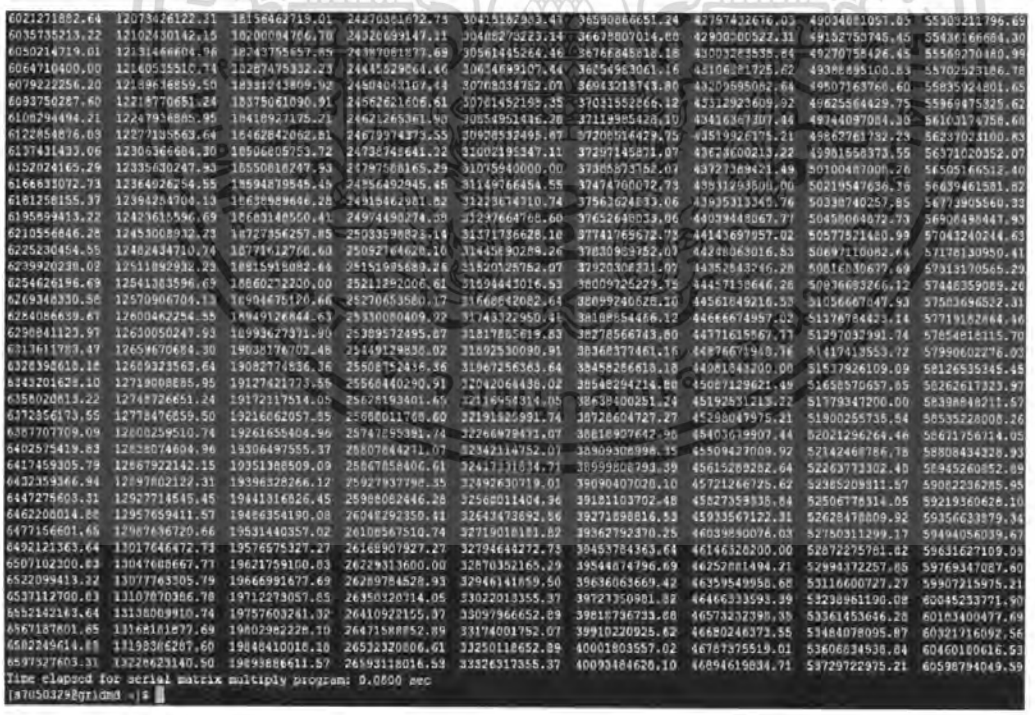
เป็นโปรแกรมคูณเมตริกซ์อย่างง่าย ซึ่งทำการนำเมตริกซ์ A เป็นตัวตั้งแล้วคูณด้วยเมตริกซ์ B และได้เมตริกซ์ผลลัพธ์ C จากนั้นทำการแสดงเมตริกซ์ผลลัพธ์ C ออกมา ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
The result matrix is

3771519.01  7559213.22  11363082.64  15183127.27  19019347.11  22871742.15  26740312.40  30625087.85  34525978.51
7559213.22  15150869.42  22774968.60  30431510.74  38120495.87  45841923.97  53595795.04  61382109.09  69200866.12
11363082.64  22774968.60  34235657.85  45745150.41  57303446.28  68910545.45  80566447.93  92271153.72  104024662.81
15183127.27  30431510.74  45745150.41  61124046.28  76568198.35  92077606.61  107652271.07  123292191.74  138997368.60
19019347.11  38120495.87  57303446.28  76568198.35  95914752.07  115343107.44  134853264.46  154445223.14  174118983.47
22871742.15  45841923.97  68910545.45  92077606.61  115343107.44  138707047.93  162169428.10  185730247.93  209389507.44
26740312.40  53595795.04  80566447.93  107652271.07  134853264.46  162169428.10  189600761.90  217147266.12  244808940.50
30625087.85  61382109.09  92271153.72  123292191.74  154445223.14  185730247.93  217147266.12  248696277.69  280377282.64
34525978.51  69200866.12  104024662.81  138997368.60  174118983.47  209389507.44  244808940.50  280377282.64  316094533.88
38443074.38  77052066.12  115826975.21  154767801.65  193874545.45  233147206.61  272585785.12  312190280.99  351960694.21
42376345.45  84935709.09  127678090.91  170603490.91  213711909.09  257003345.45  300477800.00  344135272.73  387975763.64
46325791.74  92851795.04  139578009.92  186504436.36  233631074.38  280957923.97  328484985.12  376212257.85  424139742.15
50291413.22  100800323.97  151526732.23  202470638.02  253632041.32  305010942.15  356607340.50  408421236.36  460452629.75
54273209.92  108781295.87  163524257.85  218502095.87  273714809.92  329162400.00  384844866.12  440762208.26  496914426.45
58271181.82  116794710.74  175570586.78  234598809.92  293879380.17  353412297.52  413197561.98  473235178.85  533525132.23
62285328.93  124840568.60  187665719.01  250760780.17  314125752.07  377760634.71  441665428.10  505840132.23  570284747.11
66315651.24  132918869.42  199809654.55  266988006.61  334455925.62  402207411.57  470248464.46  538877084.30  607193271.07
70362148.76  141029613.22  212002393.39  283280489.26  354863900.83  426752628.10  499946671.07  573446025.75  644250704.13
74424821.49  149172800.00  224243935.54  299638228.10  375355677.69  451396284.30  527760047.93  604446968.60  681457046.28
78503669.42  157348429.75  236534280.99  316061223.14  395929256.20  476138380.17  556688595.04  637579900.83  718812297.52
82598692.56  165556502.48  248873429.75  332549474.38  416584636.36  500978915.70  585732312.40  670844826.45  756316457.85
86708890.91  173797018.18  261261381.82  349102581.82  437321018.18  525917890.91  614891200.00  704241745.45  793969527.27
90837264.46  182069976.86  273698137.19  365721745.45  458140801.65  550955305.79  644165257.85  737770657.85  831771505.79
94980813.22  190375378.51  286186695.87  382405765.29  479041586.78  576091160.33  673554486.95  771431563.64  869722393.39
99140537.19  198713225.14  298718057.85  399155041.32  500024173.55  601325454.55  703058884.30  805224462.81  907822190.08
103316436.36  207083510.74  311301223.14  415969573.55  521080561.98  626658188.43  732676452.89  839149355.37  946070895.87
107508510.74  215486241.32  323933191.74  432849361.98  542234752.07  652089361.98  762413191.74  873206241.32  984468510.74
111716760.33  223921414.88  336613963.64  449794406.61  563462743.80  677618975.21  792263100.83  907385120.66  1023015034.71
115941185.12  232389031.40  349343538.84  466804707.44  584772537.19  703247028.10  822228180.17  941715993.39  1061710467.77
120181785.12  240889090.91  362121917.36  483880264.46  606164132.23  728973520.66  852308429.75  976168859.50  1100554809.92
124438560.33  249421593.39  374949099.17  501021077.69  627637528.93  754798452.89  882503849.59  1010753719.01  1139548061.16
128711510.74  257986538.84  387825084.30  518227147.11  649192727.27  780721824.79  912814439.67  1045470571.90  1178690221.49
133000636.36  266583927.27  400749872.73  535498472.73  670829727.27  806743636.36  943240200.00  1080319418.18  1217901290.91
137305937.19  275213758.65  413723464.46  552835054.55  692548528.93  832863887.60  973781130.58  1115300257.85  1257421269.42
141627413.22  283876033.06  426748859.50  570236292.56  714349132.23  859092576.51  1004437231.40  1150413090.91  1297010157.02
145965064.46  292570750.41  439817057.85  587703986.78  736231537.19  885399709.09  1035208582.48  1185657917.36  1336747953.72
150318890.91  301297910.74  452937059.50  605236337.19  758195742.80  911815279.34  1066094943.80  1221034737.19  1376634659.50
154688892.56  310057514.05  466105864.46  622833943.80  780241752.07  938332929.26  1097096555.37  1256543550.41  1416670274.38
159075069.42  318849560.33  479323472.73  640496806.61  802369561.98  964941738.84  1128213337.19  1292184357.02  1456854798.35
163477421.49  327674049.59  492589884.30  658224925.62  824579173.55  991652628.10  1159445289.26  1327957157.02  1497168231.40
```

ภาพที่ 3.1 ผลลัพธ์ของโปรแกรม SerialMatrix.c (1)



```
60212271882.64  120734261222.31  181564627315.01  242703814672.75  304251167903.47  366998866651.24  429764232676.03  493048081057.85  556850211796.69
60357535213.22  121024301142.15  182000004766.70  243260699147.11  304882739223.14  366788807014.68  429000300522.31  491152753745.45  553436166684.30
60502214719.01  121314466804.96  182437755657.85  243870681877.69  305614455266.46  367668458618.18  430032235535.84  492270758426.45  554568270480.99
60647140400.00  121605755510.74  182877475332.23  244482529664.46  306346893107.44  368549630061.16  431068281725.62  493388695100.83  555702523186.78
60792222256.20  121896616859.50  183318243809.92  245094043107.44  307080034782.07  369432218743.80  432099985062.64  494507163760.60  556835924801.65
60937350287.60  122187770651.24  183759061090.91  245662621606.61  307814521995.35  370315552866.12  433129236609.92  495625644229.75  557969475325.62
61082494494.21  122479336885.95  184199271375.21  246212265361.90  308549351436.20  371199985428.10  434163673007.44  496744087084.30  561031174758.60
61227654876.03  122771185563.64  184642842062.81  246799747373.55  309285532995.87  372086516429.75  435199276175.21  498862761782.23  562370231100.83
61372831433.06  123063666804.30  185086805753.72  247387434641.32  310021195347.11  372971145871.07  436236000213.22  499981558373.55  563710203352.07
61518024165.29  123356030247.93  185530816247.93  247975568165.29  310756940000.00  373856573762.07  437273389421.49  501004870001.26  565051566512.40
61663231072.73  123649026254.55  185974879545.45  248564492945.45  311492766454.55  374742800072.73  438312393800.00  502119547636.76  566394461581.82
61808258155.37  123942264700.13  186419989646.28  249154462981.82  312228674310.74  375629248833.06  439353133345.76  503238740257.85  567738905560.33
61953899413.22  124236135596.69  186863140550.41  249744908274.38  312964664668.60  376516863013.06  440394848067.77  504358004072.73  569084488447.93
62100556846.28  124530008932.23  187272356257.85  250335998825.14  313717356628.10  377417659572.73  441437697957.02  505775521480.99  570443240244.63
62252304854.55  124824434710.74  187711412768.60  250927144628.10  314458590288.26  378309639752.07  442488063016.53  506971100824.64  571781830950.41
62399220238.02  125118892932.23  188151918082.64  251519895689.26  315210128752.07  379203082231.07  443528432946.28  508168306277.69  573131705565.29
62546261196.69  125413383596.69  188602732200.00  252112892006.61  315944431016.53  380099325229.75  444571328646.28  509368832666.12  574488359083.26
62693428330.58  125709006704.13  189044875120.66  252706535080.17  316686842082.64  380992406828.10  445616848216.53  510566607847.93  575803696522.31
62840866639.67  126004462254.55  189491264864.63  253300080409.92  317433229950.41  381885544668.12  446666674957.02  511767844433.14  577191182464.46
62988411123.97  126300050247.93  189936279371.90  253895724995.87  318178855619.83  382778566743.80  447716155567.77  512970032991.74  578548181115.70
63136117783.47  126596670684.30  190381776702.48  254491298838.02  318925530098.91  383673774611.16  448766719418.76  514174135553.72  579900022776.03
63283998618.18  126893323563.64  190827774836.36  255085752436.36  319672566363.64  384568266818.18  449818433200.00  515379261109.09  581265353345.45
63432061628.10  127190088885.95  191274217773.55  255684402901.91  320420664438.02  385462942114.88  450871296621.49  516585570657.85  582626173223.97
63580202813.22  127487726651.24  191721175514.05  256278193401.65  321169558314.05  386380400251.24  451926531213.22  517793447200.00  583988848211.57
63728356173.55  127784769859.50  192166624057.85  256880011768.60  321918329991.74  387278604727.27  452980479975.21  519002555735.54  585352228008.26
63877077709.09  128082259510.74  192616554404.96  257472859381.74  322669798412.07  388168907642.96  454036199907.44  520212962764.46  586717561414.05
64025749419.83  128380074604.96  193066497555.37  258070844271.07  323423114752.07  389093088998.35  455094270009.92  521424668786.78  588084834328.93
64174593305.79  128679222142.15  193513885509.09  258667858406.61  324171311834.71  389998807793.39  456152689282.64  522637713302.48  589452060852.89
64323599366.94  128978702122.31  193963328266.12  259279737798.35  324926307719.01  390904007020.10  457212466725.62  523852098311.57  590822236285.95
64472775603.31  129277714545.45  194411618626.45  259880062446.28  325680114404.36  391811037702.48  458273598338.84  525067763314.05  592193360628.10
64622088014.88  129576659411.57  194863541190.08  260482292350.41  326443478692.56  392718988816.53  459335671722.31  526282478809.92  593566333879.34
64771566601.65  129876536720.66  195314403357.02  261085675510.74  327190181181.82  393627923370.25  460398680076.03  527503112991.17  594940056039.67
64921213363.64  130176446472.73  195765735227.27  261688907927.27  327946442272.73  394537843363.64  461463628200.00  528727275781.82  596316271109.09
65071027200.83  130476088667.77  196217591100.83  262293113600.00  328703352165.29  395448674796.69  462528814494.21  529943722576.85  597693470870.60
65220994113.32  130777463305.79  196669916677.69  262897845828.93  329461141859.50  396360063669.42  463595449958.68  531166007527.27  599071215975.21
65371112700.83  131078070386.78  197122738057.85  263503210714.05  330220013355.37  397213509081.82  464663333593.39  532389611190.08  600452533771.90
65521382183.64  131380009910.74  197576003241.32  264109221255.37  330979966652.89  398187936733.88  465732322398.35  533614536436.28  601813400477.69
65671887801.65  131681181877.69  198029622228.10  264715888452.89  331740001752.07  399102220925.62  466802416373.55  534840780095.87  603217176092.56
65822549614.88  131983886287.60  198484610016.10  265323230806.61  332501118652.89  400018303557.02  467873375519.01  536068634538.84  604601809616.53
65973276603.31  132286823140.50  198938886611.57  265931186016.53  333262317355.37  400934484620.10  468946196834.71  537297722975.21  605987794049.59
Time elapsed for serial matrix multiply program: 9.0800 sec
[s7050329@grid03 ~]$
```

ภาพที่ 3.2 ผลลัพธ์ของโปรแกรม SerialMatrix.c (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

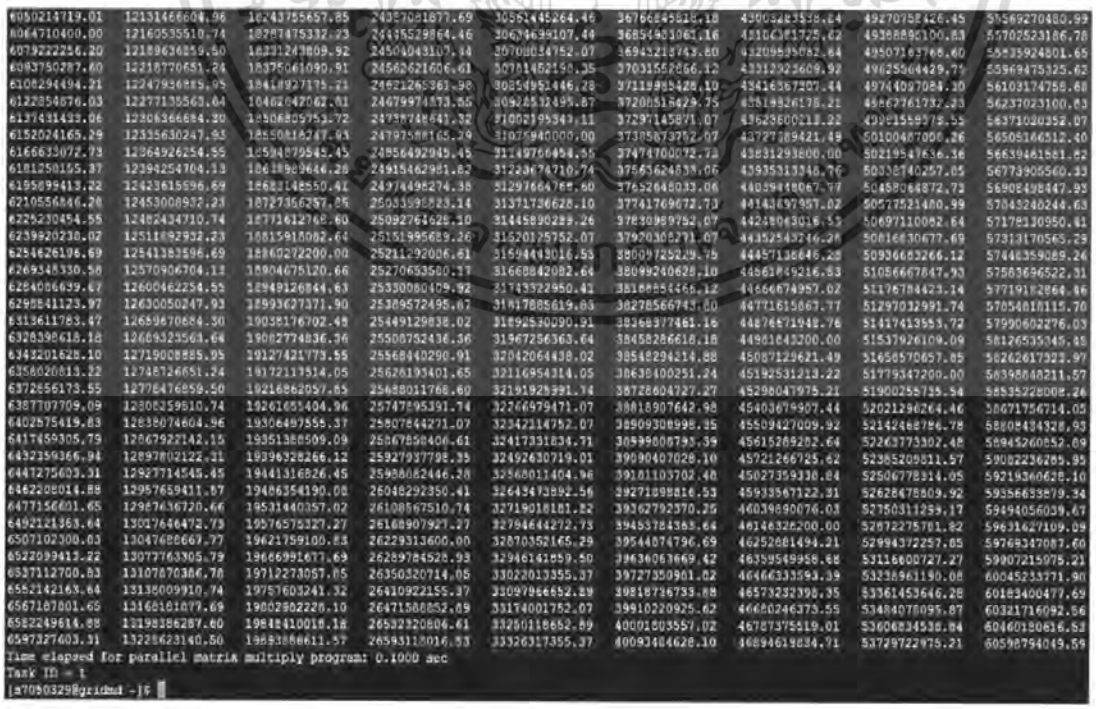
2.2 โปรแกรมคูณเมตริกซ์แบบขนานชื่อ ParallelMatrix.c

เป็นโปรแกรมคูณเมตริกซ์ที่มีความซับซ้อนมากขึ้น แต่ยังคงมีขั้นตอนหลักของการทำงานเหมือนกับโปรแกรมคูณเมตริกซ์แบบลำดับขั้น โดยที่โปรแกรมนี้จะมีการระบุจำนวนเครื่องที่ใช้ในการประมวลผล ด้วยการใช้ฟังก์ชั่น MPI_Comm_size และจะเก็บค่าของจำนวนเครื่องลงไปในตัวแปร totalTask แล้วสามารถใช้ MPI_Comm_rank กำหนดหมายเลขให้แก่เครื่องแต่ละเครื่อง ตัวแปรที่ใช้เก็บหมายเลขของเครื่อง คือ taskId

เครื่องที่ถูกกำหนดหมายเลขเครื่องให้เป็น 0 ในโปรแกรมนี้จะเรียกว่า MASTER มีหน้าที่กำหนดค่าเริ่มต้นให้แก่เมตริกซ์ A และ B แล้วทำการแบ่งงานให้แก่เครื่องอื่นๆที่ไม่ใช่เครื่องหมายเลข 0 ซึ่งเรียกว่า WORKER โดยจะส่งแถวไปให้แต่ละเครื่อง ด้วยการใช้ฟังก์ชั่น MPI_Send ซึ่งจำนวนแถวที่ส่งไปให้แต่ละเครื่องนั้นจะเป็นไปตามค่าในตัวแปรที่ชื่อว่า sendRow จากนั้นรอรับผลลัพธ์จากจากประมวลผลของแต่ละ WORKER ด้วยการใช้ฟังก์ชั่น MPI_Recv

สำหรับ WORKER แต่ละเครื่อง เมื่อได้รับงานจาก MASTER แล้วจะทำการคำนวณ แล้วทำการเก็บผลลัพธ์ลงเมตริกซ์ C แล้วทำการส่งผลลัพธ์ของแต่ละเครื่องกลับไปยังเครื่อง MASTER ด้วยการใช้ฟังก์ชั่น MPI_Send

ขั้นตอนสุดท้าย คือ เครื่อง MASTER จะทำการพิมพ์ผลลัพธ์ที่ได้รับจากเครื่อง WORKER แต่ละเครื่อง ซึ่งจะได้ผลลัพธ์เหมือนกับผลลัพธ์ที่ได้จากโปรแกรมคูณเมตริกซ์แบบลำดับขั้น ดังภาพ



ภาพที่ 3.3 ผลลัพธ์ของโปรแกรม ParallelMatrix.c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ออกแบบและพัฒนาโปรแกรมแก้ปัญหาสมการปัวส์ซองทั้งแบบลำดับขั้นและแบบขนาน โดยใช้โปรแกรม Microsoft Visual C++ ในการพัฒนาโปรแกรม

4. ทำการทดลองและเก็บรวบรวมข้อมูล

4.1 ตรวจสอบและคอมไพล์โปรแกรมทั้งแบบลำดับขั้นและแบบขนานที่เขียนด้วยชุดคำสั่ง MPI

4.2 ทำการรันโปรแกรมทั้งแบบลำดับขั้นและแบบขนาน เพื่อทำการวัดระยะเวลาที่ใช้ในการประมวลผลของโปรแกรมแบบลำดับขั้นและโปรแกรมแบบขนาน

5. วิเคราะห์ประสิทธิภาพของโปรแกรม

5.1 ศึกษาทฤษฎีการคำนวณหาประสิทธิภาพโดยใช้ ค่า Speed up และ Efficiency

5.2 ทำการคำนวณประสิทธิภาพของโปรแกรมดังกล่าว ทั้งในแบบลำดับขั้น และแบบขนาน จากผลคำตอบที่ได้ในแต่ละตัวอย่างที่ใช้ในการทดสอบ

5.3 วิเคราะห์เปรียบเทียบประสิทธิภาพที่คำนวณได้ จากโปรแกรมในแบบลำดับขั้นและแบบขนาน

6. นำเสนอผลงาน

7. ประเมินผลการทำงาน

3.2 ลักษณะการทำงานของโปรแกรม

ในหัวข้อนี้จะเป็นการอธิบายเกี่ยวกับการทำงานของโปรแกรมโดยรวม ซึ่งจะเหมือนกันทั้งโปรแกรมแบบลำดับขั้นและโปรแกรมแบบขนาน โดยโปรแกรมนี้เป็นโปรแกรมที่ใช้ในการหาผลเฉลยของสมการปัวส์ซองโดยใช้วิธีของจาโคบี ในโปรแกรมนี้จะเป็นการสมมติว่ามีแผ่นเหล็ก 2 มิติ แล้วแบ่งแผ่นเหล็กออกเป็นเอลิเมนต์ แต่ละเอลิเมนต์จะมีการกำหนดหมายเลขเอลิเมนต์ และจุดต่อของเอลิเมนต์หรือโหนดก็จะมีการกำหนดหมายเลขไว้ด้วย เมื่อทำการกำหนดหมายเลขให้แก่ทุกจุดต่อแล้ว ต่อไปจะกำหนดค่าตำแหน่ง x และ y ของแต่ละจุดต่อ จากนั้นจะเป็นการสร้างเมตริกซ์ K ของแต่ละเอลิเมนต์ สำหรับค่า x และ y ที่ได้จะนำไปใช้ในการคำนวณหาพื้นที่ (Area) ของเอลิเมนต์ และนำค่า x y และพื้นที่ของเอลิเมนต์ไปคำนวณหาค่าในแต่ละตำแหน่งของเมตริกซ์ K ของแต่ละเอลิเมนต์

ขั้นตอนถัดไป คือ นำเมตริกซ์ของแต่ละเอลิเมนต์มารวมกันเป็นเมตริกซ์ K ของระบบ นอกจากนี้ยังมีเมตริกซ์อีกประเภทหนึ่ง คือ เมตริกซ์ F ซึ่งจะเป็นเมตริกซ์ที่มี 1 คอลัมน์ การรวมเมตริกซ์ F ของระบบจะขึ้นอยู่กับรูปแบบของสมการปัวส์ซองที่เลือกไว้

จากนั้นเป็นการกำหนดค่าขอบ ซึ่งในโปรแกรมนี้จะใช้การกำหนดค่าขอบแบบ Dirichlet ซึ่งเป็นการกำหนดค่าขอบเป็นค่าคงที่ เมื่อกำหนดค่าขอบแล้ว จะเข้าสู่ขั้นตอนของการหาผลเฉลยด้วยวิธีการของจาโคบี ซึ่งหาจากเมตริกซ์ K และเมตริกซ์ F ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

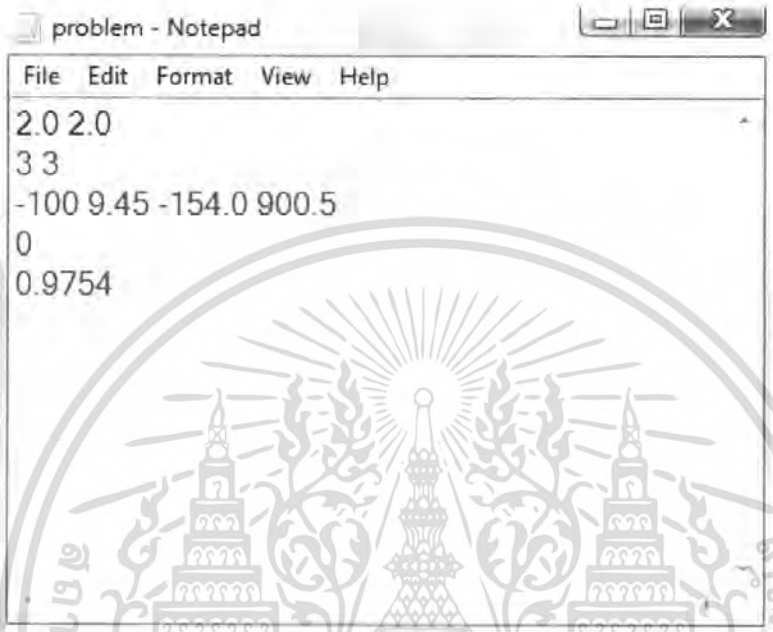
3.3 ขั้นตอนการทำงานของโปรแกรม

ภาพที่ 3.4 แสดงลำดับขั้นตอนการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ขั้นตอนหลักจะเหมือนกันทั้งโปรแกรมแบบลำดับขั้นและโปรแกรมแบบขนาน เพียงแต่โปรแกรมแบบขนานจะมีการส่งค่าไปยังเครื่องต่างๆด้วย ซึ่งจะอธิบายรายละเอียดในหัวข้อที่ 3.4

เริ่มจากโปรแกรมจะทำการรับค่าจากเทกซ์ไฟล์ (Text File) ซึ่งเป็นไฟล์ที่เป็นอินพุตชื่อว่า problem.text โดยตัวอย่างของไฟล์นี้เป็นดังภาพ



ภาพที่ 3.5 ภาพตัวอย่างของไฟล์ problem.txt

โปรแกรมจะรับต่างๆ ซึ่งจะอยู่ในแต่ละบรรทัดของไฟล์ ดังนี้

1. ค่าความยาวและความกว้างของแผ่นเหล็ก จะอยู่ในบรรทัดที่ 1 จากภาพค่าความยาวและความกว้างจะเท่ากับ 2.0 และ 2.0 ตามลำดับ

2. จำนวนจุดต่อหรือโหนดในแนวแกน x และ y จะอยู่ในบรรทัดที่ 2 จากภาพจำนวนจุดต่อในแนวแกน x และ y เท่ากับ 3 และ 3 ตามลำดับ

3. ค่าขอบของด้านทั้ง 4 ของแผ่นเหล็ก คือ บน ล่าง ซ้าย และขวา จะอยู่ในบรรทัดที่ 3 จากภาพค่าขอบของด้านบน ล่าง ซ้าย และขวา ของแผ่นเหล็กมีค่าเท่ากับ -100, 9.45, -154.0 และ 900.5 ตามลำดับ

4. รูปแบบของสมการปัวส์ซอง โดยการเลือกภาพแบบให้ระบุหมายเลขในบรรทัดที่ 4 ดังนี้

รูปแบบ constant ระบุหมายเลข 0

รูปแบบ x ระบุหมายเลข 1

รูปแบบ y ระบุหมายเลข 2

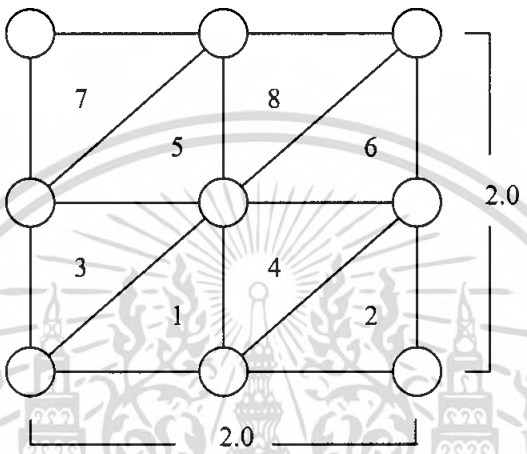
รูปแบบ xy ระบุหมายเลข 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพเป็นการเลือกสมการปัวส์ซองในรูปแบบ constant โดยระบุหมายเลข 0 ในบรรทัดที่ 4

5. ถ้าเลือกสมการปัวส์ซองในรูปแบบ constant จะมีการให้ระบุค่า G ด้วยในบรรทัดที่ 5 จากภาพค่า G คือ 0.9754

หลังจากรับค่าแล้ว จะมีการแบ่งแผ่นเหล็กออกเป็นเอลิเมนต์ตามจำนวนจุดต่อ โดยแต่ละเอลิเมนต์จะมีหมายเลขกำหนดไว้ ดังภาพ



ภาพที่ 3.6 ลักษณะของปัญหาหลังจากรับอินพุตและแบ่งเอลิเมนต์

สำหรับหัวข้อที่ 3.3 นี้ การยกตัวอย่างเพื่ออธิบายประกอบในทุกขั้นตอน จะยึดลักษณะของปัญหาตามภาพที่ 3.6 นี้

จากนั้นจะทำการกำหนดหมายเลขแก่จุดต่อทุกจุด หมายเลขของจุดต่อจะถูกเก็บไว้ในอาร์เรย์ 2 มิติที่ชื่อว่า node[][] โดยมิติที่ 1 จะเก็บหมายเลขของเอลิเมนต์ และมิติที่ 2 จะเก็บค่าที่บอกว่าเป็นจุดต่อที่เท่าใดของเอลิเมนต์นั้น เนื่องจากเอลิเมนต์ที่ใช้เป็น เอลิเมนต์แบบสามเหลี่ยมซึ่งมี 3 มุม ดังนั้นในหนึ่งเอลิเมนต์จะมีตำแหน่งของจุดต่ออยู่ 3 ตำแหน่ง (3 มุม)

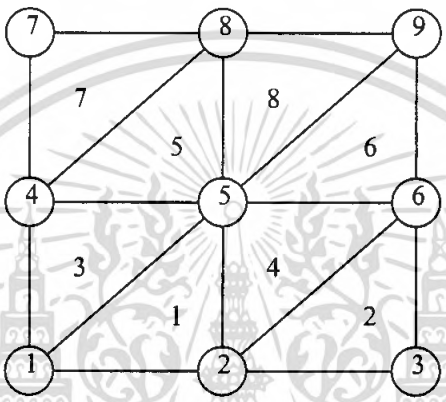
ตัวอย่างเช่นเอลิเมนต์ที่ 1 ประกอบด้วยจุดต่อหมายเลข 1 2 และ 5 สามารถเขียนได้ ดังนี้

```
node[0][0] = 1;
node[0][1] = 2;
node[0][2] = 5;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อสังเกต คือ ค่าในมิติที่ 1 มีค่าเป็น 0 ไม่ใช่ 1 ตามหมายเลขของเอลิเมนต์ เนื่องจากตำแหน่งแรกของอาร์เรย์มีค่าเป็น 0 แต่หมายเลขของเอลิเมนต์แรกมีค่าเป็น 1 ดังนั้นมิติที่ 1 ที่มีค่าเป็น 0 จะอ้างถึงเอลิเมนต์หมายเลข 1 ในทำนองเดียวกันเมื่อมิติที่ 1 มีค่าเป็น 1 จะอ้างถึงเอลิเมนต์หมายเลข 2 เป็นเช่นนี้ไปเรื่อยๆ จนกระทั่งถึงเอลิเมนต์ที่ n มิติที่ 1 จะมีค่าเท่ากับ n-1 สำหรับในมิติที่ 2 เมื่อมีค่าเป็น 0 จะอ้างถึงตำแหน่ง (มุม) ที่ 1 ของเอลิเมนต์ เมื่อมีค่าเป็น 1 จะอ้างถึงตำแหน่งที่ 2 ของเอลิเมนต์ และเมื่อมีค่าเป็น 2 จะอ้างถึงตำแหน่งที่ 3 ของเอลิเมนต์

เมื่อกำหนดหมายเลขให้แก่จุดต่อทุกจุดแล้ว ปัญหาจะมีลักษณะดังภาพ



ภาพที่ 3.7 ลักษณะของปัญหาหลังจากกำหนดหมายเลขให้แก่จุดต่อทุกจุดแล้ว

ต่อไปจะเป็นการกำหนดค่าของตำแหน่ง x และ y ให้แต่ละจุดต่อ โดยที่ค่า x และ y ที่กำหนดให้จะเป็นค่าในจตุภาคที่ 1 คือ เป็นบวกทั้งคู่ โดยที่จุดต่อหมายเลข 1 ซึ่งอยู่ทางมุมซ้ายล่างสุดของปัญหา จะมีค่า x และ y เท่ากับ 0.0 แต่ละจุดต่อจะมีระยะห่างในแนวแกน x และ y เป็นระยะที่เท่ากัน โดยระยะห่างของแต่ละจุดต่อในแนวแกน x มีค่าเท่ากับ

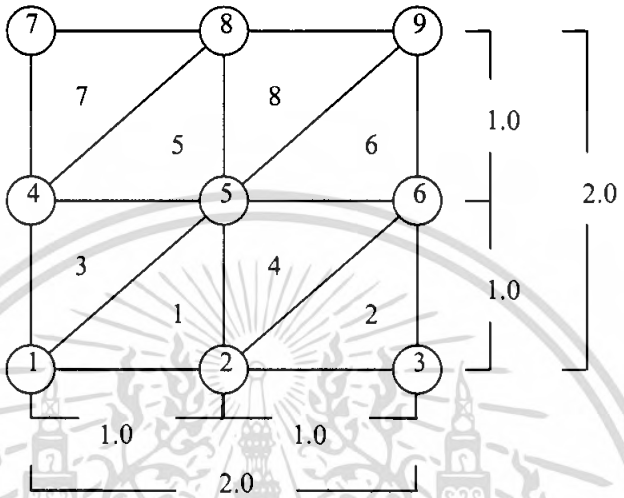
ความยาวของแผ่นเหล็ก/ (จำนวนจุดต่อในแนวแกน x -1)

และระยะห่างของแต่ละจุดต่อในแนวแกน y มีค่าเท่ากับ

ความกว้างของแผ่นเหล็ก/ (จำนวนจุดต่อในแนวแกน y -1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น จากเทกซ์ไฟล์ problem.txt ได้กำหนดความยาวและความกว้างของแผ่นเหล็กเป็น 2.0 และกำหนดจำนวนจุดต่อในแนวแกน x และ y เป็น 3 ดังนั้นระยะห่างระหว่างจุดต่อในแนวแกน x และ y มีค่าเป็น 2.0 ดังภาพ



ภาพที่ 3.8 ระยะห่างระหว่างจุดต่อในแนวแกน x และ y

จากภาพ ค่า x และ y ของจุดต่อที่ 1 คือ 0.0 จุดต่อที่ 2 ห่างจากจุดต่อที่ 1 เป็นระยะทาง 1.0 ในแนวแกน x ดังนั้นค่า x ของจุดต่อที่ 2 คือ 1.0 และสำหรับค่า y ของจุดต่อที่ 2 มีค่าเท่ากับ 0.0 เนื่องจากเมื่อพิจารณาในแนวแกน y แล้ว จุดต่อที่ 2 อยู่ในแนวกันกับจุดต่อที่ 1 ซึ่งมีค่า y เท่ากับ 0.0 เหมือนกัน และจุดต่อที่ 3 ห่างจากจุดต่อที่ 1 เป็นระยะทาง 2.0 ในแนวแกน x และสำหรับค่า y ของจุดต่อที่ 3 ก็เท่ากับ 0.0 เช่นกัน

ณ จุดต่อที่ 4 ถ้าพิจารณาในแนวแกน x จะเห็นได้ว่าอยู่แนวเดียวกันกับจุดต่อที่ 1 คือ 0.0 นั่นคือค่า x เท่ากับ 0.0 แต่เมื่อพิจารณาในแนวแกน y จะเห็นได้ว่า จุดต่อที่ 4 อยู่ห่างจากจุดต่อที่ 1 เป็นระยะทาง 1.0 ดังนั้นค่า y ของจุดต่อที่ 4 จึงมีค่าเท่ากับ 1.0 และสำหรับจุดต่ออื่นๆ ก็ใช้การพิจารณาในลักษณะเช่นนี้

เมื่อทำการกำหนดค่า x และ y ให้แก่จุดต่อทุกจุดแล้ว ต่อไปจะเป็นการสร้างเมตริกซ์ K ของแต่ละเอลิเมนต์ โดยจะอยู่ในรูปของอาร์เรย์ชื่อว่า kMatrix[][] ซึ่งจะเป็นเมตริกซ์ขนาด 3x3 ในขั้นตอนนี้จะเริ่มจากนำค่า x และ y ของจุดต่อทั้ง 3 จุดในเอลิเมนต์นั้น มาทำการคำนวณเพื่อให้ได้พื้นที่ของเอลิเมนต์นั้น จากนั้นนำค่า x y และ area มาทำการคำนวณค่าของ kMatrix[][] ของเอลิเมนต์นั้น โดยมีวิธีการคำนวณ ดังนี้

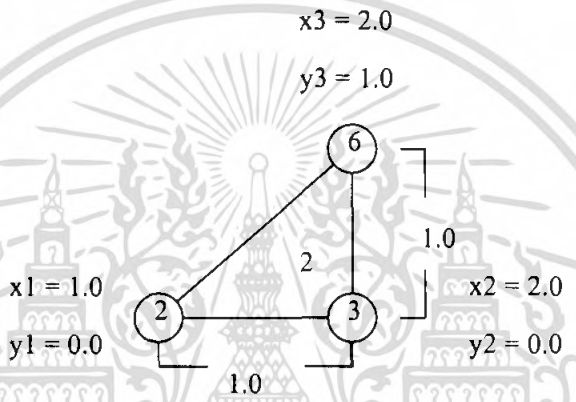
```
area = (float)(0.5*(x2*y3+x1*y2+x3*y1-x2*y1-x1*y3-x3*y2));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} 0.50 & -0.50 & 0.00 \\ -0.50 & 1.00 & -0.50 \\ 0.00 & -0.50 & 0.50 \end{bmatrix}$$

ภาพที่ 3.10 kMatrix[][] ของเอลิเมนต์ที่ 1

เอลิเมนต์ที่ 2 มีลักษณะดังภาพ



ภาพที่ 3.11 ลักษณะของเอลิเมนต์ที่ 2

kMatrix[][] ของเอลิเมนต์ที่ 2 เป็นดังภาพ

$$\begin{bmatrix} 0.50 & -0.50 & 0.00 \\ -0.50 & 1.00 & -0.50 \\ 0.00 & -0.50 & 0.50 \end{bmatrix}$$

ภาพที่ 3.12 kMatrix[][] ของเอลิเมนต์ที่ 2

ส่วนเอลิเมนต์ที่ 3 มีลักษณะดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
kMatrix[0][0] = ((x3-x2)*(x3-x2)+(y2-y3)*(y2-y3))/(4*area);
kMatrix[0][1] = ((x3-x2)*(x1-x3)+(y2-y3)*(y3-y1))/(4*area);
kMatrix[0][2] = ((x3-x2)*(x2-x1)+(y2-y3)*(y1-y2))/(4*area);
kMatrix[1][0] = kMatrix[0][1];
kMatrix[1][1] = ((x1-x3)*(x1-x3)+(y3-y1)*(y3-y1))/(4*area);
kMatrix[1][2] = ((x1-x3)*(x2-x1)+(y3-y1)*(y1-y2))/(4*area);
kMatrix[2][0] = kMatrix[0][2];
kMatrix[2][1] = kMatrix[1][2];
kMatrix[2][2] = ((x2-x1)*(x2-x1)+(y1-y2)*(y1-y2))/(4*area);
```

โดย

x1 คือ ค่า x ของจุดต่อที่ 1 ในเอลิเมนต์นั้น

x2 คือ ค่า x ของจุดต่อที่ 2 ในเอลิเมนต์นั้น

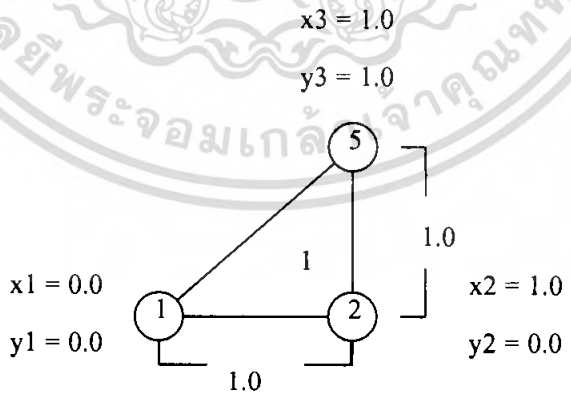
x3 คือ ค่า x ของจุดต่อที่ 3 ในเอลิเมนต์นั้น

y1 คือ ค่า x ของจุดต่อที่ 1 ในเอลิเมนต์นั้น

y2 คือ ค่า x ของจุดต่อที่ 2 ในเอลิเมนต์นั้น

y3 คือ ค่า x ของจุดต่อที่ 3 ในเอลิเมนต์นั้น

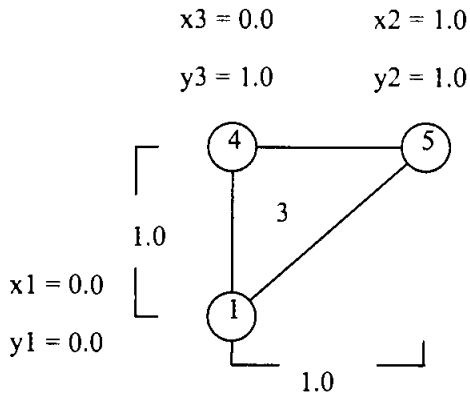
ตัวอย่างเช่น ในเอลิเมนต์ที่ 1 จะมีลักษณะเป็นดังภาพ



ภาพที่ 3.9 ลักษณะของเอลิเมนต์ที่ 1

จากวิธีการคำนวณ kMatrix[][] ที่ได้จากเอลิเมนต์นี้ เป็นดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.13 ลักษณะของเอลิเมนต์ที่ 3

และ kMatrix[][] ของเอลิเมนต์ที่ 3 เป็นดังภาพ

$$\begin{bmatrix} 0.50 & 0.00 & -0.50 \\ 0.00 & 0.50 & -0.50 \\ -0.50 & -0.50 & 1.00 \end{bmatrix}$$

ภาพที่ 3.14 kMatrix[][] ของเอลิเมนต์ที่ 3

สำหรับเอลิเมนต์อื่นๆ ให้หาค่า x y เพื่อนำมาคำนวณหา area แล้วนำค่า x y และ area มาทำการคำนวณหา kMatrix[][] ตามวิธีการที่ได้กล่าวถึง

จากนั้นจะทำการนำ kMatrix[][] ของแต่ละเอลิเมนต์ มารวมกันเป็นเมตริกซ์ K ของระบบ โดยจะอยู่ในรูปของอาร์เรย์ชื่อว่า kSys[][] ซึ่งมีขนาดเท่ากับ

จำนวนของจุดต่อทั้งหมด x จำนวนของจุดต่อทั้งหมด

เช่น จากตัวอย่างของเทกซ์ไฟล์ problem.txt ได้กำหนดจำนวนจุดต่อในแนวแกน x คือ 3 และจำนวนจุดต่อในแนวแกน y คือ 3 จำนวนของจุดต่อทั้งหมด คือ 3x3=9 ดังนั้นขนาดของ kSys[][] คือ 9x9 เป็นต้น

ในโปรแกรมจะกำหนดค่าเริ่มต้นในทุกตำแหน่งของ kSys[][] ไว้เท่ากับ 0.0 แล้วจะนำค่าใน kMatrix[][] ของแต่ละเอลิเมนต์มารวมกันใน kSys[][]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลขของแต่ละจุดต่อของเอลิเมนต์จะมีความสัมพันธ์กับตำแหน่งใน kMatrix[][] ยกตัวอย่างเอลิเมนต์ที่ 1 ซึ่งมีหมายเลขจุดต่อ คือ 1 2 และ 5 จะมี kMatrix[][] ดังภาพ

	1	2	5
1	0.50	- 0.50	0.00
2	0.00	1.00	- 0.50
5	0.00	0.00	0.50

ภาพที่ 3.15 ความสัมพันธ์ระหว่างหมายเลขของจุดต่อกับตำแหน่งใน kMatrix[][]

และตำแหน่งของ kMatrix[][] ของแต่ละเอลิเมนต์จะมีความสัมพันธ์กับตำแหน่งของ kSys[][] เมื่อทำการนำค่าใน kMatrix[][] ของเอลิเมนต์ที่ 1 มาใส่ใน kSys[][] แล้ว kSys[][] จะเป็นดังภาพ

	1	2	3	4	5	6	7	8	9
1	**0.50**	**- 0.50**	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00
2	**- 0.50**	**1.00**	0.00	0.00	**- 0.50**	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	**0.00**	**- 0.50**	0.00	0.00	**0.50**	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

ภาพที่ 3.16 kSys[][] หลังจากนำค่าใน kMatrix[][] ของเอลิเมนต์ที่ 1 มาใส่แล้ว

โดยที่ค่าที่เป็นตัวหนาในภาพ คือ ค่าที่เกิดขึ้นหลังจากที่นำ kMatrix[][] ของเอลิเมนต์ล่าสุดมารวม ซึ่งในที่นี้คือ kMatrix[][] จากเอลิเมนต์ที่ 1

จากนั้นนำ kMatrix[][] ของเอลิเมนต์ที่ 2 มารวม จะได้ kSys[][] ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9
1	0.50	- 0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	- 0.50	**1.50**	**- 0.50**	0.00	- 0.50	**0.00**	0.00	0.00	0.00
3	0.00	**- 0.50**	**1.00**	0.00	0.00	**- 0.50**	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	- 0.50	0.00	0.00	0.50	0.00	0.00	0.00	0.00
6	0.00	**0.00**	**- 0.50**	0.00	0.00	**0.50**	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

ภาพที่ 3.17 kSys[][] หลังจากนำค่าใน kMatrix[][] ของเอลิเมนต์ที่ 2 มารวมแล้ว

และสำหรับ kMatrx[][] ของเอลิเมนต์ที่ 3 เมื่อนำมารวมแล้ว จะได้ kSys[][] ดังภาพ

	1	2	3	4	5	6	7	8	9
1	**1.00**	- 0.50	0.00	**- 0.50**	**0.00**	0.00	0.00	0.00	0.00
2	- 0.50	1.50	- 0.50	0.00	- 0.50	0.00	0.00	0.00	0.00
3	0.00	- 0.50	1.00	0.00	0.00	- 0.50	0.00	0.00	0.00
4	**- 0.50**	0.00	0.00	**1.00**	**0.50**	0.00	0.00	0.00	0.00
5	**0.00**	- 0.50	0.00	**- 0.50**	**0.00**	0.00	0.00	0.00	0.00
6	0.00	0.00	- 0.50	0.00	0.00	0.50	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

ภาพที่ 3.18 kSys[][] หลังจากนำค่าใน kMatrix[][] ของเอลิเมนต์ที่ 3 มารวมแล้ว

เมื่อนำ kMatrix[][] ของแต่ละเอลิเมนต์มารวมกันจนครบแล้ว จะได้ kSys[][] ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9
1	1.00	- 0.50	0.00	- 0.50	0.00	0.00	0.00	0.00	0.00
2	0.00	2.00	- 0.50	0.00	- 1.00	0.00	0.00	0.00	0.00
3	0.00	0.00	1.00	0.00	0.00	- 0.50	0.00	0.00	0.00
4	0.00	0.00	0.00	2.00	- 0.50	0.00	- 0.50	0.00	0.00
5	0.00	0.00	0.00	- 0.50	4.00	- 0.50	0.00	- 1.00	0.00
6	0.00	0.00	0.00	0.00	- 0.50	2.00	0.00	0.00	- 0.50
7	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	- 0.50	2.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	- 0.50	1.00

ภาพที่ 3.19 kSys[][]หลังจากนำค่าใน kMatrix[][] ของแต่ละเอลิเมนต์มารวมกันแล้ว

สำหรับเมตริกซ์ F ของระบบ ซึ่งจะอยู่ในรูปของอาร์เรย์ชื่อว่า fSys[] ซึ่งเป็นเมตริกซ์ที่มีเพียง 1 คอลัมน์ ในโปรแกรมจะกำหนดค่าเริ่มต้นในทุกตำแหน่งของ fSys[] ไว้เท่ากับ 0.0 จากนั้นจะมีการรวมค่าใน fSys[] โดยการรวมค่านั้นขึ้นอยู่กับรูปแบบของสมการปัวส์ซองที่เลือกไว้ในตอนแรก ดังนี้

1. รูปแบบ constant พิจารณาจากค่า G ที่รับมา ซึ่งอยู่ในบรรทัดสุดท้ายของเทกซ์ไฟล์ problem.txt แล้วนำมาคำนวณ ดังนี้

a = ค่า G ที่รับมา * area / 3

แล้วนำค่าที่ได้ (ในที่นี้ขอเรียกว่า a) มาใส่ในเมตริกซ์ ซึ่งอยู่ในรูปอาร์เรย์ชื่อว่า fMatrix[] โดยที่ fMatrix[] เป็นเมตริกซ์ของแต่ละเอลิเมนต์มี 3 แถว 1 คอลัมน์ หมายเลขของจุดต่อในเอลิเมนต์นั้นมีความสัมพันธ์กับตำแหน่งใน fMatrix[] ตัวอย่างเช่นในเอลิเมนต์ที่ 1 จะมี fMatrix[] ดังภาพ

1	a
2	a
5	a

ภาพที่ 3.20 fMatrix[] ของเอลิเมนต์ที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และตำแหน่งใน fMatrix[] จะมีความสัมพันธ์กับตำแหน่งของ fSys[] เมื่อทำการนำค่าใน fMatrix[] ของเอลิเมนต์ที่ 1 มาใส่ใน fSys[] แล้ว fSys[] จะเป็นดังภาพ

1	**a**
2	**a**
3	0.00
4	0.00
5	**a**
6	0.00
7	0.00
8	0.00
9	0.00

ภาพที่ 3.21 fSys[] หลังจากนำค่าใน fMatrix[] ของเอลิเมนต์ที่ 1 มาใส่แล้ว

สำหรับเอลิเมนต์ที่ 2 fMatrix[] จะมีค่าเหมือนกับ fMatrix[] ของเอลิเมนต์ที่ 1 เพียงแต่ตำแหน่งการรวมลงใน fSys[] จะเปลี่ยนไป ดังภาพ

1	a
2	**2a**
3	**a**
4	0.00
5	a
6	**a**
7	0.00
8	0.00
9	0.00

ภาพที่ 3.22 fSys[] หลังจากนำค่าใน fMatrix[] ของเอลิเมนต์ที่ 2 มารวมแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเอลิเมนต์อื่นๆให้นำ fMatrix[] ของแต่ละเอลิเมนต์ซึ่งมีค่าเท่ากันอยู่แล้ว มารวมกันตามหลักการที่ได้ยกตัวอย่างไป จนได้ fSys[] ดังภาพ

1	2a
2	3a
3	a
4	3a
5	6a
6	3a
7	a
8	3a
9	2a

ภาพที่ 3.23 fSys[] หลังจากนำค่าใน fMatrix[] ของทุกเอลิเมนต์มารวมกันแล้ว

2. รูปแบบ x จะทำการนำค่า x1, x2 และ x3 ของแต่ละเอลิเมนต์มารวมกันใน fSys[] ตัวอย่างเช่น จากภาพที่ 3.9 ซึ่งแสดงค่า x1, x2 และ x3 ของจุดต่อหมายเลข 1 2 และ 5 ในเอลิเมนต์ที่ 1 การรวมค่า x1, x2 และ x3 ของเอลิเมนต์ที่ 1 ลงใน fSys[] จะเป็นดังภาพ

1	**0.00**
2	**1.00**
3	0.00
4	0.00
5	**1.00**
6	0.00
7	0.00
8	0.00
9	0.00

ภาพที่ 3.24 fSys[] หลังจากนำค่า x1, x2 และx3 ของเอลิเมนต์ที่ 1 มาใส่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเอลิเมนต์ที่ 2 เมื่อนำค่า x1, x2 และ x3 มารวมใน fSys[] แล้ว fSys[] จะเป็นดังภาพ

1	0.00
2	**2.00**
3	**2.00**
4	0.00
5	1.00
6	**2.00**
7	0.00
8	0.00
9	0.00

ภาพที่ 3.25 fSys[] หลังจากนำค่า x1, x2 และx3 ของเอลิเมนต์ที่ 2 มารวมแล้ว

และเมื่อนำค่า x1, x2 และ x3 ของเอลิเมนต์ที่เหลือ มารวมใน fSys[] แล้ว fSys[] จะเป็นดังภาพ

1	0.00
2	3.00
3	2.00
4	0.00
5	6.00
6	6.00
7	0.00
8	3.00
9	4.00

ภาพที่ 3.26 fSys[] หลังจากนำค่า x1, x2 และx3 ของเอลิเมนต์ที่เหลือมารวมแล้ว

3. รูปแบบ y รูปแบบนี้คล้ายกับรูปแบบ x ต่างกันตรงที่จะใช้ค่า y1 y2 และ y3 แทน ตัวอย่างจาก ภาพที่ 3.9 เมื่อนำค่า y1, y2 และ y3 มารวมใน fSys[] แล้ว จะได้ผลดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1	**0.00**
2	**0.00**
3	0.00
4	0.00
5	**1.00**
6	0.00
7	0.00
8	0.00
9	0.00

ภาพที่ 3.27 fSys[] หลังจากนำค่า y1, y2 และy3ของเอลิเมนต์ที่ 1 มาใส่แล้ว

จากนั้นเมื่อนำค่า y1,y2 และ y3 ของเอลิเมนต์ที่ 2 มารวมใน fSys[] แล้วจะเป็นดังภาพ

1	0.00
2	**0.00**
3	**0.00**
4	0.00
5	1.00
6	**1.00**
7	0.00
8	0.00
9	0.00

ภาพที่ 3.28 fSys[] หลังจากนำค่า y1, y2 และy3ของเอลิเมนต์ที่ 2 มารวมแล้ว

และเมื่อนำค่า y1, y2 และ y3 ของเอลิเมนต์ที่เหลือ มารวมใน fSys[] แล้วจะเป็นดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1	0.00
2	0.00
3	0.00
4	3.00
5	6.00
6	3.00
7	2.00
8	6.00
9	4.00

ภาพที่ 3.29 fSys[] หลังจากนำค่า y1, y2 และy3ของเอลิเมนต์ที่เหลือมารวมแล้ว

4. รูปแบบ xy รูปแบบนี้จะนำค่า x1*y1, x2*y2 และ x3*y3 มารวมใน fSys[] ตัวอย่างจากภาพที่ 3.9 เมื่อนำค่า x1*y1, x2*y2 และ x3*y3 มารวมใน fSys[] แล้ว จะได้ผลดังภาพ

1	**0.00**
2	**0.00**
3	0.00
4	0.00
5	**1.00**
6	0.00
7	0.00
8	0.00
9	0.00

ภาพที่ 3.30 fSys[] หลังจากนำค่า x1*y1, x2*y2 และx3*y3ของเอลิเมนต์ที่ 1 มาใส่แล้ว

ต่อมา ณ เอลิเมนต์ที่ 2 เมื่อนำค่า x1*y1, x2*y2 และ x3*y3 มารวมใน fSys[] แล้ว จะได้ผลดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1	0.00
2	**0.00**
3	**0.00**
4	0.00
5	1.00
6	**2.00**
7	0.00
8	0.00
9	0.00

ภาพที่ 3.31 fSys[] หลังจากนำค่า x1*y1, x2*y2 และx3*y3ของเอลิเมนต์ที่ 2 มารวมแล้ว

และเมื่อนำค่า x1*y1, x2*y2 และ x3*y3 ของเอลิเมนต์ที่เหลือมารวมใน fSys[] แล้ว จะได้ผลดังภาพ

1	0.00
2	0.00
3	0.00
4	0.00
5	6.00
6	6.00
7	0.00
8	6.00
9	8.00

ภาพที่ 3.32 fSys[] หลังจากนำค่า x1*y1, x2*y2 และx3*y3ของเอลิเมนต์ที่เหลือมารวมแล้ว

เมื่อได้เมตริกซ์ K ของระบบ (kSys[][]) และเมตริกซ์ F ของระบบ (fSys[]) แล้วต่อไปจะเป็นการกำหนดค่าขอบเขต (Boundary value) ให้แก่ขอบของแผ่นเหล็กทั้ง 4 ด้าน คือ ด้านบน ด้านล่าง ด้านซ้าย และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านขวา ซึ่งในโปรแกรมนี้จะใช้การกำหนดค่าขอบแบบ Dirichlet ซึ่งเป็นการกำหนดค่าขอบเป็นค่าคงที่ การกำหนดค่าจะกำหนดค่าลงไปที่ จุดต่อต่างๆที่อยู่บนขอบของแผ่นเหล็กทั้ง 4 ด้านนี้

ในการกำหนดค่าขอบเขตนั้น เริ่มจากจะกำหนดค่าขอบเขตของขอบด้านบนและขอบด้านล่างของแผ่นเหล็กก่อน โดยจะทำไปพร้อมกัน จากภาพที่ 3.7 จุดที่อยู่บนขอบด้านบน ได้แก่ จุดต่อหมายเลข 7 8 และ 9 นำหมายเลขของจุดต่อเหล่านี้มาใส่ในอาร์เรย์ชื่อ top[] และนำหมายของเลขจุดต่อที่อยู่บนขอบด้านล่าง ได้แก่ จุดต่อหมายเลข 1 2 และ 3 มาใส่ในอาร์เรย์ชื่อ bottom[] จากนั้นกำหนดค่าขอบเขตของขอบด้านซ้ายและขอบด้านขวา นำหมายเลขของจุดต่อที่อยู่บนขอบด้านซ้าย ได้แก่ จุดต่อหมายเลข 1 4 และ 7 มาใส่ในอาร์เรย์ชื่อ left[] และนำหมายเลขของจุดต่อที่อยู่บนขอบด้านขวา ได้แก่ จุดต่อหมายเลข 3 6 และ 9 มาใส่ในอาร์เรย์ชื่อ right[]

จากนั้นจะทำการปรับค่าใน kSys[][] โดยที่ค่าในตำแหน่ง kSys[i][i] จะเท่ากับ 1.00 เมื่อ i คือ ตำแหน่งในอาร์เรย์มีค่าตั้งแต่ 0 ถึง จำนวนจุดต่อทั้งหมด–1 และตำแหน่งอื่นๆซึ่งอยู่ในแถวเดียวกับ i จะมีค่าเท่ากับ 0.00 ดังภาพ

	1	2	3	4	5	6	7	8	9
1	**1.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
2	**0.00**	**1.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
3	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
4	**0.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
5	0.00	0.00	0.00	- 0.50	4.00	- 0.50	0.00	- 1.00	0.00
6	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**	**0.00**
7	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**
8	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**1.00**	**0.00**
9	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**1.00**

ภาพที่ 3.33 kSys[][] หลังจากกำหนดค่าขอบเขตแล้ว

และทำการปรับค่าใน fSys[] โดยหมายเลขของจุดต่อมีความสัมพันธ์กับตำแหน่งใน fSys[] ซึ่งค่าในแต่ละตำแหน่งใน fSys[] จะเปลี่ยนเป็นค่าใดนั้นขึ้นอยู่กับว่าจุดต่อที่สัมพันธ์กับตำแหน่งนั้น อยู่ในขอบด้านใด โดยค่าที่กำหนดให้ใหม่นั้นขึ้นอยู่กับค่าที่รับมาจากอินพุตซึ่งอยู่ในบรรทัดที่ 3 ของเทกซ์ไฟล์ problem.txt เมื่อกำหนดค่าต่างๆให้แก่ fSys[] แทนที่ค่าเก่าแล้ว จะได้ fSys[] ใหม่ ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1	**9.45**
2	**9.45**
3	**9.45**
4	**-154.00**
5	6.00
6	**900.50**
7	**- 100.00**
8	**- 100.00**
9	**- 100.00**

ภาพที่ 3.34 fSys[] หลังจากปรับค่าตามค่าขอบเขตที่รับมาแล้ว

หลังจากที่กำหนดค่าขอบเขตแล้ว ก็จะเข้าสู่ขั้นตอนการหาผลเฉลยโดยใช้วิธีของจาโคบี ระบบสมการที่จะทำการหาผลเฉลย ประกอบด้วย kSys[][] และ fSys[] ที่ได้ทำการปรับค่าแล้วและเมตริกซ์ 1 คอลัมน์ในรูปของอาร์เรย์ x[] เพื่อใช้เก็บตัวที่ไม่ทราบค่า โดยระบบสมการมีลักษณะ ดังภาพ

1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	x1		9.45
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	x2		9.45
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	x3		9.45
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	x4		-154.00
0.00	0.00	0.00	- 0.50	4.00	- 0.50	0.00	- 1.00	0.00	x5	=	6.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	x6		900.50
0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	x7		- 100.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	x8		- 100.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	x9		- 100.00

ภาพที่ 3.35 ลักษณะของระบบสมการที่จะทำการหาผลเฉลย

จากระบบสมการให้แก้ระบบสมการเพื่อหาค่าแต่ละค่าใน x[] การแก้ระบบสมการจะใช้อาร์เรย์ y[] ช่วยในการแก้ระบบสมการ โดยเริ่มแรกจะกำหนดให้ค่าในทุกตำแหน่งของ x[] เท่ากับ 0 และจากนั้นทำการหาค่า y[] เมื่อเข้าสู่ขั้นตอนการหาค่า y[] จะเริ่มนับจำนวนรอบตั้งแต่ 1 ตัวแปรที่ทำหน้าที่เก็บจำนวนรอบมีชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่า count และกำหนดให้ค่าเริ่มต้น y[i] มีค่าเท่ากับ fSys[i] เมื่อ i คือ ตำแหน่งในอาร์เรย์มีค่าตั้งแต่ 0 ถึง จำนวนจุดต่อทั้งหมด–1 เมื่อได้ค่า y[] แล้วให้นำไปทำการคำนวณ ดังนี้

$$| ((y[i]-x[i]) / y[i]) \times 100 |$$

โดยค่าที่คำนวณได้จะถูกนำไปเปรียบเทียบกับค่า tolerance ซึ่งเป็นค่าที่น้อย ในโปรแกรมนี้กำหนดค่าให้เท่ากับ 0.05 หากค่าที่คำนวณได้มีค่ามากกว่า tolerance จะมีการกำหนดให้ค่า x[i] เท่ากับ y[i] แล้วจะเข้าสู่ขั้นตอนการหาค่า y[] อีกครั้ง พร้อมกับเพิ่มค่า count ขึ้นไปอีก 1 แต่ถ้าค่าที่คำนวณได้มีค่าน้อยกว่าหรือเท่ากับค่า tolerance ก็จะมีการกำหนดให้ค่า x[i] เท่ากับ y[i] เช่นกัน แต่ว่าจะไม่เข้าสู่ขั้นตอนการหาค่า y[] แล้ว โปรแกรมจะแสดงค่าของ x[] และค่าของ count แล้วจบการทำงาน

3.4 ส่วนของการรับส่งข้อมูลสำหรับโปรแกรมแบบขนาน

สำหรับโปรแกรมแบบขนาน ในส่วนที่มีการทำงานแบบขนาน ซึ่งจะมีการรับส่งข้อมูลกันระหว่างเครื่องคอมพิวเตอร์ต่างๆที่ใช้ในการประมวลผล โดยต้องมีการเขียนโปรแกรมเพิ่มเติมเพื่อจัดการเกี่ยวกับการรับส่งข้อมูล ซึ่งในโปรแกรมนี้จะมีส่วนที่ทำงานแบบขนานอยู่ 2 ส่วน คือ

1. ส่วนที่ทำการรวมเมตริกซ์ K ของแต่ละเอลิเมนต์ (kMatrix[][]) เข้าด้วยกัน เป็นเมตริกซ์ K ของระบบ (kSys[][])
2. ส่วนที่ทำการหาผลเฉลยด้วยวิธีการของจาโคบี

ซึ่งทั้ง 2 ส่วนนี้จำเป็นต้องมีการเขียนโปรแกรมเพื่อกำหนดให้ว่าจะส่งข้อมูลอะไร ขนาดเท่าใด ส่งไปยังเครื่องไหน เป็นต้น และต้องกำหนดว่าจะให้มีการรับข้อมูลอย่างไรด้วย ในหัวข้อที่ 3.4 นี้จะอธิบายในแต่ละส่วน ดังนี้

3.4.1 ส่วนที่ทำการรวมเมตริกซ์ K ของแต่ละเอลิเมนต์เข้าด้วยกัน เป็นเมตริกซ์ K ของระบบ

เครื่องที่ถูกกำหนดให้เป็น MASTER จะทำการแบ่งจำนวนแถวของอาร์เรย์ node[][] ออกเป็นชุดๆ แต่ละชุดประกอบด้วยแถวจำนวนหนึ่ง แถวชุดแรกที่ได้ทำการแบ่งจะเก็บไว้ที่เครื่อง MASTER เอง แถวชุดอื่นๆที่ได้ทำการแบ่งจะส่งไปยังเครื่องที่เป็น WORKER ซึ่งการส่งแถวของอาร์เรย์ node[][] ก็เหมือนกับการส่งเอลิเมนต์นั่นเอง สิ่งที่ส่งไป คือ ค่า offset ซึ่งเป็นค่าที่บอกตำแหน่งแรกของชุดนั้น และค่า rows คือ จำนวนแถวที่ส่งไป จากนั้นแต่ละเครื่องที่เป็น WORKER จะรับแถว (หรืออาจจะมองว่าเป็นเอลิเมนต์ก็ได้) แล้วทำการสร้าง kMatrix[][] ตามจำนวนแถว (เอลิเมนต์) ที่รับมาไว้ในแต่ละเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 ส่วนที่ทำการหาผลเฉลยด้วยวิธีการของจาโคบี

เครื่องที่ถูกกำหนดให้เป็น MASTER จะทำการส่งอาร์เรย์ check[] ซึ่งเป็นอาร์เรย์ที่มีขนาดเท่ากับเครื่องที่ใช้ในการประมวลผล ซึ่งรายละเอียดของอาร์เรย์นี้จะอธิบายต่อไป และเครื่องที่เป็น MASTER จะส่ง x[] โดยจะแบ่ง x[] ออกเป็นชุดตามจำนวนแถว ซึ่งจำนวนแถวของ x[] ที่จะส่งไปยังเครื่องต่างๆ จะเท่ากับจำนวนแถวของ node[][] ที่เคยส่งไปยังแต่ละเครื่อง จากนั้นเครื่องที่เป็น WORKER จะรับ check[] และ x[] แล้วทำการหาผลเฉลยด้วยวิธีการของจาโคบี เมื่อทำการหาครบแล้วในแต่ละรอบ ทุกเครื่องจะทำการกำหนดค่าใน check[] ที่ได้รับ สำหรับแต่ละเครื่องแล้ว หากคำนวณได้มีค่ามากกว่า torelance ค่าใน check[] จะเท่ากับ 1 แต่ถ้ามีค่าน้อยกว่าหรือเท่ากับ tolerance ค่าใน check[] จะเท่ากับ 0 จากนั้นแต่ละเครื่องจะส่งค่า check[] และค่า x[] กลับมาที่ เครื่องที่เป็น MASTER และเครื่องที่เป็น MASTER จะทำการรวมค่าใน check[] ที่ได้รับมาจากแต่ละเครื่อง ถ้ารวมกันได้เท่ากับ 0 แสดงว่าการหาผลเฉลยเสร็จสิ้น แต่ถ้าไม่เท่ากับ 0 แสดงว่ามีอย่างน้อย 1 เครื่องที่ต้องทำใหม่ เครื่องที่เป็น MASTER จะทำการตรวจสอบว่าค่าใน check[] นี้มาจากเครื่องใด จากนั้นส่งค่า check[] และค่า x[] ของเครื่องนั้นกลับไปยังเครื่องนั้น เพื่อทำการหาผลเฉลยรอบต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและประเมินผล

4.1 การทดลองเพื่อทดสอบความถูกต้องของผลลัพธ์

การทดลองในข้อ 4.1 นี้ จะมีการกำหนดให้ปัญหามีจำนวนจุดต่อเท่ากับ 11x11 ค่า G = 0.1 และค่า tolerance = 0.05 จุดประสงค์ของการทดลองในข้อนี้ คือ เพื่อทดสอบความถูกต้องของผลเฉลยที่ได้จากโปรแกรม ขั้นแรกจะทำการหาผลเฉลยแท้จริง (Exact) ตั้งแต่ x[0] ถึง x[10] ซึ่งเป็นดังนี้

x[0]	1.000000
x[1]	0.450000
x[2]	0.000000
x[3]	-0.350000
x[4]	-0.600000
x[5]	-0.750000
x[6]	-0.800000
x[7]	-0.750000
x[8]	-0.600000
x[9]	-0.350000
x[10]	0.000000

ขั้นต่อไปจะทำการหาผลเฉลยจากโปรแกรม ซึ่งผลที่ได้เป็นดังนี้ (แสดงตั้งแต่ x[0] ถึง x[10])

```
x[0] 1.000000
x[1] 1.433056
x[2] 1.746233
x[3] 1.940540
x[4] 2.016779
x[5] 1.975467
x[6] 1.816779
x[7] 1.540540
x[8] 1.146232
x[9] 0.633056
x[10] 0.000000
```

ภาพที่ 4.1 ผลเฉลยที่ได้จากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำผลเฉลยที่ได้จากโปรแกรมมาทำการคำนวณหา % ความคลาดเคลื่อนจากวิธีการหาผลเฉลยแท้จริง

$$\left(\frac{|\text{Exact} - \text{ค่าที่ได้}|}{|\text{Exact}|} \right) \times 100$$

ซึ่ง % ความคลาดเคลื่อนเมื่อเทียบผลเฉลยที่ได้จากโปรแกรมกับผลเฉลยแท้จริง เป็นดังนี้

x[0] = 0 %

x[1] = 0.022 %

x[2] ตัวหารเป็น 0 (Exact = 0)

x[3] = 0.065 %

x[4] = 0.043 %

x[5] = 0.036 %

x[6] = 0.033 %

x[7] = 0.030 %

x[8] = 0.029 %

x[9] = 0.028 %

x[10] ตัวหารเป็น 0

จาก % ความคลาดเคลื่อนที่ได้ จะเห็นได้ว่ามี % ความคลาดเคลื่อนน้อยมาก ดังนั้นสามารถใช้โปรแกรมนี้หาผลเฉลยได้

4.2 การทดลองเพื่อวัดเวลาที่ใช้ในการประมวลผล

การทดลองในข้อที่ 4.2 นี้ จะมีการกำหนดให้ปัญหามีจำนวนจุดต่อเท่ากับ 50x50, 100x100 และ 150x150 ตามลำดับ สำหรับค่า G และ tolerance กำหนดให้คงที่ คือ 0.5 และ 0.05 ตามลำดับ

จุดประสงค์ของการทดลองในข้อนี้ คือ เพื่อวัดเวลาที่ใช้ในการประมวลผลของโปรแกรมแบบลำดับขั้นและโปรแกรมแบบขนาน โดยจะมีการให้ระบุจำนวนโปรเซส (Process) ที่ใช้ในการประมวลผล ซึ่งคอมพิวเตอร์ที่ใช้ในการประมวลผล 1 เครื่อง สามารถที่จะมีได้หลายโปรเซส แต่ในการทดลองนี้กำหนดให้คอมพิวเตอร์ 1 เครื่องมีได้ 1 โปรเซส นั่นคือ จำนวนโปรเซสที่สั่ง คือ จำนวนคอมพิวเตอร์ที่ใช้ในการประมวลผลนั่นเอง สำหรับโปรแกรมแบบลำดับขั้นจะสั่งให้ทำงาน 1 โปรเซส และสำหรับโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบขนานจะสั่งตั้งแต่ 2 โปรเซสขึ้นไป แต่ในการทดลองนี้จะสั่งให้มีการทำงาน 2, 4, 6, 8, 9 และ 10 โปรเซสตามลำดับ

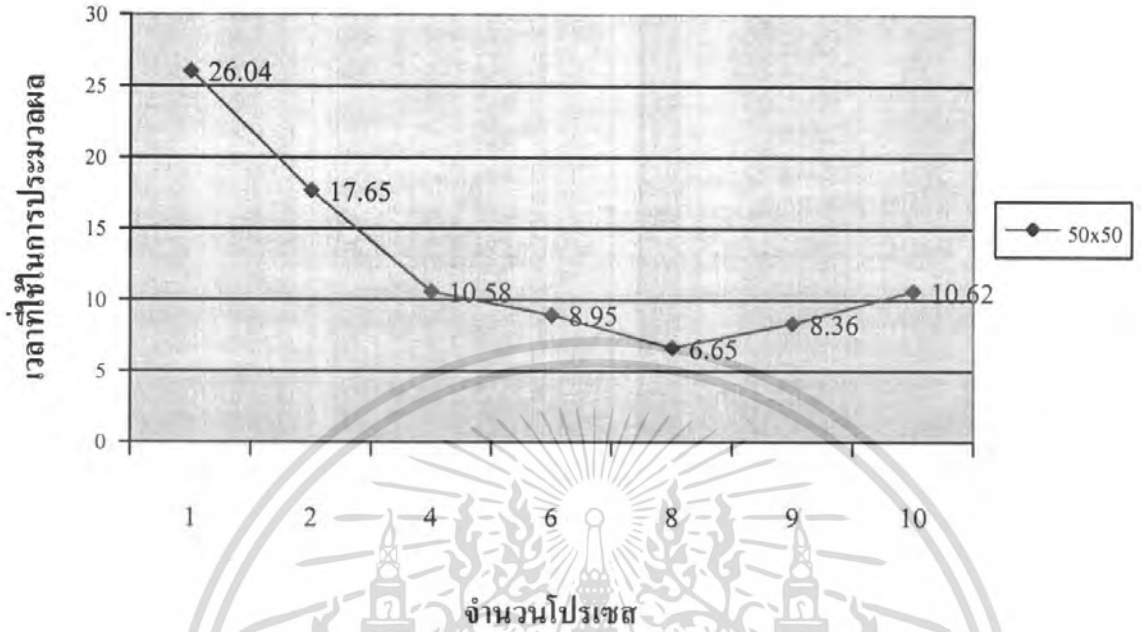
ผลการทดลองในข้อนี้เป็นดังที่แสดงไว้ในตาราง

ขนาดของปัญหา (จำนวนจุดต่อ)	จำนวนโปรเซส (เท่ากับจำนวนเครื่อง)	เวลาที่ใช้ในการประมวลผล (วินาที)
50 x 50	1	26.04
	2	17.65
	4	10.58
	6	8.95
	8	6.65
	9	8.36
	10	10.62
100 x 100	1	3202.07
	2	1419.16
	4	935.98
	6	578.13
	8	420.26
	9	372.03
	10	375.31
150 x 150	1	16301.09
	2	10964.77
	4	5079.40
	6	4726.18
	8	3220.75
	9	2175.69
	10	3050.33

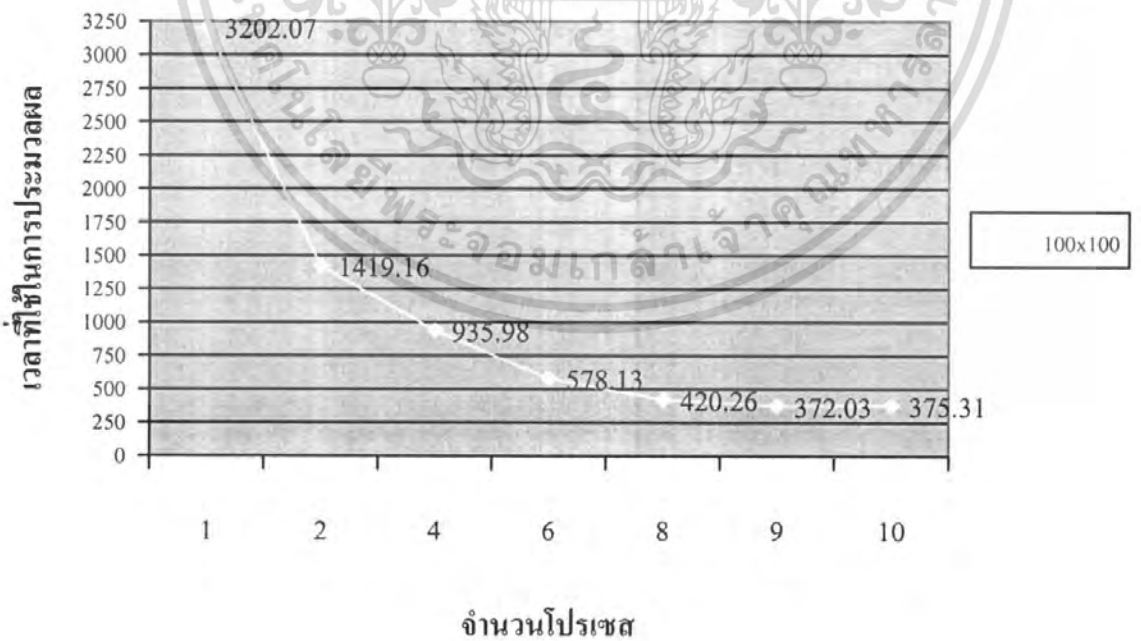
ตารางที่ 4.1 ผลการทดลองวัดเวลาที่ใช้ในการประมวลผลของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางนำเวลาที่วัดได้มาเขียนเป็นกราฟต่างๆ โดยกราฟแรกที่แสดง คือ กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับเวลาที่ใช้ในการประมวลผลของโปรแกรมสำหรับปัญหาขนาดต่างๆ ดังภาพ

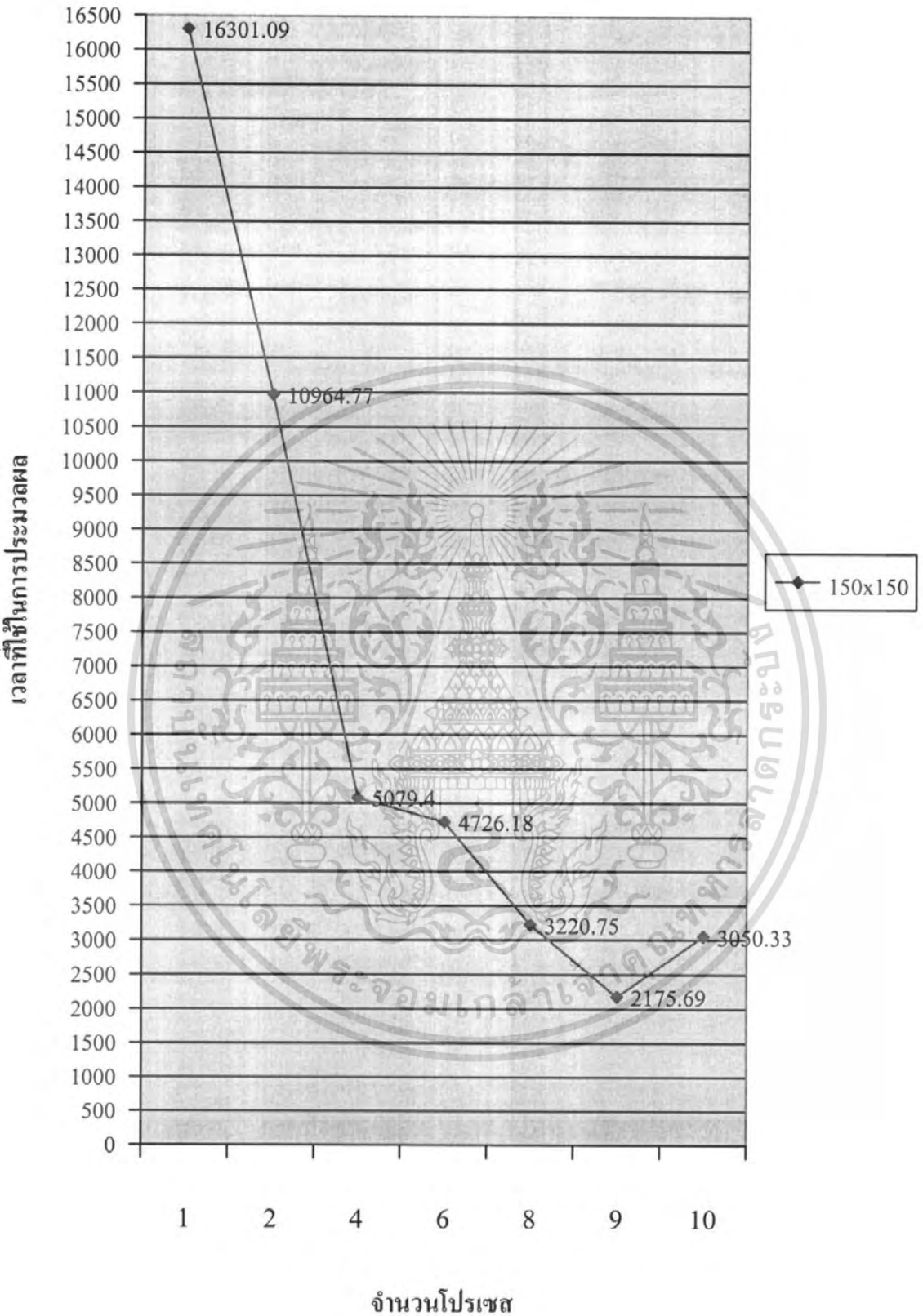


ภาพที่ 4.2 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับเวลาที่ใช้ในการประมวลผลของโปรแกรมสำหรับปัญหาขนาด 50x50



ภาพที่ 4.3 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับเวลาที่ใช้ในการประมวลผลของโปรแกรมสำหรับปัญหาขนาด 100x100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



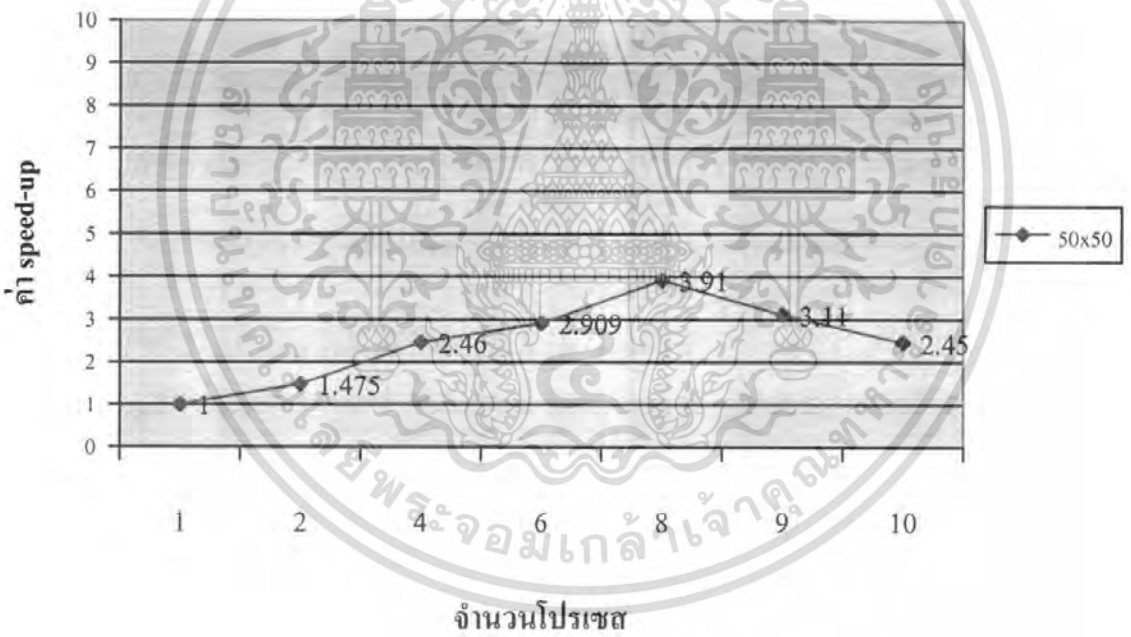
ภาพที่ 4.4 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับเวลาที่ใช้ในการประมวลผลของโปรแกรมสำหรับปัญหาขนาด 150x150

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับเวลาที่ใช้ในการประมวลผลของโปรแกรมทั้ง 3 กราฟ จะเห็นได้ว่าเมื่อมีการระบุจำนวนโปรเซสที่มากขึ้นจะทำให้โปรแกรมใช้เวลาในการประมวลผลน้อยลง แต่อย่างไรก็ตามเมื่อระบุจำนวนโปรเซสจนถึงจำนวนหนึ่งแล้ว จะทำให้เวลาที่ใช้ในการประมวลผลลดลงไปจนต่ำที่สุด สำหรับปัญหาในขนาดที่ต่างกันจะมีจำนวนโปรเซสที่ทำให้เวลาลดลงจนต่ำสุดต่างกัน จาจากภาพที่ 4.2 จะเห็นว่าสำหรับปัญหาขนาด 50x50 นั้น จะมีจำนวนโปรเซสที่ทำให้เวลาลดลงจนต่ำสุดเท่ากับ 8 และจากภาพที่ 4.3 และ 4.4 จะเห็นว่าสำหรับปัญหาขนาด 100x100 และ 150x150 นั้น จะมีจำนวนโปรเซสดังกล่าวเท่ากับ 9 ซึ่งหากระบุจำนวนโปรเซสมากกว่านั้นแล้ว จะไม่ช่วยให้เวลาที่ใช้ลดลง แต่กลับทำให้เวลาที่ใช้เพิ่มขึ้น เนื่องจากต้องใช้เวลาในการติดต่อกันระหว่างเครื่องที่ใช้ประมวลผลมากขึ้น

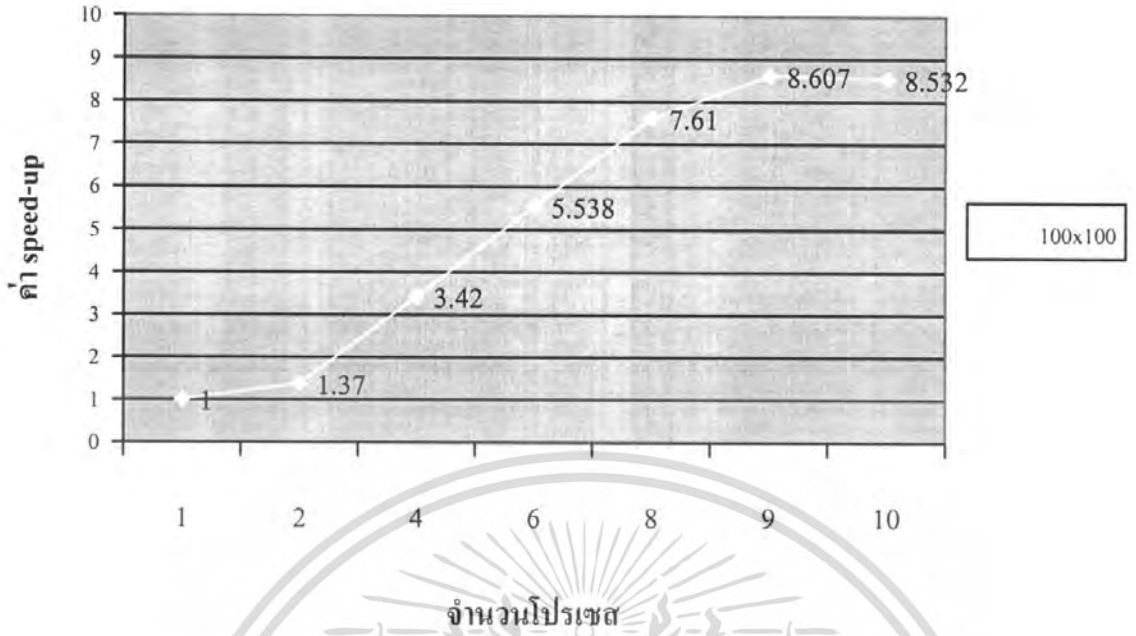
และจากกราฟทั้ง 3 จะเห็นได้ว่าสำหรับปัญหามีขนาดที่ใหญ่ขึ้น เมื่อนำการประมวลผลแบบขนานมาใช้แล้วจะทำให้เวลาที่ใช้ในการประมวลผลลดลงในอัตราที่มากขึ้นด้วย

กราฟที่สอง คือ กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่า Speed up สำหรับปัญหาขนาดต่างๆ ซึ่งมีลักษณะดังภาพ

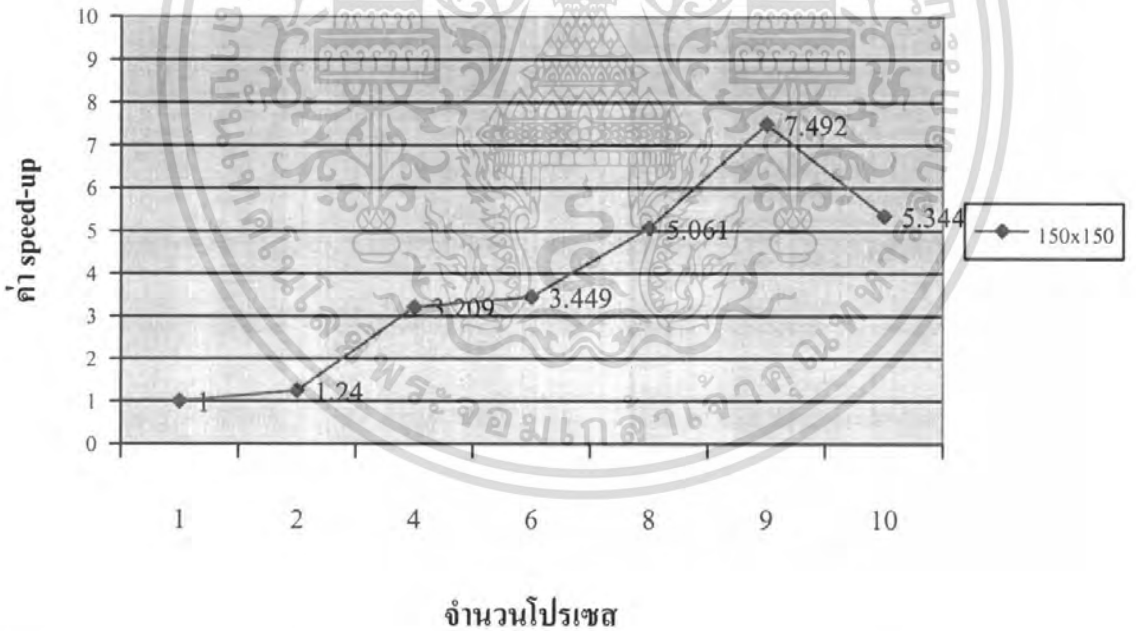


ภาพที่ 4.5 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่า Speed up สำหรับปัญหาขนาด 50x50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.6 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่า Speed up สำหรับปัญหาขนาด 100x100



ภาพที่ 4.7 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่า Speed up สำหรับปัญหาขนาด 150x150

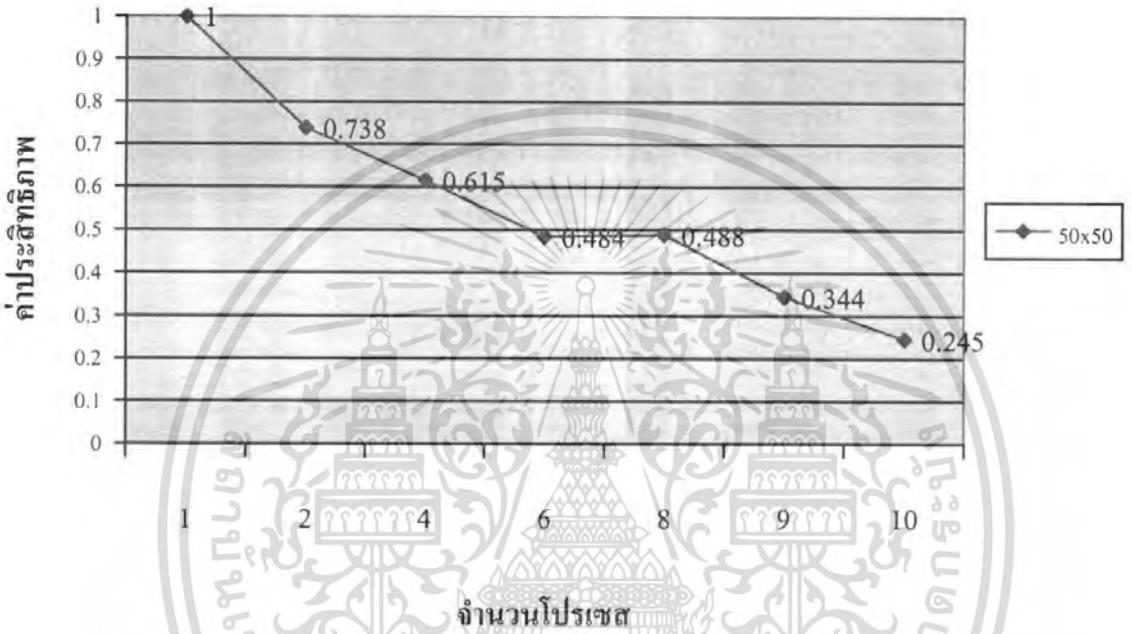
จากกราฟที่สอง จะเห็นได้ว่าเมื่อมีการระบุจำนวนโปรเซสที่มากขึ้น ค่า Speed up จะเพิ่มมากขึ้น แต่อย่างไรก็ตามเมื่อระบุจำนวนโปรเซสจนถึงจำนวนหนึ่งแล้ว จะทำให้ค่า Speed up เพิ่มขึ้นไปจนสูงที่สุด สำหรับปัญหาในขนาดที่ต่างกันจะมีจำนวนโปรเซสที่ทำให้ค่า Speed up เพิ่มขึ้นจนสูงสุดต่างกัน จากภาพที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 จะเห็นว่าสำหรับปัญหาขนาด 50x50 นั้น จะมีจำนวนโปรเซสที่ทำให้ค่า Speed up เพิ่มขึ้นจนสูงสุดเท่ากับ 8 และจากภาพที่ 4.6 และ 4.7 จะเห็นว่าสำหรับปัญหาขนาด 100x100 และ 150x150 นั้น จะมีจำนวนโปรเซสดังกล่าวเท่ากับ 9 ซึ่งหากระบุจำนวนโปรเซสมากกว่านั้นแล้ว จะทำให้ค่า Speed up ลดลง

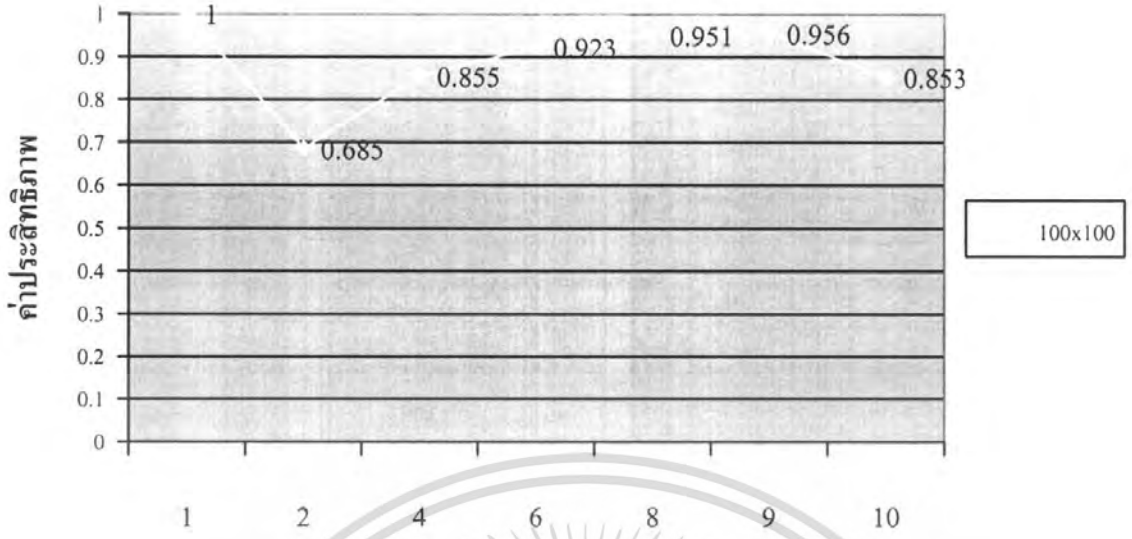
ข้อสังเกต คือ ค่า Speed up จะมีค่าน้อยกว่าหรือเท่ากับจำนวนโปรเซส

และกราฟที่สาม คือ กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับประสิทธิภาพ (Efficiency) สำหรับปัญหาขนาดต่างๆ ซึ่งเป็นดังภาพ

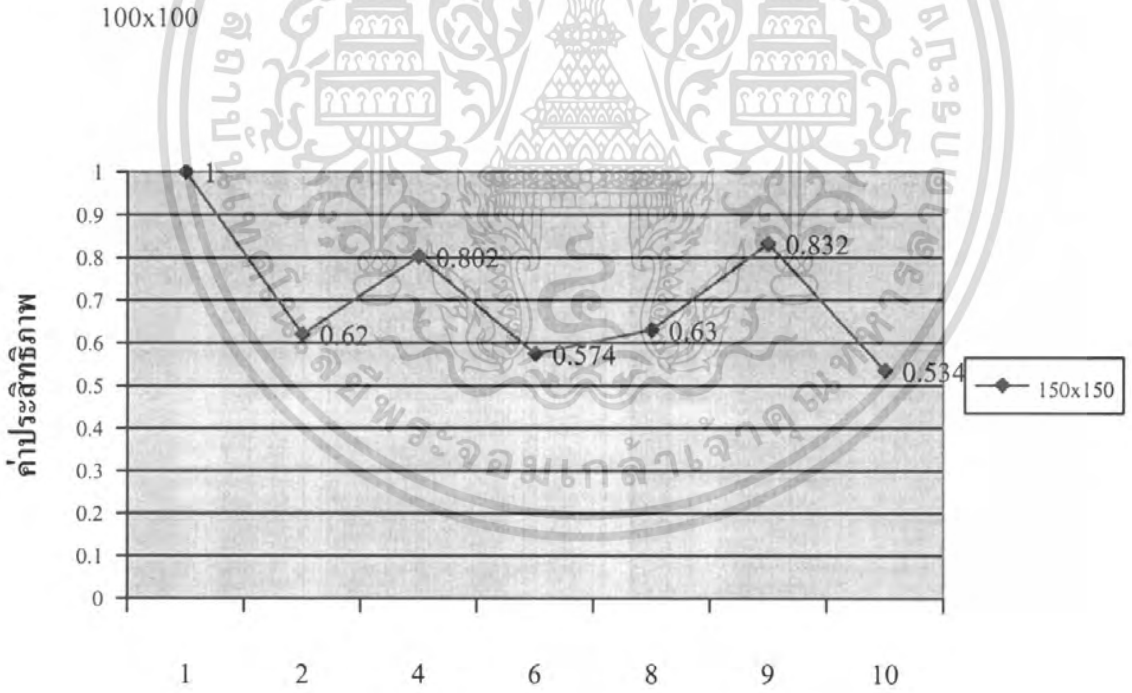


ภาพที่ 4.8 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่าประสิทธิภาพ สำหรับปัญหาขนาด 50x50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.9 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่าประสิทธิภาพ สำหรับปัญหาขนาด 100x100



ภาพที่ 4.10 กราฟแสดงความสัมพันธ์ระหว่างจำนวนโปรเซสกับค่าประสิทธิภาพ สำหรับปัญหาขนาด 150x150

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟที่สาม ข้อสังเกต คือ ค่าประสิทธิภาพจะมีค่ามากที่สุด คือ 1 และเมื่อมีการระบุจำนวนโปรเซสที่มากขึ้น ค่าประสิทธิภาพจะลดลงต่ำกว่า 1 ซึ่งค่าประสิทธิภาพจะมากหรือน้อยขึ้นอยู่กับสัดส่วนระหว่างค่า Speed up กับจำนวนโปรเซส และขึ้นอยู่กับขนาดของปัญหา กล่าวคือ สำหรับปัญหาขนาด 50x50 ถ้าพิจารณาจากกราฟแล้ว จะเห็นได้ว่าเมื่อใช้การประมวลผลแบบขนานโดยระบุจำนวนโปรเซสที่มากขึ้น ค่าประสิทธิภาพจะลดลงอย่างเห็นได้ชัด และสำหรับปัญหาที่มีขนาดใหญ่ขึ้นมา ได้แก่ ปัญหาขนาด 100x100 และ 150x150 จะเห็นว่าค่าประสิทธิภาพจะอยู่ในระดับที่ดีขึ้น (ใกล้เคียงกับ 1 มากขึ้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผล ปัญหาและอุปสรรค ข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

5.1.1 สำหรับเวลาที่ใช้ในการประมวลผล เมื่อนำการประมวลผลแบบขนานมาใช้กับปัญหาที่มีขนาดมากขึ้น จะทำให้เวลาที่ใช้ในการประมวลผลลดลงในอัตราที่มากขึ้น

5.1.2 สำหรับค่า Speed up นั้น เมื่อนำการประมวลผลแบบขนานมาใช้กับปัญหาที่มีขนาดมากขึ้นไม่จำเป็นว่าปัญหาที่มีขนาดมากกว่าจะมีค่า Speed up ที่มากกว่า

5.1.3 สำหรับค่าประสิทธิภาพนั้น เมื่อนำการประมวลผลแบบขนานมาใช้กับปัญหาที่มีขนาดมากขึ้นไม่จำเป็นว่าปัญหาที่มีขนาดมากกว่าจะมีค่าประสิทธิภาพที่มากกว่า

5.1.4 สำหรับปัญหาต่างๆ การระบุจำนวนโปรเซสที่มากขึ้นจะทำให้โปรแกรมประมวลผลได้เร็วขึ้น แต่อย่างไรก็ตามเมื่อระบุจำนวนโปรเซสจนถึงจำนวนหนึ่งแล้ว จะทำให้โปรแกรมประมวลผลได้เร็วที่สุด ซึ่งจำนวนโปรเซสนั้นเป็นจำนวนโปรเซสที่เหมาะสมสำหรับปัญหานั้น ถ้าหากระบุจำนวนโปรเซสมากกว่านั้นแล้วโปรแกรมจะประมวลผลช้าลง เนื่องจากต้องใช้เวลาในการติดต่อกันระหว่างเครื่องที่ใช้ประมวลผลมากขึ้น

5.1.5 สำหรับปัญหาเดียวกัน แต่มีขนาดที่ต่างกัน จะจำนวนโปรเซสที่เหมาะสมต่างกันด้วย จากการทดลอง ปัญหาขนาด 50x50 นั้น จะมีจำนวนโปรเซสที่เหมาะสมเท่ากับ 8 และสำหรับปัญหาขนาด 100x100 และ 150x150 นั้น จะมีจำนวนโปรเซสที่เหมาะสมเท่ากับ 9

5.2 ปัญหาและอุปสรรค

ไม่สามารถทำการประมวลปัญหาที่มีจำนวนจุดต่อมากๆ ได้ เป็นผลมาจากข้อจำกัด 3 ประการ ดังนี้

5.2.1 ข้อจำกัดทางด้านฮาร์ดแวร์ คือ มีพื้นที่ของหน่วยความจำไม่เพียงพอที่จะรองรับการประมวลผลของงานที่มีขนาดใหญ่ได้

5.2.2 ข้อจำกัดของโปรแกรม คือ ต้องมีการจัดการการส่งและรับข้อมูลให้ดี โดยที่ต้องกำหนดให้จำนวนข้อมูลที่ทำการส่งและรับมีขนาดเท่ากัน

5.2.3 ข้อจำกัดทางด้านคอมพิวเตอร์เน็ตเวิร์ก คือ คอมพิวเตอร์เน็ตเวิร์กมีขีดจำกัดในการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

เราสามารถใช้ระบบคอมพิวเตอร์ Cluster จัดการกับงานที่มีขนาดใหญ่ หรือปริมาณงานที่มีจำนวนมากได้อย่างมีประสิทธิภาพได้ หากเข้าใจหลักการทำงานของ Cluster ดังนั้น ในการเขียนโปรแกรมควรมีความรู้ ความเข้าใจในสถาปัตยกรรมของฮาร์ดแวร์ และระบบปฏิบัติการด้วย เพื่อให้เกิดประสิทธิภาพสูงสุดในตัวโปรแกรม และลดภาระงานในการคำนวณ

หากมีปัญหาทางวิทยาศาสตร์หรือคณิตศาสตร์ปัญหาหนึ่ง ในการเขียนโปรแกรมแบบขนานสำหรับปัญหานั้น จะต้องตรวจสอบดูก่อนว่าปัญหานั้นมีการทำงานที่เป็นอิสระต่อกันหรือไม่ ถ้าปัญหานั้นมีการทำงานที่ไม่เป็นอิสระต่อกัน ก็ไม่ควรที่จะนำปัญหานั้นมาเขียนโปรแกรมแบบขนาน แต่หากปัญหานั้นมีการทำงานที่เป็นอิสระต่อกันแล้ว ก็สามารถนำปัญหานั้นมาเขียนโปรแกรมแบบขนานได้ แต่ก่อนอื่นควรดูว่าจะทำการแบ่งงานสำหรับปัญหานั้นๆได้อย่างไรจึงจะเหมาะสมที่สุด

สำหรับโปรแกรมนี้เขียนขึ้นมาเพื่อช่วยในการแก้ปัญหาสมการปัวส์ซองแบบ 2 มิติโดยระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการประมวลผลแบบขนานแล้วทำการเปรียบเทียบกับการประมวลผลแบบลำดับขั้นเพียงเท่านั้น ซึ่งจะเห็นว่าโปรแกรมนี้ยังไม่มีความยืดหยุ่นไม่สามารถช่วยแก้ปัญหาทางด้านคณิตศาสตร์ได้อย่างหลากหลายเท่าที่ควร ดังนั้นในอนาคตอันใกล้นี้อาจจะสามารถพัฒนาโปรแกรมให้มีความยืดหยุ่นมากขึ้น อาทิเช่น อาจทำให้โปรแกรมสามารถแก้ปัญหาสมการปัวส์ซองแบบ 3 มิติโดยระเบียบวิธีไฟไนต์เอลิเมนต์ หรือ ทำให้โปรแกรมสามารถช่วยในการแก้ปัญหาสมการในรูปแบบอื่นๆได้ เช่น สมการของลาปลาส เป็นต้น หรืออาจจะขยายขอบเขตของการทำงานจากที่ทำระบบคลัสเตอร์ (Cluster System) ไปเป็นการทำงานบนระบบกริดซึ่งใหญ่กว่าแทนก็เป็นเรื่องที่สามารถทำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอ้างอิง

ปราโมทย์ เดชะอำไพ. 2547. **ไฟไนต์เอลิเมนต์ในงานวิศวกรรม**. พิมพ์ครั้งที่ 3 ปรับปรุงแก้ไข. กรุงเทพฯ : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.

ปราโมทย์ เดชะอำไพ. 2546. **ระเบียบวิธีเชิงตัวเลขในงานวิศวกรรม**. พิมพ์ครั้งที่ 4 ปรับปรุงแก้ไข. กรุงเทพฯ: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.

วรุณ เกียรติดุริยกุล และ นิตินัย วรรณชากร. ระบบควบคุมและจัดการทรัพยากรบนระบบคลัสเตอร์.ปัญหาพิเศษ หลักสูตรวิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2548.

ณีรนุช ลิปิพัฒนากุล,ทศพร คชานุบาล และ สุรีพร องคนิกูล . การประมวลผลแบบกริด . ปัญหาพิเศษ หลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง . 2548 .

ณัฏฐารมย์ เธียรเมธาวุฒิ,วัชรากร ธิติเลิศ และ อิงพร รุ่งโรจน์รัตนกุล . โปรแกรมสำหรับการแก้ปัญหาสมการปัวส์ซองโดยระเบียบวิธีไฟไนต์เอลิเมนต์ . ปัญหาพิเศษ หลักสูตรวิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง . 2549 .

ระบบคลัสเตอร์ : http://espuc.east.spu.ac.th/service/article/content18/index.htm

คำสั่งพื้นฐานของ UNIX : http://www.tisi.go.th/Information/internet/command.html

: http://www.vwin.co.th/document.php?node=4

Wikipedia : Computer Cluster, Available : http://en.wikipedia.org/wiki/Computer_cluster.

Sun Grid Engine : http://gridengine.sunsource.net

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้