

ห้องสมุดคณะเทคโนโลยีสารสนเทศ จอฉ.

วิธีการปรับปรุงพื้นผิวสำหรับโปรแกรมในงานอุตสาหกรรม
The adaptive grid generation method for industrial application

โดย

นายอภิชาติ พลาศิลปมณฑล

รหัส 38626062

อาจารย์ที่ปรึกษา

อาจารย์กวิน สนิธิเพิ่มพูน

Prof. Stanislav S. Makhanov



H001538

วัน เดือน ปี.....	U 7 S.A. 2549
เลขทะเบียน.....	01538
เลขเรียกหนังสือ.....	๑252๖
	2540
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ จอฉ."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2540

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการ	วิธีการปรับปรุงพื้นผิวสำหรับโปรแกรมในงานอุตสาหกรรม
นักศึกษา	นายอภิชาติ พลาศิลป์มงคล
อาจารย์ที่ปรึกษา	อาจารย์ กวิน สนธิเพิ่มพูน Prof. Stanislav S. Makhanov
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
พ.ศ.	2540

บทคัดย่อ

โครงการนี้ศึกษาถึงวิธีการปรับปรุงความคลาดเคลื่อนของการแปลงคำสั่งจากแบบที่ออกไว้ โดยเราสร้างแบบจำลองของการสร้างพื้นผิวโค้ง 3 มิติ แล้วทำการแปลงพื้นผิวโค้ง 3 มิติที่ได้เป็นคำสั่งสำหรับให้เครื่องมือใช้ในการสร้างชิ้นงานจริง โดยเราใช้วิธีการการสร้างพื้นผิวชิ้นงานจริงของเครื่องจักรอุตสาหกรรมอัตโนมัติแบบ 5 แกน

เพื่อให้ได้ประสิทธิภาพสูงสุดในนำเสนอผลงานทั้งหมด เราได้พัฒนาโปรแกรมคอมพิวเตอร์โดยมีลำดับการพัฒนาดังนี้

1. พัฒนาโปรแกรมสำหรับสร้างพื้นผิวโค้ง 3 มิติ เพื่อเป็นต้นแบบของพื้นผิว
2. พัฒนาโปรแกรมสำหรับการแปลงคำสั่งจากพื้นผิวดั้งเดิม เป็นทางเดินของเครื่องมือซึ่งเครื่องจักรจะใช้ในการทำงาน
3. แสดงความคลาดเคลื่อนของการแปลงคำสั่งทั้งหมด กำหนดขอบเขตของพื้นผิวชิ้นงานเพื่อทำการลดความคลาดเคลื่อนที่เกิดขึ้น
4. แสดงผลการปรับปรุงความคลาดเคลื่อนของชิ้นงานที่ได้จากการปรับปรุง

เพื่อให้ได้ประสิทธิภาพสูงสุดของการพัฒนาโปรแกรมเรา เลือกที่จะพัฒนาโปรแกรมภายใต้ระบบการประมวลผลแบบขนาน โดยพัฒนามนเครื่องซูเปอร์คอมพิวเตอร์ SPP 1600 โดยพัฒนาโปรแกรมโดยอาศัยโปรแกรมภาษา C ร่วมกับเครื่องสนับสนุนการแสดงผลของ motif ทำการแสดงผลผ่านระบบ X windows ของเครื่องซูเปอร์คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราได้ทำการสรุปผลการทดลองโดยวัดประสิทธิภาพในด้านของความเร็วในการประมวลผล ที่ได้จากการทำงานบนระบบเครื่องซูเปอร์คอมพิวเตอร์ และความสามารถของอัลกอริทึมที่ใช้ในการปรับปรุงความคลาดเคลื่อนของการแปลงคำสั่ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	The adaptive grid generation method for industrial application
Student	Mr. Apichart Parasilmongkol
Advisor	Mr. Kawin Sonthipaumpoon Prof. Stanislav S. Makhanov
Level of Study	Master of Science in Information Technology
Major	Information Science
Year	1997

ABSTRACT

We present a new approach to tool-path optimization of 5-axis milling robots based on global approximation of the required surface by an actual surface composed from the tool trajectories. The proposed procedure is based on concatenation of inverse kinematics techniques and an elliptic grid generation method initially proposed for computational fluid dynamics applications. The main objective of this study is an implementation of a performance of the method and a development of a user-friendly graphic interface. We demonstrate that the proposed computer package substantially increases a performance of the method. For instance as;

First, develop and implement program for generated uniform surface according to Biparametric cubic surface theory.

Second, present one easy transformation by zigzag method, which that transform the uniform surface to be tool path surface for 5 axis milling machine then initialize this surface to be generated adaptive surface.

Third, study the results error distribution between uniform surface and adaptive surface then try to optimize and adjust them to be the least variance between two surface.

Finally, we present this application on supercomputer and present performance analysis of adaptive grid generation algorithm for minimize error and speed of program process on parallel processing of SPP 1600 supercomputer.

กิตติกรรมประกาศ

กราบขอบคุณบิดา มารดา ของข้าพเจ้า ที่ได้กรุณาเมตตาสนับสนุน ส่งเสริมให้ข้าพเจ้าได้มีโอกาสศึกษาเล่าเรียนจนถึงปัจจุบัน ทั้งในด้านทุนทรัพย์ กำลังใจ และรับฟังปัญหาอย่างอดทน

ขอขอบคุณท่านอาจารย์ กวิน สนธิเพิ่มพูน และ Prof. Stanislav S. Makhanov ซึ่งได้กรุณาเป็นอาจารย์ที่ปรึกษา ถ่ายทอดความรู้อย่างอดทน ติดตามการทำงานของข้าพเจ้าอย่างใกล้ชิด รวมทั้งให้กำลังใจอยู่เสมอ ทำให้ข้าพเจ้าได้รับความรู้มากมายจากการศึกษาในโครงการนี้

ขอขอบคุณท่านคณะกรรมการควบคุมการสอบที่กรุณาตรวจสอบผลงาน ชี้แนะข้อบกพร่องของเนื้อหาในโครงการที่จัดทำอย่างละเอียด ซึ่งข้าพเจ้าได้พยายามนำมาปรับปรุงให้เกิดความชัดเจนในเนื้อหาของรายงานฉบับนี้

ขอขอบคุณเจ้าหน้าที่ทุกท่านและเพื่อนๆ ที่ให้ความร่วมมือ สนับสนุนการทำงาน ติดตามการทำงานของข้าพเจ้า และคอยเตือนถึงกำหนดเวลาต่างๆ ซึ่งข้าพเจ้ามักจะลืมอยู่เสมอ

อภิชาติ พลาศิลป์มงคล

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	1
บทคัดย่อภาษาอังกฤษ	3
กิตติกรรมประกาศ	4
สารบัญ	5
สารบัญตาราง	9
สารบัญภาพ	10
บทที่	
1 บทนำ	12
1.1 ความเป็นมาและความสำคัญของปัญหา	12
1.2 วัตถุประสงค์ของโครงการ	12
1.3 ขั้นตอนการศึกษา	12
1.4 ขอบเขตของการศึกษา	13
1.5 ประโยชน์ที่คาดว่าจะได้รับ	13
2 การสร้างแบบจำลองของพื้นผิว	14
2.1 ตัวแปรสำหรับเส้นโค้ง 3 มิติ	14
2.2 วิธีการสร้างเส้นโค้งของ Bezier	15
2.3 การหาพื้นผิวโค้ง 3 มิติ โดยวิธีการของ Bezier	16
2.4 ตัวอย่างการสร้างแบบจำลอง	19

บทที่	หน้า
3 การหาจุดอ้างอิงของเครื่องจักร	21
3.1 การหารูปทั่วไปของพื้นผิว	21
3.2 เครื่องจักรขนาด 5 แกน	22
3.3 ปัญหาจากค่าความคลาดเคลื่อน	23
3.4 การพิจารณาการเปลี่ยนแปลงจุดอ้างอิงแบบไม่เป็นเส้นตรง	24
3.5 วิธีการคำนวณเส้นทางการเคลื่อนที่ของเครื่องมือ	25
3.6 การกำหนดขอบเขตของคุณภาพชิ้นงาน	26
4 การปรับปรุงพื้นผิวของชิ้นงาน	29
4.1 สมการกำหนดรูปแบบระนาบโค้ง 3 มิติ	29
4.2 การปรับปรุงความคลาดเคลื่อนของระนาบแบบ 2 ทิศทาง	31
4.3 การแก้ปัญหาโดยวิธีการทางคณิตศาสตร์	33
4.4 ขั้นตอนการทำงานเพื่อลดความคลาดเคลื่อนจากการแปลงข้อมูล	35
4.5 การกำหนดขอบเขตของพื้นผิว	35
5 การโปรแกรมบนระบบ Supercomputer	40
5.1 การประมวลผลแบบขนาน (Parallel computing)	40
5.2 โครงสร้างพื้นฐานของเครื่องคอมพิวเตอร์แบบ SIMD และ MIMD	40
5.3 การใช้หน่วยความจำร่วมกัน และการกระจายหน่วยความจำ	41
5.4 การพิจารณาความเป็นอิสระของข้อมูลกับการทำงานแบบขนาน	42
5.5 การใช้หน่วยความจำร่วมกันกับ Message Passing	43
5.6 การกระจายงานและจัดตารางการทำงานให้กับหน่วยประมวลผล	43
5.7 การเขียนโปรแกรมบน Convex Supercomputer	43
5.8 การ Compile โปรแกรม	44

บทที่	หน้า
6 ขั้นตอนการทำงานของโปรแกรม	45
6.1 ขอบเขตของโปรแกรมที่ออกแบบ	45
6.2 ขั้นตอนการใช้งานโปรแกรม	45
6.2.1 โปรแกรมสำหรับการสร้างแบบจำลองของพื้นผิว	46
6.2.1.1 กำหนดค่าตัวแปรควบคุมการสร้างพื้นผิว 3 มิติ Px Py Pz	47
6.2.1.2 กำหนดค่าตัวแปรสำหรับการสร้างพื้นผิว	49
6.2.2 โปรแกรมสร้างพื้นผิวของชิ้นงาน	50
6.2.2.1 การกำหนดค่าตัวแปรสำหรับสร้างทางเดินของเครื่องมือ	51
6.2.2.2 การเลือกทางเดินของเครื่องมือให้เครื่องจักรทำงาน	52
6.2.3 โปรแกรมกำหนดขอบเขตของชิ้นงาน	53
6.2.4 กลุ่มโปรแกรมสำหรับการปรับปรุงความคลาดเคลื่อน	57
7 บทสรุป	59
7.1 การวัดประสิทธิภาพของโปรแกรมบน Super Computer	59
7.2 การใช้งานใน Text Mode (CXP)	59
7.3 การเขียนใช้ Cxp จาก batch mode	60
7.4 การสั่ง run cxp ร่วมกับ nohup	61
7.5 ข้อควรระวังในการใช้โปรแกรม nohup cxp	61
7.6 ตารางสรุปเปรียบเทียบผลการทำงานของโปรแกรมส่วนปรับปรุงพื้นผิว	61
7.7 เปรียบเทียบประสิทธิภาพของโปรแกรม	62
7.8 ประสิทธิภาพของวิธีการแก้ปัญหา	63
7.9 แนวทางการพัฒนา	63
บรรณานุกรม	65
ภาคผนวก ก. รายงานการวัดประสิทธิภาพของรายงานส่วนปรับปรุงพื้นผิว	67
ภาคผนวก ข. โปรแกรมที่ใช้ทดสอบบนเครื่องซูเปอร์คอมพิวเตอร์	72
ภาคผนวก ค. โปรแกรมที่ใช้สร้างพื้นผิวโค้ง 3 มิติ และแปลงพื้นผิว	78
ภาคผนวก ง หลักการเขียนโปรแกรมบนระบบ X windows	85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
7-1 ตารางการวัดประสิทธิภาพการทำงานของโปรแกรมส่วนปรับปรุงพื้นผิว	62
7-2 ตารางเปรียบเทียบความเร็วในการประมวลผลจากการ compile แบบต่างๆ	62
7-3 แสดงประสิทธิภาพในการลดความคลาดเคลื่อน	63



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้า
รูปที่ 2-1 ลักษณะของเส้นโค้ง 3 มิติ	14
รูปที่ 2-2 แสดงผลกระทบของตัวแปร P ที่มีผลต่อสวนโค้ง	
โดยการกำหนดค่าความโค้งจะกำหนดให้กับตัวแปร P0 , P1 , P2 , P3	15
รูปที่ 2-3 รูปแสดงการใช้ตัวแปรสำหรับควบคุมความโค้งของพื้นผิว 3 มิติ	17
รูปที่ 2-4 รูปแสดงตัวอย่างของการสร้างพื้นผิวโค้งเป็นรูปโดม	19
รูปที่ 2-5 รูปแสดงตัวอย่างของการสร้างพื้นผิวโค้งเป็นรูป Sine	20
รูปที่ 3-1 รูปแสดงเครื่องจักรขนาด 5 แกน	22
รูปที่ 3-2 แสดงการตัดของพื้นผิวที่ไม่เป็นเส้นตรง	23
รูปที่ 3-3 แสดงการวิธีการเปลี่ยนแปลงจากพื้นผิวที่สร้างขึ้นเป็น ทางเดินของเครื่องมือ	25
รูปที่ 3-4 รูปแสดงการเคลื่อนที่ของเครื่องมือ ชิ้นงานที่จะได้จากพื้นผิวรูปโดม	25
รูปที่ 3-5 แสดงการเคลื่อนที่ของเครื่องมือจากชิ้นงานรูป Sine	26
รูปที่ 3-6 ความแตกต่างของชิ้นงานที่จะได้กับพื้นผิวที่ออกแบบ	27
รูปที่ 3-7 แสดงตำแหน่งของจุดที่ได้จากการทำงานของเครื่องของพื้นผิวรูปโดม	27
รูปที่ 3-8 แสดงตำแหน่งของจุดที่ได้จากการทำงานของเครื่องของพื้นผิวรูป Sine	28
รูปที่ 4-1 กำหนดของเขตของพื้นผิวเพื่อทดสอบการปรับปรุง	36
รูปที่ 4-2 รูปแสดงผลที่ได้จากการโปรแกรมซึ่งทำหน้าที่ปรับปรุงพื้นผิวงาน	37
รูปที่ 4-3 รูปแสดงการกำหนดของเขตของผิวชิ้นงานที่ต้องการปรับปรุง	37
รูปที่ 4-4 แสดงบริเวณที่มีความผิดพลาดมาก	38
รูปที่ 4-5 แสดงการปรับปรุงพื้นผิวรูปโดมที่ได้จากโปรแกรมปรับปรุงพื้นผิว	38
รูปที่ 5-1 รูปแสดงการแบ่งหน่วยความจำกันใช้ของ CPU	41
รูปที่ 5-2 โครงสร้างการจัดการหน่วยความจำแบบกระจาย	41
รูปที่ 6-1 ขั้นตอนการใช้โปรแกรมในโรงงาน	45
รูปที่ 6-2 ขั้นตอนการสร้างพื้นผิวของชิ้นงาน	46
รูปที่ 6-3 แสดงโปรแกรมกำลังแสดงผลการสร้างโปรแกรม 3 มิติ	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
รูปที่ 6-4 รูปแสดงการรับค่าสำหรับตัวแปรควบคุม Px	47
รูปที่ 6-5 รูปแสดงการรับค่าสำหรับตัวแปรควบคุม Py	48
รูปที่ 6-6 รูปแสดงการรับค่าสำหรับตัวแปรควบคุม Pz	48
รูปที่ 6-7 โปรแกรมส่วนกำหนดขนาดพื้นผิวและการแสดงผล	49
รูปที่ 6-8 แสดงโปรแกรมส่วนกำหนดสีของพื้นผิวบนจอภาพ	49
รูปที่ 6-9 ขั้นตอนการสร้างพื้นผิวของชิ้นงาน	50
รูปที่ 6-10 รูปแสดงการสร้างทางเดินของเครื่องมือสำหรับเครื่องจักร	50
รูปที่ 6-11 แสดงรายละเอียดของโปรแกรมส่วนกำหนดค่าตัวแปรของเครื่องจักร	51
รูปที่ 6-12 รูปแสดงทางเดินของเครื่องมือที่เครื่องจักรแบบ 5 แกนใช้ กำหนดทิศทางตามแกน X	52
รูปที่ 6-13 รูปแสดงทางเดินของเครื่องมือที่เครื่องจักรแบบ 5 แกนใช้ กำหนดทิศทางตามแกน Y	52
รูปที่ 6-14 ขั้นตอนการกำหนดของเขตของพื้นผิวชิ้นงาน	53
รูปที่ 6-15 ชิ้นงานซึ่งออกแบบโดยอาศัยการสร้างเส้นโค้ง 3 มิติ แสดงผลใน 2 มิติ	53
รูปที่ 6-16 รูปแสดงทางเดินของเครื่องมือที่เครื่องจักรขนาด 5 แกนใช้	54
รูปที่ 6-17 รูปแสดงพื้นผิวที่จะได้จากการแปลงพื้นผิวโค้ง 3 มิติ ที่ออกแบบมาให้เครื่องจักรทำงาน	55
รูปที่ 6-18 รูปแสดงการกำหนดค่าสำหรับการปรับปรุงพื้นผิว	55
รูปที่ 6-19 กำหนดของเขตของพื้นผิวที่จะทำการปรับปรุง	56
รูปที่ 6-20 รูปแสดงการโปรแกรมส่วนที่กำหนดสีของพื้นผิวที่จะแสดง	56
รูปที่ 6-21 ขั้นตอนการปรับปรุงพื้นผิว	57
รูปที่ 6-22 รูปแสดงการพื้นผิวที่กำหนดขอบเรียบร้อยแล้ว	57
รูปที่ 6-23 รูปแสดงพื้นผิวที่ปรับปรุงไประยะหนึ่ง	57

บทที่ 1

บทนำ

Introduction

1.1 ความเป็นมาและความสำคัญของปัญหา

ในระบบอุตสาหกรรมเราจะพบว่ามีเครื่องจักรอัตโนมัติเข้ามาใช้ในการออกแบบ และควบคุมการทำงานของเครื่องจักรอุตสาหกรรมอัตโนมัติกันอย่างกว้างขวาง เราพบว่าเกิดปัญหาขึ้นในขั้นตอนของการสร้างทางเดินสำหรับเครื่องมือที่เครื่องจักรอัตโนมัติใช้ในการทำงาน ทำให้เกิดความคลาดเคลื่อนขึ้นบางส่วน ในทางปฏิบัติหากเรานำชุดคำสั่งที่มีความคลาดเคลื่อนสูงไปใช้งานจะทำให้ได้ชิ้นงานที่ด้อยคุณภาพ ทำให้ต้องเสียค่าใช้จ่ายจากความคลาดเคลื่อนต่างๆ ได้แก่ การเสียเวลาในการทำงานของเครื่องจักรมากเกินไป การที่ต้องเสียค่าใช้จ่ายในการซ่อมแซมชิ้นงานที่ผิดแบบ รวมถึงการต้องจัดหาเครื่องจักรเข้ามาการปรับปรุงแก้ไขชิ้นงาน

จากปัญหาดังกล่าวหาวิธีที่ดีในการแก้ไขความคลาดเคลื่อนที่จะเกิดจากการแปลงคำสั่งจะทำให้การทำงานของเครื่องจักรสามารถทำได้ถูกต้องที่สุดในครั้งเดียว

1.2 วัตถุประสงค์ของโครงการ

โครงการนี้จัดทำขึ้นเพื่อศึกษาถึง การลดความคลาดเคลื่อนที่เกิดจากการแปลงแบบของการสร้างพื้นผิวชิ้นงาน ไปเป็นทางเดินของเครื่องมือซึ่งเครื่องจักรอุตสาหกรรมอัตโนมัติประเภท 5 แกนใช้ โดยทำการพัฒนาโปรแกรมประยุกต์เพื่อนำเสนอวิธีการลดความคลาดเคลื่อนในการแปลงสร้างทางเดินของเครื่องมือที่เครื่องจักรใช้

นอกจากนี้ เพื่อให้ได้โปรแกรมประยุกต์ที่มีประสิทธิภาพสูง จึงได้ทำการพัฒนาโปรแกรมบนระบบการประมวลผลแบบขนานบนเครื่องซูเปอร์คอมพิวเตอร์ โดยนำเสนอผลการทดสอบโปรแกรมที่ได้จากการประมวลผลแบบขนานบนเครื่องซูเปอร์คอมพิวเตอร์ SPP 1600

1.3 ขั้นตอนการศึกษา

1.3.1 ศึกษาการสร้างแบบจำลองแทนพื้นผิวของชิ้นงานโดยอาศัยวิธีการของ Bicubic

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.3.2 ศึกษาการกำหนดทางเดินของเครื่องมือสำหรับเครื่องจักรขนาด 5 แกน
- 1.3.3 ศึกษาการเกิดความคลาดเคลื่อนของการแปลงคำสั่ง
- 1.3.4 ศึกษาวิธีการปรับปรุงความคลาดเคลื่อนของการแปลงคำสั่งด้วยวิธีการของ Adaptive Grid Generation
- 1.3.5 ศึกษาการพัฒนาโปรแกรมแบบขนาดในระบบ supercomputer
- 1.3.5 ศึกษาวิธีการวัดประสิทธิภาพของโปรแกรมบนเครื่อง supercomputer
- 1.3.6 ศึกษาการทำงานของโปรแกรมบนระบบ X windows โดยอาศัย Motif tool ช่วยในการพัฒนา

1.4 ขอบเขตของการศึกษา

- 1.4.1 ทำการแทนรูปแบบของพื้นผิวที่ออกแบบโดยสมการสร้างพื้นผิว 3 มิติ โดยสามารถสร้างพื้นผิวที่มีขนาดใหญ่ขึ้นกว่าการใช้งานบนเครื่องคอมพิวเตอร์ส่วนบุคคล
- 1.4.2 ทำการแทนรูปแบบของพื้นผิวที่เครื่องจักรทำงานโดยการใช้สมการควบคุมการทางเดินของเครื่องมือที่เครื่องจักรอุตสาหกรรมอัตโนมัติประเภท 5 แกนใช้
- 1.4.3 แสดงพื้นผิวที่มีความคลาดเคลื่อนจากแปลงคำสั่งผ่านระบบกราฟิกส์
- 1.4.4 แสดงขอบเขตของพื้นผิวที่จะทำการปรับปรุงความผิดพลาด และทำการปรับปรุงความคลาดเคลื่อนโดยอาศัยวิธีการปรับปรุงของ Adaptive Grid Generation Method
- 1.4.5 ทำการพัฒนาโปรแกรมทั้งหมดบนระบบซูเปอร์คอมพิวเตอร์ SPP 1600 โดยมีการแสดงผลกราฟิกส์ผ่านระบบ X windows ของเครื่อง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 แสดงประสิทธิภาพของการพัฒนาโปรแกรมในระบบการประมวลผลแบบขนาน
- 1.5.2 แสดงประสิทธิภาพของโปรแกรมในการลดความคลาดเคลื่อนของการแปลงคำสั่ง
- 1.5.3 ทำให้ผู้ศึกษาวิธีการปรับปรุงความคลาดเคลื่อนที่เกิดจากการกำหนดทางเดินของเครื่องมือสำหรับเครื่องจักรขนาด 5 แกน สามารถเข้าใจได้ง่ายขึ้นเมื่อศึกษาร่วมกับโปรแกรมที่พัฒนาขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การสร้างแบบจำลองของพื้นผิว

Surface Design

2.1 ตัวแปรสำหรับเส้นโค้ง 3 มิติ (Parametric Representation of cubic curve)

ประยุกต์การหาจุดอ้างอิงใหม่จากการเปลี่ยนแปลงเชิงสมการของเส้นโค้ง 3 มิติ ซึ่งมีหลักการในการนำไปประยุกต์ใช้ กล่าวคือ เส้นโค้ง 3 มิติใดๆ คือ จุดซึ่งเคลื่อนที่ไปใน space โดยกำหนดการอ้างอิงจุดใด ๆ เป็นเวกเตอร์ (r) สัมพันธ์กับการเปลี่ยนแปลงของตัวแปร เวลา (t) ในรูปของ

$$X = X(t) \quad (1)$$

$$Y = Y(t) \quad (2)$$

$$Z = Z(t) \quad (3)$$

รูปที่ 2-1 ลักษณะของเส้นโค้ง 3 มิติ

เราสามารถอธิบายเส้นโค้งโดยอาศัยตัวแปร u และ v เมื่อ u, v เป็น unit vector ที่มีค่าระหว่าง 0 ถึง 1 ในความสัมพันธ์ในลักษณะเส้นโค้ง 3 มิติ (cubic curve) ดังนี้

$$X(u) = a_x u^3 + b_x u^2 + c_x u + d_x \quad (4)$$

$$Y(u) = a_y u^3 + b_y u^2 + c_y u + d_y \quad (5)$$

$$Z(u) = a_z u^3 + b_z u^2 + c_z u + d_z \quad (6)$$

โดยที่ $0 \leq u \leq 1$ และแต่ละ X, Y และ Z เป็น cubic polynomial ของตัวแปร u

2.2 วิธีการสร้างเส้นโค้งของ Bezier (Bezier cubic curve)

วิธีการสร้างเส้นโค้งแบบนี้กำหนดว่า เส้นโค้งที่เกิดขึ้นจากสมการนี้จะประกอบด้วย จุดซึ่งทำหน้าที่ในการควบคุมความโค้งของค่าตัวแปร 4 ตัว

เราสามารถหารูปทั่วไปของเส้นโค้ง 3 มิติ (cubic curve) รวมทั้งเซตเฉพาะที่เป็นสมการควบคุม $P(P_0 - P_3)$

$$Q(u) = au^3 + bu^2 + cu + d \quad (7)$$

จากรูปนี้เราสามารถแยกออกได้เป็น

$$Q(u) = au^3 + bu^2 + cu + d \quad (8)$$

จาก

$$(1-u)^3 + 3u(1-u)^2 + 3u^2(1-u) + u^3 \quad (9)$$

สมการโพลิโนเมียลโดยอาศัยสมการควบคุม

$$Q(u) = P_0(1-u)^3 + P_13u(1-u)^2 + P_23u^2(1-u) + P_3u^3 \quad (10)$$

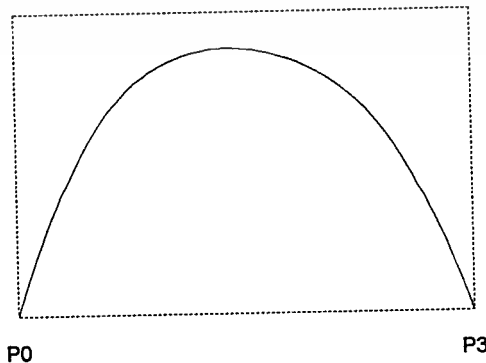
หรือเขียนในรูปทั่วไปของสมการเมตริกซ์

$$Q(u) = UBP \quad (11)$$

$$= [u^3 \ u^2 \ u \ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (12)$$

P1

P2



P0

P3

รูปที่ 2-2 แสดงผลกระทบของตัวแปร P ที่มีผลต่อส่วนโค้ง โดยการกำหนดค่าความโค้ง

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันวิจัยเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถพิจารณาสมการโดยละเอียดใช้โปรแกรม MATLAB ได้ดังนี้

```
p = [ 0.1 0.4 0.4 0.1];
u = 0:1/49:1; % เวกเตอร์ u มีขนาด 50 หน่วย
Q = ones(1,length(u));
for i=1:length(u)
    Q(i) = p(1) * (1 - u(i))^3 + p(2)*3*u(i)*(1-u(i))^2 + p(3)*3*u(i)^2*(1-u(i)) +
    p(4)*u(i)^3;
end

plot(Q);
xlabel('u');
ylabel('Q(u)');
```

ผลที่เกิดจากการกำหนดค่า ให้กับตัวแปร P จาก ตัวอย่างโปรแกรมลักษณะของเส้นโค้ง จะเปลี่ยนไปตามค่าที่กำหนด p(1) ถึง p(4)

2.3 การหาพื้นผิวโค้ง 3 มิติ โดยวิธีของ Bezier (Biparametric cubic surface)

ด้วยวิธีการของ Bezier เราสามารถสร้างพื้นผิวโค้ง 3 มิติ ได้จากการตัดกันของเส้นโค้งที่เกิดจากสมการเส้นโค้ง 3 มิติ (cubic curve) ในข้อ 2.2 เราสามารถนำมาใช้ในการพิจารณาเป็นพื้นผิวโค้ง โดยอาศัยค่าควบคุม P ดังนี้ Bizier patch กำหนดเป็น

$$Q(u,v) = \sum_{i=0}^3 \sum_{j=0}^3 P_{ij} B_i(u) B_j(v) \quad (13)$$

โดยกำหนดให้

$$P(u,v) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} B P B^T \\ v^3 \\ v^2 \\ v \\ 1 \end{bmatrix} \quad (14)$$

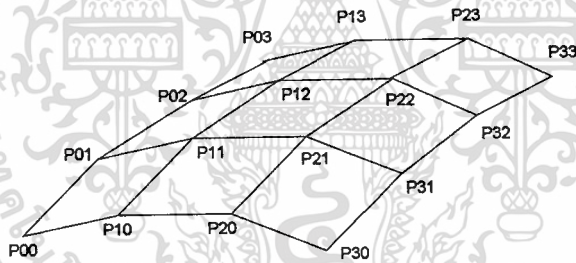
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่

$$B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

ตัวแปรที่ใช้สำหรับควบคุม

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix}$$



รูปที่ 2-3 รูปแสดงการใช้ตัวแปรสำหรับควบคุมความโค้งของพื้นผิว 3 มิติ

ทดสอบสมการโดยใช้โปรแกรม MATLAB สำหรับสร้างพื้นผิวโค้ง 3 มิติ (Biparametric cubic surfaces)

```
% biparametric cubic surfaces
P = [ 0.0  0.3  0.3  0.0,
      0.3  0.6  0.6  0.3,
      0.3  0.6  0.6  0.3,
      0.0  0.3  0.3  0.0];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
B = [-1.0  3.0 -3.0  1.0,
      3.0 -6.0  3.0  0.0,
     -3.0  3.0  0.0  0.0,
      1.0  0.0  0.0  0.0];
```

```
u = 0:1/10:1;
```

```
v = 0:1/10:1;
```

```
Q = ones(length(u),length(v));
```

```
U = ones(1,4);
```

```
V = U';
```

```
for n=1:length(u)
```

```
for x=1:length(v)
```

```
U(1) = u(n)^3;
```

```
U(2) = u(n)^2;
```

```
U(3) = u(n);
```

```
U(4) = 1;
```

```
V(1) = v(x)^3;
```

```
V(2) = v(x)^2;
```

```
V(3) = v(x);
```

```
V(4) = 1;
```

```
Q(n,x) = U*B*P*V;
```

```
end
```

```
end
```

```
mesh(Q);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

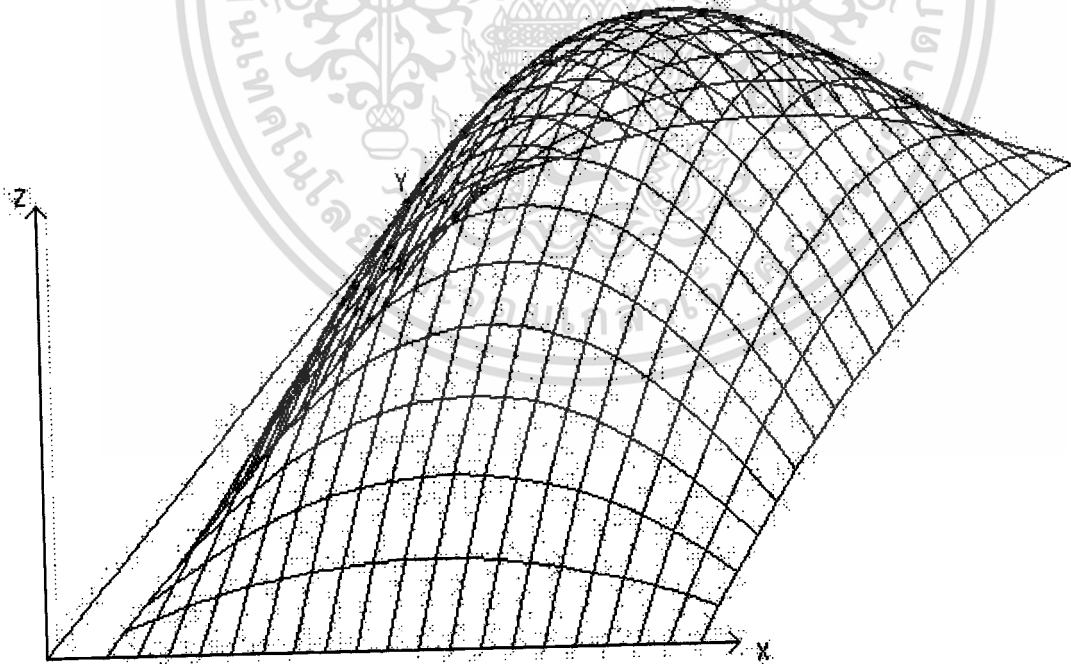
2.4 ตัวอย่างการสร้างแบบจำลอง (Biparametric Cubic Surface Examples)

จากรูปที่ 2-3 เกิดจากการกำหนดให้ ตัวแปรซึ่งทำหน้าที่ควบคุมความโค้งของพื้นผิวโค้งใน 3 มิติ โดยกำหนดให้ค่า

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 90 & 90 & 0 \\ 0 & 90 & 90 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

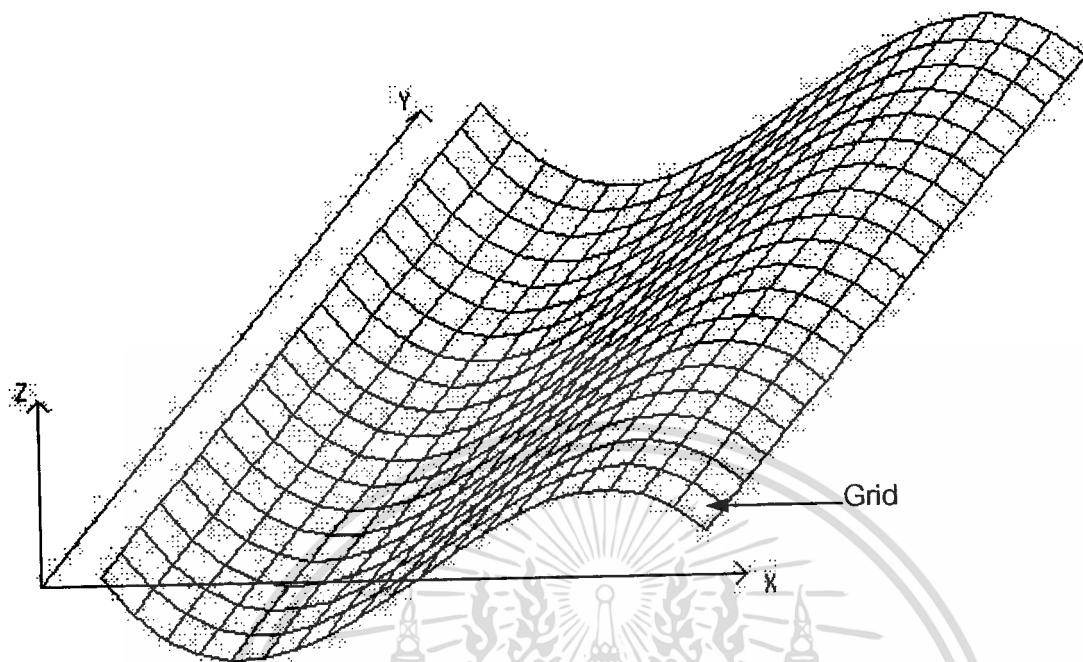
ในรูปที่ 2-3 เกิดจากเรากำหนดให้ค่าตัวแปร P ดังนี้

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -40 & -40 & 40 & -40 \\ 40 & 40 & 40 & 40 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



รูปที่ 2-4 รูปแสดงตัวอย่างของการสร้างพื้นผิวโค้งเป็นรูปโดม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-5 รูปแสดงตัวอย่างของการสร้างพื้นผิวโค้งเป็นรูป Sine

จากรูปที่ 2-4 และ 2-5 จะพบว่าการสร้างพื้นผิวโค้งโดยใช้สมการสร้างพื้นผิวโค้ง 3 มิติแบบนี้จะมีความเป็นไปได้น้อยมากที่พื้นที่ของ Grid จะมีขนาดน้อย ซึ่งถ้า Grid มีขนาดเล็กมากจะทำให้การปรับปรุงพื้นผิวอาจเกิดปัญหาได้

บทที่ 3

การหาจุดอ้างอิงของเครื่องจักร

Reference point of machine

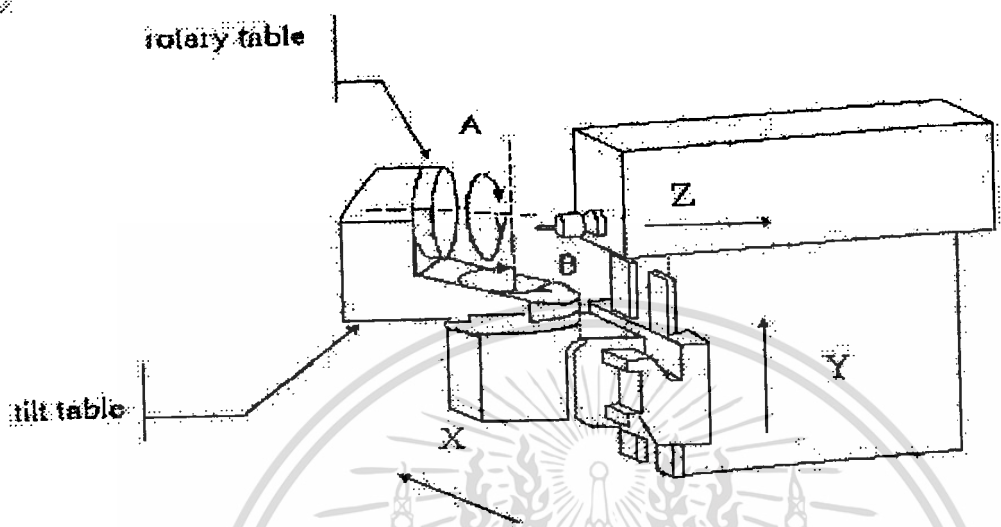
3.1 การหารูปทั่วไปของพื้นผิว (Normal form of surface)

ในระบบอุตสาหกรรมการร่างแบบและการออกแบบผลิตภัณฑ์ที่ใช้คอมพิวเตอร์ช่วยงานนั้น แยกการทำงานออกเป็น 2 ส่วน คือ ส่วนของการออกแบบ (Computer Aided Design- CAD) ซึ่งเป็นการกำหนดรูปร่าง และรูปแบบ และส่วนการกำหนดชุดคำสั่ง (Computer Aided Manufacturing - CAM) เพื่อบอกให้เครื่องจักรดำเนินการเพื่อผลิตตามรูปแบบที่กำหนดไว้ตามส่วนที่ 1 ที่ต้องการ ตัวอย่างเช่น การตัดเจาะวัสดุที่มีรูปทรงใด ๆ มีขนาด กว้าง (x) และ ยาว (Y) และมีความสูง (Z) การกำหนดจุดอ้างอิงของชิ้นงาน (workpieces coordinate system) ในรูปแบบสมการคณิตศาสตร์ เป็น (x, y, z, l, j, k) โดยที่ l, j, k คือ เวกเตอร์ขนาด 1 หน่วย ที่ขนานไปกับเครื่องมือที่เครื่องจักรอัตโนมัติแบบ 5 แกนใช้ เมื่อพิจารณาเฉพาะสมการการเคลื่อนที่ของเครื่องมือ ที่มีตำแหน่งตั้งฉากกับพื้นผิว $P(u,v)$ เป็นต้น $(l, j, k) = N$ โดยที่ $N(u,v)$ คือ เวกเตอร์ทั่วไปของ $P(u,v)$ กำหนดโดย

$$NN \equiv NN(x(u, v), y(u, v), z(u, v)) = \frac{\partial P(u, v)}{\partial u} \times \frac{\partial P(u, v)}{\partial v} \quad (1)$$

$$N(u, v) = \frac{NN(u, v)}{|NN(uv)|} \quad (2)$$

3.2 เครื่องจักรขนาด 5 แกน (5 axis milling machine)



รูปที่ 3-1 รูปแสดงเครื่องจักรขนาด 5 แกน

มุมของการหมุนของแกน A และ แกน B กำหนดโดย

$$\begin{aligned}
 A &= -\text{atan}(j/l) && \text{if } l > 0 \\
 &= (180 \text{ degree} - \text{atan}(j/l)) && \text{if } l < 0 \text{ and } j < 0 \\
 &= (180 \text{ degree} - \text{atan}(j/l)) && \text{if } l < 0 \text{ and } j > 0 \\
 &= -90 \text{ degree} && \text{if } l = 0 \text{ and } j > 0 \\
 &= 90 \text{ degree} && \text{if } l = 0 \text{ and } j < 0 \\
 &= 180 \text{ degree} && \text{if } l < 0 \text{ and } j = 0 \\
 &= 0 && \text{if } \sqrt{l^2 + j^2} < 0.01
 \end{aligned}$$

and

$$\begin{aligned}
 B &= -\text{atan}(\sqrt{l^2+j^2}/k) && \text{if } l < 0 \\
 &= \text{atan}(\sqrt{l^2+j^2}/k) && \text{if } l > 0 \\
 &= 0 && \text{if } \sqrt{l^2+j^2} < 0.01
 \end{aligned}$$

พิจารณาสมการแทนจุด P_0 และ P_1 บนพื้นผิว $P(u,v)$

$$M(p) = M(X_m, Y_m, Z_m) = X_m(p), Y_m(p), Z_m(p) \quad (3)$$

เราหา X_m, Y_m, Z_m ได้จาก

$$X_m = X_m(p) = (AB - c - z1) \cos(b) + (e + (a + x1) \cos(A) - (b + y1) \sin(A)) \sin(B) \quad (4)$$

$$Y_m = -((b + y1) \cos(A)) - (a + x1) \sin(A)$$

$$Z_m = -B4 + OF + \cos(B) + [e + (a + x1) \cos(A) - (b + y1) \sin(A)] - (AB - c - z1) \sin(B) \quad (5)$$

โดยที่ $p = (X, Y, Z)$ และ (a, b, c) กำหนดให้จุดแทนตำแหน่งของชิ้นงานบนระนาบของโต๊ะทำงาน

AB : ระยะห่างระหว่าง A และ B (กำหนด AB = 251.286)

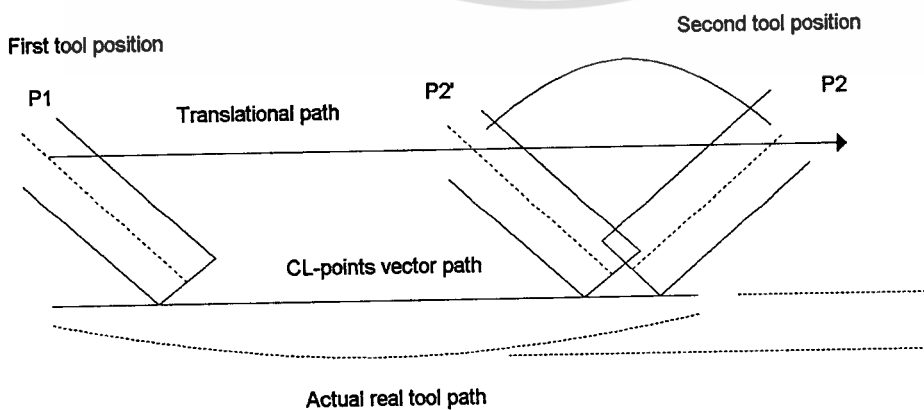
OF : ระยะห่างจากปลายแหลมของเครื่องมือ ถึง แท่นเครื่องจักร

B4 : ระยะห่างจากแนวแกน B ไปยัง แท่นเครื่องจักร (B4 = 371.943)

e : ค่าความคลาดเคลื่อนของ แกน B (e = 0.08)

3.3 ปัญหาจากค่าความคลาดเคลื่อน (Error from translation)

ปัจจุบันการใช้ค่าอ้างอิงแบบ 5 แกน (5-axis) ในการสั่งงานเครื่องจักร (CAD/CAM) สามารถแทนค่าตำแหน่งอ้างอิงต่อ ๆ กันไปด้วยสมการคณิตศาสตร์ ในเชิงสมการเส้นตรง ที่อ้างอิงถึงจุดแกนกลางในการหมุนที่คงที่ หากแต่ในความเป็นจริงการควบคุมเครื่องจักรไม่สามารถควบคุมแนวแกนกลางให้มีค่าคงที่ได้เนื่องจาก เครื่องจักรมีการสั่นตามวงรอบของการหมุน ซึ่งความเบี่ยงเบนดังกล่าวไม่ได้อยู่ในรูปของสมการเส้นตรงอีกต่อไป



รูปที่ 3-2 แสดงการตัดของพื้นผิวที่ไม่เป็นเส้นตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจักรเคลื่อนที่จาก P1 ไปยัง P2' และจาก P2' ไปยัง P2 ด้วยการหมุน ของแนวแกนเคลื่อนที่ ดังนั้นจึงเกิดแรงกระทำที่แยกออกจากกันเป็น 2 แนวทาง การพิจารณาถึงความโค้งงอของเครื่องมือ ณ ระยะเวลาสิ้นสุดของแนวเครื่องมือ เพื่อเป็นระยะเวลาใช้งานที่แม่นยำ ดังนั้นการปฏิบัติงานจริงจึงจะมีความคลาดเคลื่อนน้อยที่สุด ทำอย่างไรจึงจะให้การอ้างอิงนั้นเป็นไปในลักษณะดังกล่าวได้

3.4 การพิจารณาการเปลี่ยนแปลงจุดอ้างอิงแบบไม่เป็นเส้นตรง (Non-linear kinematics of five-axis milling machine)

ความสัมพันธ์ของตัวแปรที่มีผลกระทบกับการจุดอ้างอิงเป็นไปตาม สมการความสัมพันธ์ของตัวแปรดังนี้ โดยพิจารณารายละเอียดจากรูปที่ 3-1.

$$M = B(A(P+R)+T)+C \quad (6)$$

M : ตำแหน่งที่ทำงานได้ของเครื่องจักร(Machine coordinates)

P : ตำแหน่งบนชิ้นงาน (The workpiece coordinates)

R : ตำแหน่งของการหมุน (The rotary table coordinates)

T : จุดแสดงตำแหน่งการเอียงของโต๊ะ (The tilt table coordinates)

C : ตำแหน่งศูนย์กลางของจุดที่ตัด (The cutter center coordinates)

A,B : แทนการหมุนเมตริกซ์การหมุนของแกนหลักและแกนรอง

a : มุมที่หมุน a

b : มุมที่หมุน b

$$a = \arctan(i/j), \quad (7)$$

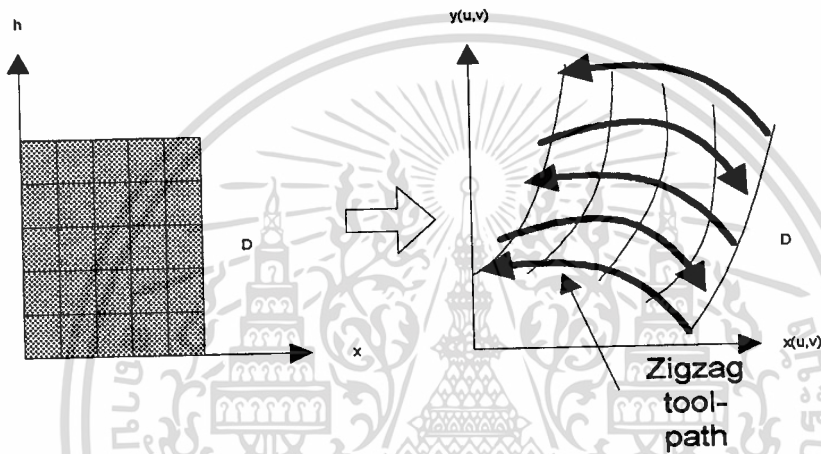
$$b = \arctan((i^2 + j^2)^{1/2} / k), \quad (8)$$

i,j,k : รูปทั่วไปของพื้นผิวที่ต้องการ

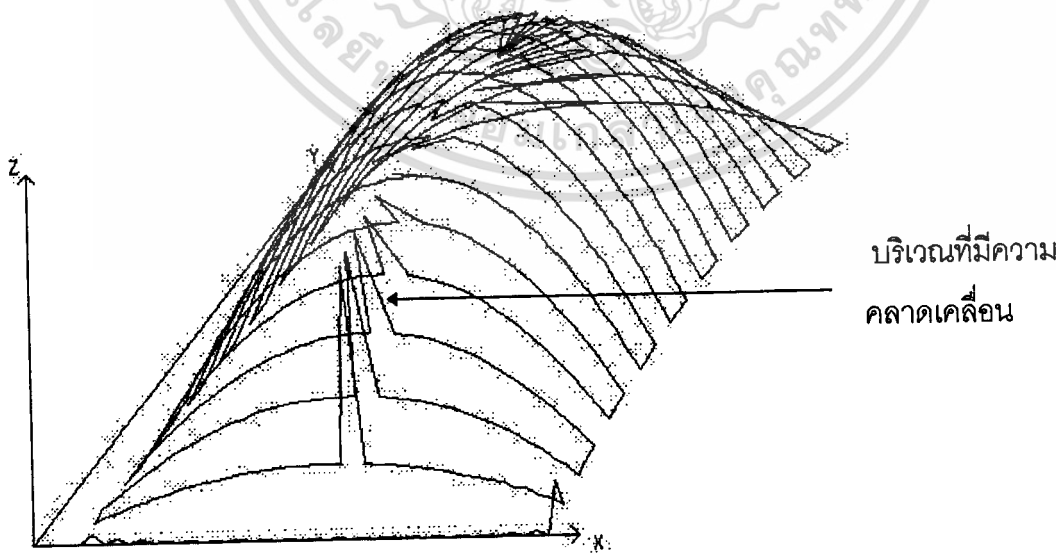
3.5 การคำนวณเส้นทางการเคลื่อนที่ของเครื่องมือ

เมื่อเราได้รูปทั่วไปของพื้นผิวแล้วเราจะนำไปคำนวณตำแหน่งที่เครื่องจักรจะต้องทำการตัดชิ้นงานให้ได้รูปทรงตามต้องการ ในโครงการนี้เราอาศัยการคำนวณการเคลื่อนที่แบบสลับฟันปลา (Zigzag) เนื่องจากเป็นวิธีที่ง่ายที่สุด และมีการเคลื่อนที่ของเครื่องมือน้อยที่สุด

คำนวณพื้นที่ (computation region) $\Delta = \{0 \leq \xi \leq N_\xi, 0 \leq \eta \leq N_\eta\}$ ภายในชิ้นงานจริง D



รูปที่ 3-3 แสดงการวิธีการเปลี่ยนแปลงจากพื้นผิวที่สร้างขึ้นเป็นทางเดินของเครื่องมือ

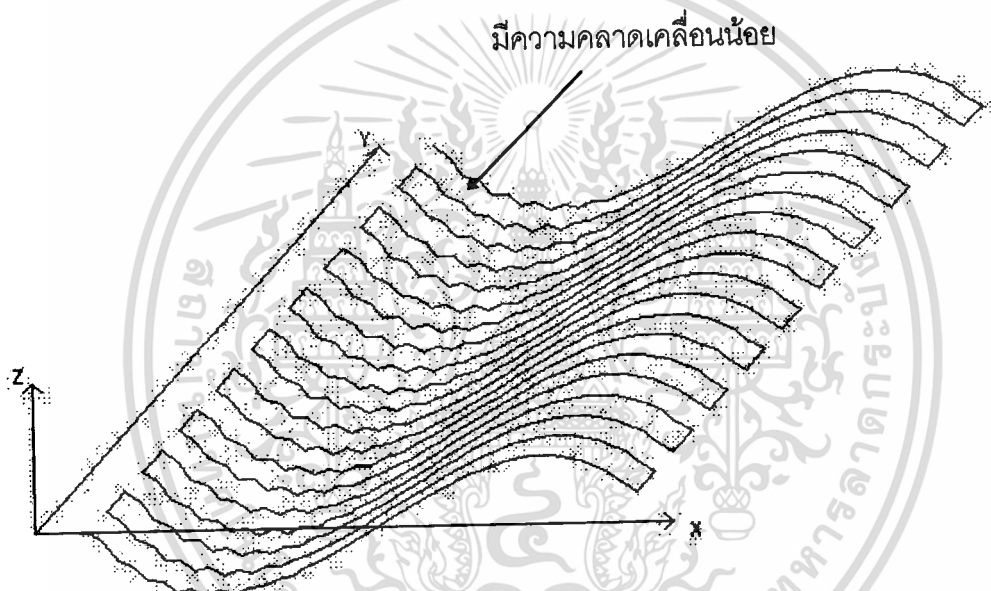


รูปที่ 3-4 รูปแสดงการเคลื่อนที่ของเครื่องมือ ชิ้นงานที่จะได้จากพื้นผิวรูปโดม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณารูปที่ 3-4 เรากำหนดค่าของตัวแปรควบคุม P จะเห็นว่า มีการเปลี่ยนแปลงของตัวควบคุมจาก 0 ไปเป็น 90 ซึ่งมีการเปลี่ยนแปลงมากทำให้เห็นพื้นที่ ๆ คลาดเคลื่อนมากขึ้นชัดเจนขึ้น

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 90 & 90 & 0 \\ 0 & 90 & 90 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



รูปที่ 3-5 แสดงการเคลื่อนที่ของเครื่องมือจากชิ้นงานรูป Sine

จากรูปที่ 3-4 และ 3-5 เราจะพบว่า ในรูปที่ 3-4 มีความผิดพลาดของพื้นผิวที่ได้มากกว่ารูปที่ 3-5 เนื่องจากชิ้นงานที่ต้องการมีความแตกต่างของพื้นผิวมาก

3.6 การกำหนดขอบเขตของคุณภาพชิ้นงาน (Quality Criterion)

$$\omega_r = |S(\xi, \eta) - T_r(\xi, \eta)| \tag{9}$$

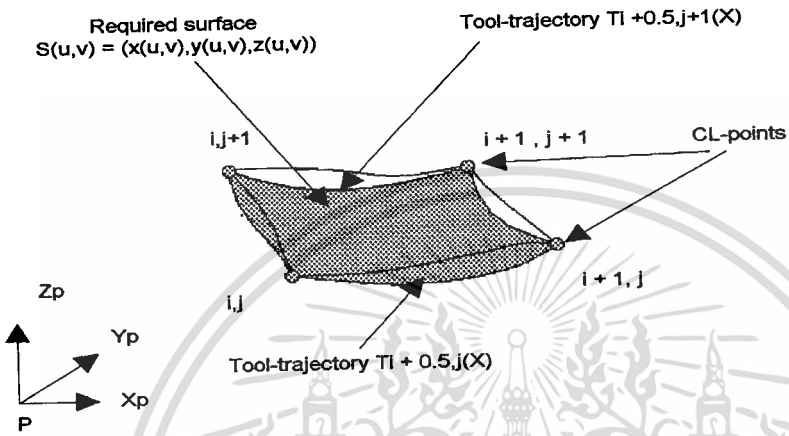
เราให้ตัวแปร S เป็นพื้นผิวที่ได้จากการออกแบบ และ T เป็นพื้นผิวที่ได้จากการออกแบบซึ่งเครื่องจักรจะใช้ในการสร้างชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ $S \equiv S(u, v) = (x(u, v), y(u, v), z(u, v))$ ต้องการพื้นผิวที่ $\tau = \xi$ หรือ $\tau = \eta$,

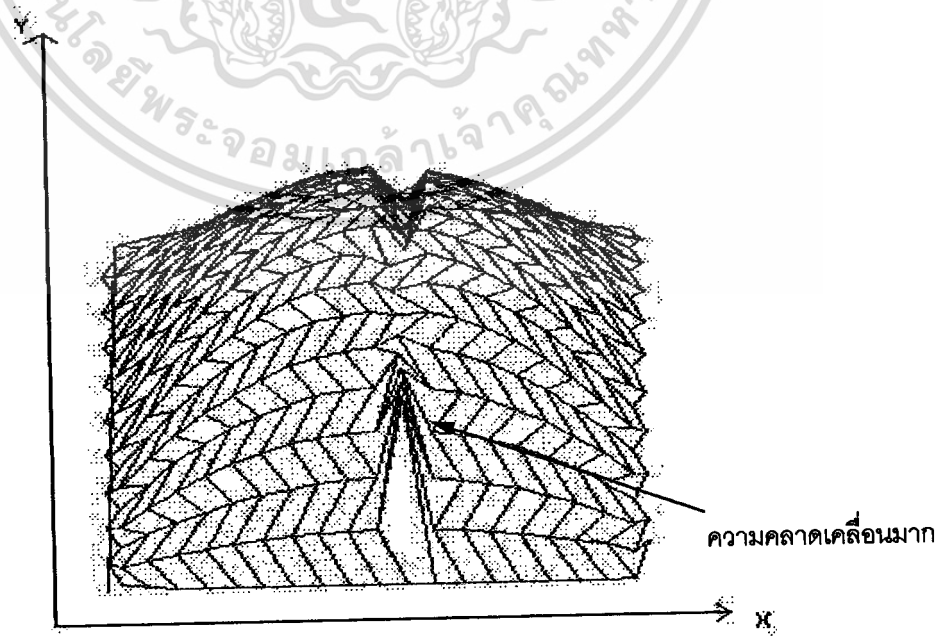
$$T_\xi(\xi, \eta) = L_\eta(T_{i+0.5, j}, T_{i+0.5, j+1}), \text{ if } i \leq \xi \leq i+1 \text{ and } j \leq \eta \leq j+1, \quad (10)$$

$$T_\eta(\xi, \eta) = L_\xi(T_{i+0.5, j}, T_{i+0.5, j+1}), \text{ if } i \leq \xi \leq i+1 \text{ and } j \leq \eta \leq j+1, \quad (11)$$

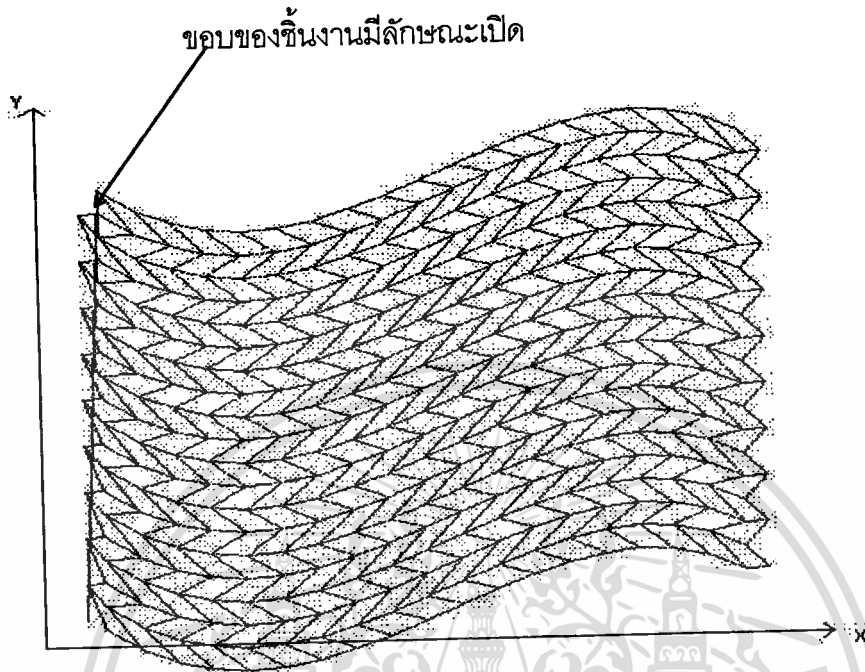


รูปที่ 3-6 ความแตกต่างของชิ้นงานที่จะได้กับพื้นผิวที่ออกแบบ

จากรูปที่ 3-6 เรากำหนดให้ตัวแปร T แทนพื้นผิวของชิ้นงานที่จะได้ เราจะพบว่าพื้นผิวของชิ้นงานที่ได้จากเครื่องจักรทำงานจะมีความแตกต่างจากแบบที่ออกไว้อยู่ระดับหนึ่ง ดังแสดงในรูปที่ 3-7 จุดที่เราเห็นเกิดจากการตัดของพื้นผิวจะไม่ตรงกับจุดซึ่งเกิดจากการออกแบบเท่ากับจุดเริ่มต้นเสมอ



รูปที่ 3-7 แสดงตำแหน่งของจุดที่ได้จากการทำงานของเครื่องของพื้นผิวรูปโดม



รูปที่ 3-8 แสดงตำแหน่งของจุดที่ได้จากการทำงานของเครื่องของพื้นผิวรูป Sine

จากรูปที่ 3-7, 3-8 เราจะได้การเคลื่อนที่ของเครื่องมือ ซึ่งพบว่าจุดที่เกิดจากการสร้างของเครื่องมือคลาดเคลื่อนไปจาก $S(\xi, \eta)$ และ $T(\xi, \eta)$ เสมอไป

บทที่ 4

การปรับปรุงพื้นผิวของชิ้นงาน

Adaptive surface approach

4.1 สมการกำหนดรูปแบบระนาบโค้ง 3 มิติ (Formulation of the mesh generator)

ในการสร้างระนาบโค้ง 3 มิติ เรากำหนดให้จัดเก็บข้อมูลที่เป็นจุดในรูปของตัวแปรชุด (array) โดยเก็บในรูปของ ตัวแปร $X(i, j)$ โดยที่ $i = 1, \dots, M; j = 1, \dots, N$ กำหนดความสัมพันธ์ของจุดหนึ่งจุดและจุดข้างเคียงของ $X(i, j)$ เป็น $X(i, j), X(i+1, j), X(i, j+1), X(i-1, j)$ และ $X(i, j-1)$

เราสามารถสร้างแบบจำลองทางคณิตศาสตร์ขึ้นได้ โดยเราจะกำหนดการแทนระนาบโค้ง 3 มิติ ทั้งหมดในความเป็นจริง ซึ่งตามโครงการนี้เรากำหนดให้เป็นพื้นผิว ที่ได้จากการแปลงสมการจากแบบจำลองของพื้นผิว เป็นแบบจำลองของเครื่องจักร โดยหลักการนี้เราจะแทนด้วย $X(\xi, \eta)$ โดยกำหนดตำแหน่งในความเป็นจริงของพื้นผิวที่ได้จากการคำนวณในสมการด้วย ξ และ η เราสามารถกำหนดได้ว่ารูปทั่วไปของพื้นผิวที่สร้างขึ้นจะมี $\Delta \xi = \Delta \eta = 1$ ในการปรับปรุงระนาบเราจะสนใจพื้นผิวที่ได้เครื่องจักรจะใช้จริงเท่านั้น $X(\xi, \eta)$

จากประโยชน์ข้างต้นเราสามารถอาศัยคุณสมบัติการเปลี่ยนแปลง (differential properties) ของการสร้างพื้นผิวโดยอาศัยสมการการเดินทางของเครื่องมือที่เครื่องจักรใช้

เช่น $\sqrt{\left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta}\right)^2 - \left(\frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}\right)^2}$ ซึ่งเป็นไปตามการเปลี่ยนแปลงของเส้นโค้ง η จะมี

ผลต่อการเปลี่ยนแปลงของปริมาตรที่เกิดจากการเลื่อนจุดของระนาบนั้น กล่าวคือ การเปลี่ยนแปลงในแนวใดแนวหนึ่งของระนาบ จะต้องกระทำในแนวเดียวกันทั้งระนาบและจะต้องอยู่ภายใต้การควบคุมของสมการควบคุมความโค้งของระนาบ 3 มิติ ในแนวใดแนวหนึ่ง เราพิจารณาได้จากการสร้างพื้นผิวโค้งในบทที่ 2 จะถูกควบคุมด้วยเวกเตอร์ 2 แนว คือ U และ V โดยเรากำหนดการเปลี่ยนแปลงของปริมาตรที่ขึ้นกับพื้นผิวให้ขึ้นกับ J (Jacobian) ที่ได้จากการแปลงพื้นผิวดันเทียบกับพื้นผิวที่เครื่องจักรใช้ในการทำงานในแต่ละจุด

$$J = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \quad (1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลรวมที่ได้จากการคำนวณทั้งหมดของระนาบ เราสามารถนำมาใช้เป็นคุณสมบัติในการวัดความเกลี้ยงเกลาของการแปลงพื้นผิว เป็นพื้นผิวสำหรับเครื่องจักร ให้ตัวแปร I_s เป็นตัวแปร ซึ่งใช้ในการวัดความเกลี้ยงเกลาของพื้นผิวโค้ง จากการเปลี่ยนแปลงของระนาบตามแนว ξ และ η กล่าวคือ I_s คือ ผลรวมของปริมาตรที่เปลี่ยนแปลงในระนาบที่เครื่องจักรทำงาน

$$I_s = \int_D [(\nabla \xi)^2 + (\nabla \eta)^2] dV \quad (2)$$

การเปลี่ยนแปลงของปริมาตรสามารถวัดได้จาก

$$I_v = \int_D wJ dV, \quad (3)$$

โดยที่ $w = w(x, y)$ ซึ่งก็คือสมการใดๆ ซึ่งแสดงการเปลี่ยนแปลงของปริมาตร

โดยเหตุนี้ ถ้าเราสามารถวัดค่าได้เราก็สามารถควบคุมการเปลี่ยนแปลงของปริมาตรได้ ความเกลี้ยงเกลาของพื้นผิวโค้ง (smoothness) จะได้รับจากการทำให้ I_v มีค่าน้อยที่สุด ตัวอย่างเช่น เราสามารถคำนวณการเปลี่ยนแปลงของปริมาตรโดยอาศัยค่าความเปลี่ยนแปลง $w \equiv 1$ หมายความว่า w เป็นค่าที่ได้จากการวัดของสมการที่ไม่เป็นแบบเส้นตรง จึงต้องกำหนดเป็นช่วง ขึ้นกับจุดที่จะพบบนระนาบ เช่น $w = \exp(|X(\xi, \eta)^2 + Y(\xi, \eta)^2|)$ ดังนั้น $wJ = J = const.$ จากการจำลองรูปแบบของปัญหานี้เราจะพบว่า ตัวแปรเวลา t สามารถหาจุดใดบนระนาบได้ เมื่อเวลา $t = 0$ ได้เป็น

$$X(\xi, \eta, t=0) = X_0(\xi, \eta),$$

$$Y(\xi, \eta, t=0) = Y_0(\xi, \eta),$$

โดยที่อัตราเร็วในการเปลี่ยนแปลงของระนาบโค้ง 3 มิติจะขึ้นอยู่กับ

$$u(\xi, \eta, t) \equiv \frac{\partial x(\xi, \eta, t)}{\partial t},$$

$$v(\xi, \eta, t) \equiv \frac{\partial y(\xi, \eta, t)}{\partial t},$$

กล่าวคือ อัตราเร็วในการปรับตัวของพื้นผิว = การหาการเปลี่ยนแปลงของปริมาตรใน 1 หน่วยเวลาและการเปลี่ยนแปลงของระนาบหาได้จาก

$$J(\xi, \eta, t) \equiv X_\xi(\xi, \eta, t)Y_\eta(\xi, \eta, t) - X_\eta(\xi, \eta, t)Y_\xi(\xi, \eta, t),$$

เราอาศัยสมการที่ (5a) - (5d) ซึ่งจะกล่าวต่อไปจะได้

$$\frac{1}{J} \frac{\partial J}{\partial t} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}.$$

นั่นคือ $J(t) = J_0$ ถ้าไม่สามารถจะทำการปรับพื้นผิวได้

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0.$$

จากสมการข้างบนที่กล่าวมานี้จะเห็นว่าเป็นการปรับตัวของระนาบในทิศทางใดทิศทางหนึ่งเท่านั้น

เราจะสามารถหาพื้นผิวที่สมบูรณ์จากการปรับปรุงพื้นผิวได้จาก

$$I = I_s + \lambda_v I_v \quad \text{โดยที่ } \lambda_v \geq 0 \quad \text{เป็นสัดส่วนที่จะปรับ} \quad (2) + \lambda_v(3) = (4)$$

4.2 การปรับปรุงความคลาดเคลื่อนของระนาบแบบ 2 ทิศทาง

จากส่วนที่แล้วเรานำเสนอวิธีการปรับปรุงระนาบใดในทิศทางใดทิศทางหนึ่งเพียงทิศทางเดียว ในส่วนนี้จะกล่าวถึงวิธีการของ Brackbill และ Saltzman ซึ่งได้ทำการปรับปรุงวิธีการข้างต้นโดยอาศัยสมการ Euler equation เข้ามาประยุกต์ใช้กับวิธีการในส่วนที่แล้ว

เพื่อความง่ายต่อความเข้าใจได้จัดรูปสมการใหม่ ดังนี้

$$\xi_x = + \frac{Y_\eta}{J}, \quad (5a)$$

$$\xi_y = - \frac{X_\eta}{J}, \quad (5b)$$

$$\eta_x = - \frac{Y_\xi}{J}, \quad (5c)$$

$$\eta_y = + \frac{X_\xi}{J}, \quad (5d)$$

จากการเปลี่ยนค่าตัวแปร สามารถเขียนสมการใหม่สำหรับระนาบที่สมบูรณ์ได้เป็น

$$I_s = \int_1^M \int_1^N \frac{x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2}{J} d\xi d\eta, \quad (6)$$

ซึ่งตรงกับสมการ

$$\left(\frac{\partial}{\partial x} - \frac{\partial}{\partial \xi} \frac{\partial}{\partial x_\xi} - \frac{\partial}{\partial \eta} \frac{\partial}{\partial \eta} \right) \left(\frac{x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2}{J} \right) = 0, \quad (7a)$$

$$\left(\frac{\partial}{\partial y} - \frac{\partial}{\partial \xi} \frac{\partial}{\partial x_\xi} - \frac{\partial}{\partial \eta} \frac{\partial}{\partial y_\eta} \right) \left(\frac{x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2}{J} \right) = 0, \quad (7b)$$

ทำการแสดง differentiation และรวมสัมประสิทธิ์จากสมการ highest derivations

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$B(\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta}) - A(\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta}) = 0, \quad (8a)$$

$$-A(\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta}) - C(\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta}) = 0, \quad (8b)$$

โดยที่

$$A = x_{\xi} y_{\xi} + x_{\eta} y_{\eta}, \quad B = y_{\xi}^2 + y_{\eta}^2, \quad C = x_{\xi}^2 + x_{\eta}^2, \quad (9)$$

และ

$$\alpha = (x_{\eta}^2 + y_{\eta}^2) / J^3, \quad \beta = (x_{\xi} x_{\eta} + y_{\xi} y_{\eta}) / J^3, \quad \gamma = (x_{\xi}^2 + y_{\xi}^2) / J^3, \quad (10)$$

จากสมการที่ (8a) และ (8b) เมื่อ $A^2 - BC \neq 0$ เราสามารถแยกประกอบและเขียนรูปของสมการใหม่ได้เป็น

$$\alpha x_{\xi\xi} + 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0, \quad (11a)$$

$$\alpha x_{\xi\xi} + 2\beta x_{\xi\eta} + \gamma y_{\eta\eta} = 0. \quad (11b)$$

แต่กรุปสมการได้เป็นรูปทั่วไปดังนี้ เพื่อกำหนดเงื่อนไขในการนำตัวแปรไปใช้

$$b_{s1} x_{\xi\xi} + b_{s2} x_{\xi\eta} + b_{s3} x_{\eta\eta} + a_{s1} y_{\xi\xi} + a_{s2} y_{\xi\eta} + a_{s3} y_{\eta\eta} = 0 \quad (12a)$$

และ

$$a_{s1} x_{\xi\xi} + a_{s2} x_{\xi\eta} + a_{s3} x_{\eta\eta} + c_{s1} y_{\xi\xi} + c_{s2} y_{\xi\eta} + c_{s3} y_{\eta\eta} = 0, \quad (12b)$$

โดยที่

$$a_{s1} = -A\alpha, \quad b_{s1} = B\alpha, \quad c_{s1} = C\alpha, \quad (13)$$

$$a_{s2} = 2A\beta, \quad b_{s2} = -2B\beta, \quad c_{s2} = -2C\beta,$$

$$a_{s3} = -A\gamma, \quad b_{s3} = B\gamma, \quad c_{s3} = C\gamma,$$

วัดค่าการเปลี่ยนแปลงของปริมาตรของระนาบ เมื่อมีการเปลี่ยนแปลงตัวแปรอิสระและไม่

อิสระ

$$I_V = \int_1^M \int_1^N d\xi d\eta w J^2, \quad (14)$$

จากสมการที่ (14) เมื่อพิจารณาเปรียบเทียบกับสมการที่ (3) เราจะเห็นว่าได้ชัดเจนว่า I_V ที่ได้จากสมการที่ (14) จะเกิดจากการเปลี่ยนแปลงของปริมาตร แต่ I_V ในสมการที่ (3) ยังเป็นการเปลี่ยนแปลงในของระนาบอยู่ ซึ่งสำหรับ Euler equation เราจะได้

$$\frac{\partial w}{\partial x} J^2 - 2 \left[J(w_\xi y_\eta - w_\eta y_\xi) + w(J_\xi y_\eta - J_\eta y_\xi) \right] = 0 \quad (15a)$$

$$\frac{\partial w}{\partial x} J^2 - 2 \left[J(w_\xi y_\eta - w_\eta y_\xi) + w(J_\xi y_\eta - J_\eta y_\xi) \right] = 0 \quad (15b)$$

รวมสัมประสิทธิ์ของ highest derivation ได้เป็นสมการ

$$2w(b_{V1}x_{\xi\xi} + b_{V2}x_{\xi\eta} + b_{V3}x_{\eta\eta} + a_{V1}y_{\xi\xi} + a_{V2}y_{\xi\eta} + a_{V3}y_{\eta\eta}) = J^2 \frac{\partial w}{\partial x} \quad (16a)$$

$$2w(a_{V1}x_{\xi\xi} + a_{V2}x_{\xi\eta} + a_{V3}x_{\eta\eta} + c_{V1}y_{\xi\xi} + c_{V2}y_{\xi\eta} + c_{V3}y_{\eta\eta}) = -J^2 \frac{\partial w}{\partial y}, \quad (16b)$$

หาค่าสัมประสิทธิ์ได้จาก

$$\begin{aligned} a_{V1} &= -x_\eta y_\eta, & b_{V1} &= y_\eta^2, & c_{V1} &= x_\eta^2, \\ a_{V2} &= x_\xi y_\eta + x_\eta y_\xi, & b_{V2} &= -2y_\xi y_\eta, & c_{V2} &= -2x_\xi x_\eta, \\ a_{V3} &= -x_\xi y_\xi, & b_{V3} &= y_\xi^2, & c_{V3} &= x_\xi^2, \end{aligned} \quad (17)$$

ในทั่วไปเมื่อมีการปรับปรุงระนาบแบบวนรอบ ค่าสัมประสิทธิ์จะมีการเพิ่มค่าโดย

$$\begin{aligned} a &= a_s + \lambda_V a_V, \\ b &= b_s + \lambda_V b_V, \\ c &= c_s + \lambda_V c_V, \end{aligned} \quad (18)$$

4.3 การแก้ปัญหาโดยวิธีการทางคณิตศาสตร์ (Numerical Solution)

โดยทั่วไปของระนาบจุดสิ้นสุดของความแตกต่างของ Euler equation จะแก้ไขได้โดยการกระทำซ้ำ เริ่มตั้งแต่ (ξ, η) และต่อเนื่องไปเรื่อย โดยตัวแปรจะกำหนดเป็น Integer ในแต่ละจุดบนระนาบเราสามารถกำหนดช่องว่างระหว่างจุด ซึ่งเกิดจากรูปสี่เหลี่ยมวางต่อเนื่องกันไปบนระนาบ การหาอนุพันธ์จากตัวแปรอิสระก็เพื่อให้ง่ายต่อการคำนวณในแต่ละ grid ของระนาบ เรากำหนดว่าในแต่ละจุด (ξ, η) ได้จากการหาอนุพันธ์ของจุดซึ่งแทนโดย

$$\begin{aligned} x_\xi &\approx \frac{1}{2}(x_{i+1,j} - x_{i-1,j}), \\ x_\eta &\approx \frac{1}{2}(x_{i,j+1} - x_{i,j-1}), \end{aligned} \quad (19)$$

และหาอนุพันธ์ลำดับที่ 2 โดย

$$x_{\xi\xi} = x_{i+1,j} - 2x_{i,j} + x_{i-1,j}, \quad (20)$$

$$x_{\xi\eta} = \frac{1}{4}(x_{i-1,j+1} + x_{i-1,j-1} - x_{i+1,j-1} - x_{i-1,j+1})$$

$$x_{\eta\eta} = x_{i,j+1} - 2x_{i,j} + x_{i,j-1}, \quad (21)$$

กระทำแบบเดียวกันนี้กับ Y . สำหรับสมการเรขาคณิตในแต่ละจุดซึ่งเป็นผลลัพธ์จากความแตกต่างของการหาอนุพันธ์ ในระบบของสมการเราแก้ปัญหาโดย Jacobi iteration ในค่าของ $X_{i,j}$ และ $Y_{i,j}$ กระทำกับตัวแปร R_X และ R_Y ซึ่งแทนค่าความคลาดเคลื่อนที่ยังเหลืออยู่ กำหนดโดย

$$R_X = (R_S)_X + \lambda_V (R_V)_X,$$

และ

$$R_Y = (R_S)_Y + \lambda_V (R_V)_Y, \quad (22)$$

โดยแทนค่า R_S และ R_V ในสมการ (12) และ (16) ตามลำดับ จำนวนรอบของการปรับปรุงได้จาก (1) และคำนวณค่าของ $X_{i,j}^{(i+1)}$ และ $Y_{i,j}^{(i+1)}$ เราหาจุดสิ้นสุดของการปรับปรุงจาก

$$0 = R_X^i + \frac{\partial R_X^i}{\partial X_{ij}} (X_{ij}^{(i+1)} - X_{ij}^i) + \frac{\partial R_X^i}{\partial Y_{ij}} (Y_{ij}^{(i+1)} - Y_{ij}^i) \quad (23a)$$

$$0 = R_Y^i + \frac{\partial R_Y^i}{\partial X_{ij}} (X_{ij}^{(i+1)} - X_{ij}^i) + \frac{\partial R_Y^i}{\partial Y_{ij}} (Y_{ij}^{(i+1)} - Y_{ij}^i), \quad (23b)$$

โดย

$$\frac{\partial R_X^{(i)}}{\partial X_{ij}} = -2(b_1 + b_3),$$

$$\frac{\partial R_X^{(i)}}{\partial Y_{ij}} = -2(a_1 + a_3) = \frac{\partial R_Y^{(i)}}{\partial X_{ij}}, \quad (24)$$

$$\frac{\partial R_Y^{(i)}}{\partial Y_{ij}} = -2(c_1 + c_3).$$

การทำซ้ำจะทำงานกระทั่ง $R_X^{(i)}$ และ $R_Y^{(i)}$ ในทุกจุดบนระนาบมีค่าน้อยกว่าค่าความคลาด

เคลื่อนสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ขั้นตอนการทำงานเพื่อลดความคลาดเคลื่อนจากการแปลงข้อมูล

(Optimization problem)

4.4.1 กำหนดพื้นผิวเริ่มต้น

4.4.2 คำนวณ $\omega_{r,i,j} \equiv \omega_r(\xi, \eta_j)$ และ ความคลาดเคลื่อน บนพื้นผิว

4.4.3 กำหนด และทำให้เรียบ (scale และ smooth the grid function)

4.4.4 สร้างพื้นผิวใหม่

4.4.5 คำนวณ $\omega_{r,i,j}^{new}, E_r^{new}$ บนพื้นผิวใหม่ ขยายและสร้าง grid ใหม่ที่สมบูรณ์

4.4.6 ถ้า $|E_r^{new} - E_r| > \delta$ (โดย δ คือค่าที่กำหนดไว้) แล้ว

4.4.7 $\omega_{r,i,j} := \omega_{r,i,j}^{new}, G := G^{new}$ ให้กลับไปทำในข้อ 2

4.5 การกำหนดขอบเขตของพื้นผิว

จากส่วนของโปรแกรมข้างล่างนี้ กำหนดให้โปรแกรมทำงานภายใต้กรอบของพื้นผิวโดยใช้เงื่อนไขดังนี้

```
for (j = 1; j < ny-1; j++)
```

```
  for (i = 1; i < nx-1; i++)
```

```
    { a11 = RXX[i][j]; a12 = RXY[i][j];
```

```
      a21 = RXY[i][j]; a22 = RYY[i][j];
```

```
      bb1 = -RX[i][j]+RXX[i][j]*X[i][j]+RXY[i][j]*Y[i][j];
```

```
      bb2 = -RY[i][j]+RXY[i][j]*X[i][j]+RYY[i][j]*Y[i][j];
```

```
    if ((-a12*a21+a22*a11) == 0)
```

```
      { Xnew[i][j] = 0;
```

```
        Ynew[i][j] = 0;
```

```
      }
```

```
    else
```

```
      { Xnew[i][j] = -(a12*bb2-a22*bb1)/(-a12*a21+a22*a11);
```

```
        Ynew[i][j] = (a11*bb2-a21*bb1)/(-a12*a21+a22*a11);
```

```
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

d1 = a11*Xnew[i][j]+a12*Ynew[i][j];
d2 = a21*Xnew[i][j]+a22*Ynew[i][j];
}

```

กำหนดพื้นผิวใหม่ที่จะทำการปรับให้อยู่ภายใต้กรอบ

```

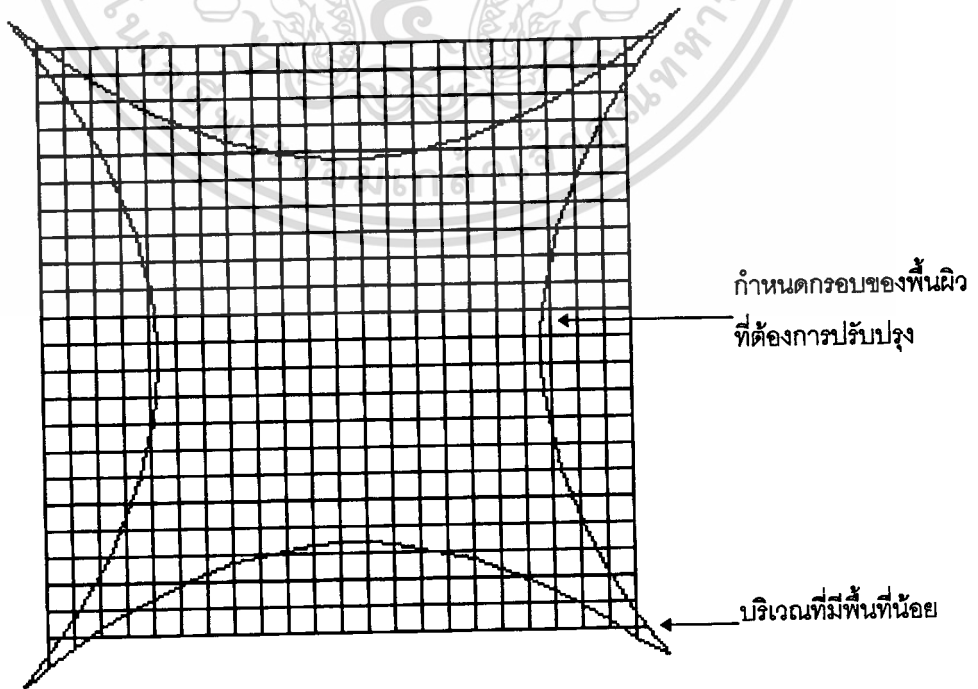
for (j = 1; j < ny-1; j++)
for (i = 1; i < nx-1; i++)
{
X[i][j] = X[i][j]*WW+Xnew[i][j]*(1-WW);
Y[i][j] = Y[i][j]*WW+Ynew[i][j]*(1-WW);
}

```

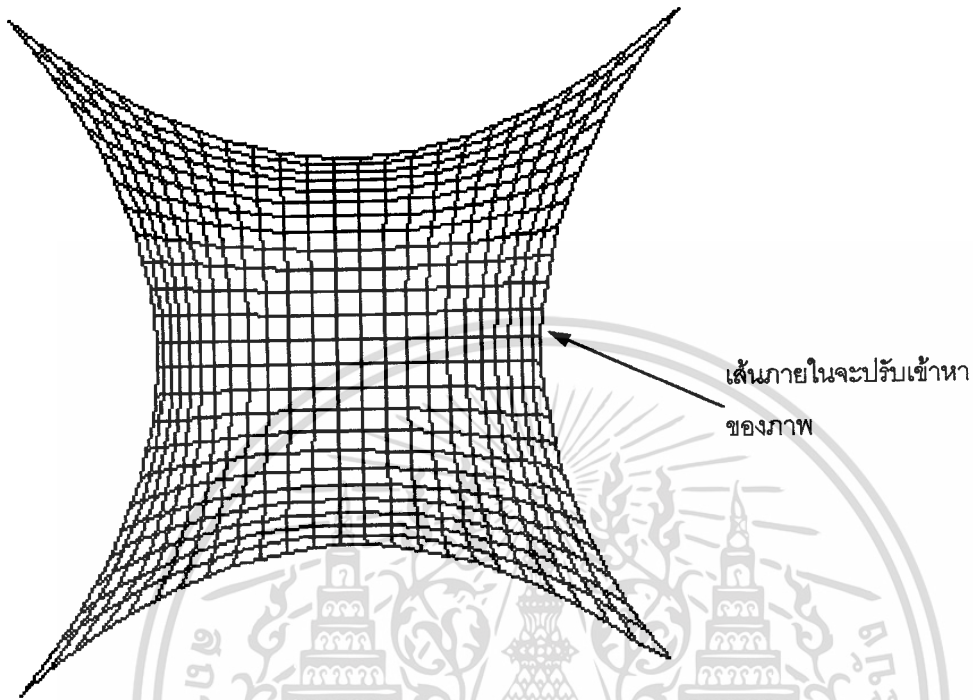
จากการคำนวณเราจะพบว่าผลลัพธ์ที่ได้ จะไม่ออกนอกขอบของพื้นผิวที่กำหนด

รูปขนาด 25 x 25 จุด

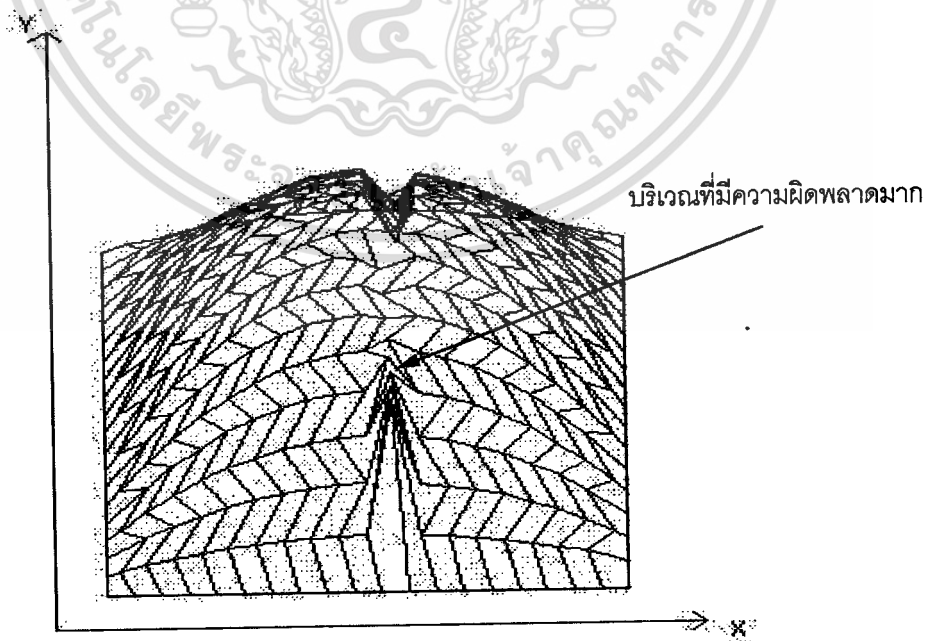
เงื่อนไข $WXY = \exp(-\text{abs}(25.0 - (X[i][j] - 10.0) * (X[i][j] - 10.0) - (Y[i][j] - 10.0) * (Y[i][j] - 10.0))) + 0.01$



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ที่ 4-1 กำหนดของเขตของพื้นผิวเพื่อทดสอบการปรับปรุง ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



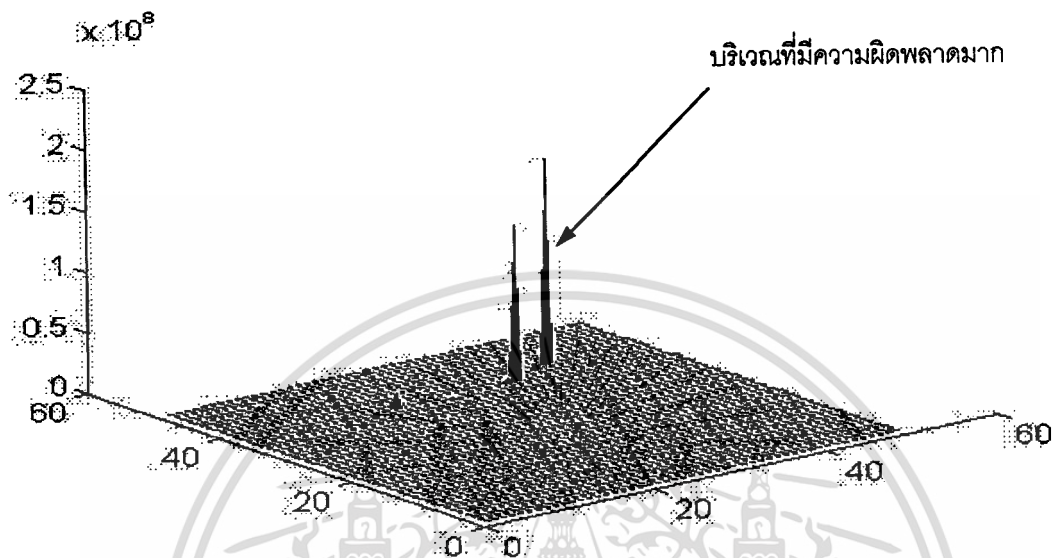
รูปที่ 4-2 รูปแสดงผลที่ได้จากการโปรแกรมซึ่งทำหน้าที่ปรับปรุงพื้นผิวงาน



รูปที่ 4-3 รูปแสดงการกำหนดของเขตของผิวชิ้นงานที่ต้องการปรับปรุง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

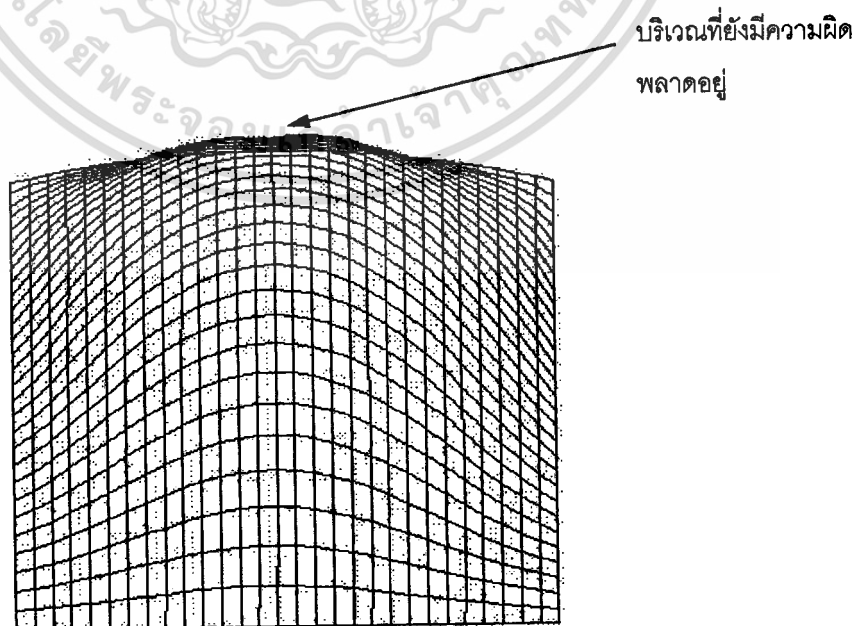
ด้วยเหตุผลข้างต้นเราจะกำหนดของเขตของพื้นผิวที่ได้จากการแปลงโดยกำหนดให้ขอบของชิ้นงานอยู่บนขอบของพื้นผิวที่ออกแบบได้ดังนี้



รูปที่ 4-4 แสดงบริเวณที่มีความผิดพลาดมาก

จากรูปที่ 4-4 เมื่อพื้นผิวที่กำหนดไว้ผ่านโปรแกรมส่วนปรับปรุง โปรแกรมจะทำการค้นหาความผิดพลาด โดยเราจะเห็นว่าจะเกิดบริเวณที่มีความผิดพลาดสูงปรากฏออกมา

เมื่อทำการปรับปรุงพื้นผิวของชิ้นงานโดยอาศัยโปรแกรมเพื่อทำการปรับปรุงชิ้นงานโดยกำหนดให้โปรแกรมทำการปรับปรุงพื้นผิวชิ้นงานเป็นจำนวน 250 ครั้งจะได้ผลดังนี้



รูปที่ 4-5 แสดงการปรับปรุงพื้นผิวรูปโดมที่ได้จากโปรแกรมปรับปรุงพื้นผิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4-5 เราจะสามารถเห็นผลลัพธ์ได้ 2 ข้อคือ

1. พื้นที่ที่มีความผิดพลาดมากจากลดลง
2. พื้นที่ที่มีความผิดพลาดน้อยจะหมดไป เราจะสังเกตจากเส้นที่ตรงขึ้นในแนวดิ่ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การโปรแกรมบนระบบซูเปอร์คอมพิวเตอร์ Supercomputer Program Implementation

5.1 การประมวลผลแบบขนาน (Parallel computing)

การเขียนโปรแกรมแบบขนานเป็นความพยายามในการให้เครื่องคอมพิวเตอร์ 1 เครื่องสามารถทำงานได้หลายงานได้ พร้อมกัน (Multitasking) โดยจะเครื่องคอมพิวเตอร์ประเภทนี้ จะมีความสามารถในการจัดการคำสั่งสำหรับการประมวลผลให้สามารถทำงานพร้อมกันหลายโปรเซสเซอร์ได้ โดยทั่วไปเครื่องคอมพิวเตอร์ซึ่งประมวลผลแบบขนานจะมีลักษณะดังนี้

- 5.1.1 มีหน่วยประมวลผล (Processor) หลายตัวทำงานร่วมกัน เป็น Massive Parallel Processor (MPP) โดยจะต้องมีจำนวนของ Processor มากๆ
- 5.1.2 มีระบบปฏิบัติการที่สามารถรองรับการทำงานแบบขนานได้
- 5.1.3 การติดต่อกับเครื่องคอมพิวเตอร์จะต้องกระทำผ่าน Workstation
- 5.1.4 งานที่สมควรเป็นงานที่มีการคำนวณสูง

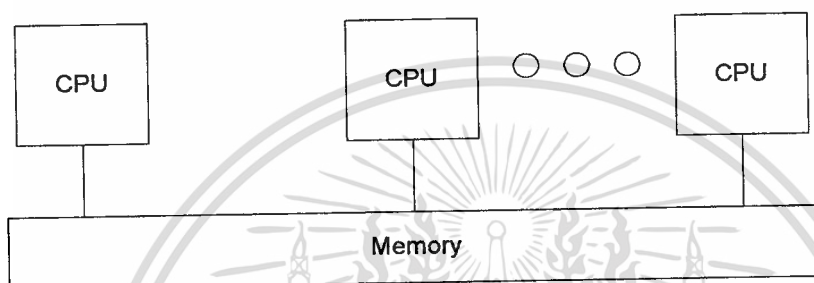
5.2 โครงสร้างพื้นฐานของเครื่องคอมพิวเตอร์แบบ SIMD และ MIMD

SIMD คือ Single Instruction multiple data parallel processor เครื่องคอมพิวเตอร์ประเภทนี้จะทำงานในลักษณะมีหน่วยควบคุมกลางทำหน้าที่จัดการและคำสั่งในการประมวลผลให้กับ Processor ต่างๆ โดยมักจะทำการกำหนดค่าให้กับรีจิสเตอร์ของแต่ละ Processor ลักษณะของงานที่ทำใน SIMD เป็นงานที่มีการจัดการข้อมูลขนาดใหญ่โดยใช้ชุดคำสั่งน้อยชุด

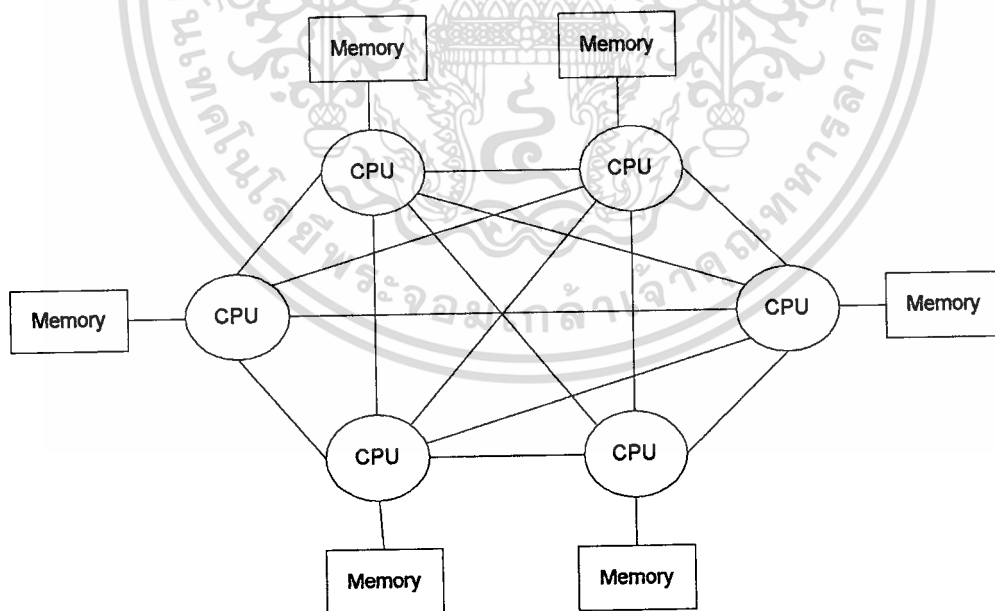
MIMD คือ Multiple Instruction multiprocessor เครื่องคอมพิวเตอร์ประเภทนี้มักจะทำงานในลักษณะที่มี Processor หลายตัวทำงานคำสั่งที่แตกต่างกัน ลักษณะงานที่ทำในเครื่องคอมพิวเตอร์แบบ นี้จะเป็นการประมวลผลที่มีขั้นตอนซับซ้อน และจะทำกับข้อมูลขนาดใหญ่

5.3 การใช้หน่วยความจำร่วมกัน และการกระจายหน่วยความจำ

เครื่องคอมพิวเตอร์แบบขนานที่สามารถทำงานแบบ MIMD จะมีการแบ่งหน่วยความจำออกเป็นหน่วยความจำส่วนกลางซึ่งหน่วยประมวลผล จะใช้งานร่วมกัน และหน่วยความจำของแต่ละ หน่วยประมวลผลเอง การจัดแบ่งหน่วยความจำของเครื่องคอมพิวเตอร์แบบขนานถูกออกแบบแตกต่างกันไป



รูปที่ 5-1 รูปแสดงการแบ่งหน่วยความจำกันใช้ของ CPU



รูปที่ 5-2 โครงสร้างการจัดการหน่วยความจำแบบกระจาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำส่วนกลางนั้นจะใช้ในการควบคุม และจัดลำดับการทำงานของคำสั่ง
รายงานการทำงานของหน่วยประมวลผลซึ่งเชื่อมต่ออยู่ผ่านระบบเครือข่าย ซึ่งอาจจะเป็นระบบ
เครือข่ายภายนอกหรือบัสภายในเครื่องเดียวกันก็ได้

หน่วยความจำของแต่ละหน่วยประมวลผลนั้นจะทำหน้าที่ในการจัดเก็บคำสั่งและ ข้อมูล
ซึ่งหน่วยประมวลผลจะต้องใช้ทำการประมวลผลรวมถึงทำหน้าที่เป็นหน่วยความจำประเภท
แคชด้วย

5.4 การพิจารณาความเป็นอิสระของข้อมูลกับการทำงานแบบขนาน

การจะพิจารณาว่าคำสั่งชุดใดสามารถทำงานแบบขนานได้หรือไม่นั้น จะต้องพิจารณาว่า
ใน 2 ระดับด้วยกันคือ

5.4.1 ชุดคำสั่งที่ต้องการลำดับขั้นตอน หมายถึง ชุดคำสั่ง B จะต้องรอให้ชุดคำสั่ง A ทำ
งานเสร็จสิ้นลงเสียก่อนจึงจะสามารถทำงานได้

ตัวอย่าง

$$A = 10;$$

$$B = A + 10;$$

$$C = A + B;$$

5.4.2 ชุดคำสั่งที่สามารถทำงานแบบขนานกันได้

ตัวอย่าง

```
for (i = 0; i < 100; i++)
```

$$a[i] = b[i] + c[i]$$

จากชุดคำสั่งข้างบนสามารถแยกออกได้เป็น

$$a[1] = b[1] + c[1];$$

$$a[2] = b[2] + c[2];$$

$$a[3] = b[3] + c[3];$$

.....

.....

$$a[100] = b[100] + c[100];$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 การใช้หน่วยความจำร่วมกันกับ Message Passing

ในเครื่องคอมพิวเตอร์ที่ทำงานแบบ MIMD การใช้หน่วยความจำร่วมกันจำเป็นต้องมีการติดต่อสื่อสารระหว่างหน่วยประมวลผล เพื่อจัดลำดับการทำงาน โดยวิธีการก็คือแต่ละหน่วยประมวลผลจะจองพื้นที่ในหน่วยความจำส่วนกลางเพื่อทำการรับและส่งข้อมูล การส่งข้อมูลจะส่งเป็นข้อความ (Message) ซึ่งจะหมายถึงรหัสคำสั่ง โดยหน่วยประมวลผลจะทำหน้าที่เป็นทั้งผู้ร้องขอและผู้ตอบสนองในเวลาเดียวกัน

ลักษณะของการโปรแกรมแบบนี้เราสามารถพบได้กับการโปรแกรมผ่านระบบ X windows โดยอาศัยโปรแกรมจัดการเหตุการณ์ที่จะตรวจสอบการติดต่อกับผู้ใช้ แล้วส่งข้อความไปกระตุ้นให้โปรแกรมย่อยๆ ทำงาน

5.6 การกระจายงานและจัดตารางการทำงานให้กับหน่วยประมวลผล

เป็นเรื่องสำคัญมากที่จะต้องมีการจัดตารางการทำงานให้กับหน่วยประมวลผลซึ่งทำงานร่วมกัน การจัดตารางการทำงานและการกระจายงานจะกระทำได้ 2 วิธีการใหญ่ก็คือ

การจัดเตรียมตารางการทำงาน (Prescheduling) กรณีนี้จะระบบจะทำการจัดการคำสั่งต่างๆ ภายในโปรแกรม โดยจะทำการจัดการกับคำสั่งซึ่งสามารถกำหนดให้ทำงานแบบขนาน และกำหนดว่าจะให้หน่วยประมวลผลใดทำการประมวลผล

การจัดตารางแบบคงที่ (Static scheduling) เป็นงานซึ่งจะต้องกำหนดไว้เป็นหลักว่าจะต้องทำโดยหน่วยประมวลผลใด โดยจะใช้ในกรณีที่ เป็น Process หรือ Task ของระบบ

5.7 การเขียนโปรแกรมบน Convex Supercomputer

ในโครงการนี้ใช้ Editor Vi และ Pico เพื่อเขียน Source และใช้โปรแกรมช่วยเหลือซึ่งมีอยู่บนเครื่อง Supercomputer 2 โปรแกรม คือ

5.7.1 Cxda (Convex debug program) ซึ่งทำหน้าที่เป็น debugger สำหรับหาข้อผิดพลาดในโปรแกรม ทำงานภายใต้ระบบ X windows เราสามารถ Compile โปรแกรมสำหรับใช้งานร่วมกับโดย

```
% cc -cxdb myprog.c
```

เราเรียกใช้งานโปรแกรม cxdb ได้โดย

```
% cxdb & หรือ cxdb -nw สำหรับ Text Mode
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.7.2 Cxpa (Convex performance analysis) เป็นโปรแกรมช่วยในการวัดประสิทธิภาพของ source code ที่ถูกเขียนขึ้นใช้งาน ดูรายละเอียดได้ในบทถัดไป

5.8 การ Compile โปรแกรม

ในโครงการนี้เราใช้ C compiler บนเครื่อง convex (spp 1600) super computer ซึ่งโดยใช้คำสั่ง

```
% cc sourcefile -o execute_file
```

การ compile โปรแกรมบนจะมี Optimize ซึ่งทำหน้าที่ในการจัดการ Compile โปรแกรมให้สามารถทำงานแบบขนานได้ โดยจะแบ่งระดับการ Optimize ออกเป็น 4 ระดับ

- O0 มีการปรับปรุงโครงสร้างน้อยที่สุด
- O1 จะทำการปรับปรุงโครงสร้างเฉพาะส่วนที่กำหนดในโปรแกรม
- O2 ทำการปรับปรุงโครงสร้างของโปรแกรมน้อย ใน Source program
- O3 จะทำการปรับปรุงแต่ละโปรแกรมน้อยใน Source program โดยจะทำการแยก

โปรแกรมส่วนที่วนรอบออกเป็นส่วน้อย (Unroll)

โดยปกติโปรแกรมจะทำการแปลโปรแกรมโดยใช้ค่าตัวเลือก -O2

บทที่ 6

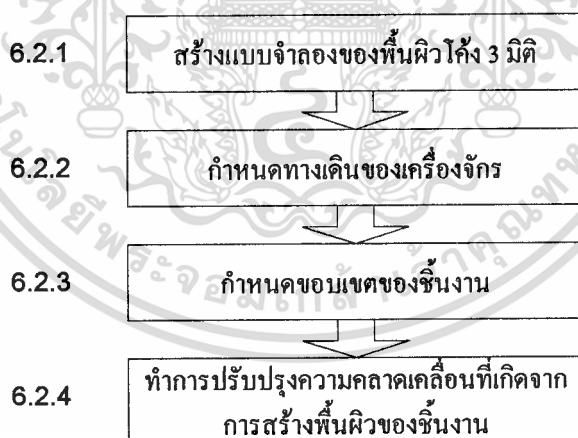
ขั้นตอนการทำงานของโปรแกรม

Program module

6.1 ขอบเขตของโปรแกรมที่ออกแบบ

- 6.1.1 โปรแกรมภายใต้ระบบปฏิบัติการ UNIX
- 6.1.2 โปรแกรมบนเครื่อง supercomputer SPP1600 ซึ่งมีหน่วยประมวลผล 8 หน่วย
โดยมีการทำงานแบบขนาน
- 6.1.3 ทำการแปลคำสั่งโดยใช้ Convex C - compiler
- 6.1.4 แสดงผลผ่านระบบ X windows โดยพัฒนาโดยใช้ Motif ช่วยในการพัฒนา

6.2 ขั้นตอนการใช้งานโปรแกรม



รูปที่ 6-1 ขั้นตอนการใช้โปรแกรมในโครงการ

ขั้นตอนการใช้โปรแกรมสำหรับประกอบด้วย

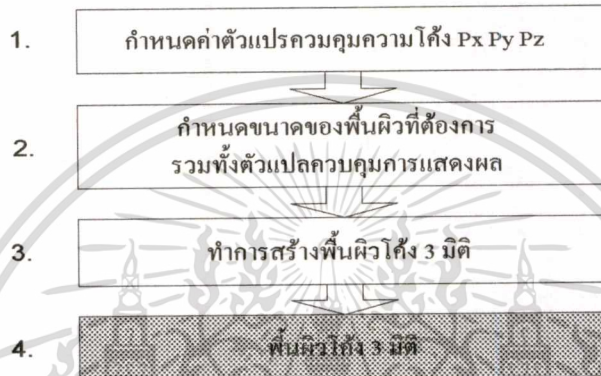
- 6.2.1 โปรแกรมสำหรับการสร้างแบบจำลองของพื้นผิว
- 6.2.2 โปรแกรมสร้างพื้นผิวของชิ้นงาน
- 6.2.3 โปรแกรมกำหนดขอบเขตของชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.4 กลุ่มโปรแกรมสำหรับการปรับปรุงความคลาดเคลื่อน ที่เกิดจากการแปลงพื้นผิวเพื่อให้เครื่องจักรใช้เป็นทางเดินของเครื่องมือ

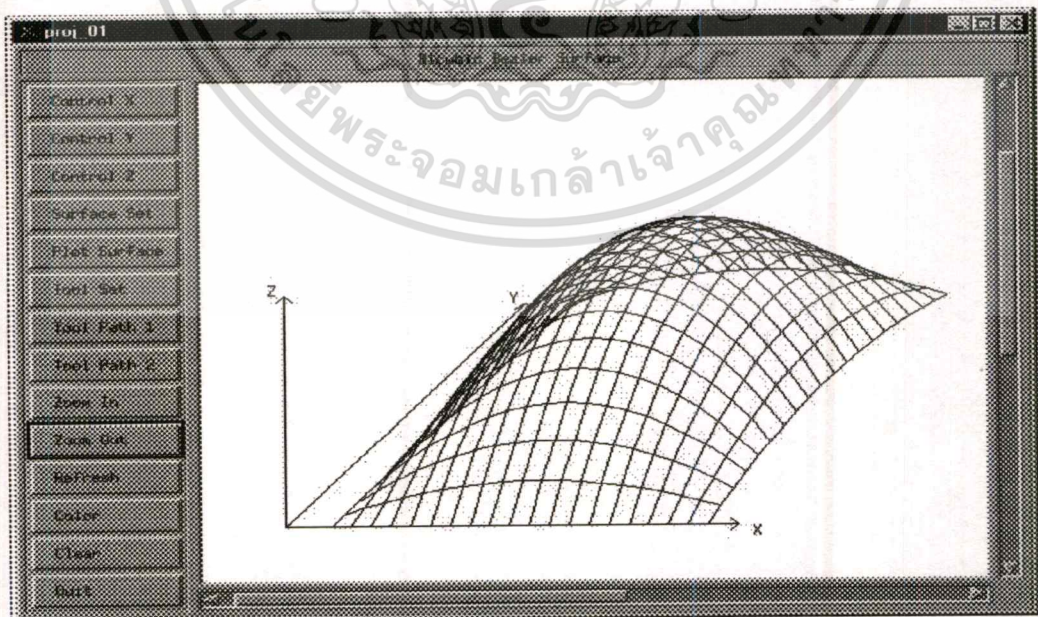
6.2.1 โปรแกรมสำหรับการสร้างแบบจำลองของพื้นผิว

การสร้างแบบจำลองของพื้นผิวมีขั้นตอนการสร้างพื้นผิวโค้ง 3 มิติ ดังนี้



รูปที่ 6-2 ขั้นตอนการสร้างพื้นผิวของชิ้นงาน

จากรูปที่ 6-2 เราต้องทำการกำหนดค่าตัวแปรต่างๆ (1) กำหนดขนาดของพื้นผิวที่ต้องการทดลอง (2) แล้วสั่งให้โปรแกรมทำการสร้างพื้นผิว (3) จะได้เป็นพื้นผิวโค้ง 3 มิติ (4)



รูปที่ 6-3 แสดงโปรแกรมกำลังแสดงผลการสร้างโปรแกรม 3 มิติ

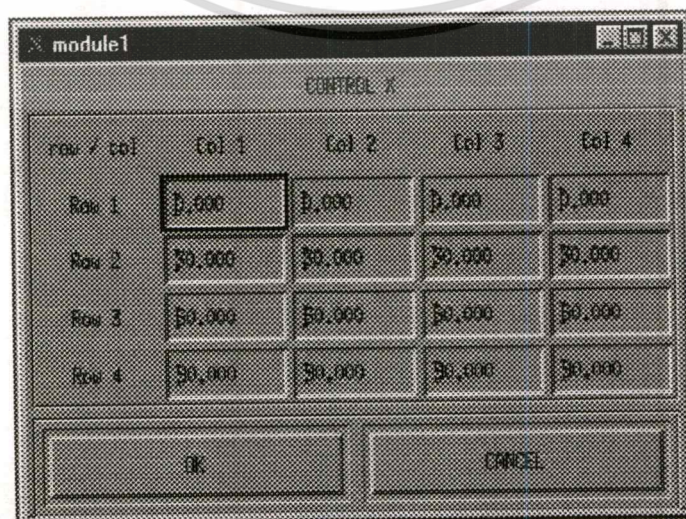
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 6-3 แสดงโปรแกรมที่ทำหน้าที่ในการออกแบบพื้นผิวซึ่งจะประกอบด้วยตัวเลือกให้ทำงานดังนี้

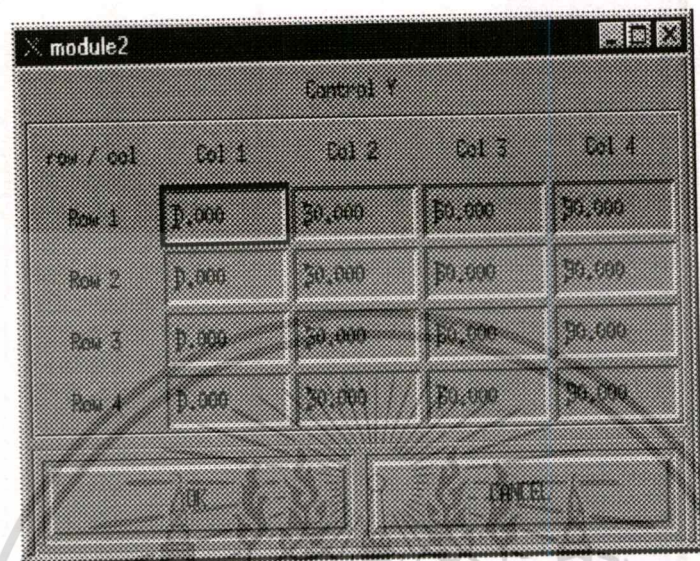
Control X	กำหนดตัวแปรควบคุมความโค้งของพื้นผิวตามแนว X ดังแสดงในรูปที่ 6-4
Control Y	กำหนดตัวแปรควบคุมความโค้งของพื้นผิวตามแนว Y ดังแสดงในรูปที่ 6-5
Control Z	กำหนดตัวแปรควบคุมความโค้งของพื้นผิวตามแกน Z ดังแสดงในรูปที่ 6-6
Surface Set	กำหนดค่าตัวแปรสำหรับการสร้างพื้นผิวโค้งและการแสดงผล ดังรูปที่ 6-7
Surface Plot	ทำการแสดงผลพื้นผิวที่ออกแบบบนจอภาพ
Tool Set	กำหนดค่าตัวแปรสร้างทางเดินของเครื่องมือ
Tool Path 1	กำหนดการสร้างทางเดินของเครื่องมือตามแนวแกน X
Tool Path 2	กำหนดการสร้างทางเดินของเครื่องมือตามแนวแกน Y
Zoom In	ลดขนาดของภาพที่แสดง
Zoom Out	ขยายภาพ
Refresh	ทำการลบและพื้นผิวใหม่
Color	กำหนดของพื้นผิวที่จะแสดงดังรูปที่ 6-8
Clear	ทำการลบจอภาพ
Quit	ออกจากโปรแกรม

โปรแกรมในกลุ่มนี้จะทำหน้าที่ในการสร้างพื้นผิว 3 มิติ โดยจะต้องกำหนดค่าตัวแปรดังนี้

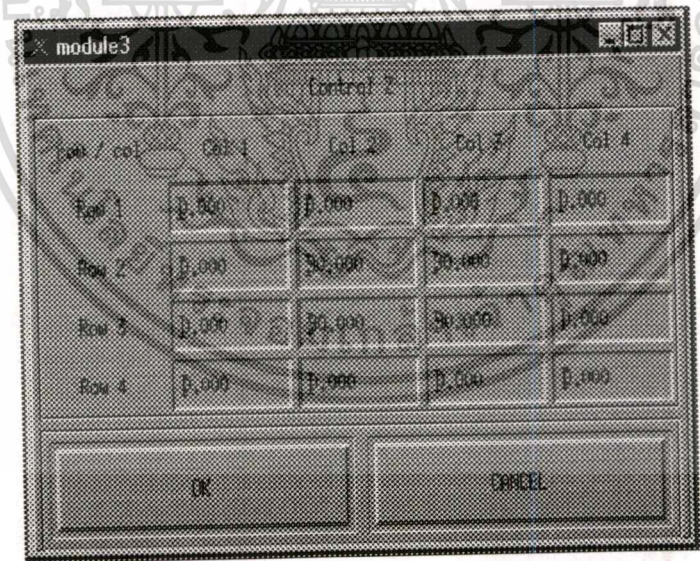
6.2.1.1 กำหนดค่าตัวแปรควบคุมการสร้างพื้นผิว 3 มิติ Px Py Pz



รูปที่ 6-4 รูปแสดงการรับค่าสำหรับตัวแปรควบคุม Px



รูปที่ 6-5 รูปแสดงการรับค่าสำหรับตัวแปรควบคุม Py



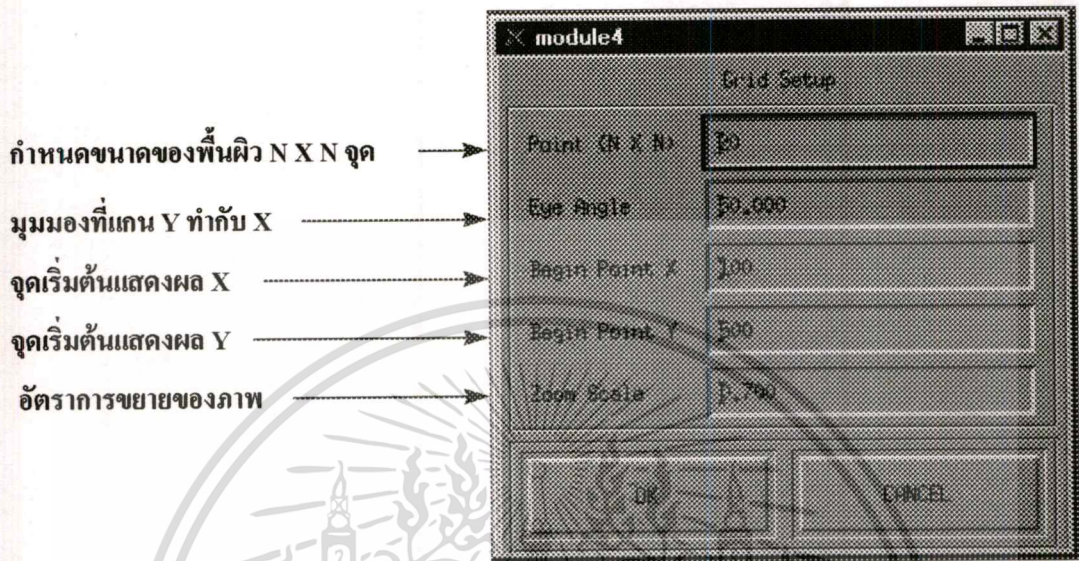
รูปที่ 6-6 รูปแสดงการรับค่าสำหรับตัวแปรควบคุม Pz

จากรูปที่ 6-4 ถึง 6-6 ผู้ใช้จะต้องกำหนดค่าให้กับตัวแปรใน 3 ระนาบให้เสร็จทั้งหมดก่อน จึงจะได้พื้นผิวโค้ง 3 มิติที่สมบูรณ์ โดยดูรายละเอียดเกี่ยวกับหน้าที่ของตัวแปรควบคุมความโค้ง

ได้จากบทที่ 2

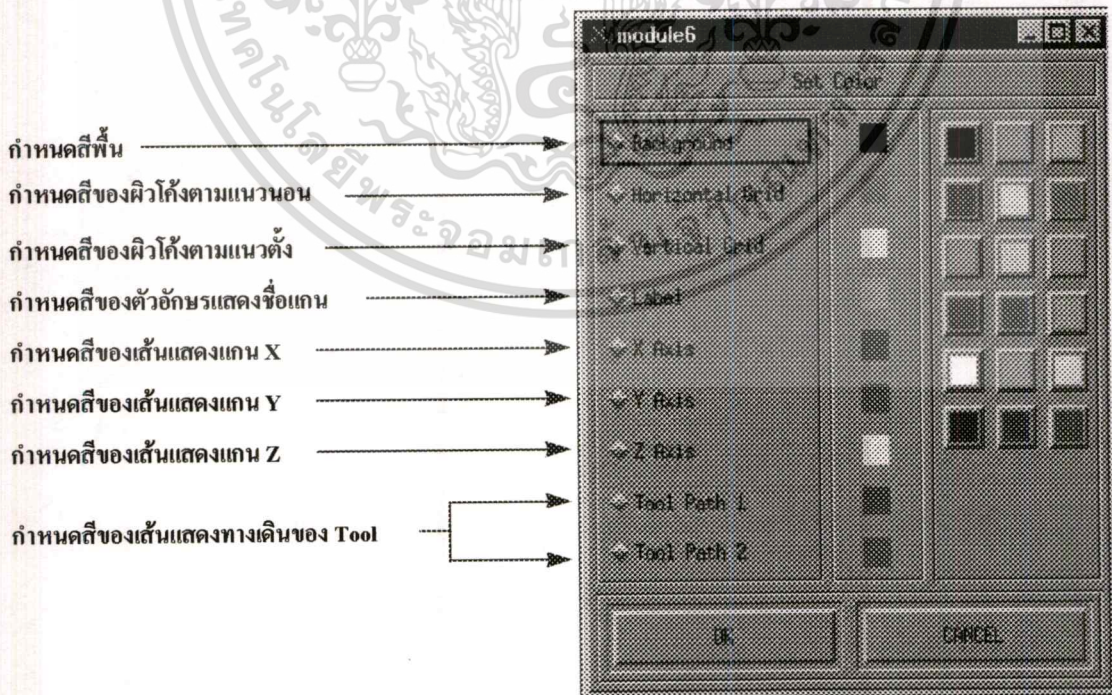
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.1.2 กำหนดค่าตัวแปรสำหรับการสร้างพื้นผิว ประกอบด้วย



รูปที่ 6-7 โปรแกรมส่วนกำหนดขนาดพื้นผิวและการแสดงผล

6.2.1.3 การกำหนดสีของพื้นผิวด้านแบบที่จะแสดงบนจอภาพ



รูปที่ 6-8 แสดงโปรแกรมส่วนกำหนดสีของพื้นผิวด้านบนจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.2 โปรแกรมสร้างพื้นผิวของชิ้นงาน

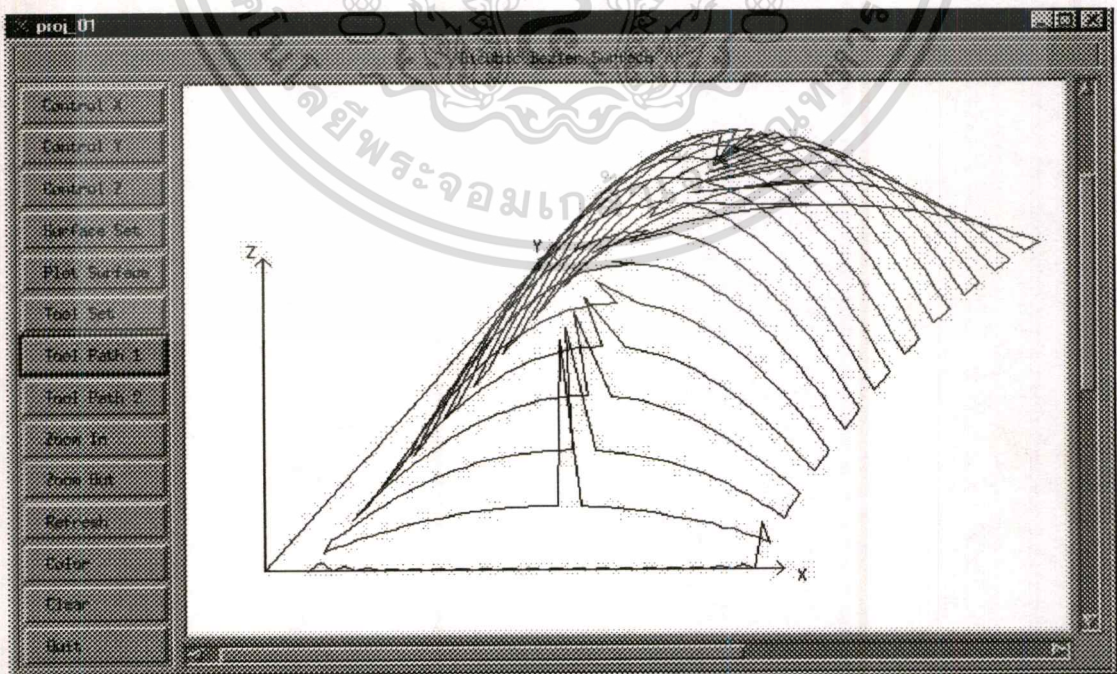
ในโครงการนี้เกิดการเลือกทางเดินของเครื่องจักรเป็นแบบสลับพื้นปลา โดยเลือกเดินใน 2 แนว โดยผู้ใช้โปรแกรมจะต้องเลือกเพียงแบบใดแบบหนึ่งเท่านั้น

1. สลับพื้นปลาตามแนวแกน X
2. สลับพื้นปลาตามแนวแกน Y



รูปที่ 6-9 ขั้นตอนการสร้างพื้นผิวของชิ้นงาน

เมื่อเราได้พื้นผิวโค้ง (1) แล้วเราจะต้องกำหนดค่าตัวแปรสำหรับสร้างทางเดินของเครื่องจักร (2) เมื่อกำหนดค่าตัวแปรต่างๆ เรียบร้อยแล้วจะทำการเลือกทางเดินของเครื่องจักร (3) เมื่อเลือกเสร็จเรียบร้อยแล้วจะแสดงทางเดินของเครื่องจักร (4)



รูปที่ 6-10 รูปแสดงการสร้างทางเดินของเครื่องมือสำหรับเครื่องจักร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 6-10 แสดงการกำหนดทางเดินของเครื่องมือให้เครื่องจักรทำงาน โดยในขั้นตอนนี้จะต้อง

6.2.2.1 การกำหนดค่าตัวแปรสำหรับสร้างทางเดินของเครื่องมือ

6.2.2.2 การเลือกทางเดินของเครื่องมือให้เครื่องจักรทำงาน

ในขั้นตอนนี้ผู้ใช้งานโปรแกรมจะต้องทำการกำหนดค่าตัวแปรสำหรับเครื่องจักรขนาด 5 แกน ซึ่งได้อธิบายรายละเอียดไว้ในบทที่ 3 ของรายงานฉบับนี้

6.2.2.1 การกำหนดค่าตัวแปรสำหรับสร้างทางเดินของเครื่องมือ

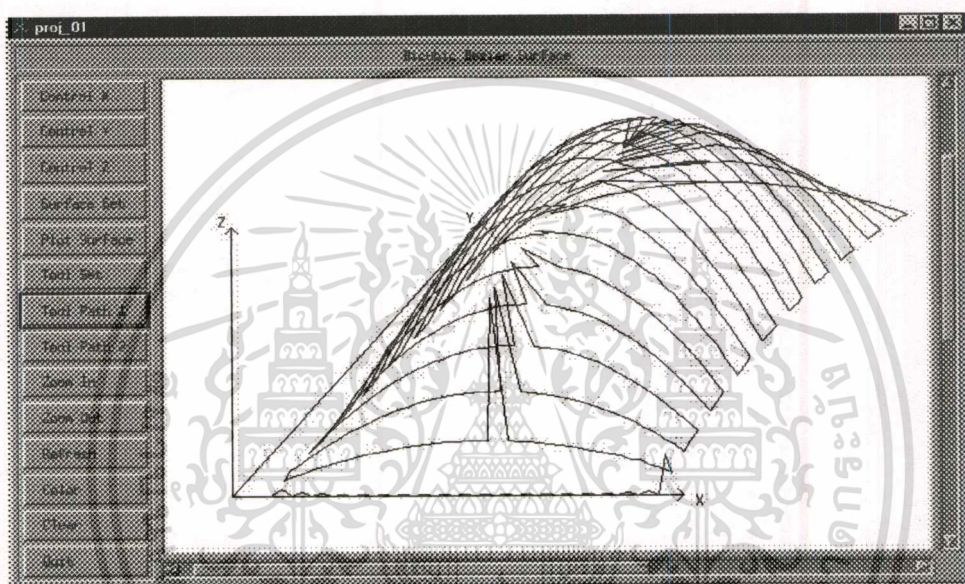
Label (Thai)	Parameter	Value
จุดเริ่มต้นบนแกน A	Origin Worktable X	115.000
จุดเริ่มต้นบนแกน B	Origin Worktable Y	0.000
จุดเริ่มต้นบนแกน C	Origin Worktable Z	125.000
ค่าความคลาดเคลื่อนของแกน B	Accuracy Y-axis	0.001
ระยะจากปลายเครื่องมือถึงจุดเริ่มต้นของเครื่อง	Tool tip to machine origin	119.000
ระยะจากแกน A ถึงแกน B	Travel X to Y-axis	251.168
ระยะจากแกน B ถึง จุดเริ่มต้นของโรงงาน	Travel Y to machine coordinate	371.313

รูปที่ 6-11 แสดงรายละเอียดของโปรแกรมส่วนกำหนดค่าตัวแปรของเครื่องจักร การกำหนดทางเดินของเครื่องจักร เป็นการกำหนดค่าตัวแปรเพื่อบอกจุดต่างๆ ซึ่งจะวางชิ้นงานลงบนแท่นวางชิ้นงานของเครื่องจักร ซึ่งได้กล่าวรายละเอียดไว้ในบทที่ 3 แล้วอย่างละเอียด การกำหนดค่าตัวแปรแต่ละตัวจะมีผลถึงการทำงานเพื่อสร้างเป็นพื้นผิวของชิ้นงาน ในบทที่ 4 ซึ่งได้กล่าวถึงหลักการสร้างปรับรูปร่างเพื่อให้ใกล้เคียงกับระนาบต้นที่ต้องการนั้น เราได้พบว่าค่าต่างๆที่เป็นสมการการสร้างชิ้นงานของเครื่องจักรนั้นขึ้นอยู่กับประเภทของเครื่องจักร แต่ในโครงการนี้จะอาศัยเพียงเครื่องจักรอุตสาหกรรมขนาด 5 แกน เป็นหลักในการสร้างแบบจำลอง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

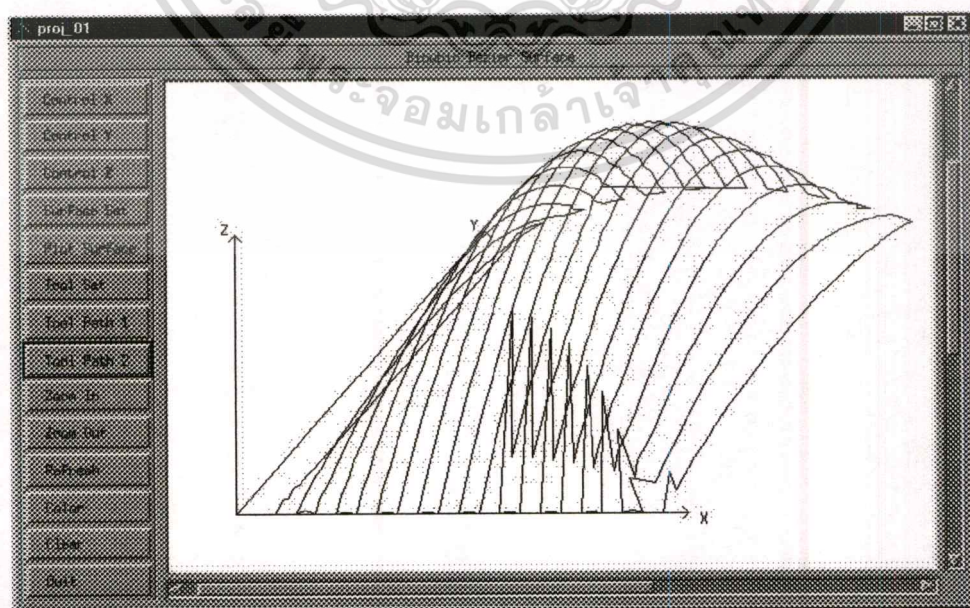
6.2.2.2 การเลือกทางเดินของเครื่องมือให้เครื่องจักรทำงาน

การกำหนดทางเดินของเครื่องจักรในโปรแกรมจะมีการกำหนดทางเดินของเครื่องจักรเป็นแบบสลัฟพื้นปลา โดยเลือกทางเดินของเครื่องจักรเป็น 2 ทางคือตามแนวแกน X และแกน Y ดังแสดงในรูปที่

6-12 และ 6-13

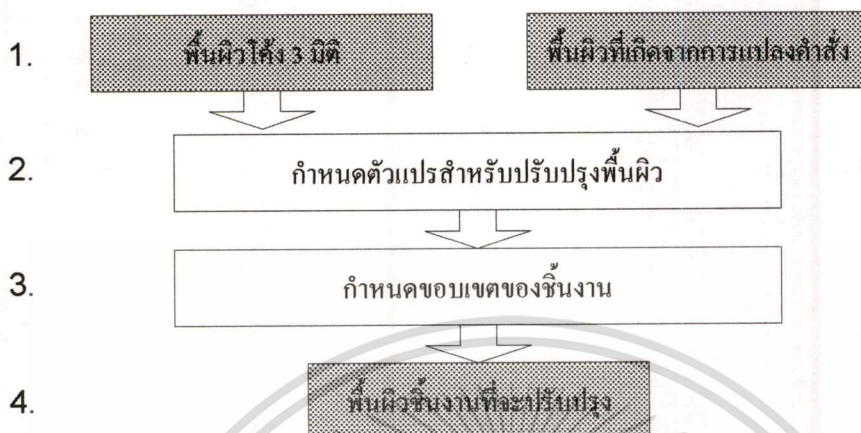


รูปที่ 6-12 รูปแสดงทางเดินของเครื่องมือที่เครื่องจักรแบบ 5 แกนใช้ กำหนดทิศทางตามแกน X



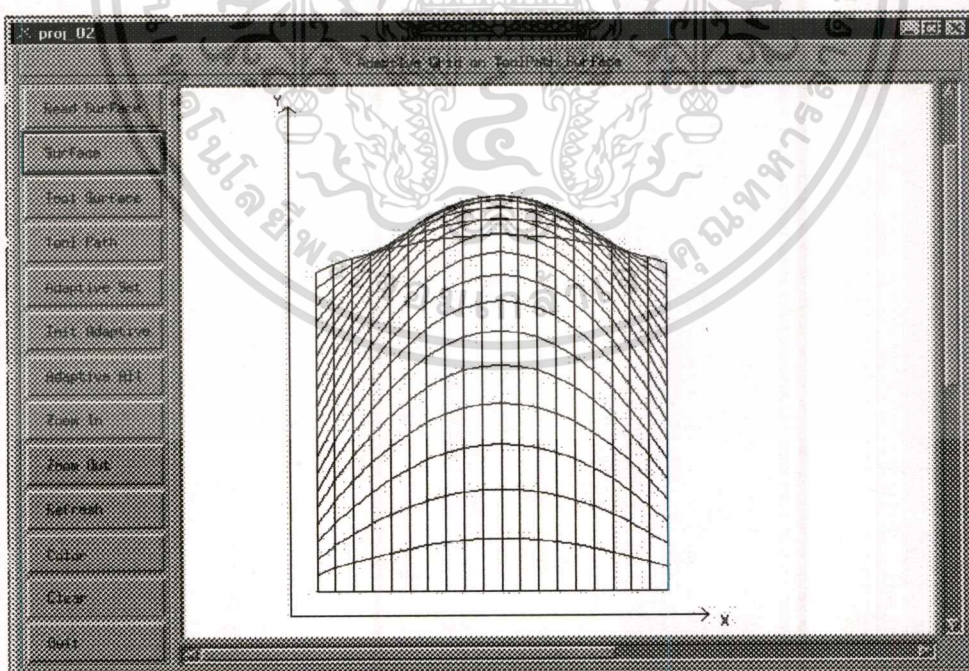
รูปที่ 6-13 รูปแสดงทางเดินของเครื่องมือที่เครื่องจักรแบบ 5 แกนใช้ กำหนดทิศทางตามแกน Y
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.3 โปรแกรมกำหนดขอบเขตของชิ้นงาน



รูปที่ 6-14 ขั้นตอนการกำหนดขอบเขตของพื้นผิวชิ้นงาน

จากรูปที่ 6-14 โปรแกรมในส่วนนี้จะนำพื้นผิวโค้ง 3 มิติต้นแบบ และพื้นผิวที่เกิดจากการแปลค่าสั่ง (1) ทำการกำหนดค่าให้ตัวแปรที่ใช้ในการปรับปรุง (2) การกำหนดขอบเขตของชิ้นงาน (3) ได้เป็นพื้นผิวที่จะทำการปรับปรุง (4) เพื่อลดความคลาดเคลื่อน

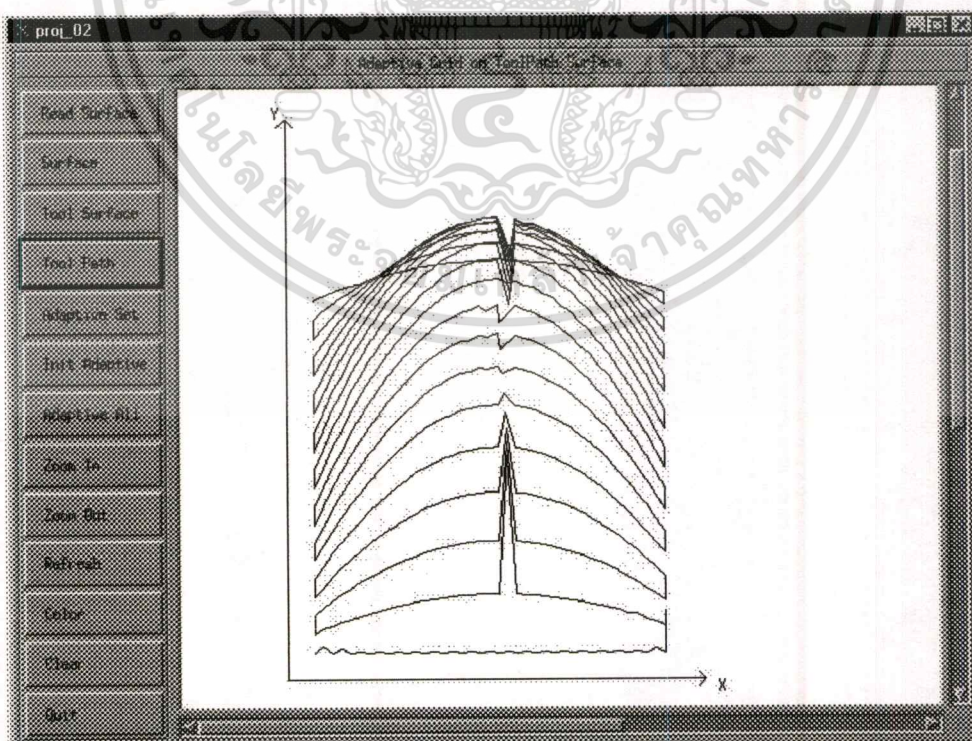


รูปที่ 6-15 ชิ้นงานซึ่งออกแบบโดยอาศัยการสร้างเส้นโค้ง 3 มิติ แสดงผลใน 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

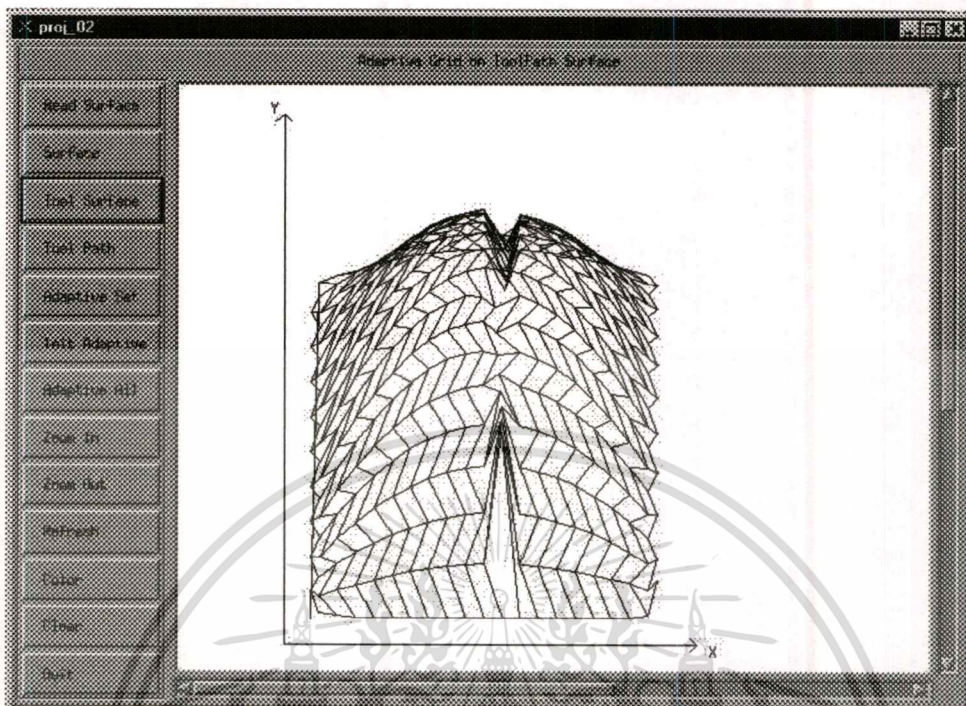
รายละเอียดของตัวเลือกที่แสดงในรูปที่ 6-15

Read Surface	รับค่าพื้นผิวโค้ง 3 มิติ ที่ออกแบบไว้และแสดงผลในแบบ 2 มิติ ดังรูปที่ 6-15
Surface	แสดงค่าพื้นผิวโค้ง 3 มิติ ในแบบ 2 มิติ
Tool Surface	แสดงพื้นผิวของชิ้นงานที่จะได้จากโปรแกรมแปลค่า ดังรูปที่ 6-17
Tool Path	แสดงทางเดินของเครื่องมือที่เครื่องจักรขนาด 5 แกนใช้ ดังรูปที่ 6-16
Adaptive Set	ทำการสำหรับการปรับปรุงพื้นผิว ดังรูปที่ 6-18
Initial Adaptive	กำหนดพื้นผิวดั้งสำหรับการปรับปรุง ดังรูปที่ 6-19
Adaptive All	เรียกโปรแกรมเพื่อทำการปรับปรุงพื้นผิว
Zoom In	ลดขนาดของภาพที่แสดง
Zoom Out	ขยายภาพ
Refresh	ทำการลบและพื้นผิวใหม่
Color	กำหนดสีของพื้นผิวที่จะแสดง ดังรูปที่ 6-20
Clear	ทำการลบจอภาพ
Quit	ออกจากโปรแกรม



รูปที่ 6-16 รูปแสดงทางเดินของเครื่องมือที่เครื่องจักรขนาด 5 แกนใช้

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์เพื่อการเรียนการสอนเท่านั้น เมื่อผู้เช่าได้เห็น ใบโฆษณาหรือโฆษณาการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-17 รูปแสดงพื้นผิวที่จะได้จากการแปลงพื้นผิวโค้ง 3 มิติ ที่ออกแบบมาให้เครื่องจักรทำงาน

อัตราการปรับปรุงพื้นผิว

ขนาดของความคลาดเคลื่อนต่ำสุด

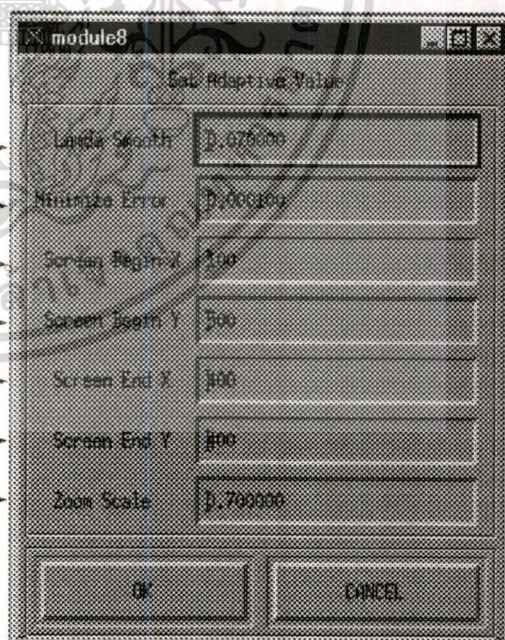
จุดเริ่มต้นแสดงผลบนจอภาพ X

จุดเริ่มต้นแสดงผลบนจอภาพ Y

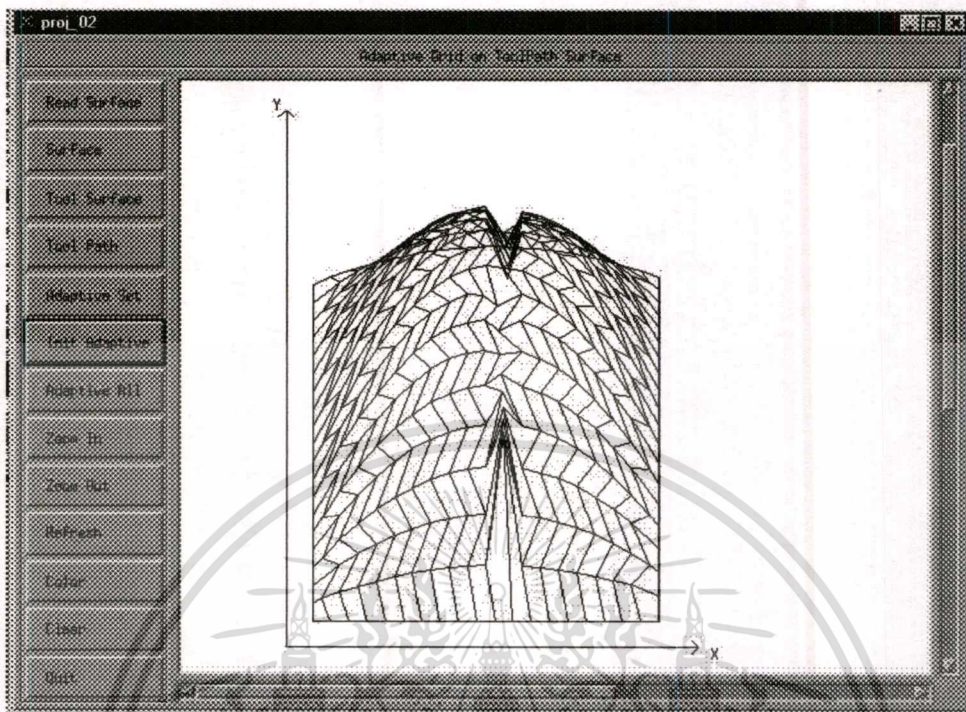
จุดสิ้นสุดการแสดงผล X

จุดสิ้นสุดการแสดงผล Y

อัตราการขยายของภาพ



รูปที่ 6-18 รูปแสดงการกำหนดค่าสำหรับการปรับปรุงพื้นผิว



รูปที่ 6-19 กำหนดของเขตของพื้นผิวที่จะทำการปรับปรุง

กำหนดสีพื้น

กำหนดสีของผิวโค้งตามแนวแกน X

กำหนดสีของผิวโค้งตามแนวแกน Y

กำหนดสีของตัวอักษรแสดงชื่อแกน

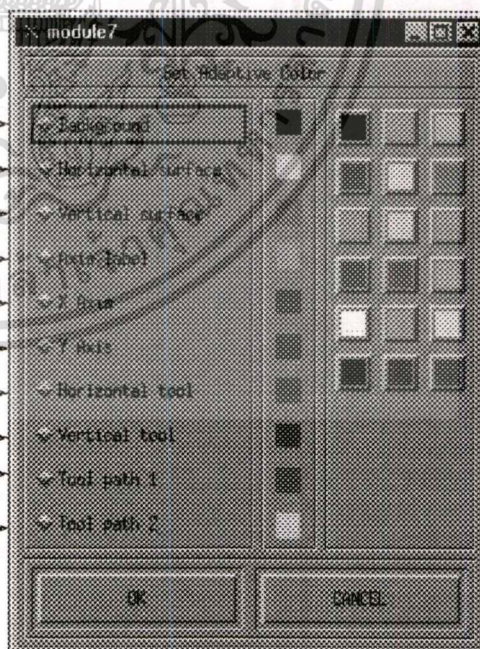
กำหนดสีของเส้นแสดงแกน X

กำหนดสีของเส้นแสดงแกน Y

กำหนดสีของ Tool ตามแนวแกน X

กำหนดสีของ Tool ตามแนวแกน Y

กำหนดสีของเส้นแสดงทางเดินของ Tool

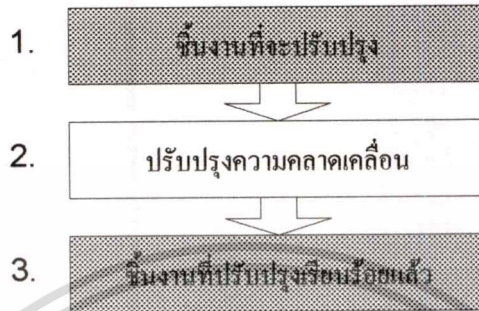


รูปที่ 6-20 รูปแสดงการโปรแกรมส่วนที่กำหนดสีของพื้นผิวที่จะแสดง

จากรูปที่ 6-20 เป็นการกำหนดสีที่ใช้ในการแสดงผลสำหรับโปรแกรมที่ใช้ในการกำหนด

ของเขตของโปรแกรมไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

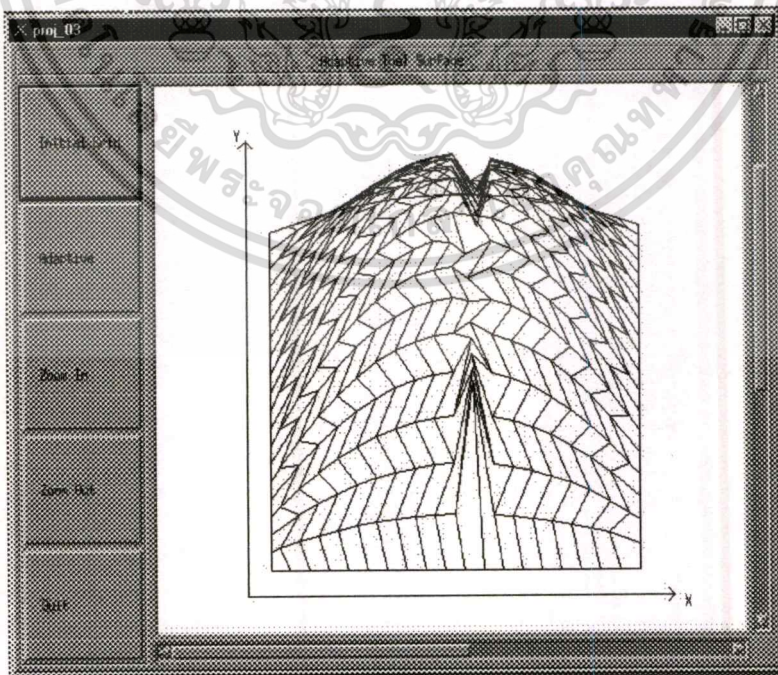
6.2.4 กลุ่มโปรแกรมสำหรับการปรับปรุงความคลาดเคลื่อน ที่เกิดจากการแปลงพื้นผิวเพื่อให้เครื่องจักรใช้เป็นทางเดินของเครื่องมือ



รูปที่ 6-21 ขั้นตอนการปรับปรุงพื้นผิว

เมื่อได้พื้นผิวที่จะสร้างเป็นชิ้นงาน (1) เราจะทำการปรับปรุงเพื่อลดความคลาดเคลื่อนของพื้นผิว (2) โดยจะทำการปรับปรุง 10 ครั้ง ต่อการแสดงผลบนจอภาพ 1 ครั้ง เมื่อทำการปรับปรุงเสร็จแล้วจะได้พื้นผิวที่กำหนดความคลาดเคลื่อนออกไป (3)

6.2.4.1 การแสดงพื้นผิวที่จะทำการปรับปรุง

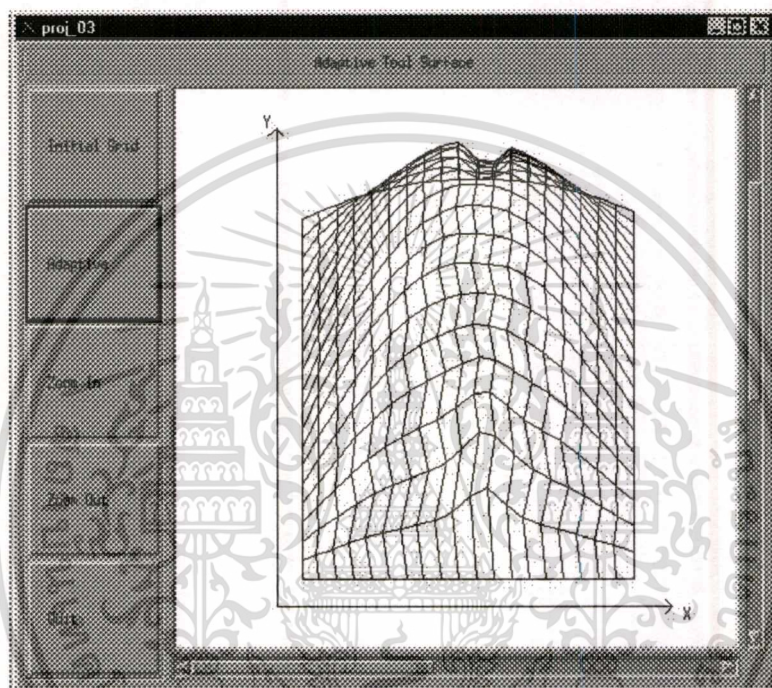


รูปที่ 6-22 รูปแสดงการพื้นผิวที่กำหนดขอบเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้พื้นผิวที่จะทำการปรับปรุงโปรแกรมในส่วนนี้จะทำการอ่านข้อมูลจากแฟ้มข้อมูล ซึ่งโปรแกรมที่ทำหน้าที่ในการกำหนดขอบของพื้นผิวกำหนดมาให้ ถ้าต้องการปรับปรุงพื้นผิวใหม่ ให้เลือกปุ่มแรกของหน้าจอ

6.2.4.2 การปรับปรุงพื้นผิว



รูปที่ 6-23 รูปแสดงพื้นผิวที่ปรับปรุงไประยะหนึ่ง

จากรูปที่ 6-23 แสดงให้เห็นการปรับปรุงพื้นผิวไประยะหนึ่งในโครงการนี้จะให้ทำการปรับปรุงพื้นผิว 10 รอบต่อการแสดงผล 1 ครั้ง

บทที่ 7

บทสรุป

Summary

7.1 การวัดประสิทธิภาพของโปรแกรมบนซูเปอร์คอมพิวเตอร์

เนื่องจากการวัดประสิทธิภาพของโปรแกรมที่ทำงานในระบบซูเปอร์คอมพิวเตอร์ ซึ่งทำงานในมีลักษณะดังนี้

1. เป็นการประมวลผลแบบขนาน (Parallel Processing)
2. เป็นการประมวลผลแบบหลายงานพร้อมกัน (Multitasking)

จากลักษณะดังกล่าวเพื่อความแน่นอนและเชื่อถือได้ ในการวัดประสิทธิภาพของโปรแกรมในโครงการนี้อาศัย โปรแกรม Cxpa (Convex Performance Analysis) เป็นโปรแกรมที่ใช้วัดประสิทธิภาพของ โปรแกรมที่พัฒนาขึ้นโดยโปรแกรมภาษา C และ โปรแกรมภาษา Fortran โดยจะแสดงใน 2 ระบบ การทำงานในระบบ X windows เนื่องจากการแสดงผลในระบบ X windows จะต้องอาศัยความสามารถในการแสดงผลสูง ผู้ใช้จะต้องทำงานบนเครื่อง X terminal เพียงอย่างเดียว มีขั้นตอนการใช้งานดังต่อไปนี้

1. ผู้ใช้จะต้องใช้คำสั่ง `setenv DISPLAY mydisplay:0.0`
2. จะต้องการ Compile โปรแกรมและ Link โปรแกรมโดยใช้ Option `-cxpa` ดังนี้
ในภาษา C จะใช้ `cc -cxpa prog.c` ในภาษา Fortran จะใช้ `fc - cxpa prog.f`
3. การเรียกใช้โปรแกรม `cxpa` จะใช้คำสั่ง `cxpa a.out &`

7.2 การใช้งานใน Text Mode (CXPA)

1. ทำการ Compile โปรแกรม โดยใช้ Option `-cxpa`
`cc -cxpa prog.c` สำหรับ Compiler ภาษา c
2. เรียกใช้โปรแกรม `cxpa` ใน Line Mode
`cxpa -nw a.out`

โปรแกรมจะแสดง prompt mode เพื่อใส่คำสั่งเป็น Text mode

Type 'help' for help.

Reading executable a.out...

Selecting profile a.out.pdf...

(Cxp) <- นี่คือนัด prompt ใช้สำหรับใส่คำสั่ง

3. ทำการเลือกพื้นที่หรือหมายเลขของคำสั่ง ที่ต้องการเริ่มต้นทำการวัดประสิทธิภาพ

(Cxp) select loop all

4. เลือกประเภทของการวัดเป็น

- CPU - ทำการรวบรวม CPU time คือเวลาที่ CPU ทำงาน

- wall clock - ทำการรวบรวมเวลาที่ใช้ในการแก้ปัญหา (Time to solution)

- events - ทำการรวบรวมเวลาที่เข้าถึงหน่วยความจำ

- vector flops ทำการรวบรวมข้อมูลที่ vector register ใช้ในการประมวลผล

5. ถ้าต้องการกำหนดให้ทำการ collect สำหรับ event ให้ใช้คำสั่ง

(Cxp) set events local_misses

6. ใช้คำสั่ง Run สำหรับให้โปรแกรมเริ่มทำการประมวลผล

(Cxp) run

7. เมื่อโปรแกรมทำงานจบลง จะแสดง prompt ให้ทำการสังเคราะห์ข้อมูล

(Cxp) analyze

8. เมื่อต้องการรายงานเพื่อทำการออกรายงานลงแฟ้มข้อมูล

(Cxp) analyze > report.file

9. เมื่อต้องการออกจากโปรแกรมให้ใช้คำสั่ง Quit

(Cxp) quit

7.3 การเขียนใช้ Cxp จาก batch mode จะใช้ คำสั่ง

```
cxp -x < batch file > a.out < <input_file> > & <output_file>
```

ตัวอย่างการสร้าง batch file สำหรับ run program ใน project นี้

```
#This is a command for adaptive grid performance report
```

```
select all
```

```
collect cpu wall_clock
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
run
analyze > cxa.report
quit
```

7.4 การสั่ง run cxa ร่วมกัน nohup (no hangup)

คำสั่ง nohup เป็นคำสั่งที่ใช้ในการสั่งให้เครื่อง super computer ทำงานการประมวลผลเป็น Daemon

ซึ่งจะสามารถ run process ทิ้งไว้โดยไม่ต้องมีการ login อยู่ที่ terminal ตลอดเวลา โดยเราจะสั่ง nohup ร่วมกัน cxa process ได้ดังนี้

```
$ nohup cxa -x cmdfile
```

เมื่อสั่งให้เครื่องทำการประมวลผลเป็นแบบ nohup โปรแกรมนี้จะสร้างแฟ้มข้อมูล Text ขึ้นมาเพื่อทำการเก็บรายงานชื่อว่า nohup.out เราสามารถสั่งให้แสดงผลทางจอภาพ ได้โดยการใช้คำสั่ง

```
$ tail nohup.out
```

7.5 ข้อควรระวังในการใช้โปรแกรม nohup cxa

1. การสั่งใน Text mode ไม่ควรโปรแกรมที่มีการติดต่อกับจอภาพ เนื่องจากการติดต่อผ่านระบบ X windows โปรแกรมจะไม่ทำงาน ซึ่งอาจเกิด process
2. ในบางกรณีโปรแกรมไม่สามารถจบการประมวลผลได้เอง หรือใช้เวลานานเกินไป ผู้สั่ง run โปรแกรมต้อง สั่ง ยกเลิก process (Kill)
3. ถ้าต้องการ nohup กับหลายแฟ้มข้อมูลเพิ่มเติมควรสั่ง redirection ลงแฟ้มข้อมูลอื่นๆ เพื่อลดความสับสนในการรายงานผลการทำงานของโปรแกรม

7.6 ตารางสรุปเปรียบเทียบผลการทำงานของโปรแกรมส่วนปรับปรุงพื้นผิว (Adaptive Surface)

โดยจะพิจารณาจาก

- CPU Time เวลาที่ใช้ในการประมวลผลของแต่ละ CPU
- Wall Clock Time เวลาที่ใช้ในการจัดเตรียมข้อมูล รวมทั้งเวลาที่ CPU ว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{ค่า } W = \sin(2.0 * \pi * X[i][j] / 25.0) * \sin(2.0 * \pi * Y[i][j] / 25.0) * 1.0 / 1000.0$$

จำนวนของจุด	CPU Time (หน่วย วินาที)	Wall Clock Time (หน่วย วินาที)
2500	479.352	1148.793
4900	992.345	1850.075
10000	2623.686	4807.246
22500	9983.924	18614.645

$$\text{ค่า } W = \exp(-\text{fabs}(25.0 - (X[i][j] - \text{nx}) * (X[i][j] - \text{nx}) - (Y[i][j] - \text{ny}) * (Y[i][j] - \text{ny}))) + 0.01);$$

จำนวนจุด	CPU Time (หน่วย วินาที)	Wall Clock Time (หน่วย วินาที)
10000	12715.816	43173.387

ตาราง 7-1 การวัดประสิทธิภาพการทำงานของโปรแกรมส่วนปรับปรุงพื้นผิว

CPU Time : เวลาที่ใช้ในการประมวลผล

Wall Clock Time : เวลาที่ใช้ในการรอการทำงานของ Process ก่อนหน้า

7.7 เปรียบเทียบประสิทธิภาพของโปรแกรม

จากทดสอบแปลโปรแกรมและปรับปรุงโปรแกรมโดยอาศัย Compiler ในแต่ตัวเลือก โดยทำการทดสอบกับโปรแกรมส่วนปรับปรุงพื้นผิว โดยวัดตั้งแต่รอบการทำงานที่ 1 ถึง 10 ได้ผล ดังนี้

ตัวเลือก (Optimization Option)	CPU Time (secs)	Wall Clock Time (secs)	Threads	รวมเวลา
-O0	165.577	169.532	1	335.089
-O1	165.577	169.532	1	335.089
-O2	165.882	168.198	1	334.080
-O3	169.448	171.057	8	340.505

7-2 ตารางเปรียบเทียบความเร็วในการประมวลผลจากการ compile แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากการวัดประสิทธิภาพการทำงานของโปรแกรมเราอาศัยความสามารถของ compiler ในการจัดการและแปลงโปรแกรมในระหว่างที่ทำการแปลโปรแกรม ซึ่งโดยสรุปเราจะพบว่า ตัวเลือก -O2 ให้ความเร็วในการประมวลผลที่เร็วที่สุด

7.8 ประสิทธิภาพของวิธีการแก้ปัญหา

เป็นการวัดประสิทธิภาพในแง่ของการลดลงของความคลาดเคลื่อน ที่เกิดจากการทำงานของโปรแกรม

ขนาดพื้นผิว	จำนวนรอบที่ปรับปรุง	ความคลาดเคลื่อนของพื้นผิวดั้ง	ความคลาดเคลื่อนของพื้นผิวที่ปรับ
100	30.0000	0.14233	0.09538
400	15.0000	0.05692	0.02845
1600	7.5000	0.01594	0.00715
6400	1.87500	0.00421	0.00268
10000	0.93750	0.00161	0.00096
14400	0.46875	0.00099	0.00041

ตาราง 7-3 แสดงประสิทธิภาพในการลดความคลาดเคลื่อน

การวัดจำนวนการปรับปรุงค่าความคลาดเคลื่อนเป็นเรื่องที่กระทำได้ยากมาก เนื่องจากในบทที่ 4 เราได้กล่าวถึงทฤษฎีการปรับปรุงระนาบโค้ง 3 มิติ ใน 2 ทิศทางพร้อมกันเราพบว่าการปรับปรุงพื้นผิวแต่ละครั้งนั้นจะเกิดค่าความคลาดเคลื่อนใหม่ขึ้นเสมอ ดังนั้นปัญหาที่พบก็คือการหาค่า w ซึ่งเป็นตัวแปรสำหรับควบคุมการสร้างพื้นผิวใหม่ โดยค่า w ต้องไม่มีค่าเป็น 1 โดยทั่วไปเราจะหาจากสมการของเครื่องจักร ซึ่งจะต้องกำหนดลักษณะของพื้นผิวให้ขึ้นกับสมการที่ไม่เป็นเชิงเส้น ด้วยเหตุผลดังกล่าวจึงมีข้อจำกัดที่ต้องระบุในภาคผนวก ข.

7.9 แนวทางการพัฒนา

1. การพัฒนาโดยการปรับปรุงพื้นผิวดั้ง ตามทฤษฎีในบทที่ 2 เรื่องการสร้างพื้นผิวยังมีทฤษฎีกล่าวเพิ่มเติมถึงการต่อระนาบ 3 มิติ ซึ่งเกิดการการพยายามสร้างรูปทรงแบบผสมผสาน
2. การเพิ่มประเภทของเครื่องจักรหรือการใช้งานเครื่องจักรประเภทอื่นที่นอกเหนือจาก

เครื่องจักรอุตสาหกรรมอัตโนมัติแบบ 5 แกน การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การวัดประยุตติใช้วิธีการของ Adaptive Grid Generation ในบทที่ 4 ของโครงการนี้ เราได้อาศัยเพียงบางส่วนของสมการ ซึ่งตามทฤษฎีได้กล่าวไว้เพิ่มเติมถึงการกำหนดความถี่ของเส้นตัดในมุมระนาบเดียวกัน

4. การเขียนโปรแกรมแบบขนานที่สมบูรณ์ เนื่องจากในโครงการนี้ได้อาศัยความสามารถของ compiler ในการแปลง Code โปรแกรมซึ่งเป็นแบบโครงสร้างอยู่แล้วไปเป็นแบบขนานเพื่อให้เครื่อง supercomputer ทำงาน แนวทางใหม่ก็คือการพัฒนาโปรแกรมโดยอาศัยโครงสร้างของโปรแกรมแบบขนาน ซึ่งน่าจะทำให้เห็นประสิทธิภาพการทำงานของอัลกอริทึมที่ใช้อย่างชัดเจน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Reference

- Adrian, Biran., Moshe Breiner : MATLAB for engineers : Addison-wesley,
first edition, 1995
- Batanov, D. , E.Bohez, S.S. Makhanov, K.Sonthipaumpoon and M.Tabucanon.
A curvilinear grid generation technique applied for tool-path planning of 5-axis
milling machine : Proc. of the 4-th Int. Conf. Computer Integrated manufacturing,
21-24 October, 1997 Singapore.
- Brackbill, J.U. and Saltzman J.S. : Adaptive zone for singular problem in two dimension :
Journal of computational physics 46,342-368,1982)
- CONVEX Computer corporation : Cxpa Reference : second edition, 1994
- CONVEX Computer corporation : Cxpb Quick Reference : third edition, 1994
- CONVEX Computer corporation : Exampnar Programming Guide : second Edition,
June 1995
- CONVEX Computer corporation : Advanced Optimization Package User' Guide :
First Edition, September 1995
- Gray, John Shapley : Interprocess communications in UNIX : Prentice hall PTR, 1997
- Heller, Dan & Paula M, Ferguson : Motif programming manual : O'Reilly & Association,
Inc, second edition, 1994
- Hull, M.E.C, Crookes D., Sweeney P.J. : Parallel Processing : Addison-wesley, 1994
- Ivenenko, S.A. : Adaptive grids and grids on surface : Comp.Maths Math.Phys,
Vol.33,No,9 pp. 1179-1193,1993, Elsevier
- Linda mui. and Eric pearce. : X Windows system administrator' guide : Volume 8,
O'Reilly & Association.inc. 1992
- Rojiani, K. B. : Programming in c with numerical method for engineerings :
Prentice-Hall, 1996

ROA, N., ISMAIL F. and BEDI S. : Tool path planning for five-axis machining using the principal axis method : Int.J.Mach. Tool Manufact.Vol.37, No.7 pp 1025-1040, 1997 Elsevier Science Ltd.

Quercia , Valerie. and Tim. : X Windows system user's guide : volume three, O'Reilly & Association.inc, 1993

Watt, Alan : 3D computer graphics : second edition, 1993



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
 รายงานการวัดประสิทธิภาพของ โปรแกรมสวนปรับปรุงพื้นผิว
 Performance Analysis Report

=====

CXpa Version 3.5 Profile

Executable : /users/s8626062/g8run/g8b/a.out
 Profile Data : /users/s8626062/g8run/g8b/a.out.pdf
 Process State : exited
 CPU Time : 992.345 secs
 Wall Clock Time : 1850.075 secs
 Architecture : CONVEX SPP-1600 (1 thread)

=====

Routine Performance Analysis

=====

Times Exec	PS	Call Counts	
		Routine Name	
13266256	DYDZ		
13266256	DYDE		
13266256	DXDZ		
13266256	DXDE		
8170608	JJ		
2098720	W		
2098720	WXY		
1054272	DDYDDZ		
1054272	DDYDDE		
1054272	DDYDZE		
1054272	DDXDZE		
1054272	DDXDDZ		
1054272	DDXDDE		
527136	al		
527136	bt		
527136	gm		
527136	C		
527136	A		
527136	B		
175712	DWDY		
175712	DWDX		
38	Nabor		
19	NNorm		
1	iteration		
1	main		
1	Output		
1	Inita		
1	OutputWXY		

(less children)		Computation (plus children)		PS Routine Name
CPU Time	%	CPU Time	%	
275.886	27.8%	452.276	45.6%	JJ
147.889	14.9%	982.242	99.0%	Nabor
71.715	7.2%	71.715	7.2%	DYDE
71.683	7.2%	71.683	7.2%	DXDE
71.477	7.2%	71.477	7.2%	DYDZ
71.448	7.2%	71.448	7.2%	DXDZ
41.309	4.2%	41.309	4.2%	WXY
36.297	3.7%	222.737	22.4%	al
29.604	3.0%	157.664	15.9%	bt
29.574	3.0%	157.622	15.9%	gm
17.742	1.8%	29.123	2.9%	B
17.728	1.8%	29.103	2.9%	A
17.715	1.8%	29.085	2.9%	C
17.605	1.8%	58.914	5.9%	W
11.645	1.2%	11.645	1.2%	DDYDDZ

เอกสารนี้เป็นลิขสิทธิ์ของสำนักงานเทคโนโลยีสารสนเทศแห่งชาติ ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.269	1.1%	11.269	1.1%	DDYDZE
11.236	1.1%	11.236	1.1%	DDXDZE
9.525	1.0%	9.525	1.0%	DDXDZDZ
9.523	1.0%	9.523	1.0%	DDXDDE
9.362	0.9%	9.362	0.9%	DDYDDE
4.226	0.4%	992.069	100.0%	iteration
3.478	0.4%	13.301	1.3%	DWDX
3.463	0.3%	13.107	1.3%	DWDY
0.670	0.1%	0.670	0.1%	NNorm
0.152	0.0%	0.152	0.0%	OutputWXY
0.118	0.0%	0.118	0.0%	Output
5.364m	0.0%	5.364m	0.0%	Inita
0.139m	0.0%	992.345	100.0%	main

Time to Solution

(less children)			(plus children)			PS	Routine Name
Wall Clock	%	CPU/Wall	Wall Clock	%	CPU/Wall		
517.340	28.0%	0.53	854.193	46.2%	0.53	JJ	
266.938	14.4%	0.55	1831.064	99.0%	0.54	Nabor	
138.888	7.5%	0.51	138.888	7.5%	0.51	DXDZ	
137.223	7.4%	0.52	137.223	7.4%	0.52	DYDE	
136.676	7.4%	0.52	136.676	7.4%	0.52	DXDE	
134.584	7.3%	0.53	134.584	7.3%	0.53	DYDZ	
71.016	3.8%	0.51	424.011	22.9%	0.53	al	
66.810	3.6%	0.62	66.810	3.6%	0.62	WXY	
56.609	3.1%	0.52	299.677	16.2%	0.53	gm	
55.546	3.0%	0.53	295.626	16.0%	0.53	bt	
35.254	1.9%	0.50	102.064	5.5%	0.58	W	
33.777	1.8%	0.52	55.708	3.0%	0.52	C	
32.664	1.8%	0.54	53.740	2.9%	0.54	A	
31.822	1.7%	0.56	54.213	2.9%	0.54	B	
21.259	1.1%	0.55	21.259	1.1%	0.55	DDYDDZ	
20.432	1.1%	0.55	20.432	1.1%	0.55	DDXDZE	
20.421	1.1%	0.55	20.421	1.1%	0.55	DDYDZE	
17.582	1.0%	0.54	17.582	1.0%	0.54	DDXDDE	
16.906	0.9%	0.55	16.906	0.9%	0.55	DDYDDE	
16.849	0.9%	0.57	16.849	0.9%	0.57	DDXDZDZ	
7.144	0.4%	0.59	1849.353	100.0%	0.54	iteration	
6.848	0.4%	0.51	23.611	1.3%	0.56	DWDY	
6.076	0.3%	0.57	22.891	1.2%	0.58	DWDX	
0.689	0.0%	0.97	0.689	0.0%	0.97	NNorm	
0.421	0.0%	0.28	0.421	0.0%	0.28	Output	
0.240	0.0%	0.63	0.240	0.0%	0.63	OutputWXY	
0.048	0.0%	0.11	0.048	0.0%	0.11	Inita	
3.019m	0.0%	0.05	1850.065	100.0%	0.54	main	

Loop Performance Analysis
For: NNorm

Optimized Loops:

Line	NL Optimization	Times Exec	Iteration Counts			Total	PS
			Min	Max	Avg		
174	0	19	68	68	68.0	1292	

Optimized Loops:

Line	NL Optimization	Times Exec	Computation		PS
			(less inner) CPU Time	(plus inner) CPU Time	
174	0	19	0.669	0.669	

Optimized Loops:

Line	NL Optimization	Times Exec	Time to Solution				PS
			(less inner)		(plus inner)		
			Wall Clock	CPU/Wall	Wall Clock	CPU/Wall	
174	0	19	0.686	0.98	0.686	0.98	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Optimized Loops:

Line	NL Optimization	Times Exec	Iteration Counts				PS
			Min	Max	Avg	Total	
188	0	1	70	70	70.0	70	
194	0	1	70	70	70.0	70	
212	0	1	70	70	70.0	70	
219	0 I	1	70	70	70.0	70	

Optimized Loops:

Line	NL Optimization	Times Exec	Computation		PS
			(less inner) CPU Time	(plus inner) CPU Time	
188	0	1	0.139m	0.139m	
194	0	1	0.500m	0.500m	
212	0	1	0.127m	0.127m	
219	0 I	1	4.516m	4.516m	

Optimized Loops:

Line	NL Optimization	Times Exec	Time to Solution				PS
			(less inner) Wall Clock CPU/Wall		(plus inner) Wall Clock CPU/Wall		
188	0	1	0.030	0.00	0.030	0.00	
194	0	1	1.901m	0.26	1.901m	0.26	
212	0	1	5.002m	0.03	5.002m	0.03	
219	0 I	1	5.376m	0.84	5.376m	0.84	

Loop Performance Analysis
For: Nabor

Optimized Loops:

Line	NL Optimization	Times Exec	Iteration Counts				PS
			Min	Max	Avg	Total	
245	0	38	68	68	68.0	2584	

Optimized Loops:

Line	NL Optimization	Times Exec	Computation		PS
			(less inner) CPU Time	(plus inner) CPU Time	
245	0	38	982.241	982.241	

Optimized Loops:

Line	NL Optimization	Times Exec	Time to Solution				PS
			(less inner) Wall Clock CPU/Wall		(plus inner) Wall Clock CPU/Wall		
245	0	38	1831.062	0.54	1831.062	0.54	

Loop Performance Analysis
For: OutputWXY

Optimized Loops:

Line	NL Optimization	Times Exec	Iteration Counts				PS
			Min	Max	Avg	Total	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
-----
322  0                1      70      70      70.0      70
329  0                1      70      70      70.0      70
```

Optimized Loops:

Line	NL Optimization	Times Exec	Computation		PS
			(less inner)	(plus inner)	
			CPU Time	CPU Time	
322	0	1	0.072	0.072	--
329	0	1	0.080	0.080	

Optimized Loops:

Line	NL Optimization	Times Exec	Time to Solution			PS
			(less inner)	(plus inner)		
			Wall Clock	Wall Clock		
			CPU/Wall	CPU/Wall		
322	0	1	0.108	0.66	0.108 0.66	
329	0	1	0.090	0.89	0.090 0.89	

=====
 Loop Performance Analysis
 For: Output
 =====

Optimized Loops:

Line	NL Optimization	Times Exec	Iteration Counts				PS
			Min	Max	Avg	Total	
342	0	1	70	70	70.0	70	
349	0	1	70	70	70.0	70	

Optimized Loops:

Line	NL Optimization	Times Exec	Computation		PS
			(less inner)	(plus inner)	
			CPU Time	CPU Time	
342	0	1	0.059	0.059	--
349	0	1	0.059	0.059	

Optimized Loops:

Line	NL Optimization	Times Exec	Time to Solution			PS
			(less inner)	(plus inner)		
			Wall Clock	Wall Clock		
			CPU/Wall	CPU/Wall		
342	0	1	0.274	0.21	0.274 0.21	
349	0	1	0.065	0.91	0.065 0.91	

=====
 Loop Performance Analysis
 For: iteration
 =====

Optimized Loops:

Line	NL Optimization	Times Exec	Iteration Counts				PS
			Min	Max	Avg	Total	
367	0	1	19	19	19.0	19	

Optimized Loops:

Line	NL Optimization	Times Exec	Computation		PS
			(less inner)	(plus inner)	
			CPU Time	CPU Time	
367	0	1	992.069	992.069	--

เอกสาร Optimized Loops: วนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 Time to Solution

Line	NL Optimization	Times Exec	(less inner)		(plus inner)		PS
			Wall Clock	CPU/Wall	Wall Clock	CPU/Wall	
367	0	1	1849.351	0.54	1849.351	0.54	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.
โปรแกรมที่ใช้ทดสอบบนเครื่องซูเปอร์คอมพิวเตอร์
Adaptive Program on Super Computer

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "point.h"

double LV,LO,LVmax,WW;

double X[nx][ny], Y[nx][ny],
       RX[nx][ny], RY[nx][ny],
       RXX[nx][ny],
       RXY[nx][ny], RYY[nx][ny],
       Xnew[nx][ny], Ynew[nx][ny],
       JJJ[nx][ny];

double Hz,He;

double DXDZ(int i,int j)
{
  if (2*Hz == 0.0) /* Detect for division by 0 */
    return 0.0;
  else
    return (X[i+1][j]-X[i-1][j])/(2*Hz);
}

double DXDE(int i,int j)
{
  if (2*He == 0.0)
    return 0.0;
  else
    return (X[i][j+1]-X[i][j-1])/(2*He);
}

double DYDZ(int i,int j)
{
  if (2*Hz == 0.0)
    return 0.0;
  else
    return (Y[i+1][j]-Y[i-1][j])/(2*Hz);
}

double DYDE(int i,int j)
{
  if (2*He == 0.0)
    return 0.0;
  else
    return (Y[i][j+1]-Y[i][j-1])/(2*He);
}

double W(int i,int j)
{
  double mg;
  mg = (double) nx;
  return exp(-fabs(mg-(X[i][j]-10.0)*(X[i][j]-10.0)-
                  (Y[i][j]-10)*(Y[i][j]-10)))+0.01);
}

double DWDZ(int i,int j)
{
  if (2*Hz == 0.0)
    return 0.0;
  else
    return (W(i+1,j)-W(i-1,j))/(2*Hz);
}

double DWDE(int i,int j)
{
  if (2*He == 0.0)
    return 0.0;
  else
    return (W(i,j+1)-W(i,j-1))/(2*He);
}

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ควรนำเอกสารนี้ไปใช้ในการเรียนการสอนเพื่อการค้าโดยไม่ขออนุญาตจากทางมหาวิทยาลัย
 ไม่ควรนำเอกสารนี้ไปใช้ในการเรียนการสอนเพื่อการค้าโดยไม่ขออนุญาตจากทางมหาวิทยาลัย

```

else
    return (W(i,j+1)-W(i,j-1))/(2*Hz);
}

double DWDX(int i,int j)
{ if ((X[i+1][j]-X[i-1][j]) == 0.0)
    return 0;
  else
    return (W(i+1,j)-W(i-1,j))/(X[i+1][j]-X[i-1][j]);
}

double DWDY(int i,int j)
{
  if ((X[i][j+1] - X[i][j-1]) == 0.0)
    return 0.0;
  else
    return (W(i,j+1)-W(i-1,j-1))/(X[i][j+1]-X[i][j-1]);
}

double DDXDDZ(int i,int j)
{ if (Hz == 0.0)
    return 0.0;
  else
    return (X[i+1][j]-2*X[i][j]+X[i-1][j])/(Hz*Hz);
}

double DDXDDE(int i,int j)
{ if (He == 0.0)
    return 0.0;
  else
    return (X[i][j+1]-2*X[i][j]+X[i][j-1])/(He*He);
}

double DDXDZE(int i,int j)
{ if (4*Hz*He == 0.0)
    return 0.0;
  else
    return (X[i+1][j+1]+X[i-1][j-1]-X[i+1][j-1]-X[i-1][j+1])/(4*Hz*He);
}

double DDYDDZ(int i,int j)
{ if (Hz*He == 0.0)
    return 0.0;
  else return (Y[i+1][j]-2*Y[i][j]+Y[i-1][j])/(Hz*Hz);
}

double DDYDDE(int i,int j)
{ if (He*He == 0.0)
    return 0.0;
  else return (Y[i][j+1]-2*Y[i][j]+Y[i][j-1])/(He*He);
}

double DDYDZE(int i,int j)
{ if (4*Hz*He == 0.0)
    return 0.0;
  else
    return (Y[i+1][j+1]+Y[i-1][j-1]-Y[i+1][j-1]-Y[i-1][j+1])/(4*Hz*He);
}

double A(int i,int j)
{
  return DXDZ(i,j)*DYDZ(i,j)+DXDE(i,j)*DYDE(i,j);
}

double B(int i,int j)
{ double DyDz,DyDe;
  DyDz = DYDZ(i,j);
  DyDe = DYDE(i,j);
  return DyDz*DyDz + DyDe*DyDe;
}

double C(int i,int j)
{ double DxDz,DxDe;
  DxDz = DXDZ(i,j);
  DxDe = DXDE(i,j);
  return DxDz*DxDz + DxDe*DxDe;
}

```

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

double JJ(int i,int j)
{
return DXDZ(i,j)*DYDE(i,j)-DXDE(i,j)*DYDZ(i,j);
}

double al(int i,int j)
{ double jj;
  jj = JJ(i,j);

  if (jj == 0)
    return 0;
  else
    return (DXDE(i,j)*DXDE(i,j)+DYDE(i,j)*DYDE(i,j))/(jj*jj*jj);
}

double bt(int i,int j)
{ double jj;
  jj = JJ(i,j);
  if (jj == 0)
    return 0;
  else
    return (DXDZ(i,j)*DXDE(i,j)+DYDZ(i,j)*DYDE(i,j))/(jj*jj*jj);
}

double gm(int i,int j)
{ double jj;
  double Dx Dz,Dy Dz;
  jj = JJ(i,j);
  if (jj == 0)
    return 0;
  else
    { Dx Dz = DXDZ(i,j);
      Dy Dz = DYDZ(i,j);
      return (DxDz*DxDz+DyDz*DyDz)/(jj*jj*jj);
    }
}

double NNorm()
{ int i,j;
  double DD = 0.0;
  for (i = 1; i < nx-1; i++)
    for (j = 1; j < ny-1; j++)
      if (DD < (fabs(RX[i][j])+fabs(RY[i][j])))
        { DD = (fabs(RX[i][j])+fabs(RY[i][j]));
        }
  return DD/2;
}

void Inita()
{ int i,j;
  FILE *fp;
  double minx,miny,maxx,maxy;

  Hz = 1;
  He = 1;
  fp = fopen(source,"r");
  for (i = 0;i < nx;i++)
    for (j = 0; j < ny; j++)
      { fscanf(fp,"%lf %lf",&X[i][j],&Y[i][j]);
      }
  fclose(fp);
}

void Nabor()
{ int i,j;
  double as1,as2,as3, bs1,bs2,bs3, cs1,cs2,cs3;
  double av1,av2,av3, bv1,bv2,bv3, cv1,cv2,cv3;
  double ao1,ao2,ao3, bo1,bo2,bo3, col,co2,co3;
  double bl,b3, a1,a3, c1,c3;
  for (i = 1;i < nx-1; i++)
    for (j = 1; j < ny-1; j++)
      /* The grid smoothness */

```

```

as1 = -A(i,j)*al(i,j);
as2 = 2*A(i,j)*bt(i,j);
as3 = -A(i,j)*gm(i,j);

bs1 = B(i,j)*al(i,j);
bs2 = -2*B(i,j)*bt(i,j);
bs3 = B(i,j)*gm(i,j);

cs1 = C(i,j)*al(i,j);
cs2 = -2*C(i,j)*bt(i,j);
cs3 = C(i,j)*gm(i,j);

/* Adaptation */
av1 = -DXDE(i,j)*DYDE(i,j);
av2 = DXDZ(i,j)*DYDE(i,j)+DXDE(i,j)*DYDZ(i,j);
av3 = -DXDZ(i,j)*DYDZ(i,j);

/* Orthogonality */
bv1 = DYDE(i,j)*DYDE(i,j);
bv2 = -2*DYDZ(i,j)*DYDE(i,j);
bv3 = DYDZ(i,j)*DYDZ(i,j);

cv1 = DXDE(i,j)*DXDE(i,j);
cv2 = -2*DXDZ(i,j)*DXDE(i,j);
cv3 = DXDZ(i,j)*DXDZ(i,j);

ao1 = DXDE(i,j)*DYDE(i,j);
ao2 = DXDZ(i,j)*DYDE(i,j)+DXDE(i,j)*DYDZ(i,j);
ao3 = DXDZ(i,j)*DYDZ(i,j);

bo1 = DXDE(i,j)*DXDE(i,j);
bo2 = 2*(DXDZ(i,j)*DXDE(i,j)+DYDZ(i,j)*DYDE(i,j));
bo3 = DXDZ(i,j)*DXDZ(i,j);

co1 = DYDE(i,j)*DYDE(i,j);
co2 = 2*(DXDZ(i,j)*DXDE(i,j)+2*DYDZ(i,j)+DYDE(i,j));
co3 = DYDZ(i,j)*DYDZ(i,j);

b1 = bs1 + LV*2*W(i,j)*bv1 + LO *bo1;
b3 = bs3 + LV*2*W(i,j)*bv3 + LO *bo3;
a1 = as1 + LV*2*W(i,j)*av1 + LO *ao1;
a3 = as3 + LV*2*W(i,j)*av3 + LO *ao3;
c1 = cs1 + LV*2*W(i,j)*cv1 + LO *co1;
c3 = cs3 + LV*2*W(i,j)*cv3 + LO *co3;

/* Rx = (Rs)x + Lamda_V(Rv)x + Lamda_o(Ro)x */
RX[i][j] = bs1*DDXDDZ(i,j)+bs2*DDXDZE(i,j)+bs3*DDXDDE(i,j)+
as1*DDYDDZ(i,j)+as2*DDYDZE(i,j)+as3*DDYDDE(i,j)+
LO*(bo1*DDXDDZ(i,j)+bo2*DDXDZE(i,j)+bo3*DDXDDE(i,j)+
ao1*DDYDDZ(i,j)+ao2*DDYDZE(i,j)+ao3*DDYDDE(i,j))+
LV*2*W(i,j)*(bv1*DDXDDZ(i,j)+bv2*DDXDZE(i,j)+bv3*DDXDDE(i,j)+
av1*DDYDDZ(i,j)+av2*DDYDZE(i,j)+av3*DDYDDE(i,j))+
LV*JJ(i,j)*JJ(i,j)*DWDX(i,j);

/* Ry = (Rs)y + Lamda_V(Rv)y + Lamda_o(Ro)y */
RY[i][j] = as1*DDXDDZ(i,j)+as2*DDXDZE(i,j)+as3*DDXDDE(i,j)+
cs1*DDYDDZ(i,j)+cs2*DDYDZE(i,j)+cs3*DDYDDE(i,j)+
LO*(ao1*DDXDDZ(i,j)+ao2*DDXDZE(i,j)+ao3*DDXDDE(i,j)+
co1*DDYDDZ(i,j)+co2*DDYDZE(i,j)+co3*DDYDDE(i,j))+
LV*2*W(i,j)*(av1*DDXDDZ(i,j)+av2*DDXDZE(i,j)+av3*DDXDDE(i,j)+
cv1*DDYDDZ(i,j)+cv2*DDYDZE(i,j)+cv3*DDYDDE(i,j))+
LV*JJ(i,j)*JJ(i,j)*DWDY(i,j);

RXX[i][j] = -2*(b1+b3);
RXY[i][j] = -2*(a1+a3);
RYY[i][j] = -2*(c1+c3);
}

void OutputWXY(void)
{ int i,j;
FILE *fp;
fp = fopen("wxy.dat","w");
for (i = 0;i < nx; i++)
for (j = 0;j < ny; j++)

```

```

        fprintf(fp,"%18.8f ",X[i][j]);
        fprintf(fp,"\n");
    }

    for (i = 0;i < nx; i++)
    {
        for (j = 0;j < ny; j++)
            fprintf(fp,"%18.8f ",Y[i][j]);
        fprintf(fp,"\n");
    }
    fclose(fp);
}

void OutputRX()
{ int i,j;
  FILE *fp;
  fp = fopen("orx.dat","w");
  for (i = 0;i < nx; i++)
  {
      for (j = 0;j < ny; j++)
          fprintf(fp,"%18.8f ",RX[i][j]);
      fprintf(fp,"\n");
  }

  for (i = 0;i < nx; i++)
  {
      for (j = 0;j < ny; j++)
          fprintf(fp,"%18.8f ",RY[i][j]);
      fprintf(fp,"\n");
  }
  fclose(fp);
}

void Output()
{ int i,j;
  FILE *fp;
  fp = fopen("gridpar1.dat","w");
  for (i = 0;i < nx; i++)
  {
      for (j = 0;j < ny; j++)
          fprintf(fp,"%18.8f ",X[i][j]);
      fprintf(fp,"\n");
  }

  for (i = 0;i < nx; i++)
  {
      for (j = 0;j < ny; j++)
          fprintf(fp,"%18.8f ",Y[i][j]);
      fprintf(fp,"\n");
  }
  fclose(fp);
}

void iteration(void)
{
    int i,j,k;
    int loop;
    double d1 = 0 ,d2 = 0;
    double it = 0, Norm;
    double a11,a12,a21,a22, bb1,bb2;
    Norm = 10000000.0;
    printf("Norm begin with %10.2f \n",Norm);
    /* while (Norm > 1000000.0) */
    for (loop = 0;(loop < 220) && (Norm > 0.001);loop++)
    {
        for (i=0;i < nx;i++)
            for (j=0;j < ny;j++)
            {
                Xnew[i][j] = X[i][j]; Ynew[i][j] = Y[i][j];
            }
        Nabor();
        for (j = 1;j < ny-1;j++)
            for (i = 1; i < nx-1;i++)
            {
                a11 = RXX[i][j]; a12 = RXY[i][j];
                a21 = RXY[i][j]; a22 = RYY[i][j];

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        bb1 = -RX[i][j]+RXX[i][j]*X[i][j]+RXY[i][j]*Y[i][j];
        bb2 = -RY[i][j]+RXY[i][j]*X[i][j]+RYY[i][j]*Y[i][j];
    if ((-a12*a21+a22*a11) == 0)
    {
        Xnew[i][j] = 0;
        Ynew[i][j] = 0;
    }
    else
    {
        Xnew[i][j] = -(a12*bb2-a22*bb1)/(-a12*a21+a22*a11);
        Ynew[i][j] = (a11*bb2-a21*bb1)/(-a12*a21+a22*a11);
    }
    d1 = a11*Xnew[i][j]+a12*Ynew[i][j];
    d2 = a21*Xnew[i][j]+a22*Ynew[i][j];
    JJJ[i][j] = JJ(i,j);
}

for (i = 1; i < nx-1;i++)
for (j = 1;j < ny-1;j++)
{
    X[i][j] = X[i][j]*WW+Xnew[i][j]*(1-WW);
    Y[i][j] = Y[i][j]*WW+Ynew[i][j]*(1-WW);
}
Nabor();
it++;
Norm = NNorm();
printf("%d %18.8f %18.8f %18.8f\n",loop+1,Norm,d1,d2);
}

void main(void)
{
    WW = 0.81;
    LV = 0.0;
    LVmax = 1.0;
    LO = 0.0;
    Inita();
    iteration();
    OutputRX();
    Output();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

โปรแกรมที่ใช้สร้างพื้นผิวโค้ง 3 มิติ และแปลงพื้นผิว

```

#include <stdio.h>
#include <math.h>

#define nx 20
#define ny 20

#define MaxCtrlX 4
#define MaxCtrlY 4

#define pi 22.0/7.0

#define aConst -15.0
#define bConst 0.0
#define cConst 175.0
#define eConst 0.08

#define OFConst 119.644
#define abConst 251.286
#define B4Const 371.943

/* Control Point Vector */
//double CtrlPointX[MaxCtrlX][MaxCtrlY];
double CtrlPointX[4][4] =
{
    10.0, 10.0, 10.0, 10.0,
    30.0, 30.0, 30.0, 30.0,
    60.0, 60.0, 60.0, 60.0,
    90.0, 90.0, 90.0, 90.0,
};

double CtrlPointY[MaxCtrlX][MaxCtrlY] =
{
    10.0, 30.0, 60.0, 90.0,
    10.0, 30.0, 60.0, 90.0,
    10.0, 30.0, 60.0, 90.0,
    10.0, 30.0, 60.0, 90.0,
};

double CtrlPointZ[MaxCtrlX][MaxCtrlY] =
{
    1.0, 1.0, 1.0, 1.0,
    1.0, 90.0, 90.0, 1.0,
    1.0, 90.0, 90.0, 1.0,
    1.0, 1.0, 1.0, 1.0,
};

/* Coordinate X Y Z */
double CoorPointX[nx][ny];
double CoorPointY[nx][ny];
double CoorPointZ[nx][ny];

/* Normal Vector X Y Z */
double NormalVectorX[nx][ny];
double NormalVectorY[nx][ny];
double NormalVectorZ[nx][ny];

/* Tool Path Calculate */
double ToolPathX[nx][ny];
double ToolPathY[nx][ny];
double ToolPathZ[nx][ny];

/* Summary of Coordinate Point */
double XSum, YSum, ZSum;
double R;

void InitialControlPoint(void)

```

เอกสารนี้เป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i = 0;i < MaxCtrlX;i++)
  for (j = 0;j < MaxCtrlY;j++)
  {
    CtrlPointX[i][j] = 0.0;
    CtrlPointY[i][j] = 0.0;
    CtrlPointZ[i][j] = 1.0;
  } */
}

double bFun(double x,int id)
{
  double result = 0,y = 1.0 - x;
  switch(id)
  {
    case 0:
      result = 1.0*y*y*y;
      break;
    case 1:
      result = 3.0*x*y*y;
      break;
    case 2:
      result = 3.0*x*x*y;
      break;
    case 3:
      result = 1.0*x*x*x;
      break;
  }
  return result;
}

/* Represent the bicubic Bizier path by
P(u,v) = (X(u,v),Y(u,v),Z(u,v)) */
double Puv(double U,double V,int Axis)
{ int i,j;
  double Result = 0.0;
  switch(Axis)
  {
    case 1: // X Axis
      for (i = 0;i < 4;i++)
        for (j = 0;j < 4;j++)
          {
            Result = Result + CtrlPointX[i][j] * bFun(U,i)*bFun(V,j);
          }
      break;
    case 2: // Y Axis
      for (i = 0;i < 4;i++)
        for (j = 0;j < 4;j++)
          {
            Result = Result + CtrlPointY[i][j] * bFun(U,i)*bFun(V,j);
          }
      break;
    case 3: // Z Axis
      for (i = 0;i < 4;i++)
        for (j = 0;j < 4;j++)
          {
            Result = Result + CtrlPointZ[i][j] * bFun(U,i)*bFun(V,j);
          }
      break;
  }
  return Result;
}

/* Calculate Coordinate of X Y Z */
/* Define max_x = nx
   Define max_y = ny */

void CoordinateCalculate(void)
{
  int i,j;
  double u,v;
  double XSum = 0.0, YSum = 0, ZSum = 0;
  double XAver = 0.0, YAver = 0.0, ZAver = 0.0;
  int PointNum;
  double val1,val2;
  PointNum = (nx)*(ny);
  for (i = 0;i < nx;i++)

```

เอกสารนี้
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  for (j =0;j < ny;j++)
  {
    u = (float) i/nx; v = (float) j/ ny;
    CoorPointX[i][j] = Puv(u,v,1); XSum = XSum + CoorPointX[i][j];
    CoorPointY[i][j] = Puv(u,v,2); YSum = YSum + CoorPointY[i][j];
    CoorPointZ[i][j] = Puv(u,v,3); ZSum = ZSum + CoorPointZ[i][j];
  }
}

XAver = XSum/PointNum;
YAver = YSum/PointNum;
ZAver = ZSum/PointNum;
val1 = 250.0/(XAver + YAver * cos(pi/6));
val2 = 200.0/(ZAver + YAver * sin(pi/6));
if (val1 < val2) R = val1; else R = val2;
}

double DB(double x,int id)
{
  double y = 0.0,result = 0.0;
  y = 1.0 - x;
  result = 0.0;
  switch(id)
  { case 0:
    result = -3.0 * y * y;
    break;
  case 1:
    result = 3.0 * y * y - 6 * x * y;
    break;
  case 2:
    result = 6.0 * x * y - 3 * x * x;
    break;
  case 3:
    result = 3.0 * x * x;
    break;
  }
  return result;
}

/* Find Difference of u */
double DPu(double x,double y,int axis)
{ double result = 0.0;
  int i,j;
  switch(axis)
  {
  case 1:
    for (i = 0;i < 4;i++)
      for (j = 0;j < 4;j++)
        { result = result + CtrlPointX[i][j]*DB(x,i)*bFun(y,j);
        }
    break;
  case 2:
    for (i = 0;i < 4;i++)
      for (j = 0;j < 4;j++)
        { result = result + CtrlPointY[i][j]*DB(x,i)*bFun(y,j);
        }
    break;
  case 3:
    for (i = 0;i < 4;i++)
      for (j = 0;j < 4;j++)
        { result = result + CtrlPointZ[i][j]*DB(x,i)*bFun(y,j);
        }
    break;
  }
  return result;
}

/* Find Difference of v */
double DPv(double x,double y,int axis)
{ double result = 0.0;
  int i,j;
  switch(axis)

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

if (x > 0) sg = 1;
if (x < 0) sg = -1;
if (x == 0) sg = 0;
return sg;
}

double Auv(double x,double y)
{
double result;
result = sqrt(x*x + y*y);
if (result > 0.01)
{
if (x != 0)
{
result = -Sgn(x)*atan(y/x);
}
else
{
if ((y < 0) && (x < 0)) result = pi - atan(y/x);
if ((y > 0) && (x < 0)) result = pi - atan(y/x);
if (y < 0) result = pi / 2.0;
if (y > 0) result = -pi / 2.0;
if ((y == 0) && (x == 0)) result = pi;
}
}
else result = 0.0;
return result;
}

double Buv(double x,double y,double z)
{
double result;
result = sqrt(x*x + y*y);
if (result > 0.01)
{
if (z != 0)
{
if (x != 0)
result = Sgn(x) * atan(result/z);
else result = atan(result/z);
} else result = pi/2;
} else result = 0;
return result;
}

void ToolPathCal(int K1,int L1,int K2,int L2)
{ double aAgel[2],mx[2],my[2],mz[2],t;
double x,y,z;
double bAgel[2],A,B;
int i;
aAgel[0] = Auv(NormalVectorX[K1][L1],NormalVectorY[K1][L1]);
aAgel[1] = Auv(NormalVectorX[K2][L2],NormalVectorY[K2][L2]);
bAgel[0] = Buv(NormalVectorX[K1][L1],NormalVectorY[K1][L1],NormalVectorZ[K1][L1]);
bAgel[1] = Buv(NormalVectorX[K2][L2],NormalVectorY[K2][L2],NormalVectorZ[K2][L2]);
if (abs(aAgel[0]) < 0.00000001) aAgel[0] = aAgel[1];
if (abs(aAgel[1]) < 0.00000001) aAgel[1] = aAgel[0];
for (i = 0;i < 2;i++)
{
if (i == 0)
{ x = CoorPointX[K1][L1];
y = CoorPointY[K1][L1];
z = CoorPointZ[K1][L1];
} else
{ x = CoorPointX[K2][L2];
y = CoorPointY[K2][L2];
z = CoorPointZ[K2][L2];
}
mx[i] = (abConst-cConst-z)*cos(bAgel[i])+(eConst+(aConst+x)*cos(aAgel[i]))-
(bConst+y)*sin(aAgel[i])*sin(bAgel[i]);
my[i] = -(bConst+y)*cos(aAgel[i])-(aConst+x)*sin(aAgel[i]);
mz[i] = -B4Const+OFConst+cos(bAgel[i])*(eConst+(aConst+x)*cos(aAgel[i]))-
(bConst+y)*sin(aAgel[i])-(abConst-cConst-z)*sin(bAgel[i]);
}
t = 0.5;
x = mx[0] + t*(mx[1]-mx[0]);
y = my[0] + t*(my[1]-my[0]);
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

z = mz[0] + t*(mz[1]-mz[0]);
A = aAgel[0] + t*(aAgel[1]-aAgel[0]);
B = bAgel[0] + t*(bAgel[1]-bAgel[0]);
mx[0] = -aConst-y*sin(A)+cos(A)*(-eConst+(B4Const-OFConst+z)*cos(B)+x*sin(B));
my[0] = -bConst-y*cos(A)-sin(A)*(-eConst+(B4Const-OFConst+z)*cos(B)+x*sin(B));
mz[0] = abConst-cConst-x*cos(B)+(B4Const-OFConst+z)*sin(B);
ToolPathX[K1][L1] = mx[0];
ToolPathY[K1][L1] = my[0];
ToolPathZ[K1][L1] = mz[0];
}

```

```

void NewToolPath(void)
{ int K1,K2,L1,L2,i;
  double x,Y,Z;
  double xe = 0.0,ye = 0.0,ze = 0.0;

  FILE *fp;
  int PointNum = (nx) * (ny);

  fp = fopen("Toolpath.dat","w");
  for (K1 = 0;K1 < nx;K1++)
    for (L1 = 0;L1 < ny;L1++)
      {
        ToolPathX[K1][L1] = 0.0;
        ToolPathY[K1][L1] = 0.0;
        ToolPathZ[K1][L1] = 0.0;
      }

  L1 = 0; K1 = L1; L2 = 1; K2 = K1;
  for (i = 0;i < PointNum - 1;i++)
  {
    if (i != 0)
    { if ((K1 == K2) && ((L1 == 0) || (L1 == ny-1)) )
      {
        L2 = L1; K2 = K1 + 1;
      }
      else
      {
        K1 = K2;
        if ((K1 % 2) == 0)
          L2 = L1 + 1;
        else
          L2 = L1 - 1;
      }
    }

    // call ToolPathCal
    ToolPathCal(K1,L1,K2,L2);
    x = 10 + R * (ToolPathX[K1][L1]+ToolPathY[K1][L1]*cos(pi/6));
    y = 400 - R * (ToolPathZ[K1][L1]+ToolPathY[K1][L1]*sin(pi/6));
    xe = 10 + R*(CoorPointX[K2][L2]+CoorPointY[K2][L2]*cos(pi/6));
    ye = 400 - R * (CoorPointZ[K2][L2]+CoorPointY[K2][L2]*sin(pi/6));
    {
      // Drawline
      fprintf(fp,"%d %d \n %d %d\n",(int) x,(int) y,(int) xe,(int) ye);
    }
    L1 = L2;
  }
  fclose(fp);
}

```

```

void PlotSurface(void)
{
  double xe = 0.0,ye = 0.0,ze = 0.0;
  double x = 0.0,y = 0.0;
  FILE *fp;
  int i,j;
  fp = fopen("Surface.dat","w");
  for (i = 0;i < nx;i++)
    for (j = 0;j < ny;j++)
      {
        xe = xe + CoorPointX[i][j];
        ye = ye + CoorPointY[i][j];
        ze = ze + CoorPointZ[i][j];
      }
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x = 10; y = 400;
for (i = 0; i < nx; i++)
{
    for (j = 0; j < ny; j++)
    {
        xe = 10 + R * (CoorPointX[i][j]+CoorPointY[i][j]*cos(pi/6));
        ye = 400 - R * (CoorPointZ[i][j]+CoorPointY[i][j]*sin(pi/6));
        if (j != 0)
        {
            // Drawline
            fprintf(fp,"%d %d %d %d\n", (int) x, (int) y, (int) xe, (int) ye);
        }
        x = xe; y = ye;
    }
}

x = 10; y = 400;
for (j = 0; j < ny; j++)
{
    for (i = 0; i < nx; i++)
    {
        xe = 10 + R*(CoorPointX[i][j]+CoorPointY[i][j]*cos(pi/6));
        ye = 400 - R * (CoorPointZ[i][j]+CoorPointY[i][j]*sin(pi/6));
        if (i != 0)
        {
            fprintf(fp,"%d %d %d %d \n", (int) x, (int) y, (int) xe, (int) ye);
        }
        x = xe; y = ye;
    }
}
fclose(fp);
}

void main(void)
{
    double i,j;
    double x,y,u,v;
    InitialControlPoint();
    CoordinateCalculate();
    PlotSurface();
    NormalVector();
    NewToolPath();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

การโปรแกรมบนระบบ X windows

How to program by X windows ?

1. X windows คืออะไร

X windows เป็นระบบการแสดงผลแบบกราฟิกส์บนระบบ Unix พัฒนาขึ้นโดย MIT ในปี ค.ศ. 1984 ปัจจุบันอยู่ที่รุ่น 11 X windows เป็นโปรแกรมซึ่งทำงานอยู่บนระบบ Client/Server โดยในระบบ X จะแบ่งออกเป็น 2 ส่วนใหญ่ด้วยกันคือ 1. Display Server จะทำแสดงผลบนจอภาพ 2. Client เป็นโปรแกรมซึ่งพัฒนาขึ้นเพื่อทำงานเฉพาะอย่างตามที่เราต้องการ

2. การใช้งาน X windows บน Convex Supercomputer

2.1 เริ่มใช้คำสั่ง telnet Nawamas.kmitl.ac.th แล้วทำการ login เข้าสู่ระบบ

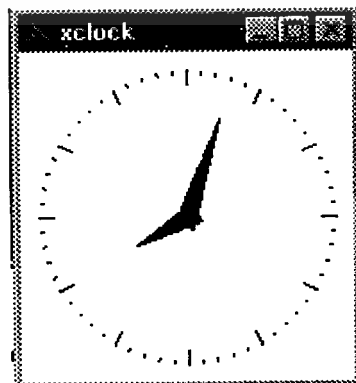
2.2 ใช้คำสั่ง export DISPLAY=[IP Address ของเครื่อง Client หรือ X-terminal]:0.0

2.3 ถ้าใช้เครื่อง X - Terminal สามารถใช้งาน X windows ได้เลย

2.4 ถ้าเป็นเครื่อง PC ที่ไม่ใช่ X windows จะต้องลงโปรแกรม X Terminal Emulator แล้วทำ

การ Run Program X server เพื่อทำการแปลงสัญญาณที่ส่งมาแสดงผลบนเครื่อง ในระบบของ X windows

2.5 เราสามารถทดสอบว่าสามารถใช้งาน X windows บนเครื่อง PC ที่เราใช้งานอยู่ได้ โดยการเรียกโปรแกรมซึ่งมีอยู่ในเครื่อง Server ซึ่งติดตั้งไว้แล้วได้แก่ xclock ซึ่งเป็นโปรแกรมนาฬิกาบน X windows




```

void DrawGraph(double XSMIN,double XSMAX,double YSMIN,double YSMAX)
{ int I, xssl,yssl,xss2,yss2;
  double xsr,ysr;
  xsr = LinearXRtoXS(XR[0],XSMIN,XSMAX);
  ysr = LinearXRtoXS(YR[0],YSMIN,YSMAX);
  xssl = (int) xsr;    yssl = (int) ysr;
  for (i = 1;i < ny;i++)
  { xsr = LinearXRtoXS(XR[i],XSMIN,XSMAX);
    ysr = LinearXRtoXS(YR[i],YSMIN,YSMAX);
    xss2 = (int) xsr;    yss2 = (int) ysr;
    XDrawLine(display,main_win,gc_1,xssl,yssl,xss2,yss2);
    xssl = xss2;
    yssl = yss2;
  }
}

void DrawGrid()
{ int i=0,j=0;
/*  ÇÒ' Grid μÒÁÁ'ÇμÑé$ */
for (i=0;i < nx;i++)
  { for (j=0;j < ny;j++)
    { XR[j] = X[j][i];
      YR[j] = Y[j][i];
      DrawGraph(XSmin,XSmax-10,YSmin,YSmax-10);
    }
  }
/*  ÇÒ' Grid μÒÁÁ'Ç'f' */
for (i=0;i<nx;i++)
  { for (j=0;j < ny;j++)
    { XR[j] = X[i][j];
      YR[j] = Y[i][j];
      DrawGraph(XSmin,XSmax-10,YSmin,YSmax-10);
    }
  }
}
void main(int argc,char *argv[])
{
  /* Open a connection to the server */
  if ((display=XOpenDisplay(NULL)) == NULL) {
    (void) fprintf(stderr, "Can't connect to display\n");
    exit(-1);
  }
  /* Get some default values of the server */
  screen_num = DefaultScreen(display);
  colormap = DefaultColormap(display,screen_num);
  black_pix = BlackPixel(display,screen_num);
  white_pix = WhitePixel(display,screen_num);

  /* Create a main window */
  main_win = XCreateSimpleWindow(display,
    DefaultRootWindow(display),
    XSmin, YSmin, /* X, y */
    XSmax, YSmax, /* width, height */
    2, /* border width */
    black_pix, white_pix);

  /* Allocate two named colors: red and green */
  if(XAllocNamedColor(display,colormap,"red", &scr_color,&exact_color)!=0)
    red_pix = scr_color.pixel;
  else
    red_pix = black_pix; /* error recovery code */

  if(XAllocNamedColor(display,colormap,"green", &scr_color,&exact_color)!=0)
    green_pix = scr_color.pixel;
  else
    green_pix = white_pix; /* error recovery code */

  /* Create two GCs with default values */
  gc_1 = XCreateGC(display,main_win,0,0);

  /* Set some values in two GCs */
  XSetState(display,gc_1,
    black_pix, /* foreground */
    green_pix, /* background */
    GXcopy, /* function */
    AllPlanes); /* plane mask */
}

```

เอกสารนี้เป็นเอกสารที่เผยแพร่ภายใต้การใช้นโยบายการเข้าถึงแบบเปิด ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

XSetLineAttributes(display,gc_1, 1, LineSolid, CapNotLast, JoinMiter);
XFlushGC(display,gc_1);

/* Set the event masks and map the window */
XSelectInput(display,main_win,ButtonPressMask);
XMapWindow(display,main_win);
if (argc == 3)
    ReadFile(argv[1]);
else
    ReadFile(source);
DrawGrid();
while(1) {
    XNextEvent(display,&the_event);
    switch (the_event.type) {
        case ButtonPress:
            switch (the_event.xbutton.button) {
                case Button1:
                    if (argc == 3)
                        ReadFile(argv[1]);
                    else
                        ReadFile(source);
                    XClearWindow(display,main_win);
                    DrawGrid();
                    break;
                case Button2:
                    if (argc == 3) ReadFile(argv[2]);
                    else ReadFile(destinate);
                    XClearWindow(display,main_win);
                    DrawGrid();
                    break;
                case Button3:
                    XCloseDisplay(display);
                    exit(0);
                    break;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ.

การเขียนโปรแกรมให้ใช้งานร่วมกัน Motif

How to program with Motif ?

Motif เป็น Library บนระบบ UNIX ซึ่งเราสามารถเขียนโปรแกรมภาษา C เพื่อเรียกใช้งาน Library ที่ได้ทำการติดตั้งไว้แล้ว Libraries เหล่านี้จะสนับสนุนการเขียนโปรแกรมเพื่อแสดงผลการทำงานในแบบ Graphic User Interface

1. ความแตกต่างของการ Program ระหว่าง Motif และ X windows Library ที่เห็นได้ชัด 2 ข้อ คือ ประการแรก User Interface ที่ได้จากใช้ Motif การเขียนโปรแกรมจะมีลักษณะ 3 มิติ ส่วน X windows เป็น 2 มิติ และประการที่สอง Motif สามารถเขียนโปรแกรมได้เข้าใจง่ายกว่า X windows

2. ลักษณะการทำงานของโปรแกรมที่พัฒนาาร่วมกัน Motif

2.1 สนับสนุนการทำงานแบบ event - driver โดยจะมีการสร้าง even manager เพื่อติดต่อกัน User Interface

2.2 การออกแบบจะเป็นแบบเชิงวัตถุ โดยเราจะต้องเขียนโปรแกรมให้สามารถทำงานเป็นกลุ่มก้อน โดยจะอาศัยตัวแปรประเภท Pointer ในการควบคุม Object ซึ่งใน Motif จะถูกเรียกว่า Widget

3. เราสามารถหา Motif ได้จากที่ใดบ้าง

เราสามารถ download program Motif ได้จาก <ftp://ftp.lesstif.org/pub/hungry/lesstif> โดยภายใน Site นี้จะให้บริการ copy program Motif ซึ่งนำไปใช้สำหรับเครื่อง PC Linux หรือ เครื่องคอมพิวเตอร์ที่ทำงานอยู่ภายในระบบปฏิบัติการ Unix

4. การติดตั้งโปรแกรม Motif โดยสังเขป

- 4.1 download program จาก <ftp://ftp.lesstif.org/pub/hungry/lesstif>
- 4.2 ทำการ uncompression โดยใช้คำสั่ง `gunzip lesstif-current.tar.gz`
- 4.3 ทำการแตกแฟ้มข้อมูล โดยใช้คำสั่ง `tar -xvf lesstif-current.tar`
- 4.4 เข้าสู่ directory ของส่วนติดตั้ง `lesstif-current`
- 4.5 เรียกใช้โปรแกรม configuration เพื่อทำการตรวจสอบระบบ
- 4.6 สั่ง `make` เพื่อทำการ compile โปรแกรม Motif
- 4.7 สั่ง `make install` เพื่อทำการติดตั้งโปรแกรม motif

5. การเขียนโปรแกรม

เราสามารถ download program ตัวอย่างสำหรับภาษา C เพื่อศึกษาได้จาก <ftp://ftp.uu.net/published/oreilly/xbook/motif>

6. การ compile โปรแกรม

เนื่องจาก โปรแกรม Motif ที่ติดตั้งอยู่บนเครื่อง Convex Super Computer มีการติดตั้ง Libraries แยกออกจาก user library และ user include เราจะต้องทำการเขียน Makefile ดังนี้

Makefile :

```
CFLAGS = -I/usr/include/X11R6 -I/usr/include/Motif1.2
```

```
LDFLAGS = -L/usr/lib/X11R6 -L/usr/lib/Motif1.2 -lXm -lXt -lX11
```

```
PROGRAMS = source_file <- ชื่อ source code ที่เราทำการเขียนขึ้น
```

```
all : $(PROGRAMS)
```

clean :

```
-$ rm -f $(PROGRAMS) <- สำหรับทำการลบโปรแกรมที่เรา compile แล้วเพื่อ compile
```

ใหม่

```

Module1.c :
#include <Xm/MainW.h>
#include <Xm/PushBG.h>
#include <Xm/PushB.h>
#include <Xm/LabelG.h>
#include <Xm/Label.h>
#include <Xm/TextF.h>
#include <Xm/Frame.h>
#include <Xm/RowColumn.h>
#include <Xm/Form.h>
#include <Xm/Text.h>
#include <ctype.h>
#include <stdio.h>
#include <math.h>

void ControlX();
void ok_pushed( );
void close_dialog( );
void help_cb();
void DestroyShell( );

Widget PbX[4][4];
double Control_X[4][4];

main
{
    String col[] = {"Col 1", "Col 2", "Col 3", "Col 4"};
    String row[] = {"Row 1", "Row 2", "Row 3", "Row 4"};

    static Widget frame,main_w,pb,label,w,parent;
    static Widget cx_shell;
    int x, y;
    String Str;

    extern void print_result();
    extern void ok_Xpushed();
    extern void check_point();
    extern void keep_data();

    cx_shell = XtVaAppInitialize(&app,"Control_X",
        &argc,XmNtitle," Control X ",NULL,NULL);

    main_w = XtVaCreateWidget(" ",
        xmFormWidgetClass, cx_shell,
        XmNwidth , 400,
        XmNheight, 245,
        NULL);

    frame = XtVaCreateManagedWidget(" Control X",
        xmLabelGadgetClass, main_w,
        XmNtopAttachment, XmATTACH_FORM,
        XmNleftAttachment, XmATTACH_FORM,
        XmNrightAttachment, XmATTACH_FORM,
        XmNtopOffset, 5,
        XmNleftOffset, 5,
        XmNrightOffset, 5,
        XmNshadowType, XmSHADOW_ETCHED_IN,
        NULL);

    frame = XtVaCreateManagedWidget("B",
        xmFrameWidgetClass, main_w,
        XmNtopAttachment, XmATTACH_WIDGET,
        XmNtopWidget, frame,
        XmNleftAttachment, XmATTACH_FORM,
        XmNrightAttachment, XmATTACH_FORM,
        XmNtopOffset, 5,
        XmNleftOffset, 5,
        XmNrightOffset, 5,
        NULL);

    parent = XtVaCreateManagedWidget ("C",
        xmFormWidgetClass, frame,
        XmNfractionBase, 5,
        NULL);
}

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

XtVaCreateManagedWidget ("row / col",
    xmLabelGadgetClass, parent,
    XmNtopAttachment, XmATTACH_POSITION,
    XmNtopPosition, 0,
    XmNleftAttachment, XmATTACH_POSITION,
    XmNleftPosition, 0,
    XmNrightAttachment, XmATTACH_POSITION,
    XmNrightPosition, 1,
    XmNbottomAttachment, XmATTACH_POSITION,
    XmNbottomPosition, 1,
    NULL);

for (x = 1; x < 5; x++) {
    for (y = 1; y < 5; y++){
        PbX[x-1][y-1] = XtVaCreateManagedWidget ("CTRX",
            xmTextFieldWidgetClass, parent,
            XmNtopAttachment, XmATTACH_POSITION,
            XmNtopPosition, Y,
            XmNleftAttachment, XmATTACH_POSITION,
            XmNleftPosition, x,
            XmNrightAttachment, XmATTACH_POSITION,
            XmNrightPosition, x+1,
            XmNbottomAttachment, XmATTACH_POSITION,
            XmNbottomPosition, y+1,
            NULL);
        XmTextFieldSetString(PbX[x-1][y-1], "0.00");
        XtAddCallback ( PbX[x-1][y-1], XmNmodifyVerifyCallback, check_point, NULL );
        XtAddCallback ( PbX[x-1][y-1], XmNactivateCallback, keep_data, NULL);
    }
    XtVaCreateManagedWidget (col[x-1],
        xmLabelGadgetClass, parent,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, 0,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, x,
        XmNrightAttachment, XmATTACH_POSITION,
        XmNrightPosition, x+1,
        XmNbottomAttachment, XmATTACH_POSITION,
        XmNbottomPosition, 1,
        NULL);
    XtVaCreateManagedWidget (row[x-1],
        xmLabelGadgetClass, parent,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, x,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, 0,
        XmNrightAttachment, XmATTACH_POSITION,
        XmNrightPosition, 1,
        XmNbottomAttachment, XmATTACH_POSITION,
        XmNbottomPosition, x+1,
        NULL);
}
XtManageChild(parent);

frame = XtVaCreateManagedWidget("E",
    xmFrameWidgetClass, main_w,
    XmNtopAttachment, XmATTACH_WIDGET,
    XmNtopWidget, frame,
    XmNleftAttachment, XmATTACH_FORM,
    XmNbottomAttachment, XmATTACH_FORM,
    XmNrightAttachment, XmATTACH_FORM,
    XmNtopOffset, 5,
    XmNleftOffset, 5,
    XmNrightOffset, 5,
    NULL);

parent = XtVaCreateManagedWidget ("F",
    xmFormWidgetClass, frame,
    XmNtopAttachment, XmATTACH_FORM,
    XmNbottomAttachment, XmATTACH_FORM,
    XmNleftAttachment, XmATTACH_FORM,
    XmNrightAttachment, XmATTACH_FORM,
    XmNfractionBase, 2,
    NULL);

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pb = XtVaCreateManagedWidget("OK",
    xmPushButtonGadgetClass, parent,
    XmNtopAttachment, XmATTACH_FORM,
    XmNbottomAttachment, XmATTACH_FORM,
    XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, 0, XmNrightAttachment,
        XmATTACH_POSITION, XmNrightPosition, 1,
    XmNshowAsDefault, True,
    XmNdefaultButtonShadowThickness, 1, NULL);

/* ON-OK Press */
/* XtAddCallback (pb, XmNactivateCallback, ok_Xpushed, NULL); */
pb = XtVaCreateManagedWidget("CANCEL",
    xmPushButtonGadgetClass, parent,
    XmNtopAttachment, XmATTACH_FORM,
    XmNbottomAttachment, XmATTACH_FORM,
    XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, 1,
    XmNrightAttachment, XmATTACH_FORM,
    XmNshowAsDefault, True,
    XmNdefaultButtonShadowThickness, 1, NULL);
/* ON-CANCEL Press */
/* XtAddCallback (pb, XmNactivateCallback, exit, NULL); */
XtManageChild(parent);
XtManageChild(main_w);
XtAppMainLoop(app);
}

void
ok_Xpushed(widget, client_data, call_data)
Widget widget;
XtPointer client_data;
XtPointer call_data;
{ int i,j;
  double val;
  String str;
  char *value ;
  for (i = 0;i < 4;i++)
    for (j = 0;j < 4;j++)
      { value = XmTextFieldGetString (PbX[i][j]);
        if (strlen(value) > 0)
          { strcpy(str,value);
            strcat(str, " ");
            sscanf(str,"%lf",&val);
            printf("%s %lf\n",value,val);
          }
        XtFree (value);
        XmTextFieldSetString(PbX[i][j],"11.00");
      }
  exit(); }
/* Check float number */
void check_point(text_w, client_data, call_data)
Widget text_w;
XtPointer client_data;
XtPointer call_data;
{
  char c;
  int sea;
  char str[20];
  int len = XmTextGetLastPosition(text_w);
  XmTextVerifyCallbackStruct *cbs =
    ( XmTextVerifyCallbackStruct *) call_data;

/* no backspacing or typing in the middle of string */
/* if (cbs->currInsert < len) {
  cbs->doit = False;
  return;
} */

if (cbs->text->length > 1) {
  cbs->doit = False;
  return;
}
c = cbs->text->ptr[0];
if ((c == '.') && (len > 0))
  strcpy(str,XmTextFieldGetString(text_w));
  for (sea = 0;sea < len;sea++)

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นโดยศูนย์เทคโนโลยีสารสนเทศฯ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        if (str[sea] == '.')
        {
            cbs->doit = False;
            return;
        }
    }

    /* don't allow non-digits or let the input exceed 12 chars */
    if (!isdigit (c) && (c != '.'))
        cbs->doit = False;
    else
    {
        if (cbs->doit == True)
        {
            cbs->text->ptr = XtRealloc (cbs->text->ptr, 1);
            cbs->text->ptr[0] = c;
        }
    }
}

void
keep_data(text_w, client_data, call_data)
Widget      text_w;
XtPointer   client_data;
XtPointer   call_data;
{
    XmProcessTraversal (text_w, XmTRAVERSE_NEXT_TAB_GROUP);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายอภิชาติ พลาศิลป์มงคล
วันเดือนปีเกิด	24 กันยายน 2515
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตร์บัณฑิต สาขาวิชา วิทยาการคอมพิวเตอร์
สถานที่สำเร็จการศึกษา	สถาบันราชภัฏสวนดุสิต
ปีที่สำเร็จการศึกษา	2536



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้