สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมสำหรับการแก้ปัญหาสมการปัวส์ซองโดยระเบียบวิธีไฟไนต์เอลิเมนต์

SOFTWARE FOR SOLVING POISSON'S EQUATION BY FINITE ELEMENT METHODS

ณัฏฐารมย์ เธียรเมธาวุฒิ
วัชรากร ธิติเลิศ
อิงพร รุ่งโรจน์รัตนกุล

วพ.
ณ3192
2549

เลขหมู่.....
เลขทะเบียน 73320
วัน,เดือน,ปี 12 ก.ค. 2550

b 11x 90507
i.....

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOFTWARE FOR SOLVING POISSON'S EQUATION BY FINITE ELEMENT METHODS

NUTTHAROM THAINMETHAWUT

WATCHARAKORN THITILERT

ENGPORN RUNGROJRATTANAKUL

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2006

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	โปรแกรมสำหรับการแก้ปัญหาสมการปัวส์ซองโดยระเบียบวิธีไฟไนต์เอลิเมนต์		
	SOFTWARE FOR SOLVING POISSON'S EQUATION BY FINITE ELEMENT METHODS		
ชื่อนักศึกษา	นางสาวณัฏฐารมย์	เธียรเมธาวุฒิ	46050318
	นายวัชรากร	ธิติเลิศ	46050732
	นางสาวอิงพร	รุ่งโรจน์รัตนกุล	46050741
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์		
สาขาวิชา	วิทยาการคอมพิวเตอร์		
อาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์	ประกอบผล	
	รศ.ภัคคินี	ชิตสกุล	

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2549

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	รศ.ดร.วีระ บุญจริง	
กรรมการ	ผศ.ดร.จีรพร ศรีสวัสดิ์	
กรรมการและอาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์ ประกอบผล	
กรรมการและอาจารย์ที่ปรึกษา	รศ.ภัคคินี ชิตสกุล	

(รองศาสตราจารย์ ดร.วีระ บุญจริง)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	โปรแกรมสำหรับการแก้ปัญหาสมการปัวส์ซองโดยระเบียบวิธีไฟไนต์เอลิเมนต์		
ชื่อนักศึกษา	นางสาวณัฏฐารมย์	เธียรเมธาวุฒิ	46050318
	นายวัชรากร	ธิติเลิศ	46050732
	นางสาวอิงพร	รุ่งโรจน์รัตนกุล	46050741
ปริญญา	วิทยาศาสตรบัณฑิต		
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์		
สาขาวิชา	วิทยาการคอมพิวเตอร์		
ปีการศึกษา	2549		
อาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์	ประกอบผล	
	รศ.ภัคคินี	ชิตสกุล	

บทคัดย่อ

ปัญหาพิเศษนี้มีจุดประสงค์เพื่อพัฒนาโปรแกรมสำหรับการแก้ปัญหาสมการปัวส์ซองโดยระเบียบวิธีไฟไนต์เอลิเมนต์ ซึ่งรูปแบบปัญหาต้องเป็นสี่เหลี่ยมมุมฉากเท่านั้น โดยที่ผลลัพธ์จะถูกแสดงด้วยกราฟฟิค และผลลัพธ์ที่ได้เมื่อเปรียบเทียบกับโปรแกรม MATLAB จะมีความแตกต่างกันไม่เกิน 2% ขึ้นอยู่กับความละเอียดที่กำหนดให้แก่โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	SOFTWARE FOR SOLVING POISSON'S EQUATION BY FINITE ELEMENT METHODS		
Students	Miss. Nuttharom	Thainmethawut	46050318
	Mr. Watcharakorn	Thitilert	46050732
	Miss. Engporn	Rongrojrattanakul	46050741
Degree	Bachelor of Science		
Department	Mathematics and Computer Science,Faculty of Science		
Programme	Computer Science		
Academic Year	2006		
Special Project Advisor	Assoc.Prof.Teerawat	Prakobphon	
	Assoc.Prof.Pakkinee	Chitsakul	

ABSTRACT

The purpose of this special project is to develop a program for solving the Poisson's equation using the Finite Element method. The problem is solved only in a form of rectangle such as hot plate. The results are shown as a graphical model. The computed results from our program are compared to those from the MATLAB software with the difference at most 2%. However, the more refinement of grid, the more improved the results.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องการพัฒนาโปรแกรมเพื่อแก้ปัญหาสมการปัวส์ซองโดยระเบียบวิธีไฟไนต์เอลิเมนต์สามารถสำเร็จลุล่วงไปด้วยดี ทางคณะผู้จัดทำต้องขอขอบพระคุณ รองศาสตราจารย์ภัคคินี ชิตสกุล และรองศาสตราจารย์ ธีรวัฒน์ ประกอบผล ซึ่งเป็นอาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่าง ๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์จนการทำปัญหาพิเศษครั้งนี้สำเร็จด้วยดี รวมทั้งเพื่อน ๆ พี่ ๆ และน้อง ๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่าง ๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ

มีนาคม 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญภาพ VII
คำอธิบายสัญลักษณ์และคำย่อ XI

บทที่ 1 บทนำ 1
1.1 ความสำคัญและที่มาของปัญหา 1
1.2 วัตถุประสงค์ของการทำ 1
1.3 ขอบเขตของปัญหา 1
1.4 ส่วนประกอบของปัญหาพิเศษ 2

บทที่ 2 ทฤษฎีและหลักการของวิธีไฟไนต์เอลิเมนต์ 4
2.1 วิธีการไฟไนต์เอลิเมนต์ 6
2.2 ข้อแตกต่างระหว่างวิธีการผลต่างสืบเนื่องและวิธีการไฟไนต์เอลิเมนต์ 7
2.3 ขั้นตอนทั่วไปของวิธีการไฟไนต์เอลิเมนต์ 7
2.4 การสร้างสมการโดยวิธีการแปรผัน 10
2.4.1 วิธีการแปรผัน 10
2.4.2 ฟังก์ชันการประมาณภายในเอลิเมนต์สองมิติ 11
2.4.3 การสร้างสมการไฟไนต์เอลิเมนต์แบบสองมิติโดยวิธีการแปรผัน 15
2.5 การสร้างสมการไฟไนต์เอลิเมนต์โดยวิธีการถ่วงน้ำหนักเศษตกค้าง 19
2.5.1 วิธีกาเลอร์คิน 19
2.5.2 วิธีการถ่วงน้ำหนักเศษตกค้างสำหรับไฟไนต์เอลิเมนต์ 20
2.5.2.1 ลำดับขั้นตอนทั่วไป 20
2.6 ฟังก์ชันการประมาณภายในเอลิเมนต์และการอินทิเกรตเอลิเมนต์เมทริกซ์เชิงตัวเลข 22
2.6.1 ฟังก์ชันการประมาณภายในเอลิเมนต์สองมิติ 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.6.1.1 เอลิเมนต์รูปสามเหลี่ยม 23
2.7 การแสดงผลทางกราฟ 25

บทที่ 3 ขั้นตอนการดำเนินงานและหลักการของโปรแกรม 28
3.1 ขั้นตอนในการดำเนินงาน 28
3.2 ลักษณะการทำงานของโปรแกรม 28
3.3 ขั้นตอนการทำงานของโปรแกรม 30
3.4 ส่วนประกอบของโปรแกรม 33
3.5 ทฤษฎีและหลักการที่ใช้ในการพัฒนาโปรแกรม 34
3.5.1 ระเบียบวิธีไฟไนต์เอลิเมนต์ 34
3.5.2 โปรแกรมระเบียบวิธีไฟไนต์เอลิเมนต์ด้วย MATLAB 51
3.5.3 Triangular decomposition 60
3.5.4 ขั้นตอนวิธีสำหรับการหาค่า z จากค่า x และ y ที่รับมาจากผู้ใช้ในหน้าแสดงผลกราฟ 66

บทที่ 4 การประเมินผล 70
4.1 วิธีการใช้งานโปรแกรม 70
4.2 การใช้งานกราฟแสดงผล 72
4.2.1 การควบคุมกราฟ 72
4.2.2 การคำนวณผลลัพธ์จากค่า x และ y ที่กำหนดให้ 72
4.3 ผลลัพธ์ที่ได้จากโปรแกรม 74
4.4 การประเมินประสิทธิภาพของโปรแกรม 80
4.4.1 ฟังก์ชันทางด้านขวาเป็นค่าคงที่ 81
4.4.2 ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x 84
4.4.3 ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน y 90
4.4.4 ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน xy 92
4.5 ข้อจำกัดของโปรแกรม 94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 5 สรุปและข้อเสนอแนะ 95

5.1 สรุปผล 95

5.2 ข้อเสนอแนะ 95

5.3 ประโยชน์ที่คาดว่าจะได้รับ 102

ภาคผนวก 103

บรรณานุกรม 221

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่ **หน้า**

รูปที่ 2.1 แสดงถึงลักษณะของแผ่นอะลูมิเนียมลักษณะหนึ่งที่ใช้เสริมความแข็งแรงของโครงสร้างภายในปีกเครื่องบิน การวิเคราะห์หาการกระจายของความเค้นบนแผ่นอะลูมิเนียมนี้ภายใต้แรงกระทำที่กำหนดให้โดยการใช้วิธีผลต่างสืบเนื่อง (finite different method) 5
รูปที่ 2.2 แสดงให้เห็นว่าตารางสี่เหลี่ยมที่ใช้วิธีผลต่างสืบเนื่องไม่สามารถจำลองรูปร่างลักษณะดั้งเดิมที่แท้จริงของแผ่นอะลูมิเนียมดังกล่าวได้โดยเที่ยงตรง หากใช้ตารางสี่เหลี่ยมให้มีขนาดเล็กลงซึ่งหมายถึงจะต้องเพิ่มจำนวนตารางสี่เหลี่ยมให้มากขึ้นก็จะสามารถจำลองรูปร่างลักษณะดั้งเดิมที่แท้จริงได้ใกล้เคียงมากยิ่งขึ้น 5
รูปที่ 2.3 การวิเคราะห์หาผลเฉลยบนแผ่นอะลูมิเนียมด้วยการใช้วิธีการไฟไนต์เอลิเมนต์ 6
รูปที่ 2.4 แสดงถึงการแบ่งรูปร่างลักษณะของปัญหาออกเป็นเอลิเมนต์แบบต่าง ๆ 8
รูปที่ 2.5 เอลิเมนต์สามเหลี่ยมแบบอย่าง ประกอบด้วยสามจุดต่อ โดยมีตัวไม่รู้ค่าอยู่ ณ ตำแหน่งที่จุดต่อ 8
รูปที่ 2.6 การแบ่งขอบเขตออกเป็นไฟไนต์เอลิเมนต์ย่อย 11
รูปที่ 2.7 เอลิเมนต์สามเหลี่ยม 12
รูปที่ 2.8 ตัวอย่างเอลิเมนต์สามเหลี่ยม 13
รูปที่ 2.9 ลักษณะการกระจายของฟังก์ชันการประมาณภายในเอลิเมนต์ 14
รูปที่ 2.10 ลักษณะการกระจายของผลเฉลยโดยประมาณภายในเอลิเมนต์ 15
รูปที่ 2.11 การแบ่งลักษณะรูปร่างของปัญหาออกเป็นเอลิเมนต์ 20
รูปที่ 2.12 ฟังก์ชันการประมาณภายในเอลิเมนต์รูปสามเหลี่ยม 24
รูปที่ 2.13 เอลิเมนต์สามเหลี่ยมอันดับสูงที่ประกอบด้วยหลายจุดต่อ 24
รูปที่ 2.14 แสดงวิธีการสร้างกราฟในแนวแกน x และแกน y 26
รูปที่ 2.15 แสดงเอลิเมนต์สามเหลี่ยมซึ่งมีพิกัด (x,y) ที่ต้องการทราบค่าอยู่ 26
รูปที่ 3.1 แสดงลำดับขั้นตอนการทำงานของโปรแกรม 30
รูปที่ 3.2 ภาพตัวอย่างแสดงการใส่ค่าต่าง ๆ โดยใช้สมการปัวส์ซองและฟังก์ชันทางด้านขวาเป็นค่าคงที่ 31
รูปที่ 3.3 ภาพที่แสดงให้เราใส่พิกัดจุดของ x และ y เพื่อหาค่าผลเฉลย ณ จุดพิกัดนั้น ๆ 32
รูปที่ 3.4 ภาพแสดงผลเฉลยของสมการปัวส์ซองเมื่อฟังก์ชันทางด้านขวาเป็นค่าคงที่ 33
รูปที่ 3.5 แสดงส่วนประกอบของโปรแกรม 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่ **หน้า**

รูปที่ 3.6 แสดงการแบ่งพื้นที่ปัญหาด้วยเอลิเมนต์สามเหลี่ยม 34

รูปที่ 3.7 แสดงเอลิเมนต์สามเหลี่ยมที่มีจุด (x,y) อยู่ และพิกัดจุดต่าง ๆ
เมื่อลากเส้นผ่านจุด (x,y) 67

รูปที่ 3.8 แสดงสมการเส้นตรงของเส้นทแยงในเอลิมนต์สามเหลี่ยม 68

รูปที่ 3.9 แสดงระยะห่าง l_s และระยะห่าง l_t 69

รูปที่ 3.10 แสดงค่าที่นำมาคำนวณ ณ ตำแหน่งต่าง ๆ ในเอลิเมนต์สามเหลี่ยม 69

รูปที่ 4.1 ภาพตัวอย่างแสดงผลการรับค่าจากผู้ใช้โดยใช้สมการปัวส์ซองและฟังก์ชันทางด้านขวา
เป็นค่าคงที่ 71

รูปที่ 4.2 แสดงวิธีใช้งานโปรแกรมในการหาค่าผลเฉลย 72

รูปที่ 4.3 แสดงผลลัพธ์ที่ได้จากการคำนวณ 73

รูปที่ 4.4 ภาพแสดงผลลัพธ์ด้วยกราฟจากสมการลาปลาซ 74

รูปที่ 4.5 ภาพแสดงผลลัพธ์ด้วยกราฟจากสมการปัวส์ซองโดยระบุฟังก์ชันทางด้านขวา
ของสมการเป็นค่าคงที่ 75

รูปที่ 4.6 ภาพแสดงผลลัพธ์ด้วยกราฟจากสมการปัวส์ซองโดยระบุฟังก์ชันทางด้านขวา
ของสมการเป็นฟังก์ชัน x 76

รูปที่ 4.7 ภาพแสดงผลลัพธ์ด้วยกราฟจากสมการปัวส์ซองโดยระบุฟังก์ชันทางด้านขวา
ของสมการเป็นฟังก์ชัน y 77

รูปที่ 4.8 ภาพแสดงผลลัพธ์ด้วยกราฟจากสมการปัวส์ซองโดยระบุฟังก์ชันทางด้านขวา
ของสมการเป็นฟังก์ชัน xy 78

รูปที่ 4.9 แสดงการเปรียบเทียบกราฟผลลัพธ์จากรูปสมการ 4 แบบ 79

รูปที่ 4.10 ภาพแสดงผลลัพธ์เป็นกราฟสามมิติที่ได้จากโปรแกรม MATLAB
โดยที่ฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่าเท่ากับ 20 80

รูปที่ 4.11 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่าเท่ากับ 20
โดยมีความละเอียดเท่ากับ 7x7 81

รูปที่ 4.12 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่าเท่ากับ 20
โดยมีความละเอียดเท่ากับ 9x9 82

รูปที่ 4.13 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่าเท่ากับ 20
โดยมีความละเอียดเท่ากับ 13x13 83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่ **หน้า**

รูปที่ 4.14 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 4x4 84

รูปที่ 4.15 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 4x9 85

รูปที่ 4.16 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 9x9 86

รูปที่ 4.17 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 9x9 87

รูปที่ 4.18 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 9x4 88

รูปที่ 4.19 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 9x4 89

รูปที่ 4.20 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน y โดยมีความละเอียดเท่ากับ 4x4 90

รูปที่ 4.21 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน y โดยมีความละเอียดเท่ากับ 9x9 91

รูปที่ 4.22 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน xy โดยมีความละเอียดเท่ากับ 4x4 92

รูปที่ 4.23 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน xy โดยมีความละเอียดเท่ากับ 9x9 93

รูปที่ 5.1 แสดงลักษณะขอบเขตของปัญหาในรูปแบบต่าง ๆ 96

รูปที่ 5.2 แสดงวิธีการกำหนดหมายเลขให้จุดต่อแต่ละจุดต่อของขอบเขตปัญหาที่เป็นสามเหลี่ยม 96

รูปที่ 5.3 แสดงส่วนพื้นที่สี่เหลี่ยมที่มุมด้านหนึ่งถูกตัดออกไป 98

รูปที่ 5.4 แสดงวิธีการกำหนดหมายเลขให้จุดต่อแต่ละจุดต่อของขอบเขตปัญหาที่เป็นห้าเหลี่ยม 98

รูปที่ 5.5 แสดงการแบ่งพื้นที่ขอบเขตปัญหาที่เป็นวงกลมออกเป็นเอลิเมนต์ย่อย ๆ 99

รูปที่ 5.6 แสดงการจัดเก็บข้อมูลค่าต่าง ๆ ทางฟิสิกส์ที่รับเข้ามา 100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่ **หน้า**

รูปที่ 5.7 แสดงการจัดเก็บข้อมูลค่าตำแหน่งของจุดต่อเริ่มต้นและจุดสิ้นสุด, ระยะห่างระหว่างจุดต่อและเงื่อนไขขอบเขต 100
รูปที่ 5.8 แสดงการนำแฟ้มข้อมูลที่ได้ทั้งหมดมาเข้าสู่การคำนวณ ด้วยระเบียบวิธีไฟไนต์เอลิเมนต์ 101
รูปที่ 5.9 นำแฟ้มข้อมูลจากรูปที่ 5.8 มาเข้าสู่กระบวนการ LU Decomposition 102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายสัญลักษณ์และคำย่อ

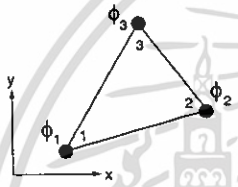
1. ∂ เป็นสัญลักษณ์แทนการอนุพันธ์บางส่วน

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

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

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

3. Φ เป็นสัญลักษณ์แทนตัวไม่ทราบค่าที่ต้องการหาคำตอบ

   เช่น 

   ต้องการหาค่า Φ_1, Φ_2 และ Φ_3 ตามจุดที่กำหนด

4. $\sum$ เป็นสัญลักษณ์แทนการบวกสะสม

5. $\int$ เป็นสัญลักษณ์แทนการหาผลรวมของค่าภายในพื้นที่ที่กำหนด

6. Ω เป็นสัญลักษณ์แทนพื้นที่ภายในที่ถูกปิดล้อม

7. Γ เป็นสัญลักษณ์แทนขอบเขตของพื้นที่ปิดล้อม

8. **const** เป็นคำย่อมาจาก constant หมายถึง ค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

คณิตศาสตร์เป็นพื้นฐานที่ทำให้เกิดคอมพิวเตอร์ ซึ่งเราสามารถนำหลักการทางคณิตศาสตร์ไปประยุกต์ใช้ในการแก้ปัญหาเชิงวิทยาศาสตร์ได้อย่างหลากหลาย ซึ่งหากต้องการนำคณิตศาสตร์ไปใช้ บางครั้งอาจจะมีความยากเกินไปที่จะนำไปใช้ได้โดยคนทั่วไปที่ไม่เคยศึกษาคณิตศาสตร์มาก่อน ดังนั้นจึงเป็นที่มาของการพัฒนาโปรแกรมคอมพิวเตอร์นี้ ซึ่งในงานนี้จะนำปัญหาเกี่ยวกับสมการปัวส์ซองมาประยุกต์สร้างเป็นโปรแกรมคอมพิวเตอร์

สมการปัวส์ซองเป็นสมการทางคณิตศาสตร์ที่นำไปใช้แก้ปัญหาในเชิงวิทยาศาสตร์ ซึ่งตัวสมการปัวส์ซองจะอยู่ในรูปของ PDE ดังนี้

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

หรือ

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

และการแก้สมการในรูปของ PDE นี้จะใช้ระเบียบวิธีไฟไนต์เอลิเมนต์ และจะแสดงผลในรูปของกราฟ

1.2 วัตถุประสงค์ของการทำ

1. พัฒนาโปรแกรมเพื่อหาผลเฉลยของสมการปัวส์ซอง 2มิติโดยระเบียบวิธีไฟไนต์เอลิเมนต์
2. ให้บุคคลที่สนใจในเรื่องของระเบียบวิธีไฟไนต์เอลิเมนต์สำหรับการแก้สมการปัวส์ซอง 2 มิติ สามารถใช้เป็นแหล่งข้อมูลในการศึกษาและนำไปพัฒนาต่อยอดได้
3. เพื่อลดความซับซ้อนในการแก้ปัญหาสมการปัวส์ซอง 2 มิติ โดยระเบียบวิธีไฟไนต์เอลิเมนต์

1.3 ขอบเขตของปัญหา

สภาพปัญหาที่ใช้ในการศึกษาเพื่อพัฒนาโปรแกรม เป็นการศึกษาการถ่ายเทความร้อนบนเนื้อวัตถุแผ่นเรียบที่มีลักษณะเป็นสี่เหลี่ยมผืนผ้า โดยวัตถุแผ่นเรียบนี้จะกำหนดพิกัดอยู่ในจตุภาคที่ 1 เท่านั้น เพราะปัญหาทางกายภาพเมื่อกล่าวถึงพิกัด จะต้องเป็นพิกัดที่มีอยู่จริง ซึ่งในความหมายทางวิทยาศาสตร์พิกัดที่มีอยู่จริงจะต้องมีค่าเป็นบวก ดังเช่นระยะทางหรือระยะห่าง จะไม่บอกค่าออกมาเป็นลบ เพราะไม่สามารถสื่อความหมายออกมาได้ โดยรูปปัญหาจะกำหนดให้ขอบสัมผัสไปกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวแกน x และแนวแกน y ซึ่งจะให้มุมเริ่มต้นอยู่ที่จุดกำเนิด และสมการปัวส์ซองที่ใช้ในการคำนวณจะมีเพียง 4 รูปแบบเท่านั้น

โดยรูปแบบของสมการปัวส์ซอง เป็นดังนี้

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

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

ซึ่งขึ้นอยู่กับฟังก์ชันทางด้านขวาของสมการตามนี้

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

ภายใต้เงื่อนไขขอบเขต

1. Dirichlet เป็นการกำหนดค่าที่ขอบเป็นค่าคงที่ โดยมีลักษณะเหมือนว่าเมื่อมีการถ่ายเทความร้อนมาที่ขอบหรือจากขอบเข้าไป จะต้องมีค่าขอบเท่าที่กำหนดให้เสมอ
2. Neuman เป็นการกำหนดค่าที่ขอบ โดยมีลักษณะเป็นอัตราการเปลี่ยนแปลง ซึ่งมีลักษณะเหมือนว่าเมื่อมีการถ่ายเทความร้อนที่ขอบแล้ว จะพุ่งผ่านออกไปในลักษณะเพิ่มขึ้นหรือลดลง หรือจากความร้อนที่ขอบจะให้พุ่งผ่านเข้าไปในลักษณะเพิ่มขึ้นหรือลดลง

ดังนั้น วัตถุที่จะทำการคำนวณจะเป็นสี่เหลี่ยมผืนผ้าซึ่งอยู่ในจตุภาคที่ 1 เท่านั้น และสมการที่ใช้ในการคำนวณเป็นสมการลาปลาซและสมการปัวส์ซอง 4 แบบตามที่กล่าวไว้ข้างต้น

1.4 ส่วนประกอบของปัญหาพิเศษ

ส่วนที่เหลือของปัญหาพิเศษ แบ่งเป็นบทต่าง ๆ ดังนี้

บทที่ 2 ทฤษฎีและหลักการของวิธีไฟไนต์เอลิเมนต์ เปรียบเทียบข้อแตกต่างระหว่างวิธีการผลต่างสืบเนื่องและวิธีการไฟไนต์เอลิเมนต์ แสดงขั้นตอนทั่วไปของวิธีการไฟไนต์เอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิเมนต์และการสร้างสมการไฟไนต์เอลิเมนต์ด้วยวิธีต่าง ๆ รวมถึงการเลือกลักษณะของฟังก์ชันการประมาณภายในเอลิเมนต์ และวิธีการสร้างการแสดงผลทางกราฟ

บทที่ 3 ขั้นตอนการดำเนินงานและหลักการของโปรแกรมรวมทั้งแสดงตัวอย่างการทำงานของโปรแกรมสำหรับการแก้ปัญหาสมการ ปัวส์ซองโดยระเบียบวิธีไฟไนต์เอลิเมนต์

บทที่ 4 การประเมินประสิทธิผลของโปรแกรมสำหรับการแก้ปัญหาสมการปัวส์ซองโดยระเบียบวิธีไฟไนต์เอลิเมนต์

บทที่ 5 สรุปผลการพัฒนาปัญหาพิเศษว่าปัญหาพิเศษฉบับนี้ ประกอบไปด้วยส่วนประกอบใดบ้าง, มีการประเมินผลอย่างไร และข้อเสนอแนะในการพัฒนาต่อยอดปัญหาพิเศษฉบับนี้ต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

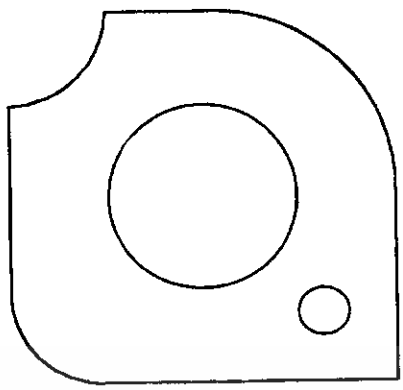
บทที่ 2
ทฤษฎีและหลักการของวิธีไฟไนต์เอลิเมนต์

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

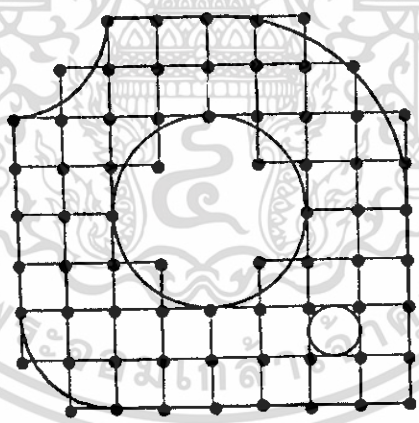
สมการเชิงอนุพันธ์ที่สอดคล้องกับปัญหาต่าง ๆ นั้น ปกติประดิษฐ์ขึ้นมาได้โดยไม่ยากเลย หากแต่ว่าผลเฉลยแม่นตรง (exact solution) ที่ต้องการและจำเป็นต้องทำการหาออกมาโดยวิธีวิเคราะห์ (analysis method) นั้นทำได้ยากลำบากมากหรืออาจจะหาไม่ได้เลยก็ได้ เหตุผลดังกล่าวก่อให้เกิดวิธีการหาผลเฉลยโดยประมาณ(approximate solution) ขึ้น วิธีการหาผลเฉลยโดยประมาณนั้นมีหลาย ๆ วิธีการ วิธีที่นิยมกันอย่างกว้างขวางในอดีตที่ผ่านมา คือ วิธีการผลต่างสืบเนื่อง (finite different method)

หลักการที่สำคัญของวิธีผลต่างสืบเนื่องก็คือการหาผลเฉลยโดยประมาณโดยเริ่มจากการเขียนสมการเชิงอนุพันธ์ให้อยู่ในรูปแบบของระบบสมการผลต่างสืบเนื่อง ข้อดีของวิธีการผลต่างสืบเนื่องนี้ก็คือ เป็นวิธีที่ง่ายแก่การศึกษาและทำความเข้าใจ รวมไปถึงความสะดวกในการเขียนโปรแกรมคอมพิวเตอร์เพื่อใช้ในการคำนวณหาผลเฉลยของปัญหานั้น ๆ และที่สำคัญที่สุด ได้กลายเป็นสิ่งที่จำเป็นเพื่อการพัฒนาปรับปรุงคุณภาพสิ่งของนั้น ๆ ให้ดียิ่งขึ้น ส่วนข้อเสียของวิธีการหาผลต่างสืบเนื่องก็มีหลายประการ เช่น ความไม่สะดวกในการกำหนดเงื่อนไขขอบเขต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



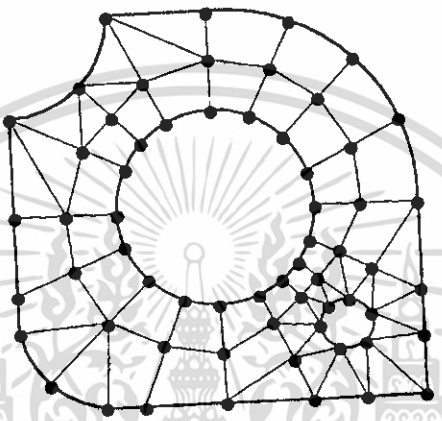
รูปที่ 2.1 แสดงถึงลักษณะของแผ่นอะลูมิเนียมลักษณะหนึ่งที่ใช้เสริมความแข็งแรงของโครงสร้างภายในปีกเครื่องบิน การวิเคราะห์หาการกระจายของความเค้น บนแผ่นอะลูมิเนียมนี้ภายใต้แรงกระทำที่กำหนดให้โดยการใช้วิธีผลต่างสืบเนื่อง (finite different method) จะเริ่มจากการแบ่งแผ่นอะลูมิเนียมนี้ออกเป็นช่องตารางสี่เหลี่ยม ดังรูปที่ 2.2 โดยที่ตารางสี่เหลี่ยมเหล่านี้ต่อกันที่จุดต่อ(grid points) ตามหัวมุมของสี่เหลี่ยมต่าง ๆ และขนาดของปัญหาหรือจำนวนตัวไม่รู้ค่า จะขึ้นอยู่กับจำนวนของจุดต่อนี้เอง



รูปที่ 2.2 นี้แสดงให้เห็นว่าตารางสี่เหลี่ยมที่ใช้วิธีผลต่างสืบเนื่องไม่สามารถจำลองรูปร่างลักษณะดั้งเดิมที่แท้จริงของแผ่นอะลูมิเนียมดังกล่าวได้โดยเที่ยงตรง หากใช้ตารางสี่เหลี่ยมให้มีขนาดเล็กลงซึ่งหมายถึงจะต้องเพิ่มจำนวนตารางสี่เหลี่ยมให้มากขึ้นก็จะสามารถจำลองรูปร่างลักษณะดั้งเดิมที่แท้จริงได้ใกล้เคียงมากยิ่งขึ้นแต่ในขณะเดียวกันจำนวนจุดต่อที่เพิ่มมากขึ้นทำให้จำนวนสมการผลต่างสืบเนื่องมากขึ้นด้วย และกระบวนการแก้ปัญหาจำเป็นต้องการหน่วยความจำบนเครื่องคอมพิวเตอร์เพิ่มขึ้นรวมถึงเวลาที่ใช้ในการคำนวณจะสูงมากขึ้นตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปข้างต้น จะแสดงให้เห็นถึงความยากลำบากในการนำไปประยุกต์วิธีการนี้เพื่อใช้กับปัญหาที่มีรูปร่างลักษณะซับซ้อนในปัจจุบันนี้ สาเหตุของความยากลำบากดังกล่าวมีส่วนก่อให้เกิดวิธีการหาผลเฉลยโดยประมาณวิธีใหม่ที่เรียกว่า วิธีไฟไนต์เอลิเมนต์ (finite element method: FEM) ซึ่งวิธีการนี้สามารถนำมาใช้กับปัญหาที่มีรูปร่างลักษณะซับซ้อนเช่นใดก็ได้ โดยสามารถจำลองรูปร่างลักษณะดั้งเดิมที่แท้จริง ได้ใกล้เคียงเที่ยงตรงกว่าหลักการของวิธีการผลต่างสืบเนื่อง กล่าวคือ เริ่มจากการแบ่งรูปร่างของปัญหาออกเป็นเนื้อที่หลาย ๆ ชิ้นที่เรียกว่า "เอลิเมนต์" โดยเอลิเมนต์ต่าง ๆ อาจจะอยู่ในรูปลักษณะของสามเหลี่ยมและสี่เหลี่ยมด้านไม่เท่าก็ได้ **ดังรูปที่ 2.3**



รูปที่ 2.3 การวิเคราะห์หาผลเฉลยบนแผ่นอะลูมิเนียมด้วยการใช้วิธีการไฟไนต์เอลิเมนต์

ผลประโยชน์สิ่งแรกที่เห็นได้ชัดเมื่อเปรียบเทียบได้ชัดเจนจากรูปที่ 2.2 โดยวิธีผลต่างสืบเนื่อง (finite different method) กับรูปที่ 2.3 โดยวิธีการไฟไนต์เอลิเมนต์ (finite element method) ประกอบด้วยเอลิเมนต์ขนาดต่าง ๆ กัน ที่สามารถจำลองรูปร่างลักษณะดั้งเดิมของแผ่นอะลูมิเนียมได้เป็นอย่างดี ซึ่งหมายถึงปัญหานี้จะถูกแก้เพื่อหาผลเฉลยโดยประมาณจากแบบจำลองทางคณิตศาสตร์ ที่มีรูปร่างลักษณะใกล้เคียงกับของจริงดั้งเดิมมากที่สุด ดังนั้นค่าผลเฉลยโดยประมาณที่จะคำนวณออกมาได้จะมีความแม่นยำมากขึ้นตามไปด้วย

2.1 วิธีการไฟไนต์เอลิเมนต์

ในการแก้ปัญหาใดปัญหาหนึ่ง ปัญหานั้นจะประกอบด้วยสมการเชิงอนุพันธ์และเงื่อนไขขอบเขตที่กำหนดมาให้ ค่าผลเฉลยแม่นตรง (exact solution) ดังกล่าวจะประกอบด้วยค่าของตัวแปรต่าง ๆ กันตามตำแหน่งต่าง ๆ บนรูปร่างลักษณะของปัญหานั้น หรือกล่าวอีกนัยหนึ่งก็คือ ค่าผลเฉลยแม่นตรงประกอบด้วยค่าต่าง ๆ ทั้งหมดนับเป็นจำนวนอนันต์ค่า แทนที่จะทำการหาค่าแม่นตรงที่ประกอบด้วยค่าต่าง ๆ จำนวนมากมายเช่นนี้ ซึ่งสำหรับปัญหาในทางปฏิบัติจะทำไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการก็คือ ทำการเปลี่ยนค่าทั้งหมดที่มีจำนวนอนันต์ค่านั้นมาเป็นค่าโดยประมาณที่มีจำนวนนับได้ (finite) ด้วยการแทนรูปร่างลักษณะของปัญหาด้วยเอลิเมนต์ (element) ซึ่งมีขนาดต่าง ๆ กัน

โดยหลักการของวิธีไฟไนต์เอลิเมนต์จะเริ่มต้นจากการพิจารณาเอลิเมนต์ทีละเอลิเมนต์ โดยทำการสร้างสมการสำหรับแต่ละเอลิเมนต์ที่ตั้งอยู่บนรากฐานที่ว่า สมการที่สร้างขึ้นมานั้นจำเป็นต้องสอดคล้องกับสมการเชิงอนุพันธ์ของปัญหาที่ทำอยู่ จากนั้นนำสมการของแต่ละเอลิเมนต์ที่สร้างขึ้นมาได้ มาประกอบกันเข้าก่อให้เกิดระบบสมการชุดใหญ่ ซึ่งในความหมายทางกายภาพ ก็คล้ายกับการนำทุกเอลิเมนต์มาประกอบรวมเข้าด้วยกันก่อให้เกิดรูปร่างลักษณะทั้งหมดของปัญหาที่แท้จริง จากนั้นจึงทำการกำหนดเงื่อนไขขอบเขตที่ให้มาลงในระบบสมการชุดใหญ่นี้แล้วจึงทำการแก้สมการดังกล่าว ซึ่งจะก่อให้เกิดผลเฉลยโดยประมาณที่ต้องการ ณ ตำแหน่งต่าง ๆ ของปัญหานั้น เป็นคำอธิบายในลักษณะกว้าง ๆ เพื่อที่จะให้เห็นภาพรวมของวิธีไฟไนต์เอลิเมนต์ ได้อธิบายในขั้นต้นว่าทำไมถึงต้องใช้วิธีไฟไนต์เอลิเมนต์ ซึ่งจะกล่าวถึงข้อแตกต่างของวิธีผลต่างสืบเนื่อง(finite different method) และวิธีไฟไนต์เอลิเมนต์ (finite element method)

2.2 ข้อแตกต่างระหว่างวิธีการผลต่างสืบเนื่องและวิธีการไฟไนต์เอลิเมนต์

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

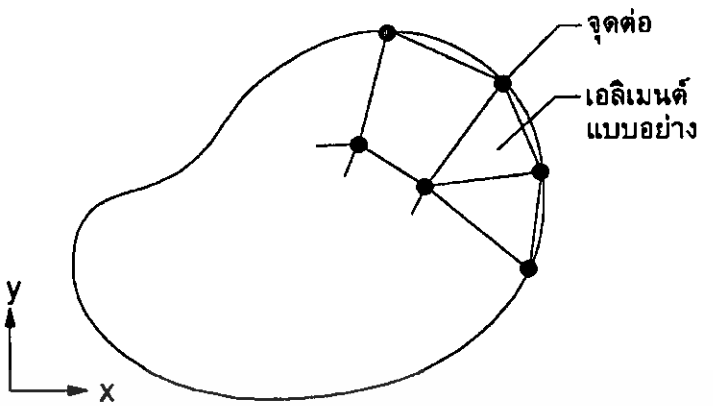
ส่วนวิธีการไฟไนต์เอลิเมนต์ เป็นการใช้ระเบียบวิธีเชิงตัวเลขเพื่อหาผลลัพธ์โดยประมาณของปัญหาที่กำหนดมาให้เช่นกัน โดยแบ่งรูปร่างลักษณะของปัญหาที่จะใช้ในการคำนวณออกเป็นชิ้นส่วนย่อย ๆ ที่เรียกว่า เอลิเมนต์ เอลิเมนต์เหล่านี้เชื่อมต่อกันที่จุดต่อ (nodes) ซึ่งเป็นตำแหน่งที่จะคำนวณหาค่าตัวแปรตาม (dependent variables)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

2.3 ขั้นตอนทั่วไปของวิธีการไฟไนต์เอลิเมนต์

วิธีการไฟไนต์เอลิเมนต์ประกอบด้วยขั้นตอนใหญ่ทั้งหมด 6 ขั้นตอน ดังต่อไปนี้

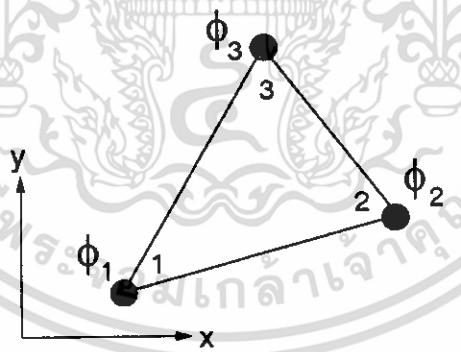
ขั้นตอนที่ 1 การแบ่งขอบเขตรูปร่างลักษณะของปัญหาที่ต้องการที่จะหาผลลัพธ์นั้นออกเป็นเอลิเมนต์ย่อย ๆ **ดังรูปที่ 2.4** ขอบเขตดังกล่าวอาจเป็นขอบเขตของปัญหาชนิดต่าง ๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงถึงการแบ่งรูปร่างลักษณะของปัญหาออกเป็นเอลิเมนต์แบบต่าง ๆ

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



รูปที่ 2.5 เอลิเมนต์สามเหลี่ยมแบบอย่าง ประกอบด้วยสามจุดต่อ โดยมีตัวไม่รู้ค่าอยู่ ณ ตำแหน่งที่จุดต่อ

ลักษณะการกระจายตัวไม่รู้ค่าบนเอลิเมนต์นี้สามารถเขียนให้อยู่ในรูปของฟังก์ชันการประมาณภายในและตัวไม่รู้ค่าที่จุดต่อดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

จากสมการที่ (2.1) สามารถเขียนให้อยู่ในรูปเมทริกซ์ได้ดังนี้

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

$$= \underset{(1x3)}{\lfloor N \rfloor} \underset{(3x1)}{\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}} \qquad (2.2)$$

โดย $\lfloor N \rfloor$ คือ เมทริกซ์ของฟังก์ชันการประมาณค่าภายในเอลิเมนต์

$\{\phi\}$ คือ เวกเตอร์เมทริกซ์ที่ประกอบด้วยตัวที่ไม่รู้ค่าที่จุดต่อของเอลิเมนต์นั้น

ซึ่งสัญลักษณ์ $\lfloor\ \rfloor$ หมายถึงเมทริกซ์ในแนวนอน และ $\{\ \}$ หมายถึงเมทริกซ์ในแนวตั้ง

<u>ขั้นตอนที่ 3</u> การสร้างสมการของเอลิเมนต์ ดังตัวอย่างเช่น สมการเอลิเมนต์สามเหลี่ยมแบบอย่างจะอยู่ในรูปแบบดังนี้

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

ซึ่งเขียนย่อได้เป็น

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

ขั้นตอนที่ 3 นี้ถือว่าเป็นหัวใจสำคัญของวิธีไฟไนต์เอลิเมนต์ การสร้างสมการของเอลิเมนต์ซึ่งอยู่ในรูปแบบของสมการที่ (2.3) สามารถทำได้โดย

ก. วิธีการโดยตรง (direct approach)

ข. วิธีการแปรผัน (variable approach)

ค. วิธีการถ่วงน้ำหนักเศษตกค้าง(method of weighted residuals)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 4 การนำสมการของแต่ละเอลิเมนต์ที่ได้มาประกอบกัน ก่อให้เกิดระบบสมการพร้อมกันขึ้น ในรูปแบบดังนี้

$$\sum(elementequation) \Rightarrow [k]_{sys}\{\phi\}_{sys} = \{F\}_{sys} \tag{2.5}$$

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

ขั้นตอนที่ 6 เมื่อคำนวณค่าต่าง ๆ ที่จุดต่อออกมาได้แล้วก็สามารถทำการหาค่าอื่นๆที่ต้องการทราบต่อไปได้ เช่น เมื่อรู้อุณหภูมิที่จุดต่าง ๆ ก็สามารถคำนวณหาปริมาณการถ่ายเทความร้อนได้ หรือเมื่อรู้ความเร็วของของไหลก็สามารถนำไปคำนวณหาปริมาณอัตราการไหลของของไหลทั้งหมดได้ เป็นต้น

จากขั้นตอนทั้ง 6 ขั้นตอนนี้จะเห็นได้ว่าวิธีการไฟไนต์เอลิเมนต์เป็นวิธีการที่มีระเบียบแบบแผนขั้นตอน โดยมีหัวใจสำคัญคือการสร้างสมการของเอลิเมนต์ จากโครงงานเรื่องนี้ได้กล่าวถึงการแก้ปัญหาในรูปแบบสองมิติซึ่งอยู่ในรูปของสมการปัวส์ซอง คือ

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

โดยต้องการที่จะทราบค่าของ u(x,y) จึงจะขอกล่าวถึงวิธีไฟไนต์เอลิเมนต์ที่ใช้ในการแก้ปัญหาสองมิติ

2.4 การสร้างสมการโดยวิธีการแปรผัน

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

ก. โดยการใช้สมการเชิงอนุพันธ์ (differential equations) ร่วมกับการใช้เงื่อนไขขอบเขตที่เหมาะสม

ข. โดยการใช้สูตรการแปรผัน (variational formulation)

หลักการสำคัญในการใช้สูตรการแปรผันก็คือ เราจำเป็นต้องทำการหาหรือสร้างฟังก์ชันซึ่งเมื่อเราทำการหาค่าต่ำสุด (minimization) ของฟังก์ชันนั้นแล้ว จะเป็นผลให้เกิดสมการเชิง

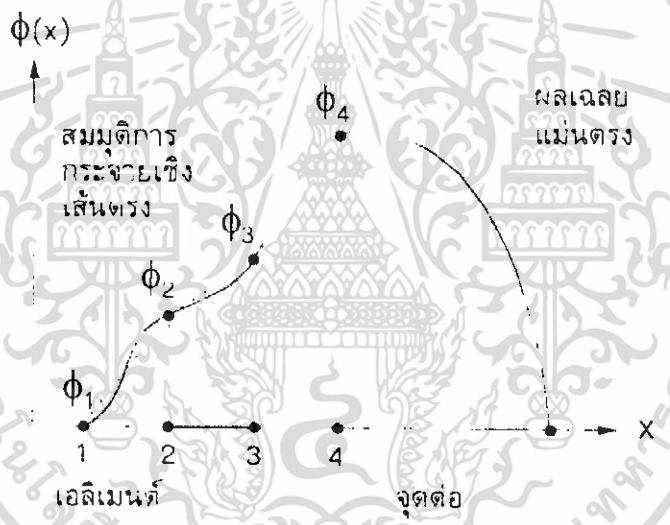
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อนุพันธ์และเงื่อนไขขอบเขตที่สอดคล้องกับปัญหาที่เราทำการแก้อยู่นั้น ดังตัวอย่างเช่น สมการอนุพันธ์ ต่อไปนี้

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

โดย $u = u(x)$ และเงื่อนไขขอบเขตที่เหมาะสมนั้นจะมีความสมมูลกัน

การสร้างสมการไฟไนต์เอลิเมนต์ในวิธีการแปรผัน จะเริ่มจากการสมมุติผลเฉลยโดยประมาณขึ้นมาซึ่งครอบคลุมทั้งขอบเขตปัญหาที่กำหนดมาให้ โดยจะมีการแบ่งขอบเขตออกเป็นเอลิเมนต์ย่อย ๆ ดังแสดงในรูปที่ 2.6 ซึ่งต่อกันที่จุดต่อต่าง ๆ โดยค่าที่จุดต่อ Φ_i, i = 1, 2, 3, ... นั้นไม่รู้ค่าและต้องการหาการกระจายของผลเฉลยระหว่างจุดต่ออาจสมมุติให้อยู่ในรูปแบบที่ง่าย เช่น ในลักษณะเชิงเส้นตรงดังแสดงในรูปที่ 2.6 เป็นต้น



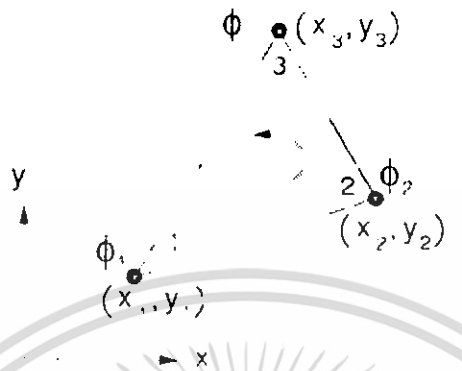
รูปที่ 2.6 การแบ่งขอบเขตออกเป็นไฟไนต์เอลิเมนต์ย่อย

2.4.2 ฟังก์ชันการประมาณภายในเอลิเมนต์สองมิติ

ปัญหาส่วนใหญ่ในสองมิติจะใช้เอลิเมนต์ที่มีลักษณะเป็นรูปสามเหลี่ยม ทั้งนี้ก็เพราะว่าลักษณะขอบเขตของปัญหาแบบสองมิติทั่ว ๆ ไปสามารถแบ่งออกเป็นเอลิเมนต์สามเหลี่ยมย่อย ๆ ได้โดยง่าย อีกทั้งสมการเอลิเมนต์รูปสามเหลี่ยมสามารถสร้างขึ้นได้โดยง่ายและนำไปใช้ในการประดิษฐ์โปรแกรมได้โดยสะดวกด้วย เอลิเมนต์สามเหลี่ยมแบบที่ง่ายที่สุดประกอบด้วยสามจุดต่อหมายเลข 1, 2, 3 ซึ่งวนอยู่ในทิศทวนเข็มนาฬิกา ดังแสดงในรูปที่ 2.7 โดยที่จุดต่อซึ่งอยู่ตำแหน่ง (x_i,y_i), i = 1, 2, 3 เป็นตำแหน่งของตัวไม่รู้ค่า ϕ_i การสร้างฟังก์ชันประมาณภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอลิเมนต์สามเหลี่ยมก็สามารถทำได้เช่นเดียวกับเอลิเมนต์ในหนึ่งมิติ ซึ่งเอลิเมนต์สามเหลี่ยมเราจะเริ่มโดยสมมุติลักษณะการกระจายของผลลัพธ์โดยประมาณแบบแผ่นเรียบบนเอลิเมนต์ นั่นคือ



รูปที่ 2.7 เอลิเมนต์สามเหลี่ยม

$$\phi\ (x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y \tag{2.7}$$

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

$$\text{จุดต่อที่ 1}: \quad \phi(x_1,y_1) = \phi_1 = \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1$$
$$\text{จุดต่อที่ 2}: \quad \phi(x_2,y_2) = \phi_2 = \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2$$
$$\text{จุดต่อที่ 3}: \quad \phi(x_3,y_3) = \phi_3 = \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3$$

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

$$\phi\,(x,y) = \underset{(1\times 3)}{\lfloor N_1(x,y) \quad N_2(x,y) \quad N_3(x,y) \rfloor} \underset{(3\times 1)}{\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}} = \underset{(1\times 3)}{\lfloor N \rfloor} \underset{(3\times 1)}{\{\phi\}} \tag{2.8}$$

โดย $\lfloor N \rfloor$ เรียกว่า เมทริกซ์ของการประมาณภายในเอลิเมนต์ (element interpolation matrix)

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในที่นี้

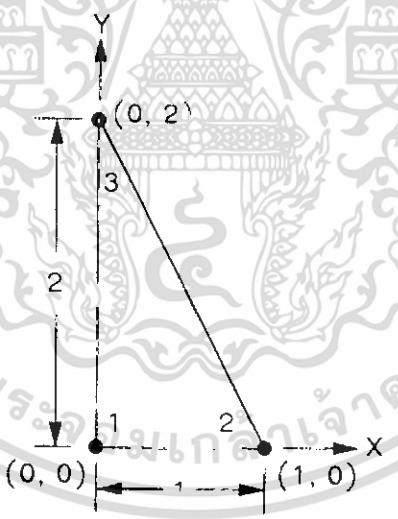
$$N_i(x,y) = \frac{1}{2A}(a_i + b_i x + c_i y) \quad i = 1, 2, 3 \tag{2.9}$$

โดย A = พื้นที่ของเอลิเมนต์สามเหลี่ยม

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

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

เพื่อให้เกิดความเข้าใจในลักษณะของฟังก์ชันการประมาณภายในเอลิเมนต์ เราจะพิจารณาเอลิเมนต์ดังแสดงในรูปที่ 2.8 ซึ่งจะมีขนาดและตำแหน่งของจุดต่อต่าง ๆ ดังนี้



รูปที่ 2.8 ตัวอย่างเอลิเมนต์สามเหลี่ยม

$$\begin{aligned} x_1 &= 0 & y_1 &= 0 \\ x_2 &= 1 & y_2 &= 0 \\ x_3 &= 0 & y_3 &= 2 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการ(2.10) พื้นที่ของเอลิเมนต์คือ

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

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

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

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

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

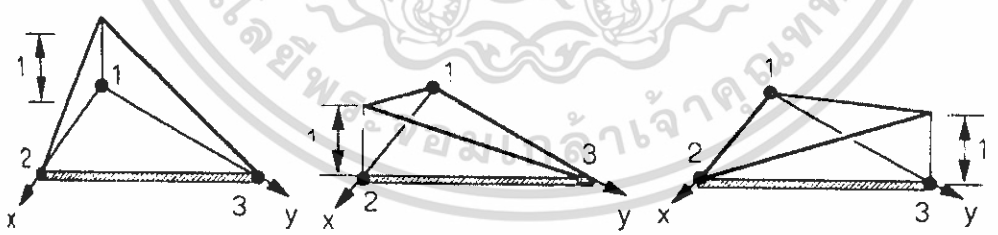
ดังนั้นฟังก์ชันการประมาณการภายในเอลิเมนต์นี้คือ

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

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

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

ซึ่งต่างมีการกระจายในลักษณะแผ่นเรียบดังแสดงในรูปที่ 2.9

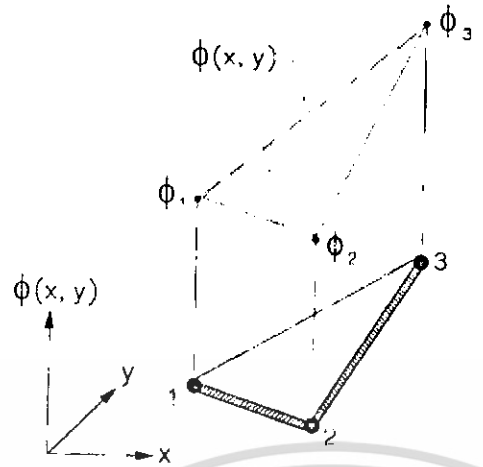


$N_1(x, y)$ $N_2(x, y)$ $N_3(x, y)$

รูปที่ 2.9 ลักษณะการกระจายของฟังก์ชันการประมาณภายในเอลิเมนต์

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ลักษณะการกระจายของผลเฉลยโดยประมาณภายในเอลิเมนต์

2.4.3 การสร้างสมการไฟไนต์เอลิเมนต์แบบสองมิติโดยวิธีการแปรผัน

เพื่อแสดงลำดับขั้นตอนการสร้างสมการไฟไนต์เอลิเมนต์แบบสองมิติโดยวิธีการแปรผัน เราจะยกตัวอย่างง่าย ๆ เช่น ปัญหาที่เป็นไปตามสมการอนุพันธ์ย่อยเชิงเส้นซึ่งอยู่ในรูปของสมการปัวส์ซอง ดังนี้

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เราเริ่มทำการหาค่าต่ำสุดของฟังก์ชันการแปรผัน J ที่เกี่ยวข้องกับ ϕ_1 จะได้

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

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

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

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

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

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

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

เมื่อแทนพจน์เหล่านี้ลงในสมการ (2.14) จะได้

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

จากนั้นจึงคูณออกมาและจัดเรียงพจน์ใหม่จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

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

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

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

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

แต่จากสมการ (2.9) ฟังก์ชันการประมาณภายในเอลิเมนต์สามเหลี่ยมที่ได้สร้างมาแล้วคือ

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

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

แทนกลับลงในสมการ (2.17) จะได้

73320

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

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

ทำนองเดียวกัน หากค่า f ในสมการ (2.18) คงที่ เราจะได้

$$F_i = f \int_A N_i dxdy$$

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

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

สมการ (2.20) นั้นสามารถอินทิเกรตได้ง่าย ๆ โดยใช้สูตรที่ว่า

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

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การสร้างสมการไฟไนต์เอลิเมนต์โดยวิธีการถ่วงน้ำหนักเศษตกค้าง

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

2.5.1 วิธีกาเลอร์คิน (Galerkin)

การถ่วงน้ำหนักเศษตกค้างโดยวิธีกาเลอร์คิน ทำโดยการคูณฟังก์ชันเศษตกค้าง R ด้วยฟังก์ชันน้ำหนัก (weighting function) W จากนั้นทำการอินทิเกรตตลอดทั้งโดเมนแล้วกำหนดผลที่ได้ให้เท่ากับศูนย์ นั่นคือ

$$\int R(t)\, W_i(t)\, dt = 0 \tag{2.22}$$

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

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

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

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

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

และจาก 2 สมการนี้ เราจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$C_1 = -0.9143 \quad \text{และ} \quad C_2 = 0.2857$$

ดังนั้น

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

อนึ่ง ด้วยเหตุผลหลาย ๆ ประการดังที่ได้อธิบายในตอนท้ายของหัวข้อย่อยนี้ เราจะใช้วิธีการ กาเลอร์คินนี้ในการสร้างสมการไฟไนต์เอลิเมนต์ในเอลิเมนต์ต่าง ๆ เช่น ลักษณะการกระจายของผลเฉลยโดยประมาณบนเอลิเมนต์หนึ่งมิติ ดังเช่นสมการดังนี้

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

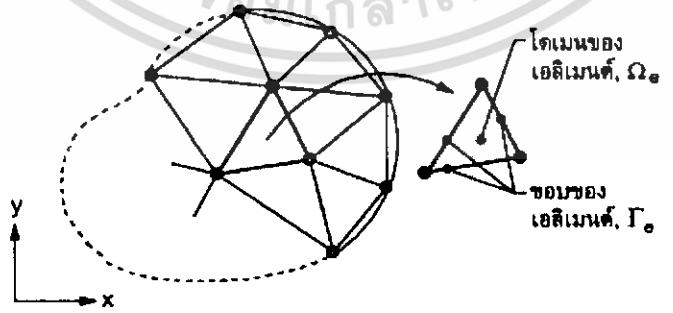
ในที่นี้ ϕ_1 และ ϕ_2 คือตัวไม่รู้ค่าเช่นเดียวกับ C_1 และ C_2 ในสมการ (2.23) ดังนั้น N_1 และ N_2 สามารถนำไปใช้เป็นฟังก์ชันน้ำหนัก W_1 และ W_2 ในการสร้างสมการไฟไนต์เอลิเมนต์ได้

2.5.2 วิธีการถ่วงน้ำหนักเศษตกค้างสำหรับไฟไนต์เอลิเมนต์

2.5.2.1 ลำดับขั้นตอนทั่วไป

การแก้ปัญหาไฟไนต์เอลิเมนต์โดยวิธีการถ่วงน้ำหนักเศษตกค้างประกอบด้วยลำดับขั้นตอน ที่สำคัญ 6 ขั้นตอนดังต่อไปนี้

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



รูปที่ 2.11 การแบ่งลักษณะรูปร่างของปัญหาออกเป็นเอลิเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นให้ทำการหาสมการเชิงอนุพันธ์ที่สอดคล้องกับปัญหาที่ต้องการแก้นั้น สมการเชิงอนุพันธ์โดยทั่วไปสามารถเขียนให้อยู่ในรูป

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

โดย L คือ ตัวดำเนินการเชิงอนุพันธ์ (differential operator)

$\bar{\phi}$ คือ ตัวแปรตามแม่นตรง

ขั้นที่ 2 สมมุติลักษณะการกระจายของผลเฉลยโดยประมาณบนเอลิเมนต์ให้อยู่ในรูป

$$\phi = \phi(x,y) \sum_{i=1}^{m} N_i \phi_i = \underset{(1\times m)}{\lfloor N \rfloor} \underset{(m\times 1)}{\{\phi\}} \tag{2.27}$$

โดย m คือ จำนวนจุดต่อของเอลิเมนต์นั้น

N_i คือ ฟังก์ชันการประมาณภายในเอลิเมนต์

ϕ_i คือ ตัวไม่รู้ค่าที่จุดต่อ

ขั้นที่ 3 สร้างสมการไฟไนต์เอลิเมนต์โดยวิธีการถ่วงน้ำหนักของเศษตกค้างหากเราแทนผลเฉลยโดยประมาณดังแสดงในสมการ (2.27) ลงในสมการเชิงอนุพันธ์ในสมการ (2.26) เราจะพบว่า

$$L(\phi) \text{ จะ } \neq 0 \quad \text{แต่จะ} \quad = \quad R$$

ซึ่ง R คือเศษตกค้าง (Residual) นั่นหมายถึง

$$R = L(\phi) = L(\lfloor N \rfloor \{\phi\}) = L\left(\sum_{i=1}^{m} N_i \phi_i\right) \tag{2.28}$$

และจากวิธีกาเลอร์คิน (Galerkin) ดังที่ได้อธิบายในหัวข้อย่อย 2.5.1 ซึ่งมีขั้นตอนโดยเริ่มจากการคูณเศษตกค้าง R ด้วยฟังก์ชันน้ำหนัก (weighting function) W จากนั้นจึงอินทิเกรตตลอดทั้งโดเมนของเอลิเมนต์นั้น แล้วกำหนดผลที่ได้ให้เท่ากับศูนย์ นั่นคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\int_{\Omega} W_i R \, d\Omega = 0 \qquad i = 1, 2, \ldots, m \tag{2.29}$$

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

ขั้นที่ 4 อินทิเกรตทีละส่วน (integrate by parts) ซึ่งหากเราแทนสมการ (2.28) ลงในสมการ (2.29) แล้วอินทิเกรตทีละส่วนจะได้

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

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

ขั้นที่ 5 แทนพจน์ที่เกี่ยวข้องกับขอบเขตของเอลิเมนต์, $\Gamma^{(e)}$, ด้วยภาวะขอบเขตอื่นๆที่เกี่ยวข้อง ซึ่งจะก่อให้เกิดสมการของเอลิเมนต์ที่สมบูรณ์สำหรับปัญหานั้น

ขั้นที 6 จากนั้นจึงเขียนสมการของเอลิเมนต์ ซึ่งมีทั้งหมด m สมการให้อยู่ในรูปของเมทริกซ์ นั่นคือ

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

โดย [K] คือ เอลิเมนต์เมทริกซ์ของความแข็งเกร็ง (element stiffness matrix)

{ϕ} คือ เวกเตอร์ซึ่งประกอบด้วยตัวไม่รู้ค่าที่ จุดต่อต่าง ๆ ของเอลิเมนต์

{F} คือ โหลดเวกเตอร์ของเอลิเมนต์นั้น

เมื่อได้สมการไฟไนต์เอลิเมนต์ดังเช่นแสดงในสมการ (2.30) แล้ว ลำดับขั้นตอนต่อไปก็ทำเช่นเดียวกันกับที่ได้อธิบายข้างต้น กล่าวคือ รวมสมการของเอลิเมนต์ย่อยเข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อให้เกิดสมการระบบรวม จากนั้นก็กำหนดกฎเกณฑ์ขอบเขต แล้วจึงแก้สมการระบบรวมเพื่อหาค่าผลลัพธ์ที่จุดต่อต่าง ๆ

2.6 ฟังก์ชันการประมาณภายในเอลิเมนต์และการอินทิเกรตเอลิเมนต์เมทริกซ์เชิงตัวเลข

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

2.6.1 ฟังก์ชันการประมาณภายในเอลิเมนต์สองมิติ

2.6.1.1 เอลิเมนต์รูปสามเหลี่ยม

เอลิเมนต์รูปสามเหลี่ยมเป็นเอลิเมนต์ที่นิยมใช้กันมากเนื่องจากแบบจำลองไฟไนต์เอลิเมนต์ที่ประกอบด้วยเอลิเมนต์สามเหลี่ยมจำนวนมากสามารถสร้างขึ้นมาได้โดยง่ายและโดยอัตโนมัติ เราได้ศึกษาเอลิเมนต์รูปสามเหลี่ยมที่ประกอบด้วย 3 จุดต่อดังแสดงในรูปที่ 2.7 และได้ทำการสร้างฟังก์ชันการประมาณภายในเอลิเมนต์รูปสามเหลี่ยมนั้นโดยเริ่มจากการสมมุติลักษณะการกระจายของผลเฉลยโดยประมาณให้อยู่ในลักษณะการกระจายแบบแผ่นเรียบ (flat plane) กล่าวคือ

$$\phi(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y \tag{2.31}$$

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

$$\phi = N_1\phi_1 + N_2\phi_2 + N_3\phi_3 \tag{2.32}$$

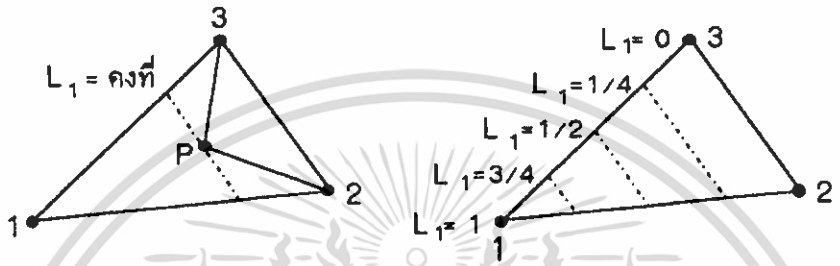
โดย N_i, i = 1, 2, 3 คือฟังก์ชันการประมาณภายในเอลิเมนต์ ซึ่งอยู่ในรูปแบบของพิกัด x, y ดังแสดงในสมการ (2.9) และต่างมีลักษณะการกระจายในรูปแบบของแผ่นเรียบดังแสดงในรูปที่ 2.9 เช่นเดียวกันกับเอลิเมนต์ในหนึ่งมิติ การกระจายของผลเฉลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยประมาณของ เอลิเมนต์รูปสามเหลี่ยมสามารถเขียนให้อยู่ในรูปแบบของพิกัดธรรมชาติ ดังนี้

$$\phi = L_1\phi_1 + L_2\phi_2 + L_3\phi_3 \tag{2.33}$$

โดย L_i , i = 1, 2, 3 สามารถอธิบายได้โดยการพิจารณารูปที่ 2.12



รูปที่ 2.12 ฟังก์ชันการประมาณภายในเอลิเมนต์รูปสามเหลี่ยม

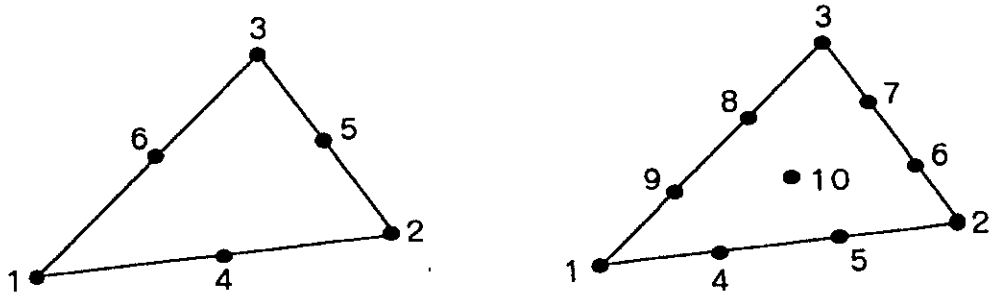
เช่น L_1 = (พื้นที่สามเหลี่ยม P23) / (พื้นที่สามเหลี่ยม 123) (2.34)

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

$$L_i = N_i \qquad i = 1, 2, 3 \tag{2.35}$$

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) เอลิเมนต์อันดับสอง (ข) เอลิเมนต์อันดับสาม

รูปที่ 2.13 เอลิเมนต์สามเหลี่ยมอันดับสูงที่ประกอบด้วยหลายจุดต่อ

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

<u>เอลิเมนต์อันดับสอง :</u>

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

จุดต่อที่ด้าน $N_4 = 4L_1L_2$ เป็นต้น (2.37)

<u>เอลิเมนต์อันดับสาม :</u>

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

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

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

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

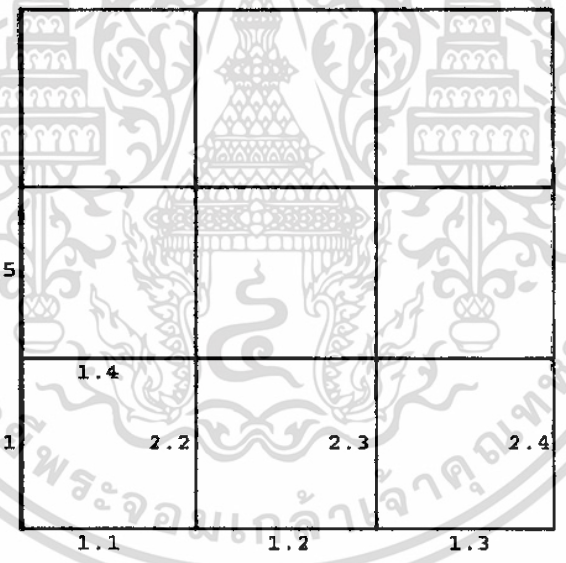
ในหัวข้อนี้เราจะมาทำการศึกษาลักษณะและขั้นตอนการสร้างฟังก์ชันการประมาณภายในเอลิเมนต์ของเอลิเมนต์ในรูปแบบสองมิติ จากนั้นเราจะได้ศึกษาวิธีการคำนวณเอลิเมนต์เมทริกซ์ต่าง ๆ โดยการใช้ฟังก์ชันการประมาณภายในที่ได้สร้างขึ้นมานี้ ซึ่งในบางกรณีเราจำเป็นต้องใช้การอินทิเกรตเชิงตัวเลข (numerical integration) เพื่อการคำนวณหาเอลิเมนต์เมทริกซ์เหล่านี้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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การแสดงผลทางกราฟ

สมมุติว่าได้รับค่าจำนวนจุดต่อในแนวแกน x เท่ากับ 4 และค่าจำนวนจุดต่อในแนวแกน y เท่ากับ 4 ฉะนั้นจำนวนจุดต่อทั้งหมดจะเท่ากับ 16 จุดต่อ

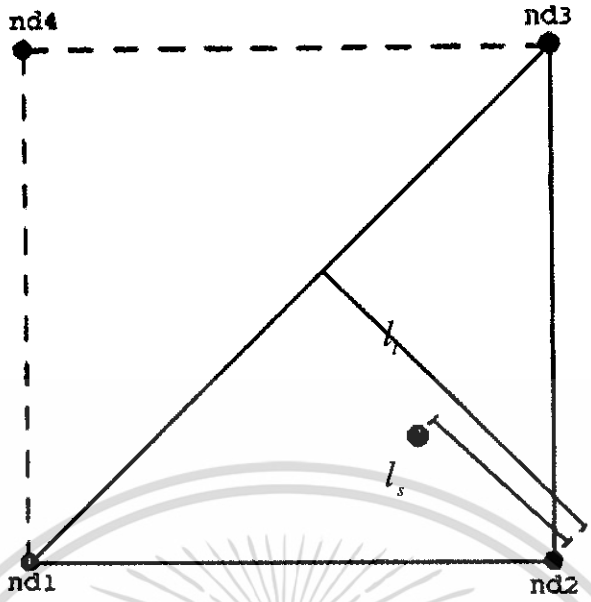
1. เริ่มต้นจะทำการสร้างกราฟ โดยจะทำการวนลูปในแนวแกน x ก่อน โดยเริ่มจุดแรกจากจุดซ้ายล่างแล้วลากเส้นเชื่อมกับจุดต่อถัดมา ค่อย ๆ วนไปจนครบตามจำนวนจุดต่อในแนวแกน x ที่ได้รับเข้ามา พอครบแล้วก็ขึ้นแถวถัดไปวนเช่นเดิมจนครบทั้ง 16 จุดต่อ ดังรูปที่ 2.14 โดยหมายเลข 1.1 , 1.2 , 1.3 , 1.4 ,... แสดงถึงลำดับในการลากเส้นเชื่อมระหว่างจุดในแนวแกน x
2. จากนั้นจะทำการวนลูปตามแนวแกน y โดยลากเส้นเชื่อมระหว่างจุดต่อที่ตรงกันในแถวที่ 1 และ ในแถวที่ 2 ตามแนวแกนตั้งจากซ้ายไปขวา เมื่อลากเส้นเชื่อมครบ 4 จุด ก็ขึ้นแถวถัดไป วนทำเช่นเดิมเป็นจำนวนเท่ากับ (จำนวนจุดต่อในแนวแกน y) - 1 รอบ ดังรูปที่ 2.14 โดยหมายเลข 2.1 , 2.2 , 2.3 , 2.4 , 2.5 , ... แสดงถึงลำดับในการลากเส้นเชื่อมระหว่างจุดในแนวแกน y



รูปที่ 2.14 แสดงวิธีการสร้างกราฟในแนวแกน x และแกน y

3. เมื่อได้กราฟในแนวแกน x และแกน y แล้ว โปรแกรมสามารถคำนวณหาผลลัพธ์ตามพิกัด (x,y) ที่ผู้ใช้กำหนดให้ได้ โดยใช้หลักการคำนวณดังต่อไปนี้
   3.1 หาว่าอยู่ในเอลิเมนต์สี่เหลี่ยมรูปไหน
   3.2 จากนั้นจึงหาต่ออีกว่า อยู่ในเอลิเมนต์สามเหลี่ยมรูปไหน (เอลิเมนต์สี่เหลี่ยม 1 รูป ประกอบด้วยเอลิเมนต์สามเหลี่ยม 2 รูป)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงเอลิเมนต์สามเหลี่ยมซึ่งมีพิกัด (x,y) ที่ต้องการทราบค่าอยู่

3.3 ต่อมาจะทำการหาค่าถ่วงน้ำหนักของแต่ละจุดต่อโดยใช้สูตร

$$w = \frac{l_s}{l_l}$$

โดย w คือ ค่าถ่วงน้ำหนัก

l_s คือ ระยะทางที่สั้นที่สุดจากพิกัด (x,y) ถึง จุดต่อใด ๆ ในเอลิเมนต์ที่พิกัดนั้นอยู่

l_l คือ ระยะทางที่ยาวที่สุดจากพิกัด (x,y) ถึง จุดต่อใด ๆ ในเอลิเมนต์ที่พิกัดนั้นอยู่

4. หาผลลัพธ์โดยใช้สูตร

$$z = \frac{nd1(w_1) + nd2(w_2) + nd3(w_3)}{w_1 + w_2 + w_3}$$

โดย z คือ ค่าผลลัพธ์

nd1, nd2, nd3 คือ หมายเลขจุดต่อที่ 1,2,3 ตามลำดับ

w_1, w_2, w_3 คือ ค่าถ่วงน้ำหนักที่คำนวณได้จากข้อ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขั้นตอนการดำเนินงานและหลักการของโปรแกรม

3.1 ขั้นตอนในการดำเนินงาน

1. วางหัวข้อและขอบเขตของโครงงาน โดยจะทำการศึกษาหลักการทางคณิตศาสตร์ของสมการปัวส์ซองในรูปแบบ

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

หรือ $$\nabla^2 u = f(x, y)$$

โดยใช้ระเบียบวิธีไฟไนต์เอลิเมนต์ แล้วจะทำการแสดงผลออกมาในรูปแบบกราฟสามมิติ

2. ศึกษาสมการปัวส์ซอง วิธีผลต่างสืบเนื่องและไฟไนต์เอลิเมนต์ โดยได้ทำการศึกษาวิธีผลต่างสืบเนื่อง ก่อนซึ่งจะทำให้เข้าใจ ไฟไนต์เอลิเมนต์ ได้ง่ายขึ้น
3. ออกแบบและพัฒนาการทำงานของโปรแกรม โดยใช้โปรแกรม Microsoft Visual C++ ในการพัฒนาโปรแกรมขึ้นมา
4. ทำการทดลองและปรับปรุงโปรแกรม
5. นำเสนอ
6. ประเมินผลการทำงาน

3.2 ลักษณะการทำงานของโปรแกรม

ในการคำนวณเอลิเมนต์เราต้องเก็บค่าพิกัดของ x และ y โดยใช้ตัวแปรอาเรย์ชื่อ xyPos เก็บค่า x และ y ของแต่ละจุดต่อ และมีตัวแปรอาเรย์ชื่อ nodes เก็บค่าหมายเลขจุดต่อของแต่ละเอลิเมนต์ และสามารถกำหนดเงื่อนไขขอบเขตได้โดยใช้ตัวแปรอาเรย์ชื่อ bcdof (boundary condition degree of freedom) ไว้เก็บหมายเลขจุดต่อที่เป็นขอบของปัญหา และใช้ตัวแปรอาเรย์ชื่อ side_value เพื่อเก็บค่าเงื่อนไขขอบเขตของปัญหาแต่ละด้าน แล้วนำไปกำหนดให้แก่ตัวแปรอาเรย์ชื่อ bcval (boundary condition value) เพื่อกำหนดค่าเงื่อนไขขอบเขตแก่จุดต่อที่เป็นขอบของปัญหาในแต่ละจุดต่อ หลังจากที่เราทราบว่าตัวแปรหลัก ๆ แต่ละตัวใช้เพื่อทำอะไรแล้ว ต่อมาเราต้องการหาเมทริกซ์ของแต่ละเอลิเมนต์ โดยกระบวนการไฟไนต์เอลิเมนต์เพื่อจะได้เมทริกซ์รวม ได้จากกระบวนการดังต่อไปนี้

```
nd1 = nodes[iel][1]
nd2 = nodes[iel][2]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
nd3 = nodes[iel][3]
```

nd1 , nd2 , nd3 จะเก็บหมายเลขจุดต่อของแต่ละเอลิเมนต์ซึ่งจะมีอาเรย์มิติที่ 1 เก็บหมายเลขเอลิเมนต์ และมิติที่ 2 เก็บตำแหน่งของจุดต่อในเอลิเมนต์ซึ่งในที่นี้เราได้ยกตัวอย่างของเอลิเมนต์ของสามเหลี่ยมจะมี 3 จุดที่ถูกพิจารณา

```
2 = nodes[1][1]
3 = nodes[1][2]
8 = nodes[1][3]
```

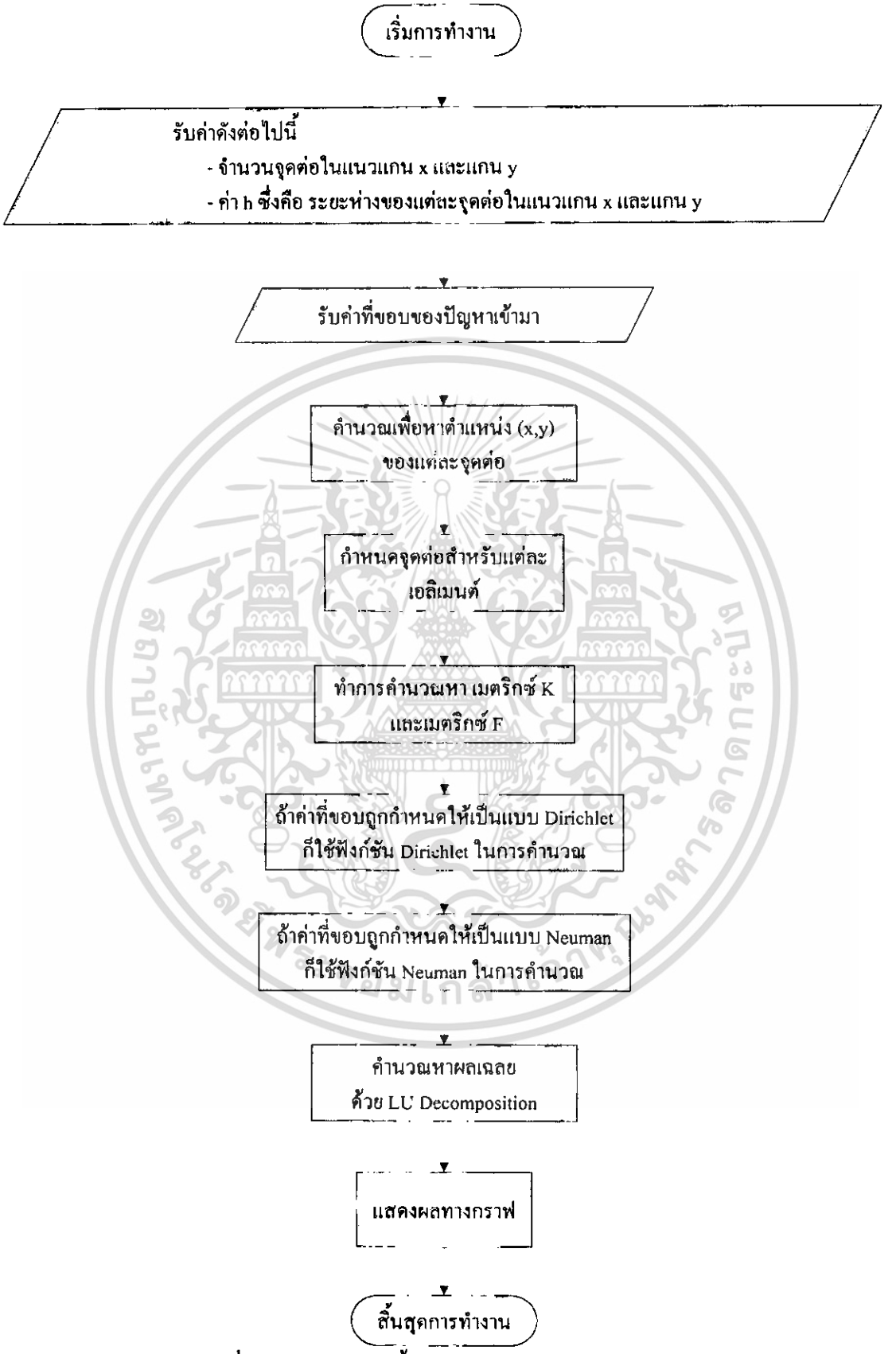
หมายความว่าเอลิเมนต์ที่ 1 ตำแหน่งจุดต่อที่ 1 เก็บหมายเลขจุดต่อ คือ 2 เอลิเมนต์ที่ 1 ตำแหน่งจุดต่อที่ 2 เก็บหมายเลขจุดต่อ คือ 3 เอลิเมนต์ที่ 1 ตำแหน่งจุดต่อที่ 3 เก็บหมายเลขจุดต่อที่ 8

```
x1 = xyPos[nd1][1]
y1 = xyPos[nd1][2]

x2 = xyPos[nd2][1]
y2 = xyPos[nd2][2]

x3 = xyPos[nd3][1]
y3 = xyPos[nd3][2]
```

ในกระบวนการนี้เราต้องการทราบค่า x และ y ของแต่ละจุดต่อ โดยที่อาเรย์ชื่อว่า xyPos โดยที่มิติที่ 1 เก็บหมายเลขจุดต่อของเอลิเมนต์ และมิติที่ 2 เก็บค่า 1 และ 2 เพื่อบ่งบอกว่า 1 คือค่า x และ 2 คือค่า y หลังจากได้ค่า x และ y แล้ว ก็นำมาเข้าฟังก์ชันคำนวณหาเมทริกซ์ k เอลิเมนต์ ที่ชื่อว่า felp2dt3 จะได้เมทริกซ์ k เอลิเมนต์ออกมา จากนั้นนำมาแอสเซมบลีเราก็จะได้เมทริกซ์ k รวมของระบบออกมาซึ่งเป็นค่าที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

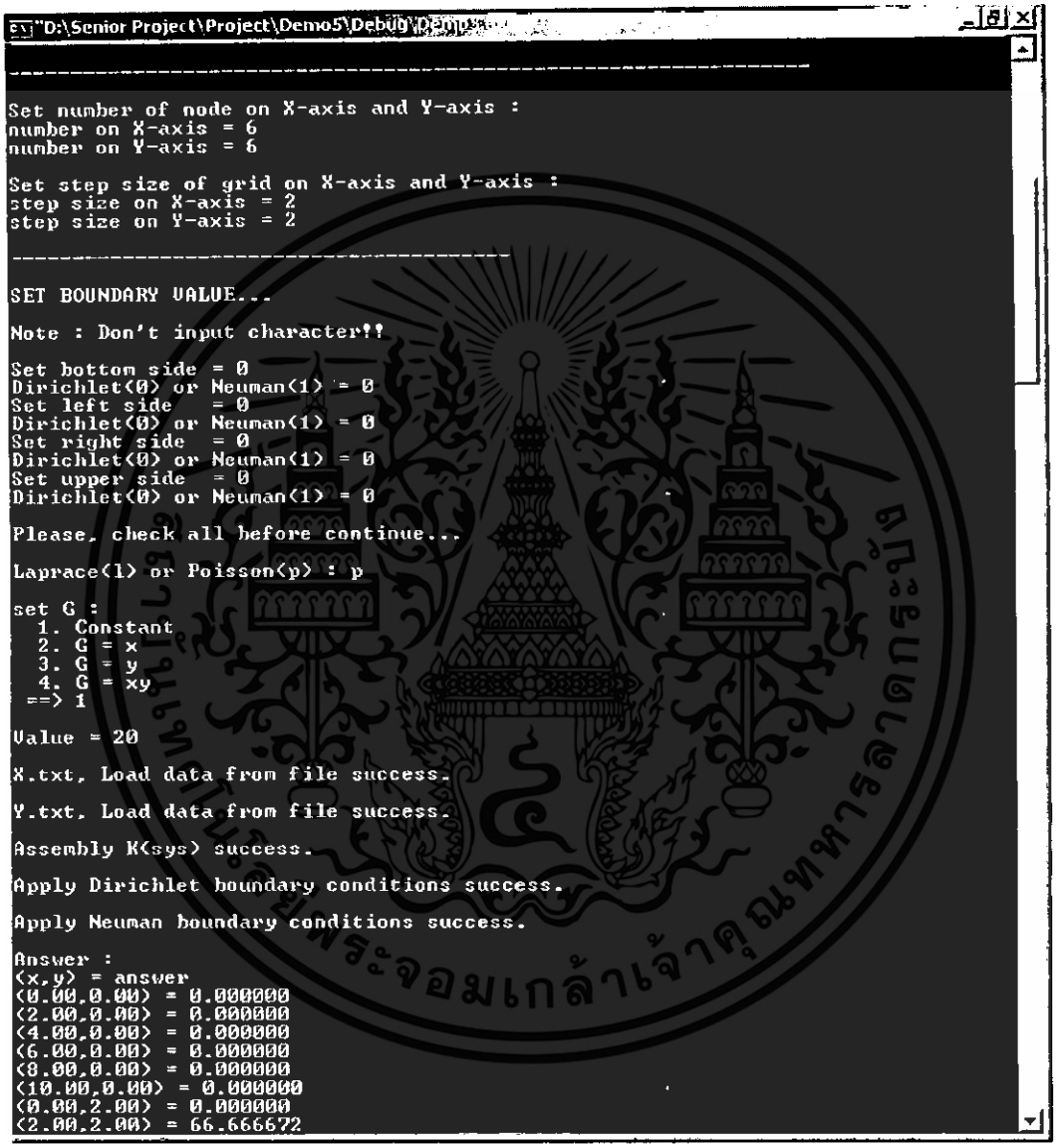
3.3 ขั้นตอนการทำงานของโปรแกรม



รูปที่ 3.1 แสดงลำดับขั้นตอนการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจะทำการรับข้อมูลตามที่โจทย์กำหนดเข้ามาดังรูป ซึ่งเราได้ทำการศึกษาปัญหาที่มีขอบเขตเป็นสี่เหลี่ยมมุมฉาก คือ จำนวนจุดต่อในแนวแกน x และแกน y , ค่า h ซึ่งคือ ระยะห่างของแต่ละจุดต่อในแนวแกน x และแกน y (grid size) และค่าที่ขอบของปัญหา (boundary) เพื่อที่จะนำมาประมาณค่าภายในขอบเขตของปัญหา ณ จุดที่เราสนใจ



รูปที่ 3.2 ภาพตัวอย่างแสดงการใส่ค่าต่าง ๆ โดยใช้สมการปัวส์ซองและฟังก์ชันทางด้านขวาเป็นค่าคงที่

หลังจากที่เราได้ใส่ค่าต่าง ๆ ครบตามที่กำหนดแล้วเราก็สามารถเลือกรูปแบบสมการที่ใช้ในการคำนวณซึ่งมี 2 วิธี คือ วิธีลาปลาซ และวิธีปัวส์ซอง ถ้าหากเราเลือกสมการปัวส์ซองเราก็สามารถเลือกฟังก์ชันทางด้านขวาของสมการได้โดยมีทั้งหมดสี่รูปแบบ คือ

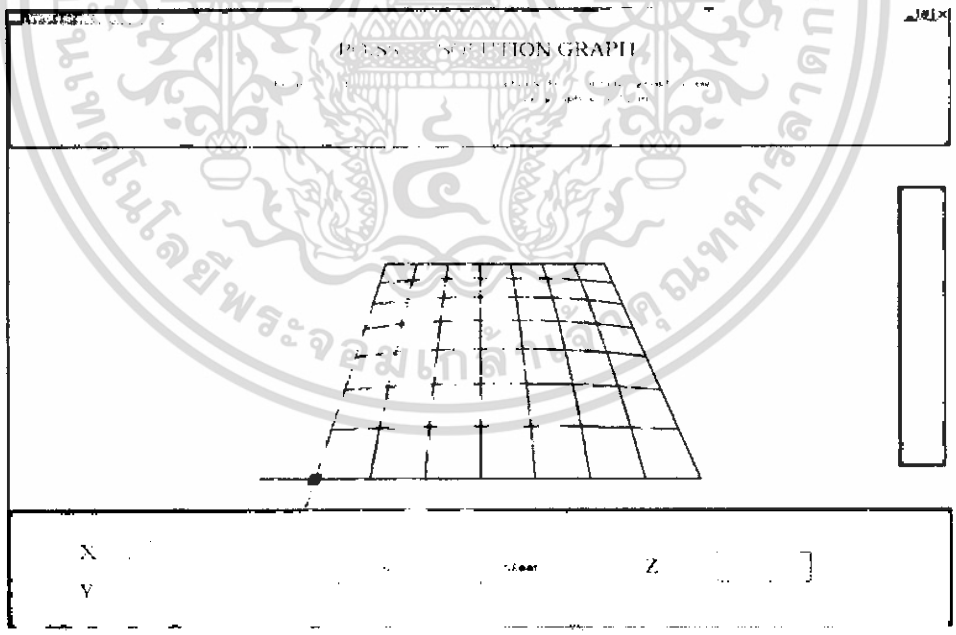
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เป็นค่าคงที่
2. ฟังก์ชัน x
3. ฟังก์ชัน y
4. ฟังก์ชัน xy

โดยในขบวนการคำนวณนี้ เราจะใช้วิธีการถ่วงน้ำหนักเศษตกค้าง (method of weighted residuals) โดยใช้วิธีกาเลอร์คิน (Galerkin) ที่เลือกใช้วิธีกาเลอร์คินนี้ก็เพราะว่า หลังจากได้เปรียบเทียบถึงข้อดีข้อเสียของวิธีการย่อยต่าง ๆ ของวิธีการถ่วงน้ำหนักเศษตกค้าง ซึ่งได้แก่ การจัดตำแหน่งจุด (point collocation) , การจัดโดเมนย่อย (subdomain collocation) , วิธีกาเลอร์คิน (Galerkin) และวิธีกำลังสองน้อยสุด (least squares) แล้วนั้น พบว่าวิธีกาเลอร์คินนั้นให้ความสะดวกหลายอย่างในการที่จะนำไปสร้างสมการไฟไนต์เอลิเมนต์

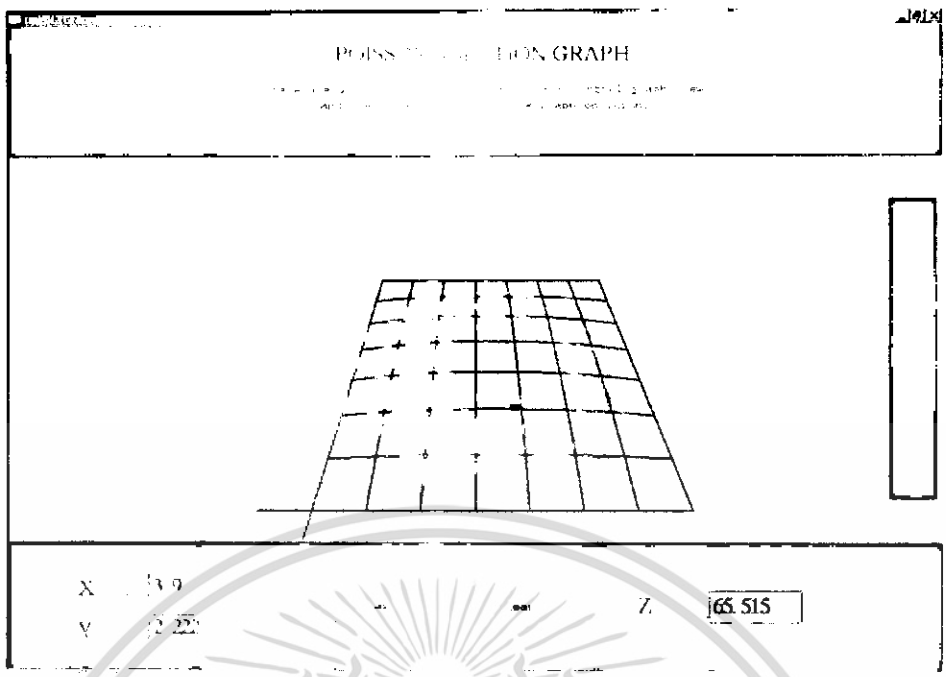
เมื่อเราได้กำหนดจำนวนจุดต่อในแนวแกน x และจำนวนจุดต่อในแนวแกน y ระยะห่างของแต่ละจุดต่อในแนวแกน x และระยะห่างของแต่ละจุดต่อในแนวแกน y แล้ว จะได้ผลลัพธ์ออกมาในรูปของไฟล์อักษร 3 ไฟล์ ซึ่งได้แก่ ไฟล์ของค่าในแนวแกน x (X.txt) , ไฟล์ของค่าในแนวแกน y (Y.txt) และไฟล์ของค่าในแนวแกน z (Z.txt) ซึ่งเป็นไฟล์ผลลัพธ์ที่ต้องการ

เมื่อเราใส่ค่าต่าง ๆ ครบถ้วนแล้วก็จะปรากฏภาพดังรูป



รูปที่ 3.3 ภาพที่แสดงให้เราใส่พิกัดจุดของ x และ y เพื่อหาค่าผลเฉลย ณ จุดพิกัดนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ภาพแสดงผลเฉลยของสมการปัวส์ซองเมื่อฟังก์ชันทางด้านขวาเป็นค่าคงที่

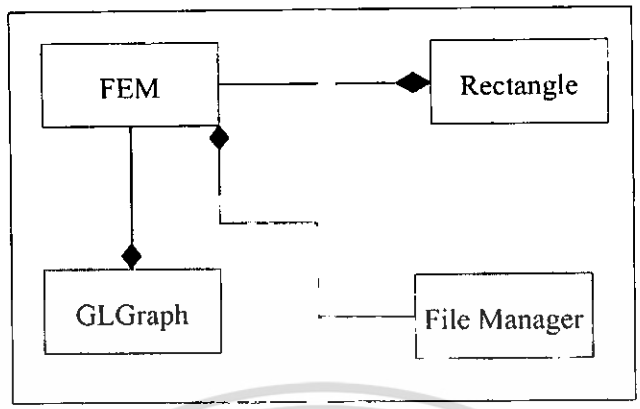
3.4 ส่วนประกอบของโปรแกรม

โปรแกรมประกอบไปด้วย 4 ส่วนหลักคือ

1. คลาส FEM เป็นคลาสที่ประกอบไปด้วยฟังก์ชันที่ใช้ในการหาผลเฉลยของไฟไนต์เอลิเมนต์ ซึ่งผลลัพธ์ที่ได้จะเก็บไว้ในแฟ้มข้อมูล Z.txt
2. คลาส File Manager เป็นคลาสที่ใช้ในการติดต่อกับแฟ้มข้อมูล ทั้งในลักษณะการอ่าน และการเขียนลงแฟ้มข้อมูล
3. คลาส GLGraph เป็นคลาสที่ใช้ในการจัดการสร้างกราฟและแสดงผลกราฟ รวมทั้งสามารถรับข้อมูลตำแหน่งของ x และ y จากผู้ใช้ แล้วส่งให้คลาส FEM ทำการประมวลผล
4. คลาส Rectangle เป็นคลาสที่ใช้รับข้อมูลจากผู้ใช้เพื่อใช้ในการกำหนดขอบเขตปัญหาเป็นสี่เหลี่ยมมุมฉากจากนั้นก็จะไปทำการเรียกคลาส FEM เพื่อจะทำการคำนวณหาผลเฉลยของไฟไนต์เอลิเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งส่วนประกอบทั้ง 4 สามารถแสดงได้ดังรูปด้านล่างนี้



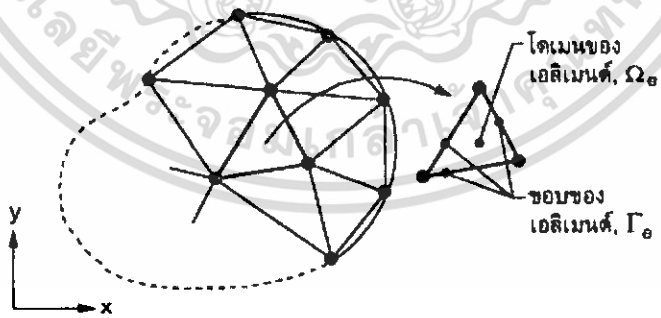
รูปที่ 3.5 แสดงส่วนประกอบของโปรแกรม

3.5 ทฤษฎีและหลักการที่ใช้ในการพัฒนาโปรแกรม

3.5.1 ระเบียบวิธีไฟไนต์เอลิเมนต์

เป็นทฤษฎีหลักที่ใช้ในการแก้ปัญหาสมการปัวส์ซอง ซึ่งได้กล่าวไว้โดยละเอียดแล้วในบทที่ 2 สำหรับในหัวข้อนี้จะแสดงให้เห็นขั้นตอนการคำนวณสำหรับเตรียมสมการเมทริกซ์ที่จะใช้ในการพัฒนาโปรแกรม ว่าได้มาได้อย่างไรตามระเบียบวิธีไฟไนต์เอลิเมนต์ 6 ขั้นตอน

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



รูปที่ 3.6 แสดงการแบ่งพื้นที่ปัญหาด้วยเอลิเมนต์สามเหลี่ยม

ซึ่งเมื่อเราทราบลักษณะของเอลิเมนต์ที่เราต้องการแล้วก็จะเข้าสู่ขั้นตอนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 2 สมมุติลักษณะการกระจายของผลเฉลยโดยประมาณบนเอลิเมนต์ให้อยู่ในรูป

$$\phi = \phi(x,y) \sum_{i=1}^{m} N_i \phi_i = \underset{(1\times m)}{\lfloor N \rfloor} \underset{(m\times 1)}{\{\phi\}}$$

โดย m คือ จำนวนจุดต่อของเอลิเมนต์นั้น

N_i คือ ฟังก์ชันการประมาณภายในเอลิเมนต์

ϕ_i คือ ตัวไม่รู้ค่าที่จุดต่อ

ฟังก์ชันการประมาณภายในของเอลิเมนต์สามเหลี่ยม คือ

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

ฟังก์ชันการประมาณภายใน คือ

$$\phi = \lfloor N_i \rfloor \{\phi_i\}$$

$$\frac{\partial N_i}{\partial x} = \frac{b_i}{2A} \quad , \quad \frac{\partial N_i}{\partial y} = \frac{c_i}{2A}$$

$$\frac{\partial \phi}{\partial x} = \frac{1}{2A} \lfloor b_i \rfloor \{\phi_i\} \quad , \quad \frac{\partial \phi}{\partial y} = \frac{1}{2A} \lfloor c_i \rfloor \{\phi_i\}$$

ขั้นที่ 3 สร้างสมการไฟไนต์เอลิเมนต์โดยวิธีการถ่วงน้ำหนักของเศษตกค้าง และจากวิธีกาเลอร์คิน (Galerkin) ซึ่งมีขั้นตอนโดยเริ่มจากการคูณเศษตกค้าง R ด้วยฟังก์ชันน้ำหนัก (weighting function) W จากนั้นจึงอินทิเกรตตลอดทั้งโดเมนของเอลิเมนต์นั้น แล้วกำหนดผลที่ได้ให้เท่ากับศูนย์ นั่นคือ

$$\int_{\Omega} W_i R \, d\Omega = 0 \qquad i = 1, 2, \ldots, m$$

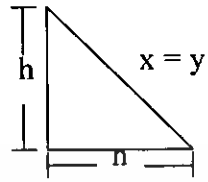
จะได้ $$\iint W_i \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) dx dy = 0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\int_0^h\int_y^h W_i\left(\frac{\partial^2 V}{\partial x^2}\right)dxdy+\int_0^h\int_0^x W_i\left(\frac{\partial^2 \phi}{\partial y^2}\right)dydx = 0 \qquad (3.1)$$

เพราะว่าเลือกเอลิเมนต์สามเหลี่ยม ซึ่งมีฟังก์ชันการประมาณภายในในรูป

$$N_{i(x,y)}=\frac{1}{2A}\left(a_i+b_i x+c_i y\right)$$



ขั้นที่ 4 อินทิเกรตทีละส่วน (integrate by parts)

แก้อินทิเกรตพจน์แรกของสมการที่ (3.1)

$$\int_0^h\int_y^h W_i\left(\frac{\partial^2 \phi}{\partial x^2}\right)dxdy$$

$$u = v_i \qquad dv=\frac{\partial^2 \phi}{\partial x^2}dx$$

$$du=\frac{dw_i}{dx}dx \qquad v=\frac{\partial \phi}{\partial x}$$

$$\int_0^h\int_y^h W_i\left(\frac{\partial^2 \phi}{\partial x^2}\right)dxdy=\int_0^h\left[\left[W_i\frac{\partial \phi}{\partial x}\right]_y^h-\int_y^h\frac{\partial \phi}{\partial x}\frac{dW_i}{dx}dx\right]dy \qquad (3.2)$$

จากสมการที่ (3.2) จะได้

$$=\int_0^h\left[W_i\frac{\partial \phi}{\partial x}\right]_y^h dy-\int_0^h\int_y^h\frac{\partial \phi}{\partial x}\frac{dW_i}{dx}dxdy \qquad (3.3)$$

แก้อินทิเกรตพจน์ที่ 2 ของสมการที่ (3.1)

$$\int_0^h\int_0^x W_i\left(\frac{\partial^2 \phi}{\partial y^2}\right)dydx$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$u = w_i \qquad dv = \frac{\partial^2 \phi}{\partial y^2}$$

$$du = \frac{dw_i}{dy} dy \qquad v = \frac{\partial \phi}{\partial y}$$

$$\int_0^h \int_0^x W_i \left(\frac{\partial^2 \phi}{\partial y^2} \right) dy dx = \int_0^h \left[\left[W_i \frac{\partial \phi}{\partial y} \right]_0^x - \int_0^x \frac{\partial \phi}{\partial y} \frac{dW_i}{dy} dy \right] dx$$

$$= \int_0^h \left[W_i \frac{\partial \phi}{\partial y} \right]_0^x - \int_0^h \int_0^x \frac{\partial \phi}{\partial y} \frac{dW_i}{dy} dy dx \qquad (3.4)$$

แทนสมการที่ (3.3) และ (3.4) ลงในสมการที่ (3.1) จะได้

$$\int_0^h \int_y^h \frac{dW_i}{dx} \frac{\partial \phi}{\partial x} dx dy + \int_0^h \int_0^x \frac{dW_i}{dy} \frac{\partial \phi}{\partial y} = \int_0^h \left[W_i \frac{\partial \phi}{\partial x} \right]_y^h dy + \int_0^h \left[W_i \frac{\partial \phi}{\partial y} \right]_0^x dx \qquad (3.5)$$

แทน i=1, 2, 3 และแทน $W_i = N_i$ จากบับโนฟ-กาเลอร์คิน (Bubnov-Galerkin) จะได้

$$\int_0^h \int_y^h \left\{ \frac{dN_i}{dx} \right\} \left[\frac{dN_i}{dy} \right] dx dy \{\phi_i\} + \int_0^h \int_0^x \left\{ \frac{dN_i}{dy} \right\} \left[\frac{dN_i}{dy} \right] dy dx \{\phi_i\} = \int_0^h \left[\{N_i\} \frac{\partial \phi}{\partial x} \right]_y^h dy + \int_0^h \left[\{N_i\} \frac{\partial \phi}{\partial y} \right]_0^x dx$$

ให้สมการข้างต้นเป็นสมการที่ (3.6) จากฝั่งซ้ายของสมการที่ (3.6) จะได้

$$= \int_0^h \int_y^h dx dy \left\{ \frac{dN_i}{dx} \right\} \left[\frac{dN_i}{dx} \right] \{\phi_i\} + \int_0^h \int_0^x dy dx \left\{ \frac{dN_i}{dy} \right\} \left[\frac{dN_i}{dy} \right] \{\phi_i\}$$

$$= \left[hy - \frac{y^2}{2} \right]_0^h \left\{ \frac{dN_i}{dx} \right\} \left[\frac{dN_i}{dx} \right] \{\phi_i\} + \left[\frac{x^2}{2} \right]_0^h \left\{ \frac{dN_i}{dy} \right\} \left[\frac{dN_i}{dy} \right] \{\phi_i\}$$

$$= \frac{h^2}{2} \left\{ \frac{dN_i}{dx} \right\} \left[\frac{dN_i}{dx} \right] \{\phi_i\} + \frac{h^2}{2} \left\{ \frac{dN_i}{dy} \right\} \left[\frac{dN_i}{dy} \right] \{\phi_i\}$$

$$= \left\{ \frac{dN_i}{dx} \right\} \left[\frac{dN_i}{dx} \right] \{\phi_i\} + \left\{ \frac{dN_i}{dy} \right\} \left[\frac{dN_i}{dy} \right] \{\phi_i\}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากฝั่งขวาของสมการที่ (3.6) จะได้

$$= \int_0^h \left[\left\{ \frac{1}{2A}(a_i + b_i x + c_i y) \right\} \right]_y^h dy \frac{\partial \phi}{\partial x} + \int_0^h \left[\left\{ \frac{1}{2A}(a_i + b_i x + c_i y) \right\} \right]_0^x dx \frac{\partial \phi}{\partial y}$$

$$= \int_0^h \left\{ \frac{1}{2A} b_i (h - y) \right\} dy \frac{\partial \phi}{\partial x} + \int_0^h \left\{ \frac{1}{2A} c_i x \right\} dx \frac{\partial \phi}{\partial y}$$

$$= \frac{1}{2A} \left\{ b_i \left[hy - \frac{y^2}{h} \right]_0^h \right\} \frac{\partial \phi}{\partial x} + \frac{1}{2A} \left\{ c_i \frac{h^2}{2} \right\} \frac{\partial \phi}{\partial y}$$

ขั้นที่ 5 แทนพจน์ที่เกี่ยวข้องกับขอบเขตของเอลิเมนต์ , $\Gamma^{(e)}$,ด้วยภาวะขอบเขตอื่นๆที่เกี่ยวข้อง ซึ่งจะก่อให้เกิดสมการของเอลิเมนต์ที่สมบูรณ์สำหรับปัญหานั้น

ขั้นที่ 6 จากนั้นจึงเขียนสมการของเอลิเมนต์ ซึ่งมีทั้งหมด m สมการให้อยู่ในรูปของเมทริกซ์ นั่นคือ

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

โดย [K] คือ เอลิเมนต์เมทริกซ์ของความแข็งเกร็ง (element stiffness matrix)

$\{\phi\}$ คือ เวกเตอร์ซึ่งประกอบด้วยตัวไม่รู้ค่าที่ จุดต่อต่าง ๆ ของเอลิเมนต์

{F} คือ โหลดเวกเตอร์ของเอลิเมนต์นั้น

จาก

$$\left\{ \frac{dN_i}{dx} \right\} \left\lfloor \frac{dN_i}{dx} \right\rfloor \{\phi_i\} + \left\{ \frac{dN_i}{dy} \right\} \left\lfloor \frac{dN_i}{dy} \right\rfloor \{\phi_i\} = \frac{1}{2A} \{b_i\} \frac{\partial \phi}{\partial x} + \frac{1}{2A} \{c_i\} \frac{\partial \phi}{\partial y}$$

$$\left[\frac{1}{4A^2} \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix} \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} + \frac{1}{4A^2} \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \end{Bmatrix} \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} \right] \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}$$

$$= \frac{1}{2A} \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix} \frac{\partial \phi}{\partial x} + \frac{1}{2A} \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \end{Bmatrix} \frac{\partial \phi}{\partial y} \qquad (3.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก $A = \frac{1}{2}$

$a_1 = 1$, $b_1 = -1$, $c_1 = 0$

$a_2 = 0$, $b_2 = 1$, $c_2 = -1$

$a_3 = 0$, $b_3 = 0$, $c_3 = 1$

แทนค่าลงในสมการที่ (3.7) จะได้

$$\begin{Bmatrix}-1\\1\\0\end{Bmatrix}\begin{bmatrix}-1 & 1 & 0\end{bmatrix}+\begin{Bmatrix}0\\-1\\1\end{Bmatrix}\begin{bmatrix}0 & -1 & 1\end{bmatrix}\begin{Bmatrix}\phi_1\\\phi_2\\\phi_3\end{Bmatrix}=\begin{Bmatrix}-1\\1\\0\end{Bmatrix}\frac{\partial\phi}{\partial x}+\begin{Bmatrix}0\\-1\\1\end{Bmatrix}\frac{\partial\phi}{\partial y}$$

$$\left[\begin{bmatrix}1 & -1 & 0\\-1 & 1 & 0\\0 & 0 & 0\end{bmatrix}+\begin{bmatrix}0 & 0 & 0\\0 & 1 & -1\\0 & -1 & 1\end{bmatrix}\right]\begin{Bmatrix}\phi_1\\\phi_2\\\phi_3\end{Bmatrix}=\begin{Bmatrix}-\frac{\partial\phi}{\partial x}\\\frac{\partial\phi}{\partial x}-\frac{\partial\phi}{\partial y}\\\frac{\partial\phi}{\partial y}\end{Bmatrix}$$

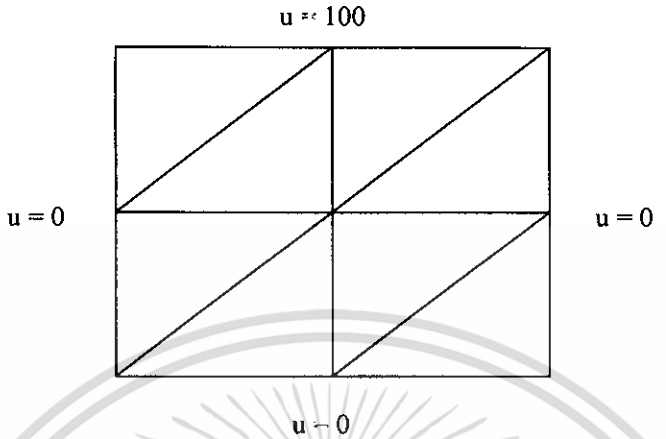
$$\begin{bmatrix}1 & -1 & 0\\-1 & 2 & -1\\0 & -1 & 1\end{bmatrix}\begin{Bmatrix}\phi_1\\\phi_2\\\phi_3\end{Bmatrix}=\begin{Bmatrix}-\frac{\partial\phi}{\partial x}\\\frac{\partial\phi}{\partial x}-\frac{\partial\phi}{\partial y}\\\frac{\partial\phi}{\partial y}\end{Bmatrix}$$

จากกระบวนการตามหลักทฤษฎีของไฟไนต์เอลิเมนต์ข้างต้น เป็นการหาระบบสมการเมทริกซ์จากเอลิมนต์สามเหลี่ยมเพียงหนึ่งเอลิเมนต์ ก็จะได้สมการเมทริกซ์ดังที่เห็น ซึ่งการจะหาผลเฉลยทั้งหมดของขอบเขตปัญหา จำเป็นต้องทำตามกระบวนการเช่นนี้จนกระทั่งได้ครบทุกเอลิเมนต์ แล้วจากนั้นก็นำสมการเมทริกซ์ในแต่ละเอลิเมนต์มารวมเข้าด้วยกัน กำหนดค่าเงื่อนไขขอบเขตให้แก่สมการระบบรวม แล้วนำสมการเมทริกซ์ระบบรวมที่ได้ไปแก้หาผลเฉลยด้วยกระบวนการ Triangular Decomposition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

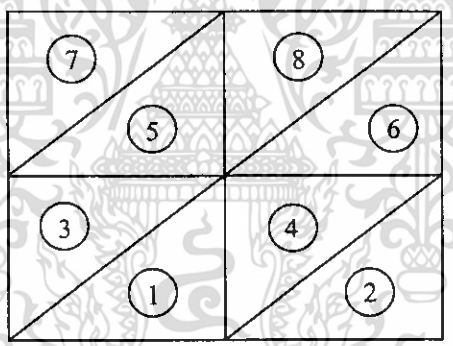
ตัวอย่าง ให้ $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$

เมื่อขอบเขตปัญหาแสดงดังรูป

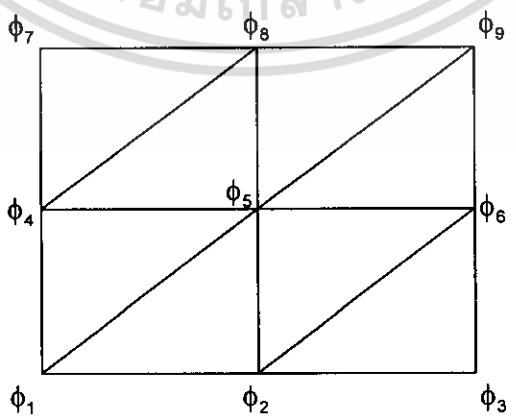


จงหา u(i,j) เมื่อ i = 0,1,2 และ j = 0,1,2

<u>วิธีทำ</u> ทำการแบ่งขอบเขตปัญหาด้วยเอลิเมนต์สามเหลี่ยม



จะได้เอลิเมนต์สามเหลี่ยมทั้งหมด 6 เอลิเมนต์

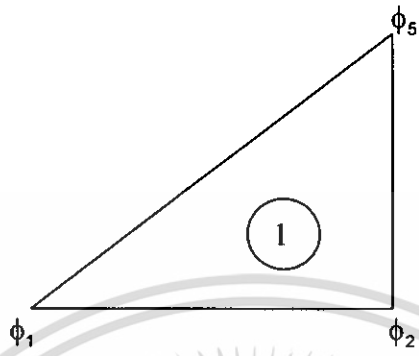


จะมีตัวไม่ทราบค่า (ϕ) ที่แต่ละจุดต่อ 9 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการหาสมการเมทริกซ์ในแต่ละเอลิเมนต์

<u>เอลิเมนต์ที่ 1</u>

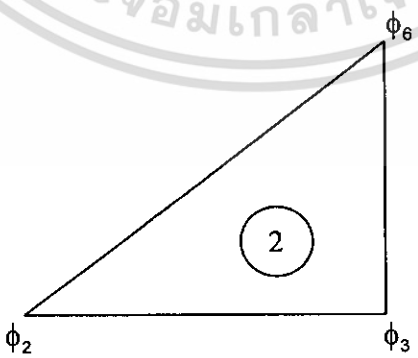


จากระเบียบวิธีไฟไนต์เอลิเมนต์ที่แสดงไว้ จะได้ว่า

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_5 \end{Bmatrix} = \begin{Bmatrix} \dfrac{-\partial\phi}{\partial x} \\ \dfrac{\partial\phi}{\partial x} - \dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial y} \end{Bmatrix} \tag{3.8}$$

เป็นสมการเมทริกซ์ของเอลิเมนต์ที่ 1

<u>เอลิเมนต์ที่ 2</u>



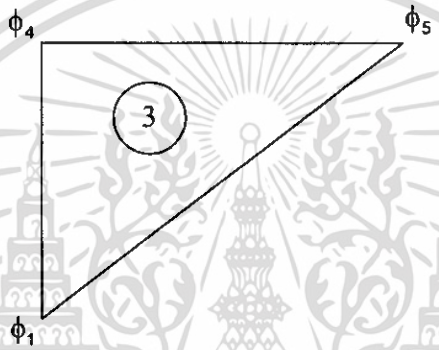
จากระเบียบวิธีไฟไนต์เอลิเมนต์ที่แสดงไว้ จะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_6 \end{Bmatrix} = \begin{Bmatrix} \dfrac{-\partial\phi}{\partial x} \\ \dfrac{\partial\phi}{\partial x} - \dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial y} \end{Bmatrix} \tag{3.9}$$

เป็นสมการเมทริกซ์ของเอลิเมนต์ที่ 2

<u>เอลิเมนต์ที่ 3</u>



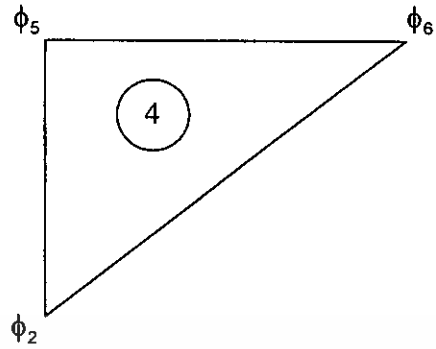
จากระเบียบวิธีไฟไนต์เอลิเมนต์ที่แสดงไว้ จะได้ว่า

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_5 \\ \phi_4 \end{Bmatrix} = \begin{Bmatrix} \dfrac{-\partial\phi}{\partial x} \\ \dfrac{\partial\phi}{\partial x} - \dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial y} \end{Bmatrix} \tag{3.10}$$

เป็นสมการเมทริกซ์ของเอลิเมนต์ที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>เอลิเมนต์ที่ 4</u>

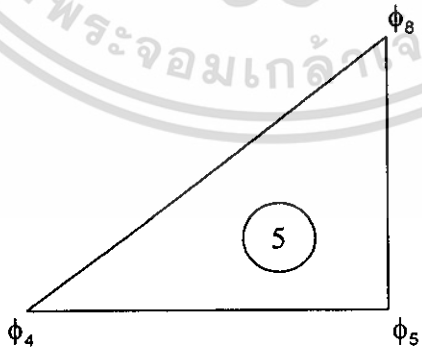


จากระเบียบวิธีไฟไนต์เอลิเมนต์ที่แสดงไว้ จะได้ว่า

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} \phi_2 \\ \phi_6 \\ \phi_5 \end{Bmatrix} = \begin{Bmatrix} \dfrac{-\partial\phi}{\partial x} \\ \dfrac{\partial\phi}{\partial x} - \dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial y} \end{Bmatrix} \tag{3.11}$$

เป็นสมการเมทริกซ์ของเอลิเมนต์ที่ 4

<u>เอลิเมนต์ที่ 5</u>



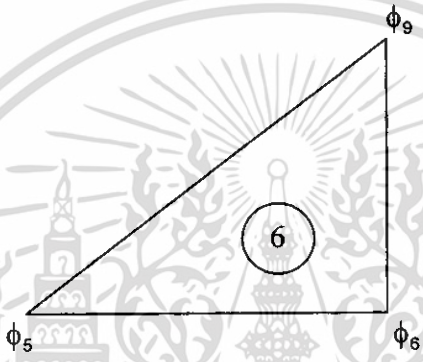
จากระเบียบวิธีไฟไนต์เอลิเมนต์ที่แสดงไว้ จะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} \phi_4 \\ \phi_5 \\ \phi_8 \end{Bmatrix} = \begin{Bmatrix} \dfrac{-\partial\phi}{\partial x} \\ \dfrac{\partial\phi}{\partial x} - \dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial y} \end{Bmatrix} \tag{3.12}$$

เป็นสมการเมทริกซ์ของเอลิเมนต์ที่ 5

<u>เอลิเมนต์ที่ 6</u>



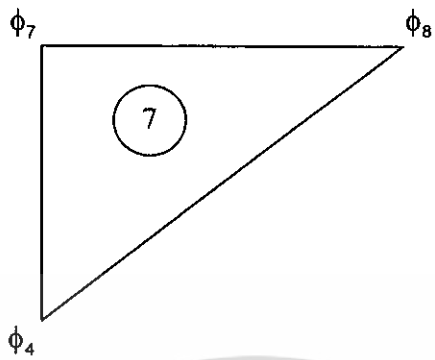
จากระเบียบวิธีไฟไนต์เอลิเมนต์ที่แสดงไว้ จะได้ว่า

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} \phi_5 \\ \phi_6 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} \dfrac{-\partial\phi}{\partial x} \\ \dfrac{\partial\phi}{\partial x} - \dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial y} \end{Bmatrix} \tag{3.13}$$

เป็นสมการเมทริกซ์ของเอลิเมนต์ที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>เอลิเมนต์ที่ 7</u>

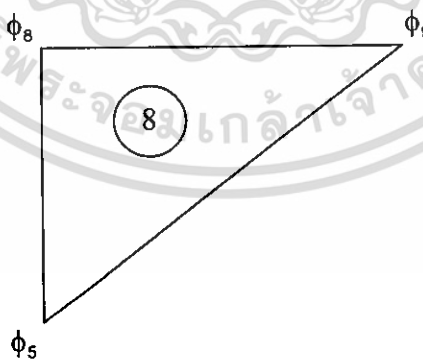


จากระเบียบวิธีไฟไนต์เอลิเมนต์ที่แสดงไว้ จะได้ว่า

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} \phi_4 \\ \phi_8 \\ \phi_7 \end{Bmatrix} = \begin{Bmatrix} \dfrac{-\partial\phi}{\partial x} \\ \dfrac{\partial\phi}{\partial x} - \dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial y} \end{Bmatrix} \qquad (3.14)$$

เป็นสมการเมทริกซ์ของเอลิเมนต์ที่ 7

<u>เอลิเมนต์ที่ 8</u>



จากระเบียบวิธีไฟไนต์เอลิเมนต์ที่แสดงไว้ จะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} \phi_5 \\ \phi_9 \\ \phi_8 \end{Bmatrix} = \begin{Bmatrix} \dfrac{-\partial\phi}{\partial x} \\ \dfrac{\partial\phi}{\partial x} - \dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial y} \end{Bmatrix} \tag{3.15}$$

เป็นสมการเมทริกซ์ของเอลิเมนต์ที่ 8

นำสมการเมทริกซ์ของแต่ละเอลิเมนต์มารวมกันเป็นสมการเมทริกซ์ระบบรวม ดังนี้

จากสมการที่ (3.8)

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{bmatrix} 1 & -1 & & & 0 & & & & \\ -1 & 2 & & & -1 & & & & \\ & & & & & & & & \\ & & & & & & & & \\ 0 & -1 & & & 1 & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{bmatrix} \end{array} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} \dfrac{-\partial\phi}{\partial x} \\ \dfrac{\partial\phi}{\partial x} - \dfrac{\partial\phi}{\partial y} \\ \\ \\ \dfrac{\partial\phi}{\partial y} \\ \\ \\ \\ \\ \end{Bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (3.9)

$$
\begin{array}{c}
\\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
\begin{bmatrix}
1 & -1 & & & 0 & & & & \\
-1 & 3 & -1 & & -1 & 0 & & & \\
 & -1 & 2 & & & -1 & & & \\
 & & & & & & & & \\
0 & -1 & & & 1 & & & & \\
 & 0 & -1 & & & 1 & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{bmatrix}
\end{array}
\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix}
=
\begin{Bmatrix} \dfrac{-\partial\phi}{\partial x} \\ -\dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial x}-\dfrac{\partial\phi}{\partial y} \\ \\ \dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial y} \\ \\ \\ \\ \end{Bmatrix}
$$

จากสมการที่ (3.10)

$$
\begin{array}{c}
\\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
\begin{bmatrix}
2 & -1 & & 0 & -1 & & & & \\
-1 & 3 & -1 & & -1 & 0 & & & \\
 & -1 & 2 & & & -1 & & & \\
0 & & & 1 & -1 & & & & \\
-1 & -1 & & -1 & 3 & & & & \\
 & 0 & -1 & & & 1 & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{bmatrix}
\end{array}
\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix}
=
\begin{Bmatrix} \dfrac{-2\partial\phi}{\partial x} \\ -\dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial x}-\dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial y} \\ \dfrac{\partial\phi}{\partial x} \\ \dfrac{\partial\phi}{\partial y} \\ \\ \\ \\ \end{Bmatrix}
$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (3.11)

$$
\begin{array}{c}
\\ 1\\ 2\\ 3\\ 4\\ 5\\ 6\\ 7\\ 8\\ 9
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array}\\
\left[\begin{array}{ccccccccc}
2 & -1 & & 0 & -1 & & & & \\
-1 & 4 & -1 & & -1 & -1 & & & \\
 & -1 & 2 & & & -1 & & & \\
0 & & & 1 & -1 & & & & \\
-1 & -1 & & -1 & 4 & -1 & & & \\
 & -1 & -1 & & -1 & 3 & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{array}\right]
\end{array}
\begin{Bmatrix} \phi_1\\ \phi_2\\ \phi_3\\ \phi_4\\ \phi_5\\ \phi_6\\ \phi_7\\ \phi_8\\ \phi_9 \end{Bmatrix}
=
\begin{Bmatrix}
\frac{-2\partial\phi}{\partial x}\\
-\frac{\partial\phi}{\partial y}-\frac{\partial\phi}{\partial x}\\
\frac{\partial\phi}{\partial x}-\frac{\partial\phi}{\partial y}\\
\frac{\partial\phi}{\partial y}\\
\frac{\partial\phi}{\partial x}+\frac{\partial\phi}{\partial y}\\
\frac{\partial\phi}{\partial x}\\
\\
\\
\\
\end{Bmatrix}
$$

จากสมการที่ (3.12)

$$
\begin{array}{c}
\\ 1\\ 2\\ 3\\ 4\\ 5\\ 6\\ 7\\ 8\\ 9
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array}\\
\left[\begin{array}{ccccccccc}
2 & -1 & & 0 & -1 & & & & \\
-1 & 4 & -1 & & -1 & -1 & & & \\
 & -1 & 2 & & & -1 & & & \\
0 & & & 2 & -2 & & & 0 & \\
-1 & -1 & & -2 & 6 & -1 & & -1 & \\
 & -1 & -1 & & -1 & 3 & & & \\
 & & & & & & & & \\
 & & & 0 & -1 & & & 1 & \\
 & & & & & & & &
\end{array}\right]
\end{array}
\begin{Bmatrix} \phi_1\\ \phi_2\\ \phi_3\\ \phi_4\\ \phi_5\\ \phi_6\\ \phi_7\\ \phi_8\\ \phi_9 \end{Bmatrix}
=
\begin{Bmatrix}
\frac{-2\partial\phi}{\partial x}\\
-\frac{\partial\phi}{\partial y}-\frac{\partial\phi}{\partial x}\\
\frac{\partial\phi}{\partial x}-\frac{\partial\phi}{\partial y}\\
\frac{\partial\phi}{\partial y}-\frac{\partial\phi}{\partial x}\\
2\frac{\partial\phi}{\partial x}\\
\frac{\partial\phi}{\partial x}\\
\\
\frac{\partial\phi}{\partial y}\\
\\
\end{Bmatrix}
$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (3.13)

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
\begin{bmatrix}
2 & -1 & & 0 & -1 & & & & \\
-1 & 4 & -1 & & -1 & -1 & & & \\
 & -1 & 2 & & & -1 & & & \\
0 & & & 2 & -2 & & & 0 & \\
-1 & -1 & & -2 & 7 & -2 & & -1 & 0 \\
 & -1 & -1 & & -2 & 5 & & & -1 \\
 & & & & & & & & \\
 & & & 0 & -1 & & & 1 & \\
 & & & & 0 & -1 & & & 1
\end{bmatrix}
\end{array}
\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix}
=
\begin{Bmatrix}
\dfrac{-2\partial\phi}{\partial x} \\
-\dfrac{\partial\phi}{\partial y}-\dfrac{\partial\phi}{\partial x} \\
\dfrac{\partial\phi}{\partial x}-\dfrac{\partial\phi}{\partial y} \\
\dfrac{\partial\phi}{\partial y}-\dfrac{\partial\phi}{\partial x} \\
\dfrac{\partial\phi}{\partial x} \\
2\dfrac{\partial\phi}{\partial x}-\dfrac{\partial\phi}{\partial y} \\
 \\
\dfrac{\partial\phi}{\partial y} \\
\dfrac{\partial\phi}{\partial y}
\end{Bmatrix}
$$

จากสมการที่ (3.14)

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
\begin{bmatrix}
2 & -1 & & 0 & -1 & & & & \\
-1 & 4 & -1 & & -1 & -1 & & & \\
 & -1 & 2 & & & -1 & & & \\
0 & & & 3 & -2 & & 0 & -1 & \\
-1 & -1 & & -2 & 7 & -2 & & -1 & 0 \\
 & -1 & -1 & & -2 & 5 & & & -1 \\
 & & & 0 & & & 1 & -1 & \\
 & & & -1 & -1 & & -1 & 3 & \\
 & & & & 0 & -1 & & & 1
\end{bmatrix}
\end{array}
\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix}
=
\begin{Bmatrix}
\dfrac{-2\partial\phi}{\partial x} \\
-\dfrac{\partial\phi}{\partial y}-\dfrac{\partial\phi}{\partial x} \\
\dfrac{\partial\phi}{\partial x}-\dfrac{\partial\phi}{\partial y} \\
\dfrac{\partial\phi}{\partial y}-2\dfrac{\partial\phi}{\partial x} \\
\dfrac{\partial\phi}{\partial x} \\
2\dfrac{\partial\phi}{\partial x}-\dfrac{\partial\phi}{\partial y} \\
\dfrac{\partial\phi}{\partial y} \\
\dfrac{\partial\phi}{\partial x} \\
\dfrac{\partial\phi}{\partial y}
\end{Bmatrix}
$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก (3.15)

$$\begin{bmatrix} 2 & -1 & & 0 & -1 & & & & \\ -1 & 4 & -1 & & -1 & -1 & & & \\ & -1 & 2 & & & -1 & & & \\ 0 & & & 3 & -2 & & 0 & -1 & \\ -1 & -1 & & -2 & 8 & -2 & & -1 & -1 \\ & -1 & -1 & & -2 & 5 & & & -1 \\ & & & 0 & & & 1 & -1 & \\ & & & -1 & -1 & & -1 & 4 & -1 \\ & & & & -1 & -1 & & -1 & 3 \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} \frac{-2\partial\phi}{\partial x} \\ -\frac{\partial\phi}{\partial y} - \frac{\partial\phi}{\partial x} \\ \frac{\partial\phi}{\partial x} - \frac{\partial\phi}{\partial y} \\ \frac{\partial\phi}{\partial y} - 2\frac{\partial\phi}{\partial x} \\ 0 \\ 2\frac{\partial\phi}{\partial x} - \frac{\partial\phi}{\partial y} \\ \frac{\partial\phi}{\partial y} \\ \frac{\partial\phi}{\partial x} + \frac{\partial\phi}{\partial y} \\ \frac{\partial\phi}{\partial x} \end{Bmatrix}$$

ดังนั้น สมการเมทริกซ์ระบบรวม คือ

$$\begin{bmatrix} 2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & -2 & 0 & 0 & -1 & 0 \\ -1 & -1 & 0 & -2 & 8 & -2 & 0 & -1 & -1 \\ 0 & -1 & -1 & 0 & -2 & 5 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 3 \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} \frac{-2\partial\phi}{\partial x} \\ -\frac{\partial\phi}{\partial y} - \frac{\partial\phi}{\partial x} \\ \frac{\partial\phi}{\partial x} - \frac{\partial\phi}{\partial y} \\ \frac{\partial\phi}{\partial y} - 2\frac{\partial\phi}{\partial x} \\ 0 \\ 2\frac{\partial\phi}{\partial x} - \frac{\partial\phi}{\partial y} \\ \frac{\partial\phi}{\partial y} \\ \frac{\partial\phi}{\partial x} + \frac{\partial\phi}{\partial y} \\ \frac{\partial\phi}{\partial x} \end{Bmatrix}$$

จากนั้นกำหนดเงื่อนไขขอบเขตให้กับสมการเมทริกซ์ระบบรวม โดยที่จุดต่อ 1, 2, 3, 4, 6, 7 และ 9 มีค่าเท่ากับ 0 ส่วนจุดต่อที่ 8 มีค่าเท่ากับ 100 และกำหนดแบบ Dirichlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
\left[\begin{array}{ccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 0 & -2 & 8 & -2 & 0 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]
\end{array}
\left\{\begin{array}{c} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{array}\right\}
=
\left\{\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 100 \\ 0 \end{array}\right\}
$$

จะได้ว่า

$$\phi_1 = 0$$
$$\phi_2 = 0$$
$$\phi_3 = 0$$
$$\phi_4 = 0$$
$$\phi_5 = 12.5$$
$$\phi_6 = 0$$
$$\phi_7 = 0$$
$$\phi_8 = 100$$
$$\phi_9 = 0$$

แต่จากที่ได้แสดงให้เห็นกระบวนการดังกล่าวข้างต้น จะพบว่าการทำเช่นนี้มีความยุ่งยากมาก เพื่อความเหมาะสมแก่เวลาและขอบเขตในการศึกษาปัญหาพิเศษ จึงได้ทำการค้นคว้าเอกสารและหนังสือต่าง ๆ มากมาย จนกระทั่งพบว่าได้มีผู้นำเสนอโปรแกรมในการคำนวณด้วยระเบียบวิธีไฟไนต์เอลิเมนต์ไว้ในขั้นต้นแล้ว จึงได้ทำการศึกษาตัวโปรแกรมนั้น แล้วนำมาพัฒนาให้ดีขึ้นในขอบเขตเนื้อหาของปัญหาพิเศษนี้

3.5.2 โปรแกรมระเบียบวิธีไฟไนต์เอลิเมนต์ด้วย MATLAB

เพื่อความเหมาะสมแก่เวลาในการทำปัญหาพิเศษชิ้นนี้ จึงได้ทำการศึกษาโปรแกรมที่มีอยู่แล้ว เพื่อนำมาพัฒนาให้ดียิ่งขึ้นและลดความยุ่งยากที่ต้องทำการสร้างสมการเมทริกซ์ด้วยกลไกทางคณิตศาสตร์อย่างที่ได้นำเสนอไว้ โดยที่โปรแกรมที่หามาได้นั้นใช้ไวยากรณ์ของ MATLAB ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-------------------------------------------------------------------------
%Example5.9.1.m
%to solve the two-dimensional Laplace equation given  as
%u,xx+u,yy=0,0 < x < 5 ,0 < y < 10
%u(x,0) = 0,u(x,10) = 100sin(pi*x/10),
%u(0,y) = 0,u,x(5,y) = 0
%using linear triangle elements
%(see Fig.5.9.1 for the finite element mesh)
%
%Variable descriptions
%k = element matrix
%f = element vector
%kk= system matrix
%ff= system vector
%gcoord = coordinate value of each node
%nodes = nodal connectivity of each element
%index = a vector containing system dofs associated with each element
%bcdof = a vector containing  dofs associated with boundary conditions
%bcval = a vector containing  boundary conditions  values associated with
%       the dofs in bcdof
%-------------------------------------------------------------------------
%
%---------------------------------
%input data for control parameter
%---------------------------------
nel=32;                 % number of elements
nnel=3;                 % number of node per element
ndof=1;                 % number of dofs per node
nnode=25;               % total number of nodes in system
sdof=nnode*ndof;        % total system dofs
%
%-------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% input data for nodal coordinate values
% gcoord(i,j)where i-> node no. and j-> x or y
%-----------------------------------------------
gcoord(1,1)=0.0;gcoord(1,2)=0.0;
gcoord(2,1)=0.0;gcoord(2,2)=0.0;
gcoord(3,1)=0.0;gcoord(3,2)=0.0;
gcoord(4,1)=0.0;gcoord(4,2)=0.0;
gcoord(5,1)=0.0;gcoord(5,2)=0.0;
gcoord(6,1)=0.0;gcoord(6,2)=0.0;
gcoord(7,1)=0.0;gcoord(7,2)=0.0;
gcoord(8,1)=0.0;gcoord(8,2)=0.0;
gcoord(9,1)=0.0;gcoord(9,2)=0.0;
gcoord(10,1)=0.0;gcoord(10,2)=0.0;
gcoord(11,1)=0.0;gcoord(11,2)=0.0;
gcoord(12,1)=0.0;gcoord(12,2)=0.0;
gcoord(13,1)=0.0;gcoord(13,2)=0.0;
gcoord(14,1)=0.0;gcoord(14,2)=0.0;
gcoord(15,1)=0.0;gcoord(15,2)=0.0;
gcoord(16,1)=0.0;gcoord(16,2)=0.0;
gcoord(17,1)=0.0;gcoord(17,2)=0.0;
gcoord(18,1)=0.0;gcoord(18,2)=0.0;
gcoord(19,1)=0.0;gcoord(19,2)=0.0;
gcoord(20,1)=0.0;gcoord(20,2)=0.0;
gcoord(21,1)=0.0;gcoord(21,2)=0.0;
gcoord(22,1)=0.0;gcoord(22,2)=0.0;
gcoord(23,1)=0.0;gcoord(23,2)=0.0;
gcoord(24,1)=0.0;gcoord(24,2)=0.0;
gcoord(25,1)=0.0;gcoord(25,2)=0.0;
%
%-----------------------------------------------------
% input data for nodal connectivity for each element
% nodes(i,j)where i-> element no. and j-> connected nodes
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-------------------------------------------------------
nodes(1,1)=1;nodes(1,2)=2;nodes(1,3)=7;
nodes(2,1)=2;nodes(2,2)=3;nodes(2,3)=8;
nodes(3,1)=3;nodes(3,2)=4;nodes(3,3)=9;
nodes(4,1)=4;nodes(4,2)=5;nodes(4,3)=10;
nodes(5,1)=1;nodes(5,2)=7;nodes(5,3)=6;
nodes(6,1)=2;nodes(6,2)=8;nodes(6,3)=7;
nodes(7,1)=3;nodes(7,2)=9;nodes(7,3)=8;
nodes(8,1)=4;nodes(8,2)=10;nodes(8,3)=9;
nodes(9,1)=6;nodes(9,2)=7;nodes(9,3)=12;
nodes(10,1)=7;nodes(10,2)=8;nodes(10,3)=13;
nodes(11,1)=8;nodes(11,2)=9;nodes(11,3)=14;
nodes(12,1)=9;nodes(12,2)=10;nodes(12,3)=15;
nodes(13,1)=6;nodes(13,2)=12;nodes(13,3)=11;
nodes(14,1)=7;nodes(14,2)=13;nodes(14,3)=12;
nodes(15,1)=8;nodes(15,2)=14;nodes(15,3)=13;
nodes(16,1)=9;nodes(16,2)=15;nodes(16,3)=14;
nodes(17,1)=11;nodes(17,2)=12;nodes(17,3)=17;
nodes(18,1)=12;nodes(18,2)=13;nodes(18,3)=18;
nodes(19,1)=13;nodes(19,2)=14;nodes(19,3)=19;
nodes(20,1)=14;nodes(20,2)=15;nodes(20,3)=20;
nodes(21,1)=11;nodes(21,2)=17;nodes(21,3)=16;
nodes(22,1)=12;nodes(22,2)=18;nodes(22,3)=17;
nodes(23,1)=13;nodes(23,2)=19;nodes(23,3)=18;
nodes(24,1)=14;nodes(24,2)=20;nodes(24,3)=19;
nodes(25,1)=16;nodes(25,2)=17;nodes(25,3)=20;
nodes(26,1)=17;nodes(26,2)=18;nodes(26,3)=23;
nodes(27,1)=18;nodes(27,2)=19;nodes(27,3)=24;
nodes(28,1)=19;nodes(28,2)=20;nodes(28,3)=25;
nodes(29,1)=16;nodes(29,2)=22;nodes(29,3)=21;
nodes(30,1)=17;nodes(30,2)=23;nodes(30,3)=22;
nodes(31,1)=18;nodes(31,2)=24;nodes(31,3)=23;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
nodes(32,1)=19;nodes(32,2)=25;nodes(32,3)=24;
%
%-------------------------------------------------
% input data for boundary conditions
%-------------------------------------------------
bcdof(1)=1;          % first node is constrian
bcval(1)=0;          % whose describe value is 0
bcdof(2)=2;          % second node is constrian
bcval(2)=0;          % whose describe value is 0
bcdof(3)=3;          % thrid node is constrian
bcval(3)=0;          % whose describe value is 0
bcdof(4)=4;          % fourth node is constrian
bcval(4)=0;          % whose describe value is 0
bcdof(5)=5;          % fifth node is constrian
bcval(5)=0;          % whose describe value is 0
bcdof(6)=6;          % sixth node is constrian
bcval(6)=0;          % whose describe value is 0
bcdof(7)=11;         % 11th node is constrian
bcval(7)=0;          % whose describe value is 0
bcdof(8)=16;         % 16th node is constrian
bcval(8)=0;          % whose describe value is 0
bcdof(9)=21;         % 21th node is constrian
bcval(9)=0;          % whose describe value is 0
bcdof(10)=22;        % 22th node is constrian
bcval(10)=38.2683;   % whose describe value is 38.2683
bcdof(11)=23;        % 23th node is constrian
bcval(11)=70.7107;   % whose describe value is 70.7107
bcdof(12)=24;        % 24th node is constrian
bcval(12)=92.3880;   % whose describe value is 92.3880
bcdof(13)=25;        % 25th node is constrian
bcval(13)=100;       % whose describe value is 100
%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-------------------------------------------------------------
% initialization of matrices and vectors
%-------------------------------------------------------------
ff=zeros(sdof,1);          % initialization of system force vector
kk=zeros(sdof,sdof);       % initialization of index vector
index=zeros(nnel*ndof,1);  % initialization of index vector
%
%-------------------------------------------------------------
% computation of element matrices and vectors and their assembly
%-------------------------------------------------------------
for iel=1:nel          % loop for the total number of elements
%
nd(1)=nodes(iel,1);    % 1st connected node for (iel)-th element
nd(2)=nodes(iel,2);    % 2nd connected node for (iel)-nd element
nd(3)=nodes(iel,3);    % 3rd connected node for (iel)-rd element
x1=gcoord(nd(1),1);y1=gcoord(nd(1),2);      % coord values of 1st node
x2=gcoord(nd(2),1);y2=gcoord(nd(2),2);      % coord values of 2nd node
x3=gcoord(nd(3),1);y3=gcoord(nd(3),2);      % coord values of 3rd node
%
index=feeldof(nd,nnel,ndof);               % extract system dofs for the element
%
k=felp2dt3(x1,y1,x2,y2,x3,y3);             % compute element matrix
%
kk=feasmbl1(kk,k,index);                   % assemble element matrices
%
end
%
%---------------------------------------
% apply boundary conditions
%---------------------------------------
[kk,ff]=feaplyc2(kk,ff,bcdof,bcval);
%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%----------------------------------------
% solve the matrix equation
%----------------------------------------
fsol=kk\ff;
%
%-----------------------------
% analytical solution
%-----------------------------
for i=1:nnode
   x=gcoord(i,1);y=gcoord(i,2);
   esol(i)=100*sinh(0.31415927*y)*sin(0.31415927*x)/sinh(3.1415927);
end
%
%---------------------------------
% print both exact and fem solutions
%---------------------------------
num=1:1:sdof;
store=[num' fsol esol']
%
%-------------------------------------------------------------------
function [kk]=feasmbl1(kk,k,index)
%-------------------------------------------------------------------
%Purpose:
%Assenbly of element matrices into the system matrix
%
%Synopsis:
%[kk]=feasmbl1(kk,k,index)
%
%Variable Description:
%kk - system matrix
%k - element matrix
%index - dof vector associated with an element
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-----------------------------------------------------------------------
%
edof = length(index);
for i=1:edof
   ii=index(i);
   for j=1:edof
      jj=index(j);
      kk(ii,jj)=kk(ii,jj)+k(i,j);
   end
end
%-----------------------------------------------------------------------

function [index]=feeldof(nd,nnel,ndof)
%-----------------------------------------------------------------------
%Purpose:
%Compute system dofs associated with each element
%
%Synopsis:
%[index]=feeldof(nd,nnel,ndof)
%
%Variable Description:
%index - system dof vector associated with element iel
%nd - element node numbers whose system dofs are to be determine
%nnel - number of nodes per element
%ndof - number of dofs per node
%-----------------------------------------------------------------------
%
edof=nnel*ndof;
k=0;
for i=1:nnel
   start=(nn(i)-1)*ndof;
   for j=1:ndof
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      k=k+1;
      index(k)=start+j:
   end
end
%-------------------------------------------------------------------------

function [k]=felp2dt3(x1,y1,x2,y2,x3,y3)
%-------------------------------------------------------------------------
%Purpose:
%element matrix for two-dimensional Laplace's equation
%using three-node linear triangular element
%
%Synopsis:
%[k]=felp2dt3(x1,y1,x2,y2,x3,y3)
%
%Variable Description:
%k - element stiffness matrix (size of 3*3)
%x1,y1 - x and y coordinate values of the first node of element
%x2,y2 - x and y coordinate values of the second node of element
%x3,y3 - x and y coordinate values of the third node of element
%-------------------------------------------------------------------------
%
%element matrix
%
A=0.5*(x2*y3+x1*y2+x3*y1-x2*y1-x1*y3-x3*y2);
%                                  %area of the triangle
k(1,1)=((x3-x2)*(x3-x2)+(y2-y3)*(y2-y3))/(4*A);
k(1,2)=((x3-x2)*(x1-x3)+(y2-y3)*(y3-y1))/(4*A);
k(1,3)=((x3-x2)*(x2-x1)+(y2-y3)*(y1-y2))/(4*A);
k(2,1)=k(1,2);
k(2,2)=((x1-x3)*(x1-x3)+(y3-y1)*(y3-y1))/(4*A);
k(2,3)=((x1-x3)*(x2-x1)+(y3-y1)*(y1-y2))/(4*A);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
k(3,1)=k(1,3);
k(3,2)=k(2,3);
k(3,3)=((x2-x1)*(x2-x1)+(y1-y2)*(y1-y2))/(4*A);
%-------------------------------------------------------------------------
```

จากโปรแกรมข้างต้นนี้ เป็นโปรแกรมอย่างง่ายที่ใช้คำนวณผลเฉลยในขอบเขตปัญหาที่เป็นสี่เหลี่ยมมุมฉาก ซึ่งมีเงื่อนไขและมีการกำหนดค่าไว้อย่างตายตัว ทำให้สามารถคำนวณหาผลเฉลยได้เพียงเงื่อนไขปัญหาอย่างหนึ่งเท่านั้น

ปัญหาพิเศษนี้ได้นำเอาอัลกอริทึมจากโปรแกรม MATLAB ข้างต้นมาศึกษาและพัฒนา ให้สามารถสร้างขอบเขตปัญหาในลักษณะสี่เหลี่ยมมุมฉากที่ยืดหยุ่นได้ และสามารถกำหนดเงื่อนไขขอบเขตได้ ทำให้โปรแกรมที่ได้พัฒนาขึ้นมามีประสิทธิภาพสูงกว่ามาก อีกทั้งยังสามารถแสดงผลเฉลยด้วยกราฟฟิคที่ทำให้ง่ายต่อการวิเคราะห์ผลลัพธ์อีกด้วย

3.5.3 Triangular decomposition

หลังจากที่ได้โปรแกรมในการคำนวณหาค่าต่าง ๆ สำหรับสมการเมทริกซ์ระบบรวมแล้ว ก็จะนำสมการเมทริกซ์ที่ได้เข้าสู่กระบวนการในการหาผลเฉลย ซึ่งในการหาผลเฉลยของสมการเมทริกซ์นั้นมีอยู่ด้วยกันหลายวิธี

ในปัญหาพิเศษนี้ได้นำเสนอวิธีการหาผลเฉลยของสมการเมทริกซ์ด้วยวิธี Triangular decomposition สาเหตุที่ใช้กระบวนการนี้ก็เพราะว่าสามารถลดการใช้หน่วยความจำลงได้ถึงครึ่งหนึ่งจากปกติ เพราะการคำนวณในลักษณะเช่นนี้จะทำการแยกเมทริกซ์ออกเป็นสองส่วน คือ เมทริกซ์สามเหลี่ยมบน และ เมทริกซ์สามเหลี่ยมล่าง ซึ่งทั้งสองเมทริกซ์นั้นจะสมมาตรกัน ทำให้เวลาขอพื้นที่ในหน่วยความจำสามารถลดการใช้พื้นที่ในหน่วยความจำลงได้ครึ่งหนึ่ง จึงเป็นสาเหตุให้เลือกกระบวนการดังกล่าว

สำหรับรายละเอียดของวิธีการ Triangular decomposition นั้น จะขอนำเสนอไว้ ดังนี้

<u>Triangular decomposition</u>

เมทริกซ์ใด ๆ สามารถเขียนให้อยู่ในรูปผลคูณ

$$A = LU$$

เมื่อ $U = [u_{ij}]$ เป็นเมทริกซ์สามเหลี่ยมบนขนาด $n \times n$

และ $L = [l_{ij}]$ เป็นเมทริกซ์สามเหลี่ยมล่างขนาด $n \times n$

โดย $l_{ij} = 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น

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

$$L=\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \quad \text{และ} \quad U=\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

หรือ $U=[u_{ij}]$ เป็นเมทริกซ์สามเหลี่ยมบนขนาด $n\times n$

โดยที่ $u_{ij}=1$

และ $L=[l_{ij}]$ เป็นเมทริกซ์สามเหลี่ยมล่างขนาด $n\times n$

$$L=\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \quad \text{และ} \quad U=\begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

ตัวอย่างที่ 1 $A=\begin{bmatrix} 1 & -3 & 7 \\ 2 & 4 & -3 \\ -3 & 7 & 2 \end{bmatrix}$ จงหา L และ U

$$LU=\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}=\begin{bmatrix} 1 & -3 & 7 \\ 2 & 4 & -3 \\ -3 & 7 & 2 \end{bmatrix}$$

$$=\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12}+u_{22} & l_{21}u_{13}+u_{23} \\ l_{31}u_{11} & l_{31}u_{12}+l_{32}u_{22} & l_{31}u_{13}+l_{32}u_{23}+u_{33} \end{bmatrix}$$

$u_{11}=a_{11}=1$, $u_{12}=a_{12}=-3$, $u_{13}=a_{13}=7$

$l_{21}u_{21}=a_{21}=2$, $l_{31}u_{11}=a_{31}=-3$

$l_{21}=2$, $l_{31}=-3$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$l_{21}u_{12}+u_{22}=a_{22}=4 \qquad l_{21}u_{13}+u_{23}=a_{23}=-3$$
$$u_{22}=4-2\times(-3) \quad , \qquad u_{23}=-3-2\times 7$$
$$=10 \qquad =-17$$

$$l_{31}u_{12}+l_{32}u_{22}=a_{32}=7 \qquad l_{31}u_{13}+l_{32}u_{23}+u_{33}=a_{33}=2$$
$$l_{32}=[7-(-3)(-3)]/10 \quad , \qquad u_{33}=2-(-3)(7)-(-1/5)(-17)$$
$$=-1/5 \qquad =98/5$$

จะได้ $$A=\begin{bmatrix}1 & -3 & 7\\ 2 & 4 & -3\\ -3 & 7 & 2\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ -3 & -1/5 & 1\end{bmatrix}\begin{bmatrix}1 & -3 & -7\\ 0 & 10 & -17\\ 0 & 0 & 98/5\end{bmatrix}$$

หรือ

$$LU=\begin{bmatrix}l_{11} & 0 & 0\\ l_{21} & l_{22} & 0\\ l_{31} & l_{32} & l_{33}\end{bmatrix}\begin{bmatrix}1 & u_{12} & u_{13}\\ 0 & 1 & u_{23}\\ 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}1 & -3 & 7\\ 2 & 4 & -3\\ -3 & 7 & 2\end{bmatrix}$$

$$=\begin{bmatrix}l_{11} & l_{11}u_{12} & l_{11}u_{13}\\ l_{21} & l_{21}u_{12}+l_{22} & l_{21}u_{13}+l_{22}u_{23}\\ l_{31} & l_{31}u_{12}+l_{32} & l_{31}u_{13}+l_{32}u_{23}+l_{33}\end{bmatrix}$$

$$l_{11}=a_{11}=1 \quad , \qquad l_{12}=a_{21}=2 \quad , \qquad l_{13}=a_{31}=-3$$

$$l_{11}u_{12}=a_{12}=-3 \qquad l_{11}u_{13}=a_{13}=7$$
$$u_{12}=-3 \quad , \qquad u_{13}=7$$

$$l_{21}u_{12}+l_{22}=a_{22}=4 \qquad l_{21}u_{13}+l_{22}u_{23}=a_{23}=-3$$
$$l_{22}=4-2\times(-3) \quad , \qquad u_{23}=[-3-2\times 7]/10$$
$$=10 \qquad =-17/10$$

$$l_{31}u_{12}+l_{32}=a_{32}=7 \qquad l_{31}u_{13}+l_{32}u_{23}+l_{33}=a_{33}=2$$
$$l_{32}=7-(-3)(-3) \quad , \qquad l_{33}=2-(-3)(7)-(-2)(-17/10)$$
$$=-2 \qquad =98/5$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ $$A=\begin{bmatrix}1 & -3 & 7\\ 2 & 4 & -3\\ -3 & 7 & 2\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ 2 & 10 & 0\\ -3 & -2 & 98/5\end{bmatrix}\begin{bmatrix}1 & -3 & -7\\ 0 & 1 & -17/10\\ 0 & 0 & 1\end{bmatrix}$$

หรือ $$=\begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ -3 & -1/5 & 1\end{bmatrix}\begin{bmatrix}1 & -3 & -7\\ 0 & 10 & -17\\ 0 & 0 & 98/5\end{bmatrix}$$

ระบบสมการเชิงเส้น n ตัวแปร $x_1, x_2, ..., x_n$

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n &= b_1\\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n &= b_2\\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n &= b_n\end{aligned}$$

สามารถเขียนในรูปเมทริกซ์ AX = B

โดย $$\begin{bmatrix}a_{11} & ... & a_{1n}\\ ... & ... & ...\\ a_{n1} & .. & a_{nn}\end{bmatrix}\begin{bmatrix}x_1\\ ...\\ x_n\end{bmatrix}=\begin{bmatrix}b_1\\ ...\\ b_n\end{bmatrix}$$

ถ้า A = LU แล้ว LUX = B

ให้ UX = Y แล้ว LY = B

<u>ตัวอย่างที่ 2</u> $$\begin{bmatrix}1 & -3 & 7\\ 2 & 4 & -3\\ -3 & 7 & 2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\begin{bmatrix}2\\ -1\\ 3\end{bmatrix}$$ หรือ AX = B

$$\begin{bmatrix}1 & -3 & 7\\ 2 & 4 & -3\\ -3 & 7 & 2\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ -3 & -1/5 & 1\end{bmatrix}\begin{bmatrix}1 & -3 & -7\\ 0 & 10 & -17\\ 0 & 0 & 98/5\end{bmatrix}$$

A = LU จะได้ L(UX) = B และ UX = Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{UX = Y} \quad \text{จะได้} \quad \begin{bmatrix} 1 & -3 & -7 \\ 0 & 10 & -17 \\ 0 & 0 & 98/5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$\text{LY = B} \quad \text{จะได้} \quad \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -1/5 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$$

ดังนั้น

$$y_1 = 2$$
$$y_2 = -1 - 2(2) = -5$$
$$y_3 = 3 - (-3)(2) - (-1/5)(-5) = 8$$

เพราะฉะนั้น $\begin{bmatrix} 1 & -3 & -7 \\ 0 & 10 & -17 \\ 0 & 0 & 98/5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -5 \\ 8 \end{bmatrix}$

และจะได้ว่า

$$x_3 = 8/(98/5) = 20/44$$
$$x_2 = [-5 - (-17)(20/49)]/10 = 19/98$$
$$x_1 = 2 - (-3)(19/98) - 7(20/49) = -27/98$$

ตัวอย่างที่ 3 จงใช้ LU decomposition หาผลเฉลยของระบบสมการ

$$2x_1 + 3x_2 + 4x_3 = 23$$
$$4x_1 + 10x_2 + 9x_3 = 59$$
$$6x_1 + 17x_2 + 20x_3 = 101$$

จะได้ว่า

$$\begin{bmatrix} 2 & 3 & 4 \\ 4 & 10 & 9 \\ 6 & 17 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 23 \\ 59 \\ 101 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} 2 & 3 & 4 \\ 4 & 10 & 9 \\ 6 & 17 & 20 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

$$= \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12}+u_{22} & l_{21}u_{13}+u_{23} \\ l_{31}u_{11} & l_{31}u_{12}+l_{32}u_{22} & l_{31}u_{13}+l_{32}u_{23}+u_{33} \end{bmatrix}$$

$u_{11}=2$, $u_{12}=3$, $u_{13}=4$

$l_{21}u_{11}=4$, $l_{31}u_{11}=6$

$l_{21}=2$, $l_{31}=3$

$l_{21}u_{12}+u_{22}=10$

$u_{22}=10-(2)(3)$

$=4$

$l_{21}u_{13}+u_{23}=9$

$u_{23}=9-(2)(4)$

$=1$

$l_{31}u_{12}+l_{32}u_{22}=17$

$l_{32}=[17-(3)(3)]/4$

$=2$

$l_{31}u_{13}+l_{32}u_{23}+u_{33}=20$

$u_{33}=20-(3)(4)-(2)(1)$

$=6$

$$\begin{bmatrix} 2 & 3 & 4 \\ 4 & 10 & 9 \\ 6 & 17 & 20 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \\ 0 & 4 & 1 \\ 0 & 0 & 6 \end{bmatrix}$$

จาก AX = B ถ้า A = LU แล้ว LUX = B

ถ้า UX = Y แล้ว LY = B

ดังนั้นจะได้

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 23 \\ 59 \\ 101 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจะได้ว่า

$$y_1 = 23$$
$$y_2 = 59 - 2\times 23 = 13$$
$$y_3 = 101 - 3\times 23 - 2\times 13 = 6$$

เพราะฉะนั้น
$$\begin{bmatrix} 2 & 3 & 4 \\ 0 & 4 & 1 \\ 0 & 0 & 6 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 23 \\ 13 \\ 6 \end{bmatrix}$$

และจะได้ว่า

$$x_3 = 6/6 = 1$$
$$x_2 = [13 - 1\times 1]/4 = 3$$
$$x_1 = (23 - 3\times 3 - 4\times 1)/2 = 10/2 = 5$$

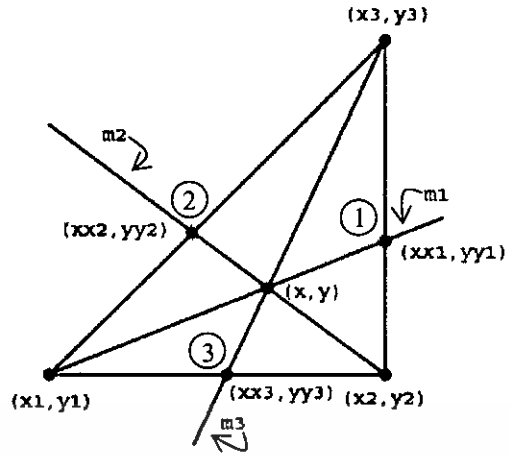
ดังนั้นจะได้
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix}$$

3.5.4 ขั้นตอนวิธีสำหรับการหาค่า z จากค่า x และ y ที่รับมาจากผู้ใช้ในหน้าแสดงผลกราฟ

ในการหาค่าผลลัพธ์จากจุด x, y ที่กำหนดให้จากผู้ใช้นั้น ได้กล่าวไว้พอสังเขปในบทที่ 2 แล้ว สำหรับในบทนี้จะนำเสนอขั้นตอนและหลักการโดยละเอียดที่ใช้ในการหาค่าผลลัพธ์ดังกล่าว

ขั้นตอนที่ 1 เมื่อรับค่า x และ y ที่ต้องการทราบค่าเข้ามา และทราบแล้วว่า ณ ตำแหน่ง x,y ที่รับเข้ามานั้นอยู่ในเอลิเมนต์สามเหลี่ยมใด ดังได้กล่าวไว้แล้วในบทที่ 2 หัวข้อที่ 2.7 การแสดงผลทางกราฟ หลักในการคำนวณเพื่อหาผลลัพธ์นั่นก็คือ ให้ทำการลากเส้นตรงจากมุมทั้งสามของเอลิเมนต์รูปสามเหลี่ยมผ่านตำแหน่ง x,y ที่ต้องการทราบค่าให้ไปตัดกับด้านของสามเหลี่ยมที่อยู่ตรงข้ามกับมุมนั้น ๆ จะได้จุดตัดทั้งหมดสามจุด ดังรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงเอลิเมนต์สามเหลี่ยมที่มีจุด (x,y) อยู่ และพิกัดจุดต่าง ๆ เมื่อลากเส้นผ่านจุด (x,y)

ขั้นตอนที่ 2 เมื่อได้จุดตัดทั้งสามดังรูปที่ 3.7 ก็ทำการคำนวณหาค่าตำแหน่ง x,y ที่จุดนั้น ๆ ซึ่งได้แก่ค่า xx1, xx2, xx3, yy1, yy2, yy3 โดยใช้สมการในการคำนวณดังนี้

$$xx1 = x2$$

$$yy1 = m1(x2 - x1) + y1 \tag{3.16}$$

$$xx2 = \frac{(m2x2) - (m1x1) + (y1 - y2)}{(m2 - m1)}$$

$$yy2 = \frac{((m1m2)(x2 - x1)) + (m2y1) - (m1y2)}{(m2 - m1)} \tag{3.17}$$

และ

$$xx3 = \frac{(y1 - y3)}{(m3 + x3)}$$

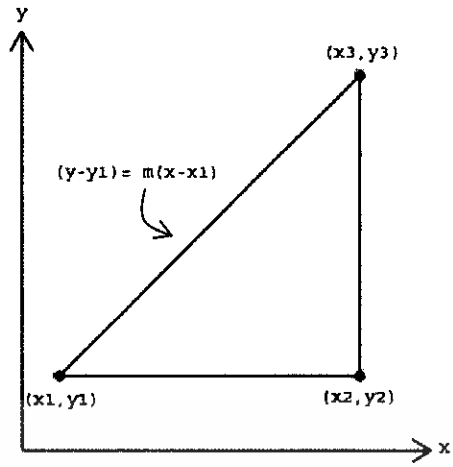
$$yy3 = y1 \tag{3.18}$$

โดยตำแหน่งค่า x และ ค่า y บนเส้นทแยงในสมการที่ (3.9) นั้นจะใช้ตำแหน่งค่า x และค่า y ที่คำนวณจากสมการนี้ในการคำนวณ

$$y = m(x - x1) + y1$$

$$x = \frac{(y - y1)}{m} + x1 \tag{3.19}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงสมการเส้นตรงของเส้นทแยงในเอลิมนต์สามเหลี่ยม

ขั้นตอนที่ 3 ทำการหาค่าถ่วงน้ำหนักของแต่ละจุดต่อโดยใช้สมการต่อไปนี้

$$l_1 = \sqrt{(xx1-x1)^2+(yy1-y1)^2}$$

$$l_s = \sqrt{(x-x1)^2+(y-y1)^2}$$

$$w_1 = l_s / l_1 \qquad (3.20)$$

$$l_1 = \sqrt{(xx2-x2)^2+(yy2-y2)^2}$$

$$l_s = \sqrt{(x-x2)^2+(y-y2)^2}$$

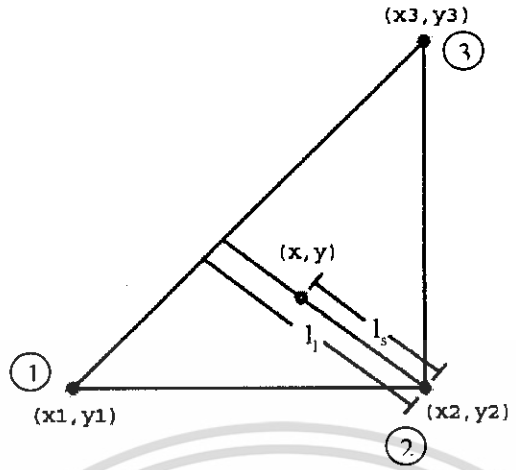
$$w_2 = l_s / l_1 \qquad (3.21)$$

$$l_1 = \sqrt{(xx3-x3)^2+(yy3-y3)^2}$$

$$l_s = \sqrt{(x-x3)^2+(y-y3)^2}$$

$$w_3 = l_s / l_1 \qquad (3.22)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดงระยะห่าง l_s และระยะห่าง l_l

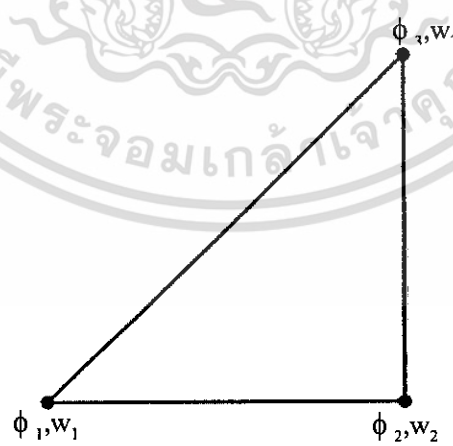
โดย w คือ ค่าถ่วงน้ำหนัก

l_s คือ ระยะทางที่สั้นที่สุดจากพิกัด(x,y) ถึงจุดต่อใด ๆ ในเอลิเมนต์ที่พิกัดนั้นอยู่

l_l คือ ระยะทางที่ยาวที่สุดจากพิกัด (x,y) ถึง จุดต่อใด ๆ ในเอลิเมนต์ที่พิกัดนั้นอยู่

<u>ขั้นตอนที่ 4</u> ทำการคำนวณหาค่า z โดยใช้สมการที่ (3.15)

$$Z = \frac{|\phi_1 \times w_1| + |\phi_2 \times w_2| + |\phi_3 \times w_3|}{|w_1| + |w_2| + |w_3|} \tag{3.23}$$



รูปที่ 3.10 แสดงค่าที่นำมาคำนวณ ณ ตำแหน่งต่าง ๆ ในเอลิเมนต์สามเหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การประเมินผล

จากการศึกษาเนื้องานที่ผ่านมา ได้ทำการพัฒนาโปรแกรมจากกระบวนการคำนวณทางคณิตศาสตร์ด้วยระเบียบวิธีไฟไนต์เอลิเมนต์เพื่อแก้ปัญหาสมการปัวส์ซอง ซึ่งได้โปรแกรมขึ้นมาตัวหนึ่งเมื่อนำค่าผลลัพธ์ที่คำนวณได้มาเปรียบเทียบกับโปรแกรม MATLAB ซึ่งเป็นที่ยอมรับอย่างกว้างขวางนั้นปรากฏว่ามีค่าใกล้เคียงกันอย่างมาก จึงอาจกล่าวได้ว่าโปรแกรมที่พัฒนาขึ้นมานี้ให้ผลลัพธ์ที่สามารถเชื่อถือได้

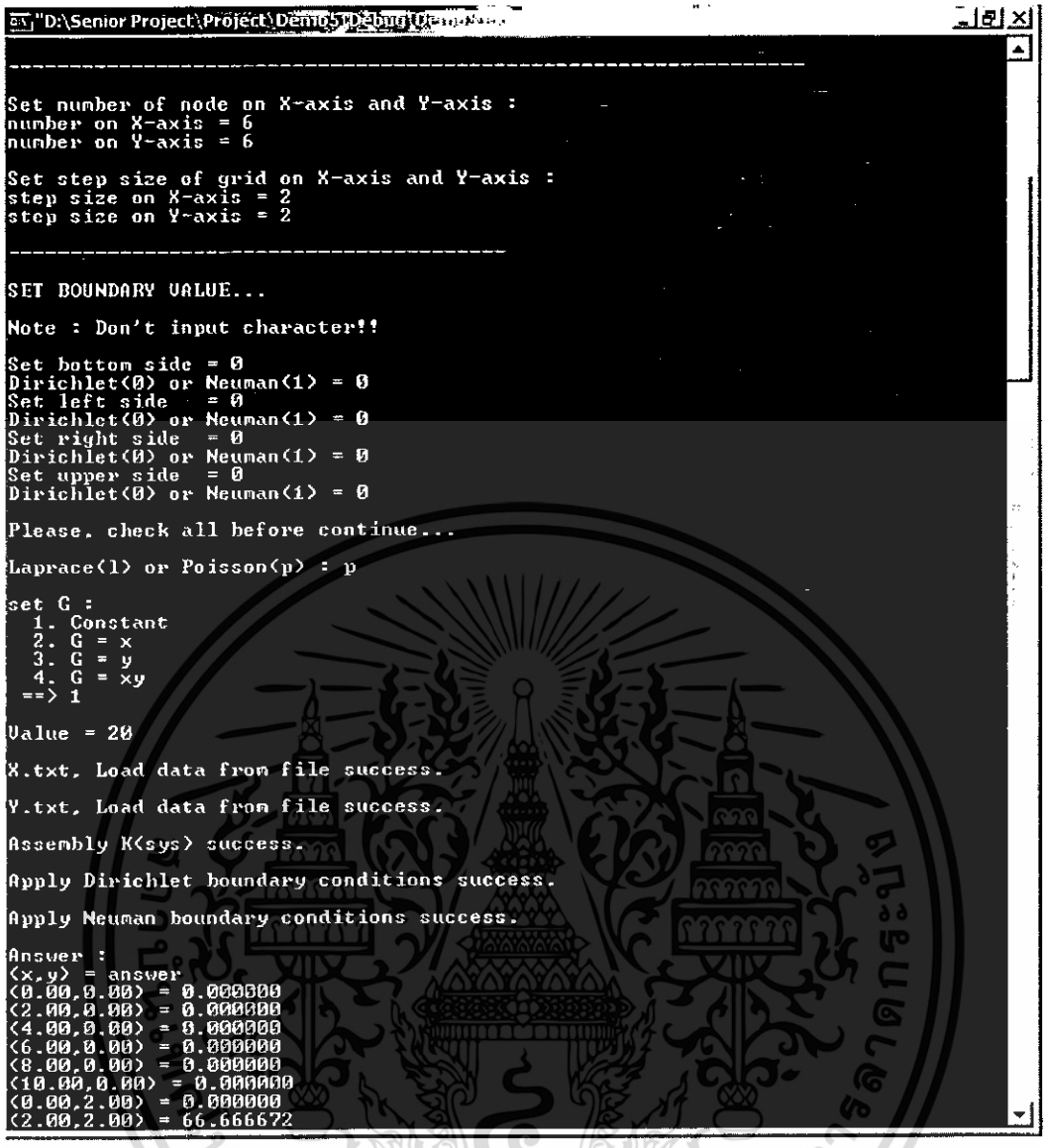
4.1 วิธีการใช้งานโปรแกรม

1. กำหนดค่าเหล่านี้ให้กับโปรแกรม
   1.1 จำนวนจุดต่อในแนวแกน x
   1.2 จำนวนจุดต่อในแนวแกน y
   1.3 ระยะห่างของแต่ละจุดต่อในแนวแกน x
   1.4 ระยะห่างของแต่ละจุดต่อในแนวแกน y
2. กำหนดค่าที่ขอบของปัญหาทั้ง 4 ด้านพร้อมทั้งกำหนดว่าจะให้เป็นแบบค่าคงที่ Dirichlet หรือ มีลักษณะเป็นอัตราการเปลี่ยนแปลงแบบ Neuman
3. เลือกว่าจะทำการคำนวณด้วยสมการลาปลาซหรือสมการปัวส์ซอง
4. ถ้าเลือกสมการปัวส์ซอง โปรแกรมจะถามให้เลือกว่าจะกำหนดฟังก์ชันทางด้านขวาของสมการเป็นอะไรจาก 4 ตัวเลือกที่กำหนดให้ คือ
   - ค่าคงที่
   - ฟังก์ชัน x
   - ฟังก์ชัน y
   - ฟังก์ชัน xy

   ถ้าเลือกเป็นค่าคงที่ก็จะต้องระบุค่าคงที่ด้วย

เมื่อกำหนดทุกอย่างครบตามข้างต้น โปรแกรมก็จะทำการคำนวณค่าผลลัพธ์ออกมาให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 ภาพตัวอย่างแสดงผลการรับค่าจากผู้ใช้โดยใช้สมการปัวส์ซองและฟังก์ชันทางด้านขวาเป็นค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การใช้งานกราฟแสดงผล

4.2.1 การควบคุมกราฟ

โปรแกรมแสดงผลด้วยกราฟนี้สามารถใช้แป้นอักษรควบคุมมุมมองต่าง ๆ ได้โดยมีปุ่มควบคุมต่าง ๆ ดังนี้

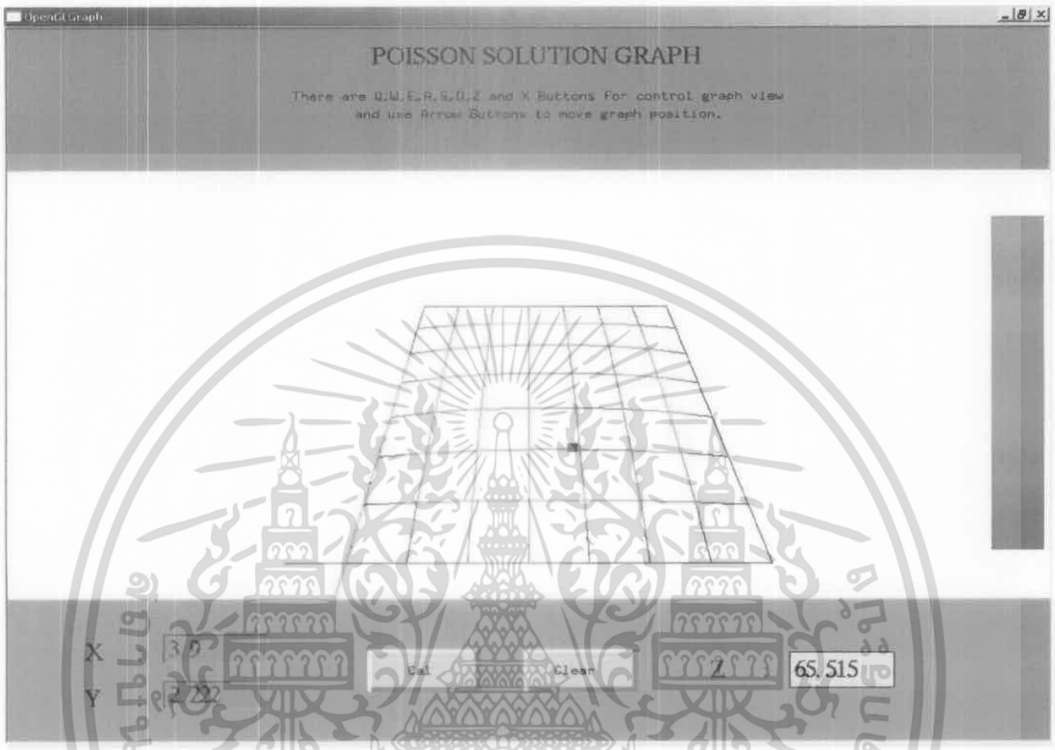
Up Arrow:	ใช้เลื่อนกราฟขึ้น
Down Arrow:	ใช้เลื่อนกราฟลง
Left Arrow	ใช้เลื่อนกราฟไปทางขวา
Right Arrow:	ใช้เลื่อนกราฟไปทางซ้าย
Key W/S:	เพิ่ม/ลด ค่าการหมุนในแกนตั้ง (Y-Axis)
Key Q/E:	เพิ่ม/ลด องศาในการหมุน
Key A/D:	เพิ่ม/ลด ค่าการหมุนในแกนนอน (X-Axis)
Key Z/X:	ซูม เข้า/ออก (Z-Axis)

4.2.2 การคำนวณผลลัพธ์จากค่า x และ y ที่กำหนดให้

รูปที่ 4.2 แสดงวิธีใช้งานโปรแกรมในการหาค่าผลเฉลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.2 ถ้าจะทำการกำหนดค่า x และ y เพื่อทำการหาค่าผลลัพธ์หรือค่า z นั้น ให้เลื่อนเมาส์ไปที่กล่องข้อความที่อยู่หลังอักษร x แล้วทำการคลิกที่กล่องข้อความนั้น กล่องข้อความจะเปลี่ยนเป็นพื้นสีขาวให้สามารถระบุค่าที่ต้องการได้ กำหนดค่าที่ต้องการทั้งในกล่องข้อความของ x และ y จากนั้นให้กดปุ่ม Cal เพื่อคำนวณผล



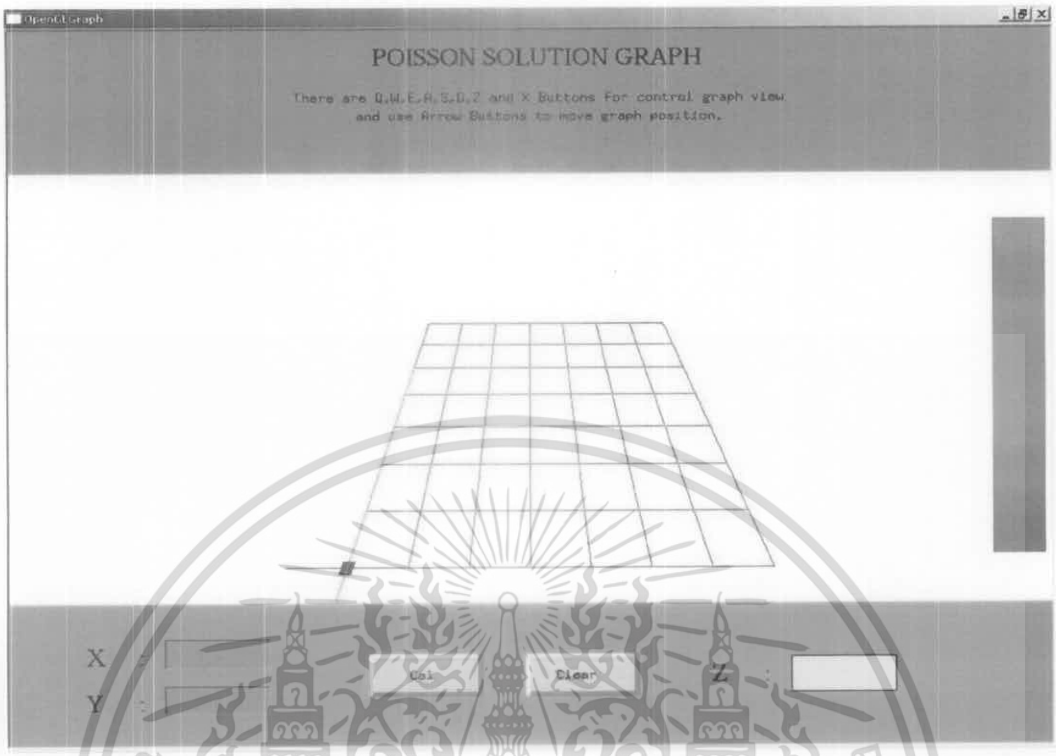
รูปที่ 4.3 แสดงผลลัพธ์ที่ได้จากการคำนวณ

เมื่อทำการกดปุ่ม Cal แล้ว โปรแกรมจะคำนวณผลลัพธ์หรือค่า z ออกมาให้เราในกล่องข้อความสีเทาด้านขวา โดยที่ในรูปที่ 4.3 โปรแกรมจะมีจุดสีดำแสดงตำแหน่งของ x และ y ที่กำหนดให้ว่าอยู่ตำแหน่งใดในกราฟ

หากต้องการระบุค่าเพื่อคำนวณใหม่ ให้กดปุ่ม Clear โปรแกรมจะล้างข้อความในกล่องข้อความทั้งหมดให้ และพร้อมที่จะรับค่าเพื่อการคำนวณได้ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลลัพธ์ที่ได้จากโปรแกรม



รูปที่ 4.4 ภาพแสดงผลลัพธ์ด้วยกราฟจากสมการลาปลาซ

จากรูปที่ 4.4 เป็นผลลัพธ์ที่ได้จากการคำนวณด้วยสมการลาปลาซผ่านฟังก์ชันของโปรแกรม โดยจะเห็นว่ารูปที่ได้มีขนาดในแต่ละด้านเป็น 7x7 ซึ่งจุดตัดจะเป็นจุดที่ระบุค่าผลลัพธ์ที่ได้ว่า ณ จุดนั้นมีอุณหภูมิเท่าไหร่ ซึ่งในปัญหานี้เราจะกำหนดให้ที่ขอบทุกด้านของปัญหาเป็น 0 ด้วยเงื่อนไขแบบ Dirichlet

ดังนั้น เมื่อไม่มีความร้อนหรือค่าใด ๆ ในเนื้อวัตถุ ค่าผลลัพธ์จึงเป็น 0 ทุกจุด และผลลัพธ์ที่ได้ก็จะเป็นดังกราฟข้างต้น

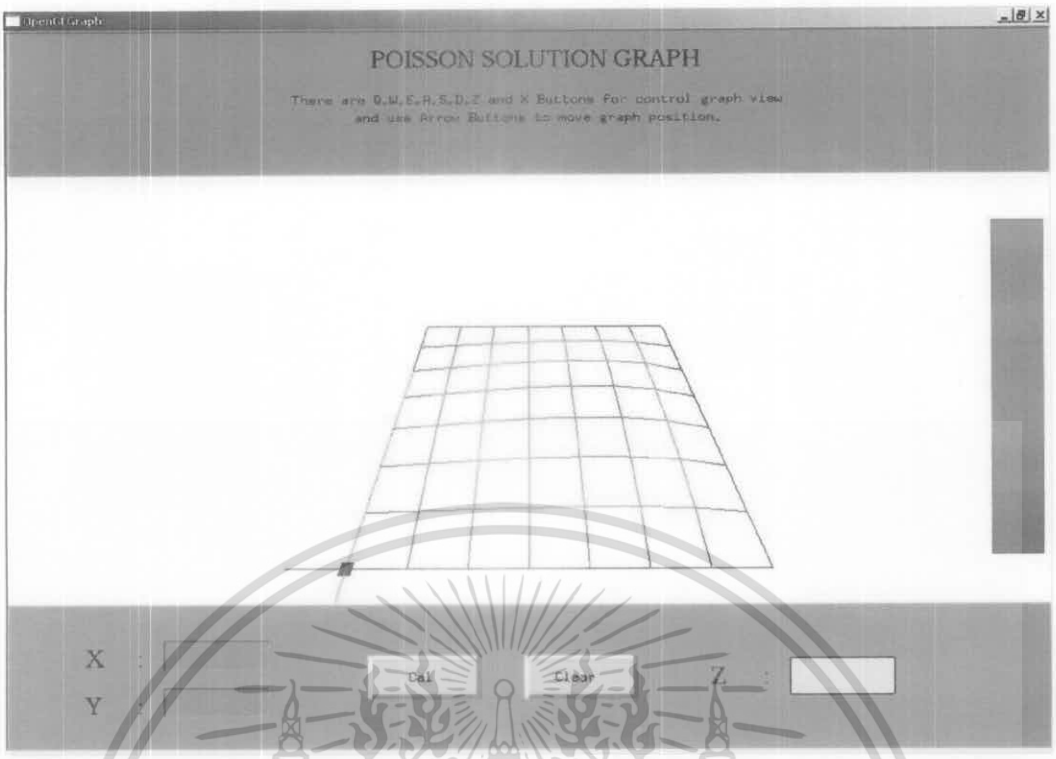
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 ภาพแสดงผลลัพธ์ด้วยกราฟจากสมการปัวส์ซองโดยระบุฟังก์ชันทางด้านขวาของสมการเป็นค่าคงที่

จากรูปที่ 4.5 เป็นผลลัพธ์ที่ได้จากการคำนวณด้วยสมการปัวส์ซองโดยระบุฟังก์ชันทางด้านขวาของสมการเป็นค่าคงที่ โดยใช้ฟังก์ชันของโปรแกรม โดยจะเห็นว่ารูปที่ได้มีขนาดในแต่ละด้านเป็น 7x7 ซึ่งจุดตัดจะเป็นจุดที่ระบุค่าผลลัพธ์ที่ได้ว่า ณ จุดนั้นมีอุณหภูมิเท่าไหร่ ซึ่งในปัญหานี้เราจะกำหนดให้ที่ขอบทุกด้านของปัญหาเป็น 0 ด้วยเงื่อนไขแบบ Dirichlet

ดังนั้น เมื่อมีการกำหนดฟังก์ชันทางด้านขวาให้กับสมการ ก็เปรียบเหมือนกับว่าเราได้ใส่ความร้อนเข้าไปยังเนื้อวัตถุ โดยความร้อนจากภายในก็จะแผ่ออกไปยังขอบทั้ง 4 ด้านของวัตถุ โดยมีรูปการแผ่ความร้อนเป็นพาราโบลา ดังแสดงตามกราฟข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



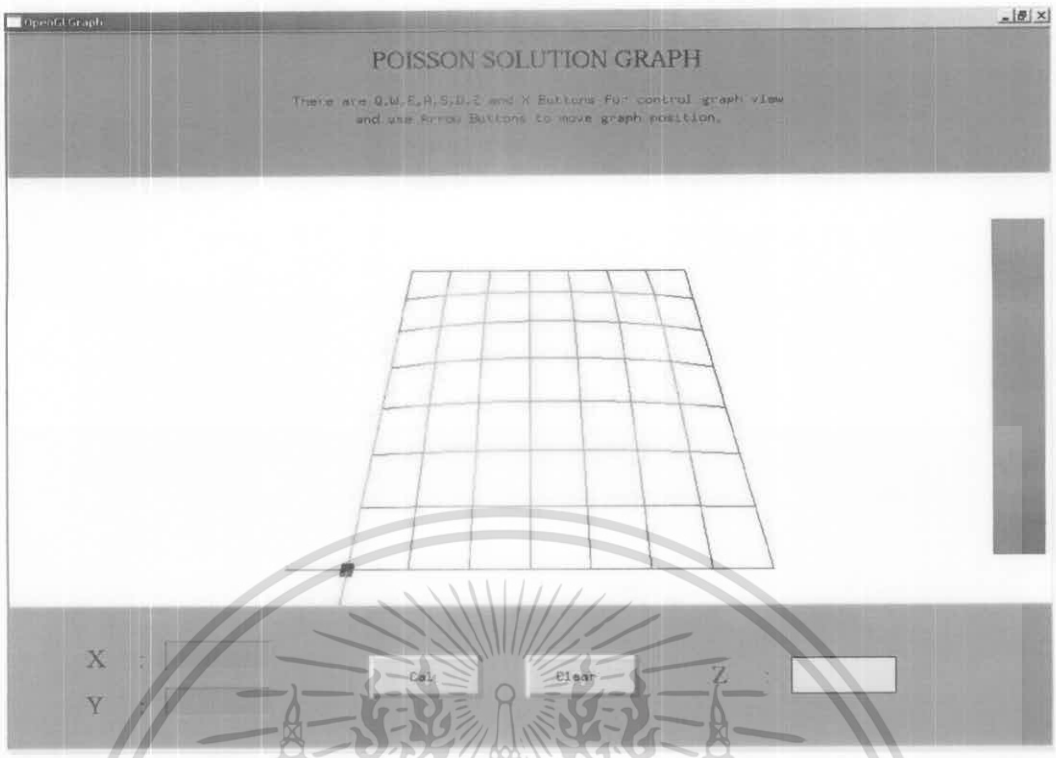
รูปที่ 4.6 ภาพแสดงผลลัพธ์ด้วยกราฟจากสมการปัวส์ซองโดยระบุฟังก์ชันทางด้านขวาของสมการเป็นฟังก์ชัน x

จากรูปที่ 4.6 เป็นผลลัพธ์ที่ได้จากการคำนวณด้วยสมการปัวส์ซอง โดยระบุฟังก์ชันทางด้านขวาของสมการเป็นฟังก์ชัน x โดยจะเห็นว่ารูปที่ได้มีขนาดในแต่ละด้านเป็น 7x7 ซึ่งจุดตัดจะเป็นจุดที่ระบุค่าผลลัพธ์ที่ได้ว่า ณ จุดนั้นมีอุณหภูมิเท่าไหร่ ซึ่งในปัญหานี้เราจะกำหนดให้ที่ขอบทุกด้านของปัญหาเป็น 0 ด้วยเงื่อนไขแบบ Dirichlet

ดังนั้น เมื่อกำหนดฟังก์ชันทางด้านขวาเป็นฟังก์ชันของ x จะมีความหมายว่าในเนื้อวัตถุจะมีความร้อนแฝงอยู่ในแต่ละจุดไม่เท่ากัน โดยขึ้นกับตำแหน่งซึ่งถ้าค่า x น้อย ๆ ก็จะมีค่าน้อย และมากขึ้นเมื่อค่า x มากขึ้น ดังที่เห็นตามกราฟ

โดยสมการปัวส์ซองนี้เป็นการศึกษาลักษณะของฟังก์ชันทางด้านขวาที่แปรผันตามเทอมของ x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ภาพแสดงผลลัพธ์ด้วยกราฟจากสมการปัวส์ซองโดยระบุฟังก์ชันทางด้านขวาของสมการเป็นฟังก์ชัน y

จากรูปที่ 4.7 เป็นผลลัพธ์ที่ได้จากการคำนวณด้วยสมการปัวส์ซองโดยระบุฟังก์ชันทางด้านขวาของสมการเป็นฟังก์ชันของ y โดยจะเห็นว่ารูปที่ได้มีขนาดในแต่ละด้านเป็น 7x7 ซึ่งจุดตัดจะเป็นจุดที่ระบุค่าผลลัพธ์ที่ได้ว่า ณ จุดนั้นมีอุณหภูมิเท่าไหร่ ซึ่งในปัญหานี้เราจะกำหนดให้ที่ขอบทุกด้านของปัญหาเป็น 0 ด้วยเงื่อนไขแบบ Dirichlet

ดังนั้น เมื่อกำหนดฟังก์ชันทางด้านขวาเป็นฟังก์ชันของ y จะมีความหมายว่าในเนื้อวัตถุจะมีความร้อนแฝงอยู่ในแต่ละจุดไม่เท่ากัน โดยขึ้นกับตำแหน่งซึ่งถ้าค่า y น้อย ๆ ก็จะมีค่าน้อย และมากขึ้นเมื่อค่า y มากขึ้น ดังที่เห็นตามกราฟ

โดยสมการปัวส์ซองนี้เป็นการศึกษาลักษณะของฟังก์ชันทางด้านขวาที่แปรผันตามเทอมของ y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

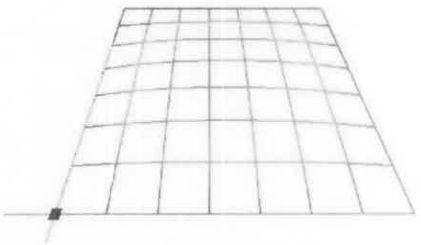
รูปที่ 4.8 ภาพแสดงผลลัพธ์ด้วยกราฟจากสมการปัวส์ซองโดยระบุฟังก์ชันทางด้านขวาของสมการเป็นฟังก์ชัน xy

จากรูปที่ 4.8 เป็นผลลัพธ์ที่ได้จากการคำนวณด้วยสมการปัวส์ซองโดยระบุฟังก์ชันทางด้านขวาของสมการเป็นฟังก์ชันของ xy โดยจะเห็นว่ารูปที่ได้มีขนาดในแต่ละด้านเป็น 7x7 ซึ่งจุดตัดจะเป็นจุดที่ระบุค่าผลลัพธ์ที่ได้ว่า ณ จุดนั้นมีอุณหภูมิเท่าไหร่ ซึ่งในปัญหานี้เราจะกำหนดให้ที่ขอบทุกด้านของปัญหาเป็น 0 ด้วยเงื่อนไขแบบ Dirichlet

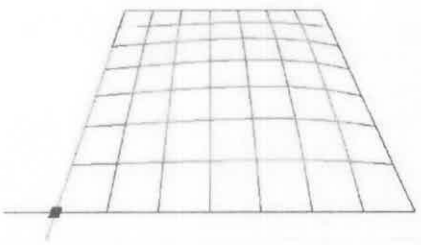
ดังนั้น เมื่อกำหนดฟังก์ชันทางด้านขวาเป็นฟังก์ชันของ xy จะมีความหมายว่าในเนื้อวัตถุจะมีความร้อนแฝงอยู่ในแต่ละจุดไม่เท่ากัน โดยขึ้นกับตำแหน่งซึ่งถ้าค่า xy น้อย ๆ ก็จะมีค่าน้อย และมากขึ้นเมื่อค่า xy มากขึ้น ดังที่เห็นตามกราฟ

โดยสมการปัวส์ซองนี้เป็นการศึกษาลักษณะของฟังก์ชันทางด้านขวาที่แปรผันตามเทอมของ x และ y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$\nabla^2\phi = const$



$\nabla^2\phi = x$

$\nabla^2\phi = y$

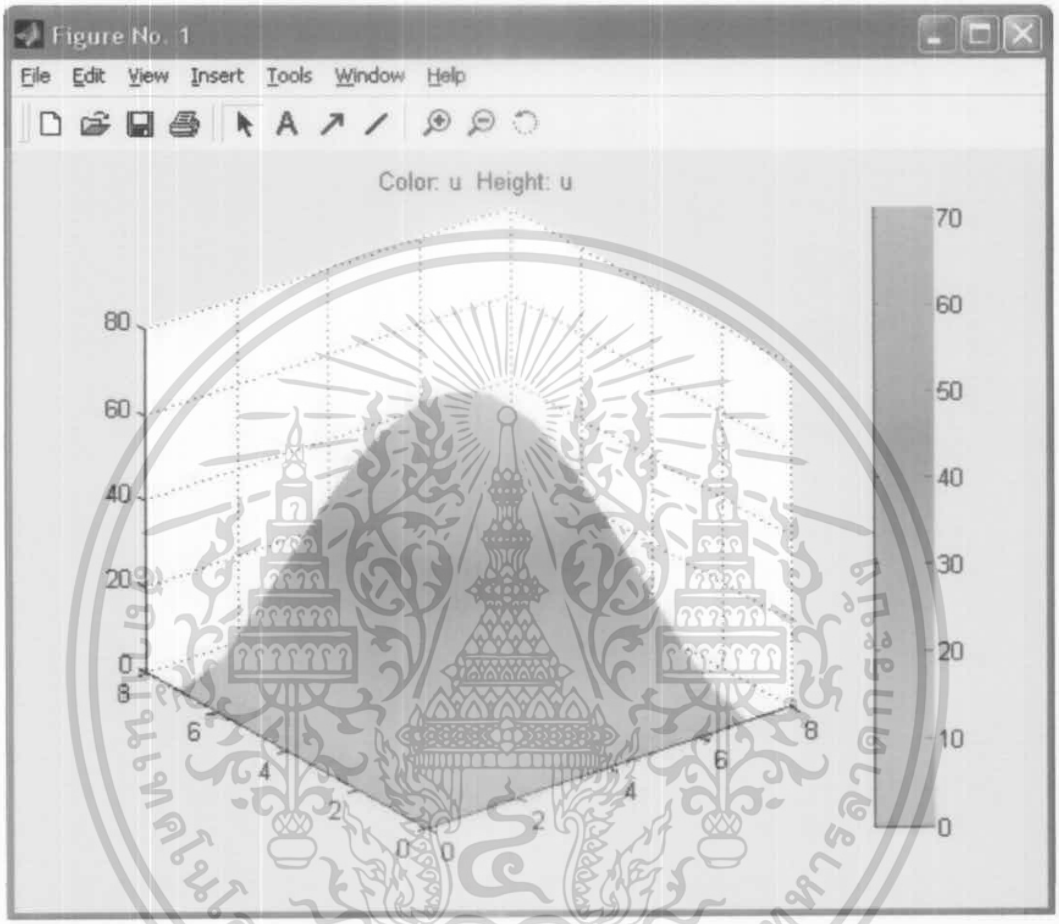
$\nabla^2\phi = xy$

รูปที่ 4.9 แสดงการเปรียบเทียบกราฟผลลัพธ์จากรูปสมการ 4 แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การประเมินประสิทธิภาพของโปรแกรม

เพื่อพิสูจน์ให้เห็นว่าค่าผลลัพธ์ที่ได้จากการคำนวณโดยโปรแกรมที่ได้พัฒนาขึ้นมานี้มาความน่าเชื่อถือ จึงได้ทำการเปรียบเทียบผลลัพธ์กับโปรแกรมที่เป็นที่ยอมรับกันอย่างกว้างขวาง นั่นคือโปรแกรม MATLAB ซึ่งให้ผลลัพธ์ดังต่อไปนี้

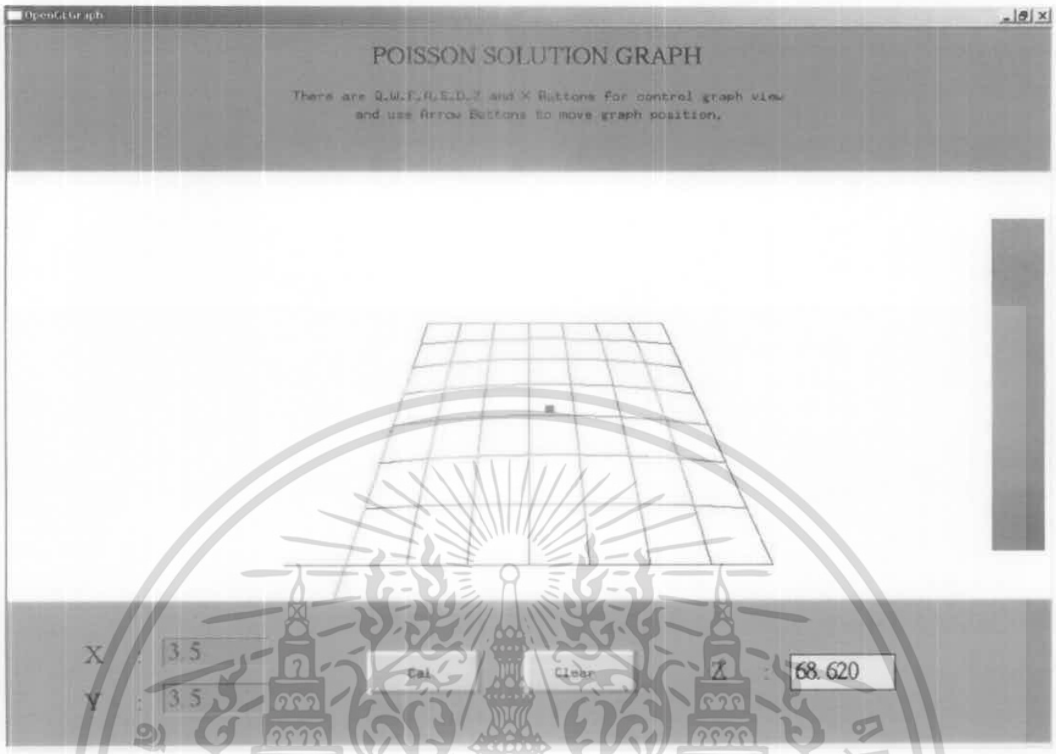


รูปที่ 4.10 ภาพแสดงผลลัพธ์เป็นกราฟสามมิติที่ได้จากโปรแกรม MATLAB โดยที่ฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่าเท่ากับ 20

รูปที่ 4.10 แสดงผลลัพธ์ที่คำนวณได้จากโปรแกรม MATLAB เป็นกราฟสามมิติ ที่มีขนาด 7x7 โดยมีฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่าเท่ากับ 20 และมีค่าสูงสุดประมาณ 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.1 ฟังก์ชันทางด้านขวาเป็นค่าคงที่



รูปที่ 4.11 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่าเท่ากับ 20 โดยมีความละเอียดเท่ากับ 7x7

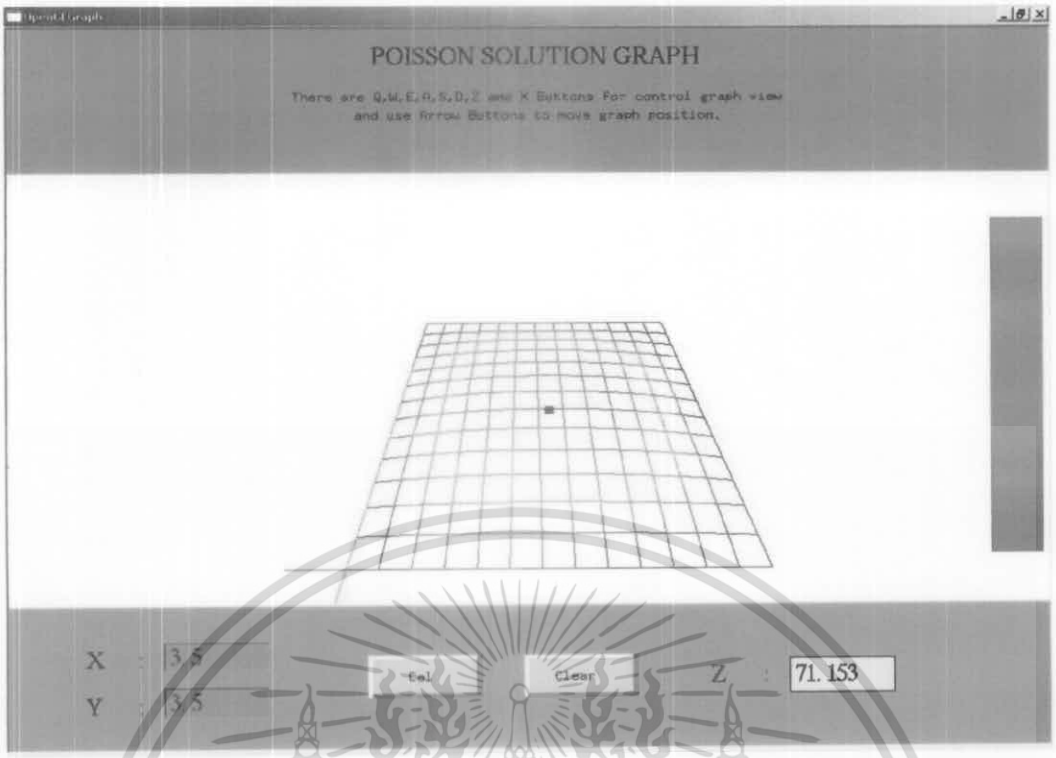
จากรูปที่ 4.11 กำหนดให้ฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่ากับ 20 มีความละเอียดเท่ากับ 7x7 ขอบทั้งสี่ด้านเป็นแบบ Dirichlet และเพื่อที่จะทำการเปรียบเทียบกับโปรแกรม MATLAB ที่มีขนาด 7x7 เช่นเดียวกัน จึงได้กำหนดให้ระยะห่างระหว่างจุดต่อ เป็น 1 ทำให้ได้ค่าสูงสุดที่ใช้ในการเปรียบเทียบกับโปรแกรม MATLAB คือ 68.620

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.12 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่าเท่ากับ 20 โดยมีความละเอียดเท่ากับ 9x9

จากรูปที่ 4.12 กำหนดให้ฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่ากับ 20 มีความละเอียดเท่ากับ 9x9 ขอบทั้งสี่ด้านเป็นแบบ Dirichlet และเพื่อที่จะทำการเปรียบเทียบกับโปรแกรม MATLAB ที่มีขนาด 7x7 จึงต้องกำหนดให้ระยะห่างระหว่างจุดต่อเป็น 0.7778 ทำให้ได้ค่าสูงสุดที่ใช้ในการเปรียบเทียบกับโปรแกรม MATLAB คือ 70.018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่าเท่ากับ 20 โดยมีความละเอียดเท่ากับ 13x13

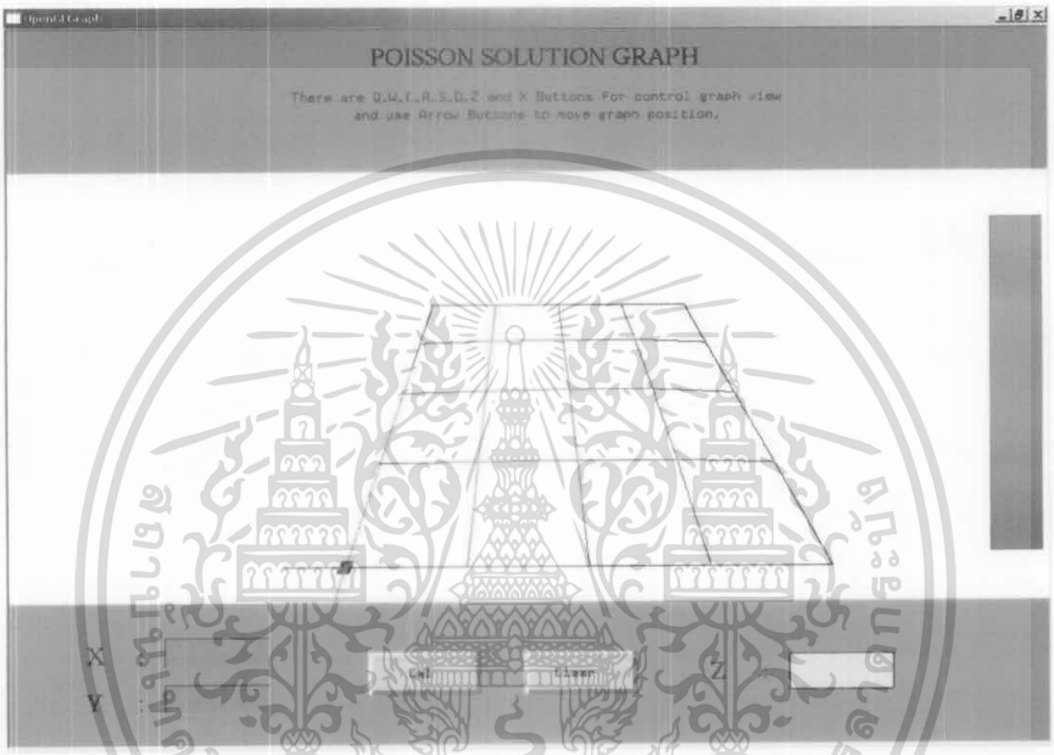
จากรูปที่ 4.13 กำหนดให้ฟังก์ชันทางด้านขวาเป็นค่าคงที่ มีค่ากับ 20 มีความละเอียดเท่ากับ 13x13 ขอบทั้งสี่ด้านเป็นแบบ Dirichlet และเพื่อที่จะทำการเปรียบเทียบกับโปรแกรม MATLAB ที่มีขนาด 7x7 จึงต้องกำหนดให้ระยะห่างระหว่างจุดต่อ เป็น 0.5385 ทำให้ได้ค่าสูงสุดที่ใช้ในการเปรียบเทียบกับโปรแกรม MATLAB คือ 71.153

จากรูปที่ 4.10 ถึงรูปที่ 4.13 จะเห็นได้ว่า โปรแกรมที่ได้ทำการพัฒนาขึ้นมานี้เมื่อทำการเพิ่มจำนวนจุดต่อให้มีความละเอียดมากขึ้น ค่าคำตอบที่ได้เมื่อนำมาเปรียบเทียบกับโปรแกรม MATLAB ซึ่งเป็นโปรแกรมที่เป็นที่ยอมรับกันอย่างกว้างขวางนั้น ค่าผลลัพธ์ที่ได้จากโปรแกรมทั้งสองมีค่าใกล้เคียงกัน จึงอนุมานได้ว่าโปรแกรมที่ได้พัฒนาขึ้นมานี้มีความน่าเชื่อถือ โดยโปรแกรมที่ได้พัฒนานี้ เป็นเพียงบางส่วนของฟังก์ชันในโปรแกรม MATLAB จึงมีขนาดที่เล็กกว่าโปรแกรม MATLAB มาก จึงเหมาะกับลักษณะของปัญหาที่ต้องอยู่ในรูปแบบของสมการปัวส์ซอง โดยไม่จำเป็นต้องใช้ทุกฟังก์ชันในโปรแกรม MATLAB ซึ่งจะเป็นการประหยัดเนื้อที่ในหน่วยความจำอย่างมากและเป็นการประหยัดค่าใช้จ่ายในเรื่องของค่าลิขสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสิ่งที่โปรแกรมที่พัฒนาขึ้นมานี้แตกต่างจากโปรแกรม MATLAB นั่นก็คือ โปรแกรม MATLAB ไม่สามารถที่จะระบุให้ฟังก์ชันทางด้านขวาแปรผันตามเทอมของ x , y และ xy ได้ และไม่สามารถกำหนดระยะห่างระหว่างจุดต่อทั้งในแนวแกน x และแกน y ได้ ซึ่งโปรแกรมที่พัฒนาขึ้นมานี้สามารถทำได้

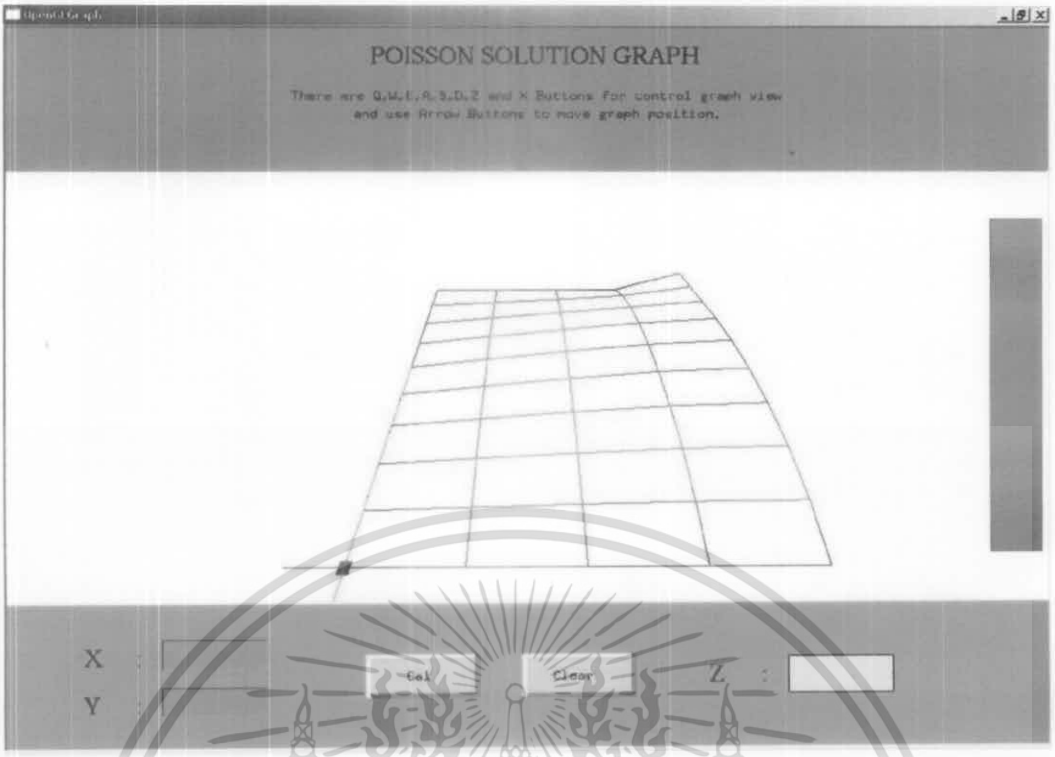
4.4.2 ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x



รูปที่ 4.14 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 4x4

จากรูปที่ 4.14 กำหนดให้ฟังก์ชันทางด้านขวาเป็นเป็นฟังก์ชัน x มีความละเอียดเท่ากับ 4x4 ขอบทั้งสี่ด้านเป็นแบบ Dirichlet

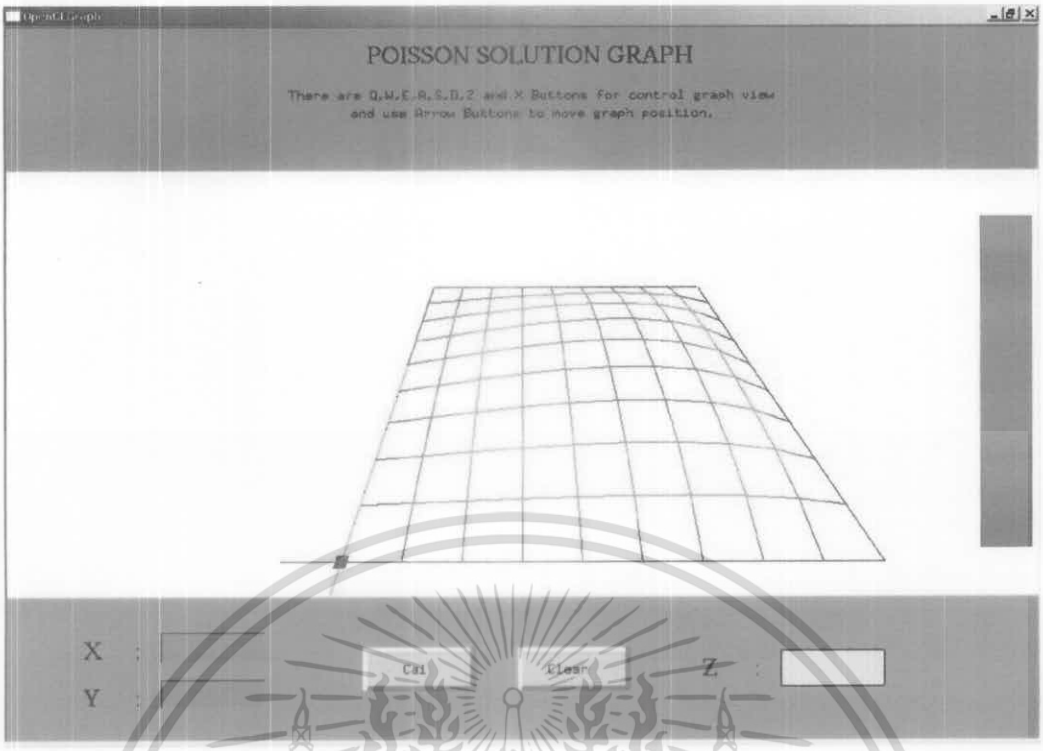
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 4x9

จากรูปที่ 4.15 กำหนดให้ฟังก์ชันทางด้านขวาเป็นเป็นฟังก์ชัน x มีความละเอียดเท่ากับ 4x9 กำหนดให้ขอบทางด้านขวาเป็นแบบ Neuman ส่วนขอบอีกสามด้านเป็นแบบ Dirichlet

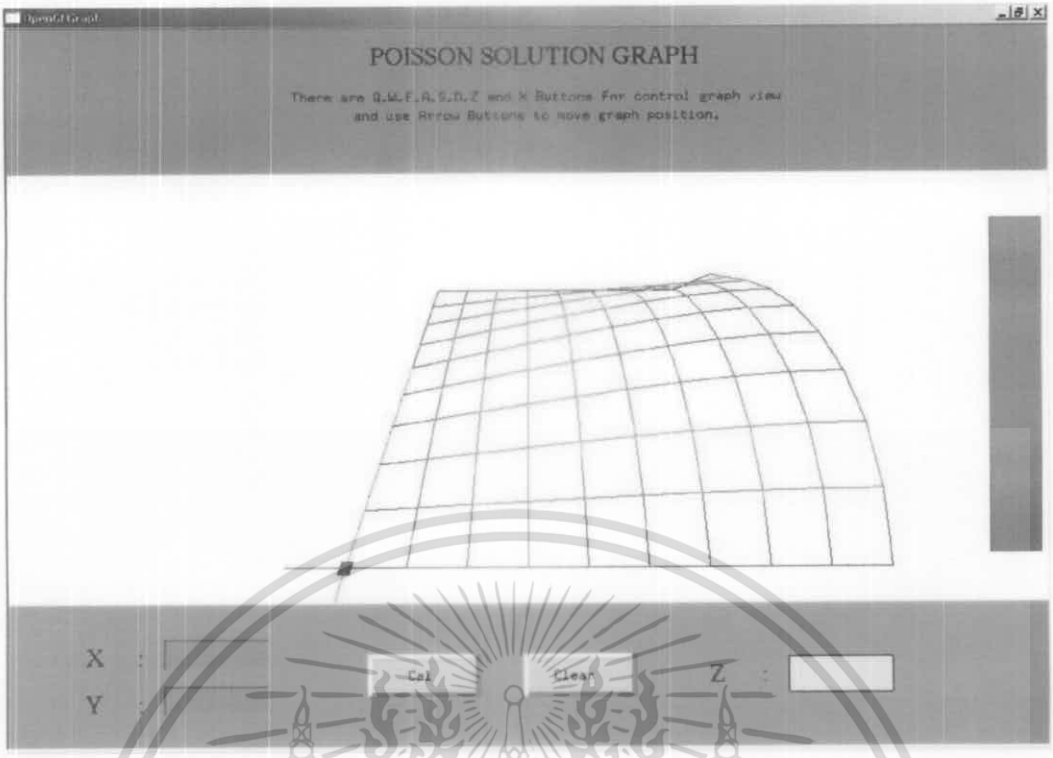
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 9x9

จากรูปที่ 4.16 กำหนดให้ฟังก์ชันทางด้านขวาเป็นเป็นฟังก์ชัน x มีความละเอียดเท่ากับ 9x9 ขอบทั้งสี่ด้านเป็นแบบ Dirichlet

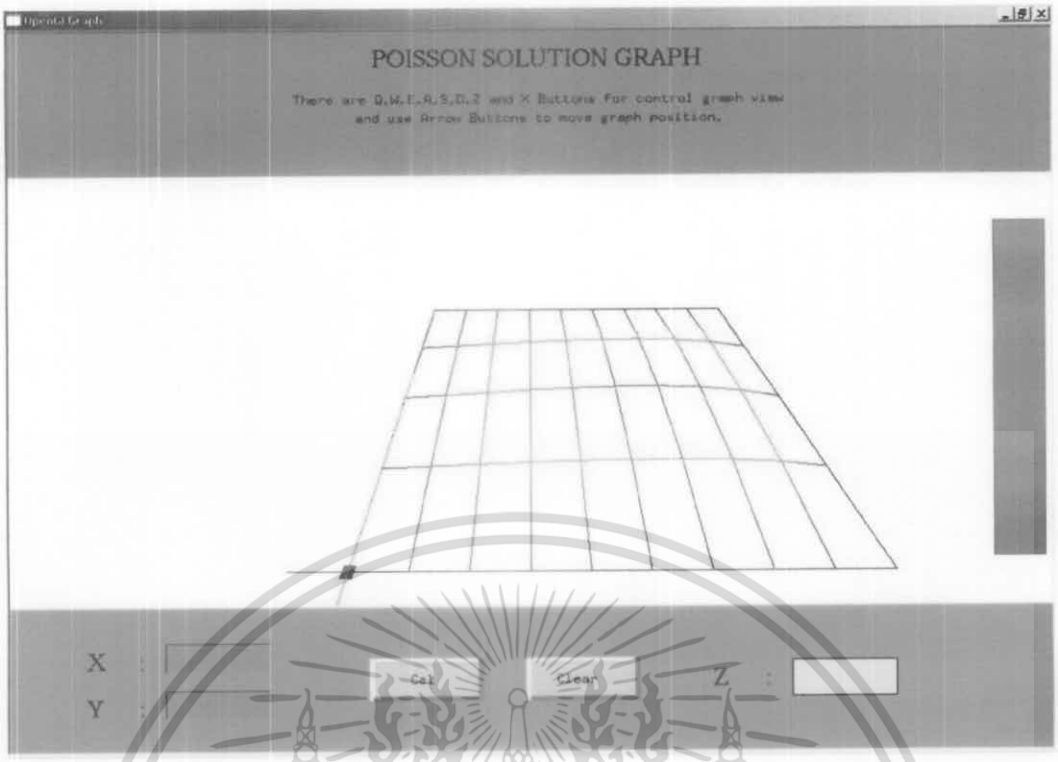
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 9x9

จากรูปที่ 4.17 กำหนดให้ฟังก์ชันทางด้านขวาเป็นเป็นฟังก์ชัน x มีความละเอียดเท่ากับ 9x9 กำหนดให้ขอบทางด้านขวาเป็นแบบ Neuman ส่วนขอบอีกสามด้านเป็นแบบ Dirichlet

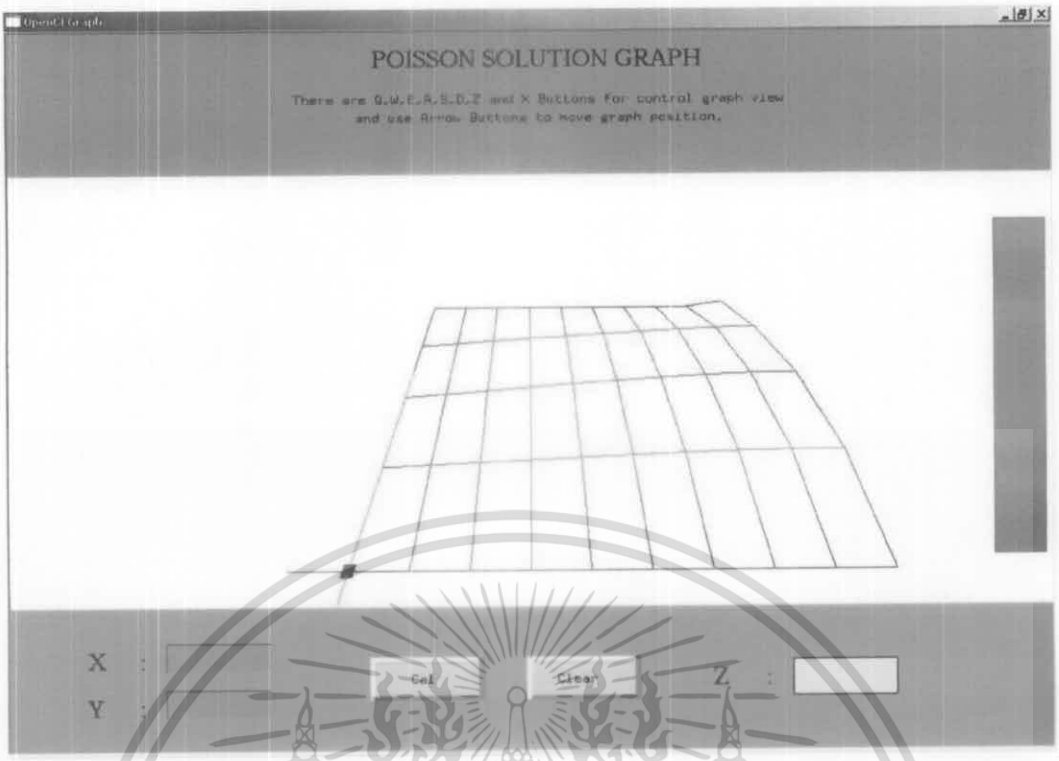
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 9x4

จากรูปที่ 4.18 กำหนดให้ฟังก์ชันทางด้านขวาเป็นเป็นฟังก์ชัน x มีความละเอียดเท่ากับ 9x4 ขอบทั้งสี่ด้านเป็นแบบ Dirichlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



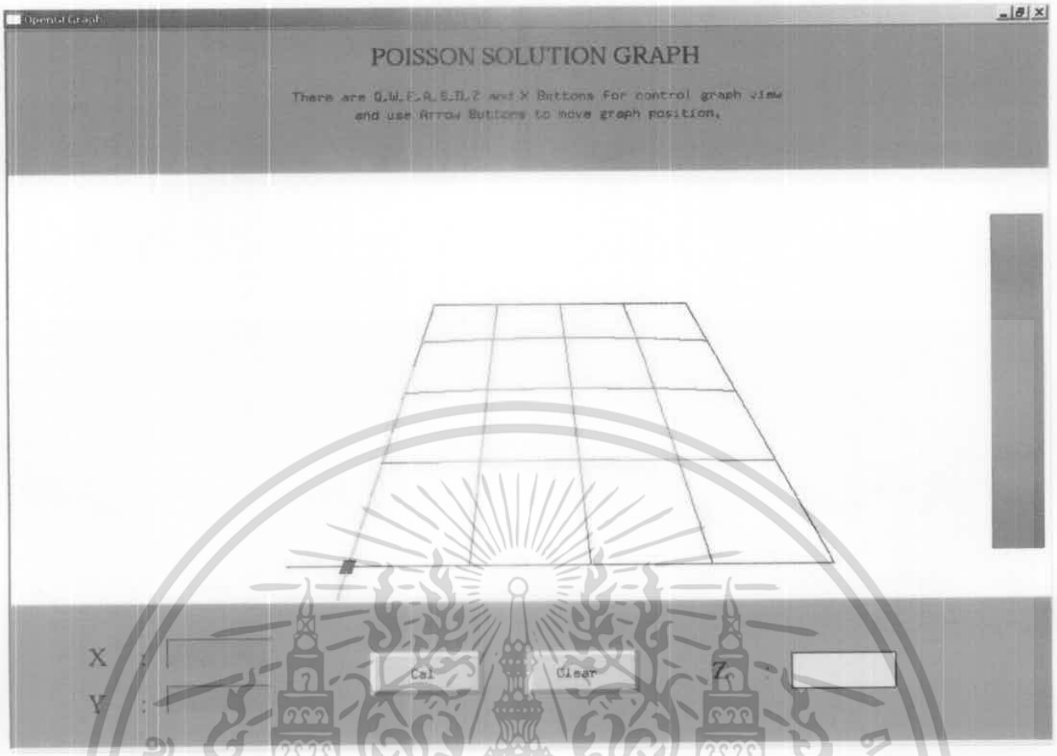
รูปที่ 4.19 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยมีความละเอียดเท่ากับ 9x4

จากรูปที่ 4.19 กำหนดให้ฟังก์ชันทางด้านขวาเป็นเป็นฟังก์ชัน x มีความละเอียดเท่ากับ 9x4 กำหนดให้ขอบทางด้านขวาเป็นแบบ Neuman ส่วนขอบอีกสามด้านเป็นแบบ Dirichlet

รูปที่ 4.14 ถึงรูปที่ 4.19 นั้นแสดงให้เห็นถึงกราฟผลลัพธ์โดยที่กำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน x โดยกราฟแสดงผลนั้นจะมีการใช้สีในการบ่งบอกถึงปริมาณความร้อนซึ่งจะไล่จากความร้อนสูงไปความร้อนต่ำ โดยสีแดงจะแสดงว่าบริเวณนั้นมีความร้อนสูงส่วนสีน้ำเงินจะแสดงว่าบริเวณนั้นมีความร้อนต่ำ จากรูปจะเห็นได้ว่าเมื่อค่า x มีค่าเพิ่มขึ้นบริเวณที่เป็นสีแดงก็จะเปลี่ยนแปลงไปตามค่า x ที่พิ่มขึ้นด้วย และยังแสดงให้เห็นว่าโปรแกรมที่พัฒนาขึ้นมานี้สามารถคำนวณและแสดงผลลัพธ์ที่มีขนาด mxm และ mxn ได้ และสามารถกำหนดให้ค่าที่ขอบของปัญหาเป็นแบบ Neuman และ Dirichlet ได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

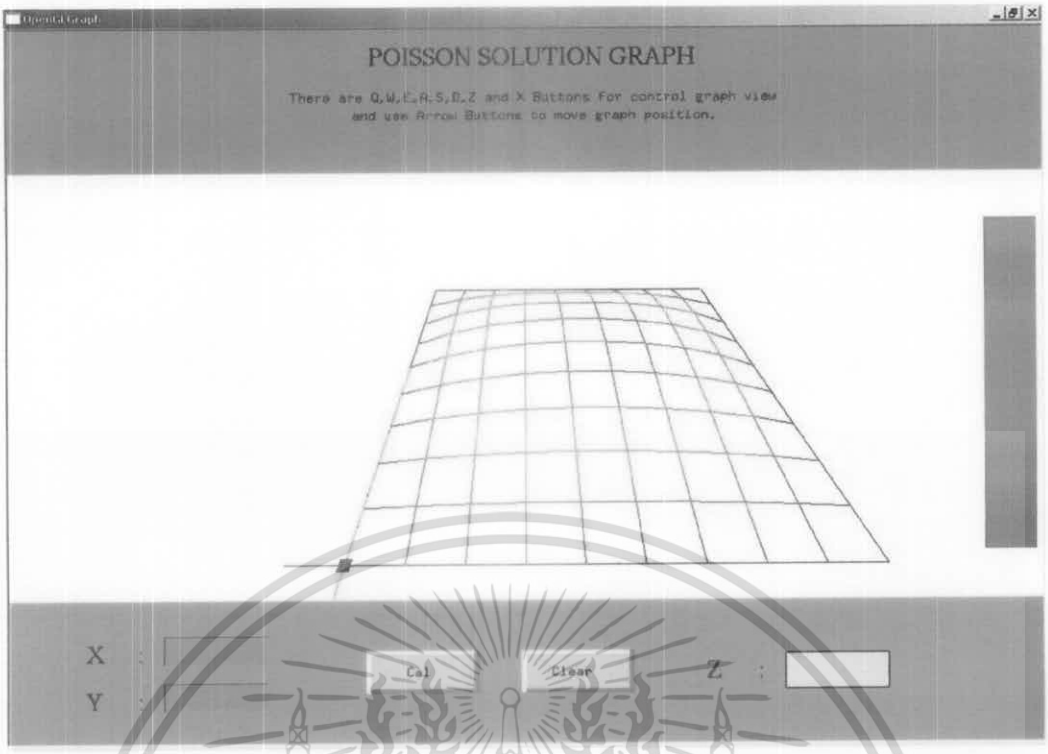
4.4.3 ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน y



รูปที่ 4.20 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน y โดยมีความละเอียดเท่ากับ 4x4

จากรูปที่ 4.20 กำหนดให้ฟังก์ชันทางด้านขวาเป็นเป็นฟังก์ชัน y มีความละเอียดเท่ากับ 4x4 ขอบทั้งสี่ด้านเป็นแบบ Dirichlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



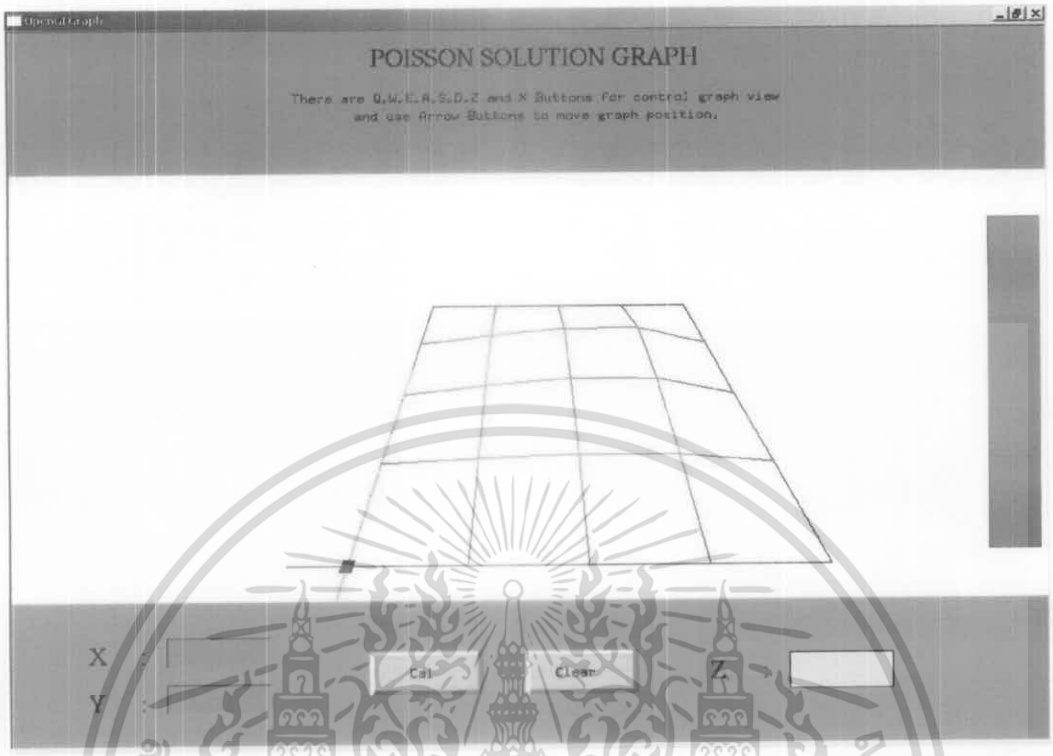
รูปที่ 4.21 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน y โดยมีความละเอียดเท่ากับ 9x9

จากรูปที่ 4.21 กำหนดให้ฟังก์ชันทางด้านขวาเป็นเป็นฟังก์ชัน y มีความละเอียดเท่ากับ 9x9 ขอบทั้งสี่ด้านเป็นแบบ Dirichlet

รูปที่ 4.20 และรูปที่ 4.21 นั้นแสดงให้เห็นถึงกราฟผลลัพธ์โดยที่กำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน y โดยกราฟแสดงผลนั้นจะมีการใช้สีในการบ่งบอกถึงปริมาณความร้อนซึ่งจะไล่จากความร้อนสูงไปความร้อนต่ำ โดยสีแดงจะแสดงว่าบริเวณนั้นมีความร้อนสูงส่วนสีน้ำเงินจะแสดงว่าบริเวณนั้นมีความร้อนต่ำ จากรูปจะเห็นได้ว่าเมื่อค่า y มีค่าเพิ่มขึ้นบริเวณที่เป็นสีแดงก็จะเปลี่ยนแปลงไปตามค่า y ที่พิ่มขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

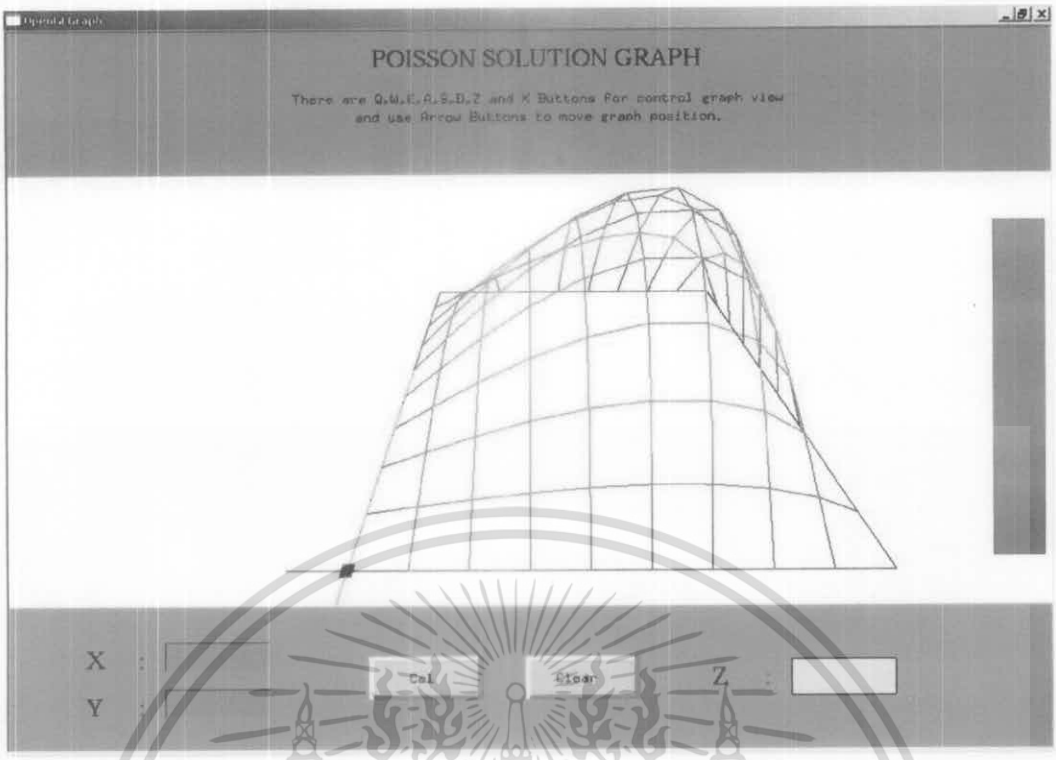
4.4.4 ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน xy



รูปที่ 4.22 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน xy โดยมีความละเอียดเท่ากับ 4x4

จากรูปที่ 4.22 กำหนดให้ฟังก์ชันทางด้านขวาเป็นเป็นฟังก์ชัน xy มีความละเอียดเท่ากับ 4x4 ขอบทั้งสี่ด้านเป็นแบบ Dirichlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 แสดงผลลัพธ์ของกราฟเมื่อกำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน xy โดยมีความละเอียดเท่ากับ 9x9

จากรูปที่ 4.23 กำหนดให้ฟังก์ชันทางด้านขวาเป็นเป็นฟังก์ชัน xy มีความละเอียดเท่ากับ 9x9 ขอบทั้งสี่ด้านเป็นแบบ Dirichlet

รูปที่ 4.22 และรูปที่ 4.23 นั้นแสดงให้เห็นถึงกราฟผลลัพธ์โดยที่กำหนดให้ฟังก์ชันทางด้านขวาเป็นฟังก์ชัน xy โดยกราฟแสดงผลนั้นจะมีการใช้สีในการบ่งบอกถึงปริมาณความร้อนซึ่งจะไล่จากความร้อนสูงไปความร้อนต่ำ โดยสีแดงจะแสดงว่าบริเวณนั้นมีความร้อนสูงส่วนสีน้ำเงินจะแสดงว่าบริเวณนั้นมีความร้อนต่ำ จากรูปจะเห็นได้ว่าเมื่อค่า xy มีค่าเพิ่มขึ้นบริเวณที่เป็นสีแดงก็จะเปลี่ยน แปลงไปตามค่า xy ที่พิ่มขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุปผล

ในการทำงานทางด้านวิทยาศาสตร์นั้น จำเป็นอย่างยิ่งที่ต้องใช้สมการคณิตศาสตร์มาช่วยในการคำนวณเพื่อให้ได้ผลลัพธ์ที่สามารถนำไปอธิบายสภาพปัญหาต่าง ๆ ได้ ซึ่งการคำนวณทางด้านคณิตศาสตร์นั้นมีความซับซ้อนมากและมีขั้นตอนในการคำนวณหลายขั้นตอน เพื่อเป็นการลดความยุ่งยากในการคำนวณทางด้านคณิตศาสตร์ จึงได้ทำการพัฒนาโปรแกรมขึ้นมาเพื่อใช้ในการคำนวณหาผลเฉลยของสมการปัวส์ซองซึ่งเป็นรูปสมการหนึ่งของสมการทางคณิตศาสตร์ โดยใช้ระเบียบวิธีไฟไนต์เอลิเมนต์ในการหาผลเฉลยโดยโปรแกรมคอมพิวเตอร์ และได้จัดทำโครงงานนี้ไว้เพื่อให้บุคคลที่สนใจหรือมีความจำเป็นต้องใช้สมการปัวส์ซองในการแก้ปัญหา สามารถนำโครงงานไปศึกษาและพัฒนาต่อยอดได้

โครงงานนี้ได้ทำการศึกษาสมการปัวส์ซองและใช้ระเบียบวิธีไฟไนต์เอลิเมนท์ในการหาผลเฉลย แล้วพัฒนาโปรแกรมขึ้นมาเพื่อหาผลเฉลยดังกล่าว ซึ่งโปรแกรมที่พัฒนาขึ้นมาใช้รองรับปัญหาที่เป็นสี่เหลี่ยมมุมฉาก และมีการแสดงผลลัพธ์ด้วยกราฟฟิคให้เข้าใจง่าย รวมทั้งหาผลลัพธ์ ณ จุด (x,y) ใด ๆ ในขอบเขตปัญหาได้

ผลลัพธ์ที่ได้จากโปรแกรมเมื่อนำไปเปรียบเทียบกับโปรแกรม MATLAB ซึ่งเป็นซอฟต์แวร์ที่เป็นที่ยอมรับ จะมีความแตกต่างกันเล็กน้อย ขึ้นอยู่กับความละเอียดที่กำหนดให้แก่โปรแกรม

5.2 ข้อเสนอแนะ

1. เนื่องจากฟังก์ชันในการคำนวณในโปรแกรมที่ได้พัฒนาขึ้นมานั้น มีขอบเขตของปัญหาซึ่งทำได้เฉพาะกับขอบเขตของปัญหาที่เฉพาะ ขึ้นอยู่กับฟังก์ชันที่แปรผันไปของฟังก์ชันทางด้านขวาของสมการ ซึ่งต่อไปอาจจะพัฒนาให้ค่าทางด้านขวาของสมการเป็น Polynomial ที่มี n ดีกรี

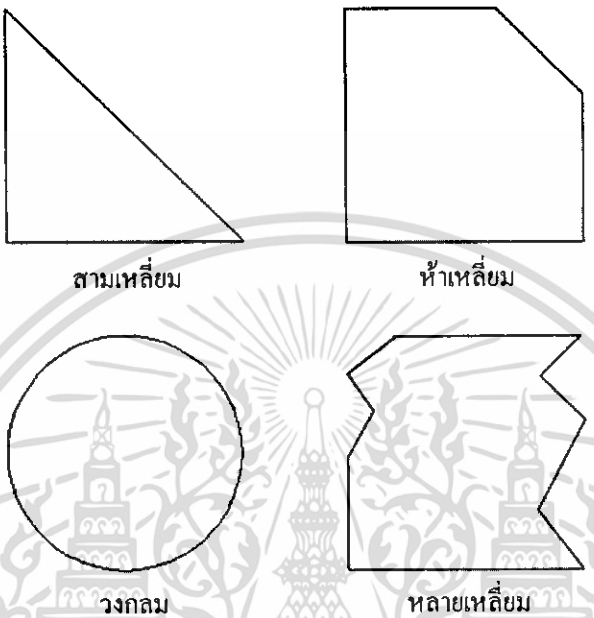
$$G_1 = x^2 + 2xy + y^2$$
$$G_2 = (x + y)^n$$

หรือ เพิ่มค่าทางด้านขวาของสมการเป็น ฟังก์ชันใด ๆ
หรือ เพิ่มค่าทางด้านขวาของสมการให้สอดคล้องกับปัญหาทางด้านวิทยาศาสตร์ เป็นต้น

2. ในเรื่องของการแบ่งเอลิเมนต์ย่อย ๆ ก็ยังมีข้อจำกัดอยู่ เพราะฉะนั้นหากเป็นไปได้ในอนาคตสามารถนำโปรแกรมที่ถูกพัฒนานี้นำไปพัฒนาต่อให้เข้ากับลักษณะของขอบเขต

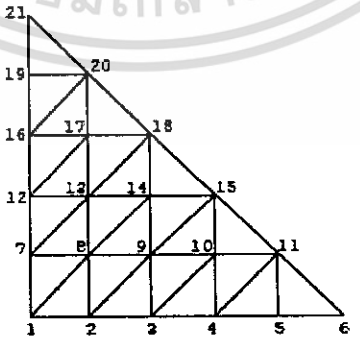
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาอื่น ๆ ได้โดยสามารถคำนวณได้หลายรูปแบบปัญหา อีกทั้งยังไม่ต้องมาเริ่มต้นศึกษาทฤษฎีใหม่ทั้งหมด ซึ่งในตอนนี้โปรแกรมที่พัฒนาขึ้นสามารถคำนวณหาผลลัพธ์ได้ในขอบเขตปัญหาที่เป็นสี่เหลี่ยมมุมฉากเท่านั้น ซึ่งอาจจะสามารถเพิ่มขอบเขตปัญหาให้อยู่ในรูปแบบของสามเหลี่ยม , ห้าเหลี่ยม , วงกลม และหลายเหลี่ยม



รูปที่ 5.1 แสดงลักษณะขอบเขตของปัญหาในรูปแบบต่าง ๆ

โดยวิธีการหาผลเฉลยของขอบเขตปัญหาที่เป็นสามเหลี่ยมนั้น จะมีหลักการคล้ายกับการหาผลเฉลยของขอบเขตปัญหาที่เป็นสี่เหลี่ยมมุมฉาก คือจะต้องทำการกำหนดหมายเลขจุดต่อแต่ละจุดต่อโดยเรียงตามลำดับจากซ้ายไปขวา และจากล่างขึ้นบนดังรูปที่ 5.2

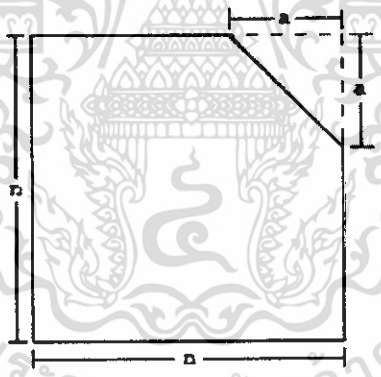


รูปที่ 5.2 แสดงวิธีการกำหนดหมายเลขให้จุดต่อแต่ละจุดต่อของขอบเขตปัญหาที่เป็นสามเหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

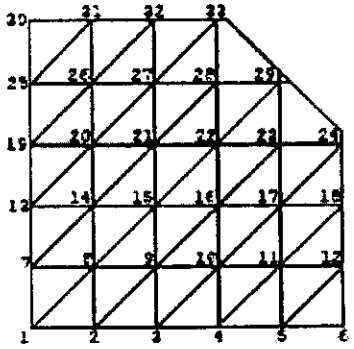
ซึ่งจะมีตัวแปรอาเรย์ชื่อ nodes เก็บค่าหมายเลขจุดต่อของแต่ละเอลิเมนต์ และสามารถกำหนดเงื่อนไขขอบเขตได้โดยใช้ตัวแปรอาเรย์ชื่อ bcdof (boundary condition degree of freedom) ไว้เก็บหมายเลขจุดต่อที่เป็นขอบของปัญหา และใช้ตัวแปรอาเรย์ชื่อ side_value เพื่อเก็บค่าเงื่อนไขขอบเขตของปัญหาแต่ละด้าน แล้วนำไปกำหนดให้แก่ตัวแปรอาเรย์ชื่อ bcval (boundary condition value) เพื่อกำหนดค่าเงื่อนไขขอบเขตแก่จุดต่อที่เป็นขอบของปัญหาในแต่ละจุดต่อ แต่สิ่งที่แตกต่างไปจากการหาผลเฉลยของขอบเขตปัญหาที่เป็นสี่เหลี่ยมมุมฉากนั่นก็คือ วิธีในการคำนวณการเก็บเลขลำดับของแต่ละจุดต่อ และการหาจุดต่อที่เป็นขอบของปัญหา

สำหรับวิธีการหาผลเฉลยของขอบเขตปัญหาที่เป็นห้าเหลี่ยมนั้น ก็มีหลักการคล้ายกับการหาผลเฉลยของขอบเขตปัญหาที่เป็นสี่เหลี่ยมมุมฉากดังที่ได้กล่าวไว้แล้วข้างต้น แต่จะมีจุดที่แตกต่างตรงที่ วิธีการหาผลเฉลยของขอบเขตปัญหาที่เป็นห้าเหลี่ยมนั้น จะมีการรับค่าตัวแปรอีกตัวนึงเข้ามา สมมุติว่าเป็นตัวแปร a ซึ่งตัวแปร a จะแสดงถึงจำนวนจุดต่อของพื้นที่สามเหลี่ยมที่ตัดออกไปซึ่งจะต้องมีค่าเท่ากัน ดังรูปที่ 5.3



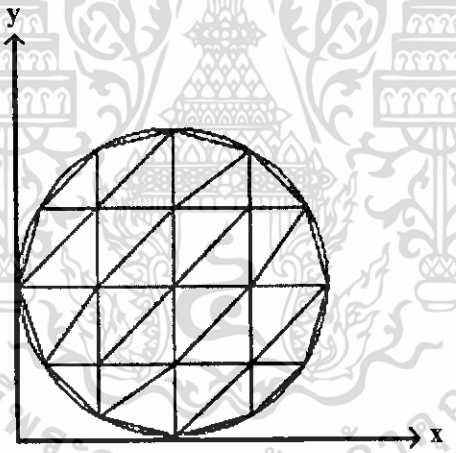
รูปที่ 5.3 แสดงส่วนพื้นที่สี่เหลี่ยมที่มุมด้านหนึ่งถูกตัดออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 แสดงวิธีการกำหนดหมายเลขให้จุดต่อแต่ละจุดต่อของขอบเขตปัญหาที่เป็นห้าเหลี่ยม

ส่วนวิธีการหาผลเฉลยของขอบเขตปัญหาที่เป็นวงกลม เราจะกำหนดให้ขอบเขตของปัญหาอยู่ชิดแนวเส้นสัมผัสในแนวแกน x และในแนวแกน y และทำการแบ่งพื้นที่ปัญหาออกเป็นเอลิเมนต์ย่อย ๆ ดังรูปที่ 5.5



รูปที่ 5.5 แสดงการแบ่งพื้นที่ขอบเขตปัญหาที่เป็นวงกลมออกเป็นเอลิเมนต์ย่อย ๆ

ซึ่งการหาผลเฉลยของขอบเขตปัญหาที่เป็นวงกลมนั้นทำได้ลำบาก จากรูปที่ 5.5 จะเห็นได้ว่าการคำนวณเพื่อกำหนดหมายเลขจุดต่อแต่ละจุดต่อ จะทำได้ไม่ง่ายเท่ากับขอบเขตปัญหาที่ได้นำเสนอไว้ ดังนั้นในอนาคตหากมีผู้สนใจจะพัฒนาในรูปแบบขอบเขตปัญหาที่เป็นวงกลม ก็สามารถนำโครงงานนี้ไปศึกษาและพัฒนาต่อไปได้

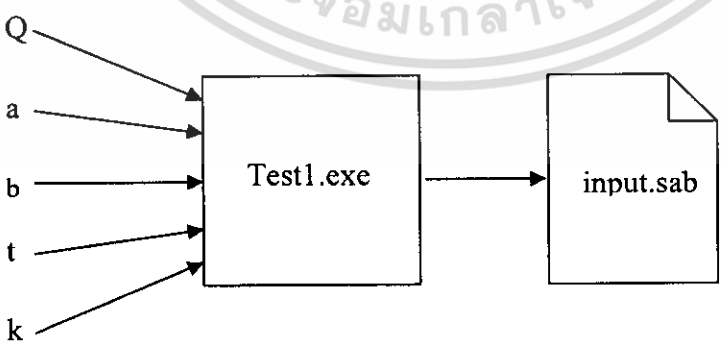
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่วนในเรื่องของกราฟฟิคนั้นก็สามารถพัฒนาให้รองรับกับขอบเขตของปัญหาในรูปแบบสามมิติได้ โดยผลลัพธ์จะอยู่ในมิติที่สี่ ซึ่งจะใช้สีในการแสดงผลลัพธ์ที่ให้ความหมายที่ทำให้เข้าใจได้ง่ายในแต่ละขอบเขตปัญหานั้น ๆ ต่างจากการแสดงผลแบบสองมิติ ที่ใช้กราฟแสดงผลลัพธ์ได้ชัดเจน

4. ในปัญหาพิเศษฉบับนี้ ควรจะเลือกใช้ภาษาซีในการพัฒนาเนื่องจากภาษาซีมีประสิทธิภาพในการประมวลได้เร็วกว่าภาษาอื่นโปรแกรมที่ได้พัฒนานี้ยังสามารถปรับปรุงประสิทธิภาพได้ดีขึ้นได้ อย่างเช่น
   - การใช้ชุดคำสั่งบางอย่างอาจเขียนด้วยการวนลูปแทนได้
   - ยังมีการประกาศตัวแปรมากเกินความจำเป็น
   - เมื่อประกาศตัวแปรแล้วไม่มีการคืนหน่วยความจำให้แก่ระบบ
   - การอ้างอิงถึงข้อมูลสามารถเปลี่ยนเป็นการอ้างอิงแบบพอยเตอร์ได้ซึ่งจะช่วยลดการใช้ทรัพยากร และทำให้โปรแกรมมีประสิทธิภาพสูงขึ้น

5. สามารถนำโปรแกรมนี้ไปพัฒนาต่อเพื่อให้ใช้กับปัญหาที่มีอยู่จริงได้ โดยลักษณะที่ต้องปรับปรุงให้เหมาะสมกับปัญหาที่จะนำไปใช้นั้น เราจะต้องทราบลักษณะฟังก์ชันทางด้านขวาของสมการ ตัวอย่างเช่น
   - ทางด้านการแพทย์ สามารถนำไปใช้ดูลักษณะการแตกหักของกระดูก
   - ทางด้านวิศวกรรม สามารถนำไปใช้ดูสภาพการเปลี่ยนแปลงของวัตถุเมื่อเกิดการชน หรือ ดูการเปลี่ยนแปลงสภาพการไหลของอากาศผ่านชิ้นวัตถุ

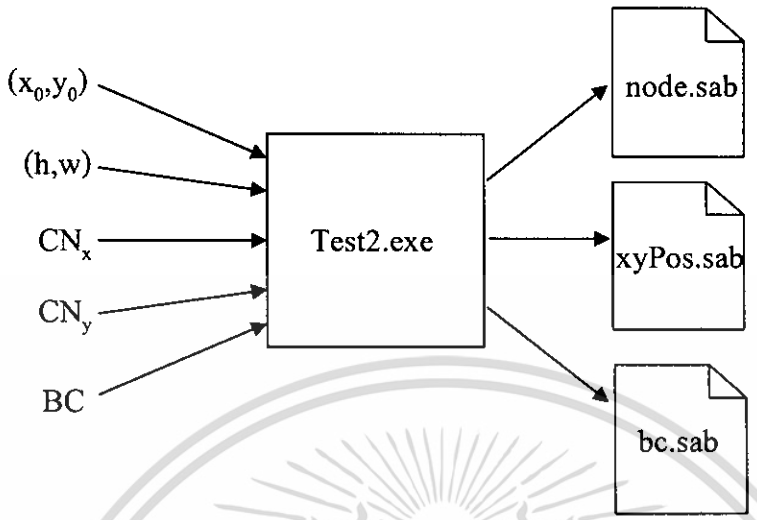
6. ควรจะมีการจัดเก็บข้อมูลโดยมีการแบ่งแยกแฟ้มข้อมูลอย่างชัดเจนดังนี้



รูปที่ 5.6 แสดงการจัดเก็บข้อมูลค่าต่าง ๆ ทางฟิสิกส์ที่รับเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยโปรแกรม Test1 จะทำการรับค่าต่าง ๆ ทางฟิสิกส์เข้ามา แล้วทำการจัดเก็บลงไฟล์ input.sab

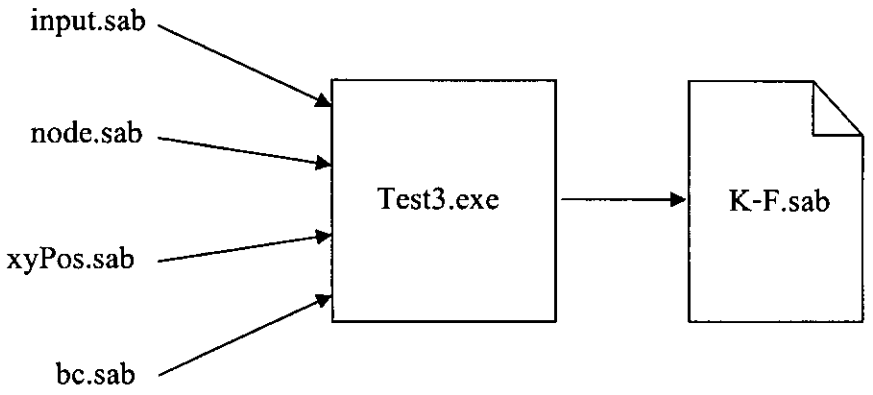


รูปที่ 5.7 แสดงการจัดเก็บข้อมูลค่าตำแหน่งของจุดต่อเริ่มต้นและจุดสิ้นสุด, ระยะห่างระหว่างจุดต่อและเงื่อนไขขอบเขต

โดย (x_0,y_0) คือ ตำแหน่ง(x,y) เริ่มต้นของขอบเขตปัญหา
(h,w) คือ ตำแหน่ง(x,y) สิ้นสุดของขอบเขตปัญหา
CN_x คือ จำนวนจุดต่อในแนวแกน x
CN_y คือ จำนวนจุดต่อในแนวแกน y
BC คือ เงื่อนไขค่าขอบเขต

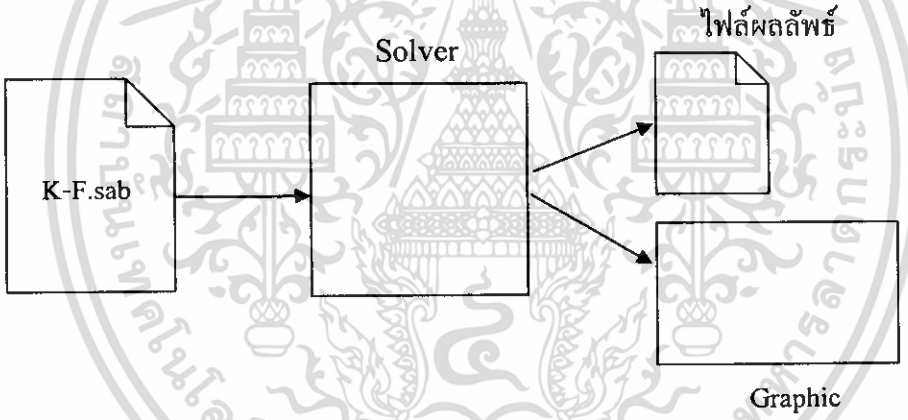
ซึ่งเมื่อโปรแกรม Test2 ทำการรับค่าต่าง ๆ ข้างต้นเรียบร้อยแล้วก็จะทำการจัดเก็บลงไฟล์ node.sab , xyPos.sab และ bc.sab ซึ่งไฟล์ bc.sab เก็บค่า bcdof และ bcval , ไฟล์ xyPos.sab เก็บค่าตำแหน่ง (x,y) ของแต่ละจุดต่อ และไฟล์ node.sab เก็บค่าหมายเลขจุดต่อของแต่ละเอลิเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8 แสดงการนำแฟ้มข้อมูลที่ได้ทั้งหมดมาเข้าสู่การคำนวณด้วยระเบียบวิธีไฟไนต์เอลิเมนต์

จากนั้นนำไฟล์ input.sab , node.sab , xyPos.sab และ bc.sab มาทำการคำนวณหาค่าเมทริกซ์ K และ เมทริกซ์ F ด้วยโปรแกรม Test3 โดยใช้กระบวนการไฟไนต์เอลิเมนต์ แล้วทำการจัดเก็บค่า เมทริกซ์ K และ เมทริกซ์ F ลงไฟล์ K-F.sab เพื่อใช้ในการคำนวณต่อไป



รูปที่ 5.9 นำแฟ้มข้อมูลจากรูปที่ 5.8 มาเข้าสู่กระบวนการ LU Decomposition

ในขั้นสุดท้ายจะนำเอาไฟล์ K-F.sab มาทำการหาผลเฉลยโดยวิธี LU Decomposition แล้วแสดงผลลัพธ์ออกมาในรูปแบบกราฟ 3 มิติ

5.3 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเห็นผลลัพธ์ได้จากการคำนวณและประมวลผลในโปรแกรมได้อย่างชัดเจน
2. เป็นแนวทางในการพัฒนาต่อ ๆ ไป โดยมีโครงงานที่ได้พัฒนานี้เป็นแหล่งข้อมูลศึกษาได้
3. ช่วยลดความซับซ้อนในการทำงานเกี่ยวกับปัญหาทางคณิตศาสตร์ในเชิงวิทยาศาสตร์ได้อย่างสะดวกและรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ประหยัดค่าใช้จ่ายและไม่เปลืองหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

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

ชัชวาล เรืองประพันธ์. 2539. **สถิติพื้นฐาน**. พิมพ์ครั้งที่ 2 ปรับปรุงแก้ไข. ขอนแก่น : หจก.โรงพิมพ์คลังนานาวิทยา.

กิตติ ภักดีวัฒนะกุล และ กิตติพงษ์ กลมกล่อม. 2547. **UML วิเคราะห์และออกแบบระบบเชิงวัตถุ**. พิมพ์ครั้งที่ 2. กรุงเทพฯ : หจก.ไทยเจริญการพิมพ์.

Robert W. Sebesta. 2006. **Concepts of Programming Language**. Seventh Edition. Bangkok : Pearson Education Indochina Ltd.

Edward Angel. 2006. **Interactive Computer Graphics**. Fourth Edition. Bangkok : Pearson Education Indochina Ltd.

William J. Palm. 2005. **Introduction to MATLAB 7 for Engineers**. Singapore : McGRAW-HILL.

Bill Blunden. 2003. **Memory Management Algorithms and Implementation in C/C++**. Wordware Publishing, Inc.

Bjarne Stroustrup. 1997. **The C+ + Programming Language**. Third Edition. Addison Wesley An Imprint of Addison Wesley Longman, Inc.

Young W. Kwon and Hyochoong Bang. 2000. **The Finite Element Method Using MATLAB**. Second Edition. Florida : CRC Press LLC.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

Source Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Demo5.cpp

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

#include "Constant.h"
#include "FEM.h"
#include "Rectangle.h"
#include "GLGraph.h"
#include "FileManager.h"

FEM fem;
class Rectangle rect;
class FileManager read;

/********************************** MAIN ****************************************/
void main()
{
        do {
                read.showReadMe();
                rect.run(&fem);
                printf("In main.\n");
                printf("max answer = %f\n\n",fem.findMaxAns());

        showGraph(fem.getSizeX(),fem.getSizeY(),fem.findMaxAns(),fem.findMinAns(),&fem);
        } while (true);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Constant.h

```
const float OUT_OF_POINTER = (float)(-pow(10,11));
const bool END_POINTER = true;
const float G = 1.0;
const int ARRAY_SIZE = 1000;
// constant below use for specify array size of 'bcdof' and 'bcval'
const int INDEX = 100;
const int SIDE = 10;
const int TYPE = 2;
const int DIRICHLET = 0;
const int NEUMAN = 1;
const int SIDE_OF_RECTANGLE = 4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEM.h

```
class FEM
{
private :
        // these variables below use for calculate node member
        int size_x;
        int size_y;
        int nodePos;
        int nodeSize;
        int epol;                                   // number of elements per one
line
        int elem_size;
        int elem;
        int nodes[2*ARRAY_SIZE][4];
        int bcdof[ARRAY_SIZE];
        float bcval[ARRAY_SIZE];
        int type[ARRAY_SIZE];
        // these variables below use for calculate matrix equation
        float ks[ARRAY_SIZE][ARRAY_SIZE];
        float ke[4][4];
        float xyPos[ARRAY_SIZE][2+1];
        float fe[ARRAY_SIZE];
        float fs[ARRAY_SIZE];
        // these variables below use for calculate matrix equation answer
        float tempX[ARRAY_SIZE];
        // special variables
        bool isPoisson;
        bool isX;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        bool isY;
        bool isXY;
        float gconst;
        float diff_x;
        float diff_y;
        //
        int number_of_nodes;
        int number_of_elements;
        int size_of_k;
        int size_of_element;
        int size_of_bc_d;
        int size_of_bc_n;
        // gmsh for graph
        int begin_element_index;
        int number_of_sides;
public :
        FEM();
        // SETTER
        void setSizeX(int sizeX);
        void setSizeY(int sizeY);
        void setNodePos(int ndPos);
        void setNodeSize(int ndSize);
        void setEPOL(int elem_per_one_line);
        void setElemSize(int elemSize);
        void setElem(int elements);
        void setNodes(int elemen_index,int node_index,int value);
        void setKs(int i,int j,float value);
        void setKe(int i,int j,float value);
        void setXYPos(int i,int j,float value);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void setFe(int i,float value);
void setFs(int i,float value);
void setAns(int i,float value);
void setIsPoisson(bool b);
void setIsX(bool b);
void setIsY(bool b);
void setIsXY(bool b);
void setGConst(float g);
void setDiffX(float x);
void setDiffY(float y);
void setNumberOfNodes(int n);
void setNumberOfElements(int e);
void setSizeOfK(int k);
void setSizeOfElement(int e);
void setSizeOfBCD(int d);
void setSizeOfBCN(int n);
void setBcdof(int index,int node);
void setBcval(int index,float data);
void setBeginElementIndex(int);
void setNumberOfSides(int);
// GETTER
int getSizeX();
int getSizeY();
int getNodePos();
int getNodeSize();
int getEPOL();
int getElemSize();
int getElem();
int getNodes(int element_index,int node_index);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float getKs(int i,int j);
float getKe(int i,int j);
float getXYPos(int i,int j);
float getFe(int i);
float getFs(int i);
float getAns(int i);
bool getIsPoisson();
bool getIsX();
bool getIsY();
bool getIsXY();
float getGConst();
float getDiffX();
float getDiffY();
int getNumberOfNodes();
int getNumberOfElements();
int getSizeOfK();
int getSizeOfElement();
int getSizeOfBCD();
int getSizeOfBCN();
int getBcdof(int index);
float getBcval(int index);
int getBeginElementIndex();
int getNumberOfSides();
// Process
void calSize();
void setNodeMember();
bool setXYPosition(char[],char[]);
void setRectBC(float side_value[],int type_arg[]);
void aplyDirichlet();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        void aplyNeuman();
        void felp2dt3(float x1,float y1,float x2,float y2,float x3,float y3);
        void assembly();
        void triDecomp();
        float findMaxAns();
        float findMinAns();
        float calAnsFromRectXY(float inputed_x,float inputed_y);
};
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FileManager.h

```
class FileManager
{
private :
public :
        FileManager();
        void loadPositionFile(char fileName[20],float *pl,int number_of_nodes);
        void writeFile(char fileName[20],float *pw,int size_of_k);
        void loadPositionFile_gmsh(char fileName[20],float *pl,float nod,float elm);
        void writeFile_gmsh(char fileName[20],float *pw,float nod);
        void showReadMe();
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GLGraph.h

```
#ifndef _GLGRAPH_H_
#define _GLGRAPH_H_

#include <windows.h>
#include <winuser.h>
//#include <gl/gl.h>
//#include <gl/glu.h>
#include <gl/glut.h>
#include "vertexhandler.h"
#include "vertex.h"

/* forword declaration */
LRESULT CALLBACK WndProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM
lParam);

/* create window & initialize opengl */
bool createOpenGLWindow(char *title, int width, int height, int bits);

/* initialize opengl */
bool initOpenGL();

/* release all context & destroy window */
void destroyOpenGLWindow();

/* directive for show error message */
#define errmsg(msg) MessageBox (NULL, msg, "ERROR!", MB_OK |
MB_ICONINFORMATION);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* key checker function */
void checkKeys();

void mouse(int button,int state,int x,int y);

void Begin2D(void);

void DrawColor64Scale(void);

void drawHeader();
void drawCalButton(void);
void drawClearButton(void);
void drawInputPanel(void);

void End2D(void);

void InitColorTable();

void GetColor64f(GLfloat* dest_color, GLfloat color, GLfloat vMin, GLfloat vMax);

char chgToC(int iin);

int showGraph(int x,int y,float maxAns,float minAns,FEM *finite);
int showGraph_gmsh(FEM *fini);

/* directive for visual key */
#define VK_W  0x57
#define VK_A  0x41
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define VK_S    0x53
#define VK_D    0x44
#define VK_Q    0x51
#define VK_E    0x45
#define VK_Z    0x5A
#define VK_X    0x58
#define VK_C    0x43
#define VK_V    0x56
#define VK_O    0x4F
#define VK_P    0x50

extern HINSTANCE hInstance;
extern HWND hWnd;
extern HDC hDC;
extern HGLRC hRC;
extern bool keys[256];
extern const char appName[];

//load vertex from file
extern VertexHandler* vertexhandler;

// rotation
extern GLfloat angle;
extern GLfloat rotx;
extern GLfloat roty;
extern GLfloat rotz;

// translation
extern GLfloat transx;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
extern GLfloat transy;
extern GLfloat transz;

// wireframe mode
extern bool isWireFrame;

extern GLfloat colorTable[64][3];// color table for specific color range

// color of text box
extern GLfloat txt_on_color[3], txt_off_color[3];

#endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rectangle.h

```
class Rectangle
{
public :
        Rectangle();
        void run(FEM *p_fem);
};
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vector3.h

```
/*
        Vector3 Class
        Author Peeba_Merlin
*/

#ifndef _VECTOR3_
#define _VECTOR3_

#ifndef _CMATH_
#include <cmath>
#endif

#include "vertex.h"

class Vector3 {
public:
        float x;
        float y;
        float z;

        /* Constructor */
        /* Vector3 Default Constructor */
        Vector3() { }

        /* Vector3 Copy Constructor */
        Vector3(Vector3& v) : x(v.x), y(v.y), z(v.z) {}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Constructor Given three values */
Vector3(float _x, float _y, float _z) : x(_x), y(_y), z(_z) {}

/* Constructor create Vector from two vertices */
Vector3(const Vertex& v1,const Vertex& v2) {
        x = v1.x - v2.x;
        y = v1.y - v2.y;
        z = v1.z - v2.z;
}

/* Vector3 Operator Overloading */
/* Assign vector value */
Vector3 &operator=(const Vector3& v) {
        x = v.x; y = v.y; z = v.z;
        return *this;
}

/* Compare for equality */
bool operator==(const Vector3& v) const {
        return x == v.x && y == v.y && z == v.z;
}

/* Compare for differency */
bool operator!=(const Vector3& v) const {
        return x != v.x || y != v.y || z != v.z;
}

/* Set to  zero vector */
void zero(void) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        x = y = z = 0;
}

/* Unary minus returns the negative of the vector */
Vector3 operator- () const {
        return Vector3(-x, -y, -z);
}

/* Binary + and - operator */

Vector3 operator-(const Vector3 & v) const {
        return Vector3(x - v.x, y - v.y, z - v.z);
}

Vector3 operator+(const Vector3& v) const {
        return Vector3(x + v.x, y + v.y, z + v.z);
}

/* Multiplication & Division by scalar */
Vector3 operator*(const float a) {
        return Vector3(x * a, y * a, z * a);
}

Vector3 operator/(const float a) {
        float inv = 1.0f / a; // no check divide by zero here
        return Vector3(x * inv, y * inv, z * inv);
}

/* Conbined assignment operators to conform to
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  C notation convention */
Vector3 &operator+=(const Vector3& v) {
        x = x + v.x; y = y - v.y; z = z - v.z;
        return *this;
}

/* Minus */
Vector3 &operator-=(const Vector3& v) {
        x = x - v.x; y = y - v.y; z = z - v.z;
        return *this;
}

/* Multiply */
Vector3 &operator*=(const float a) {
        x = x * a; y = y * a; z = z * a;
        return *this;
}

/* Devide */
Vector3 &operator/=(const float a) {
        float inv = 1.0f / a; // no check divide by zero here

        x = x * inv; y = y * inv; z = z * inv;
        return *this;
}

/* normalize vector */
void normalize() {
        float magSq = x*x + y*y + z*z;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (magSq > 0.0f) { // Check for divide by zero
                float inv = 1.0f / (float)sqrt(magSq);
                x *= inv;
                y *= inv;
                z *= inv;
            }
        }

        /* Vector dot product uses '*' as operator */
        float operator*(const Vector3& v) {
            return (x * v.x) + (y * v.y) + (z * v.z);
        }
};

/* Nonmember function uses as utility */
/* compute the magnitude of a vector */
inline float vectorMag(const Vector3& v) {
        return 1.0f / (float)sqrt(v.x * v.x + v.y * v.y + v.z * v.z);
}

inline Vector3 crossProduct(const Vector3& u, const Vector3& v) {
        return Vector3(u.y * v.z - u.z * v.y,
                       u.z * v.x - u.x * v.z,
                       u.x * v.y - u.y * v.x);
}

/* Scalar on the left multiplication for symetry */
inline Vector3 operator*(const float k, const Vector3& v) {
        return Vector3(v.x * k, v.y * k, v.z * k);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

/* Compute the distance between two points */
inline float distance(const Vector3& u, const Vector3& v) {
        float dx = u.x - v.x;
        float dy = u.y - v.y;
        float dz = u.z - v.z;
        return (float)sqrt(dx * dx + dy * dy + dz * dz);
}

/*********************
 GLOBAL ZERO VECTOR
 *********************/
extern const Vector3 ZERO_VECTOR(0.0f , 0.0f, 0.0f);

#endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VERTEX.h

```
/*
 * vertex.h - vertex descriptor class
 * Author Peeba_Merlin
*/

#ifndef _VERTEX_H
#define _VERTEX_H

class Vertex {
public :
        float x;
        float y;
        float z;

        /* Constructor */
        /* Vector3 Default Constructor */
        Vertex(void);

        /* Vertex copy constructor */
        Vertex(const Vertex& vertex) {
                x = vertex.x;
                y = vertex.y;
                z = vertex.z;
        }

        /* Constructor given three values */
        Vertex(float _x, float _y, float _z) : x(_x), y(_y), z(_z) { }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
};

#endif
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

vertexhandler.h

```
/*
 * vertex.h - vertex descriptor class
 * Author Peeba_Merlin
*/

#ifndef _VERTEX_H
#define _VERTEX_H

class Vertex {
public :
        float x;
        float y;
        float z;

        /* Constructor */
        /* Vector3 Default Constructor */
        Vertex(void);

        /* Vertex copy constructor */
        Vertex(const Vertex& vertex) {
                x = vertex.x;
                y = vertex.y;
                z = vertex.z;
        }

        /* Constructor given three values */
        Vertex(float _x, float _y, float _z) : x(_x), y(_y), z(_z) { }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
};

#endif
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEM.cpp

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

#include "Constant.h"
#include "FileManager.h"
#include "FEM.h"

FileManager fm;

/* constructor */
FEM::FEM()
{
        setSizeX(0);
        setSizeY(0);
        setNodePos(1);
        setElem(1);
        setIsPoisson(false);
        setIsX(false);
        setIsY(false);
        setIsXY(false);
        setSizeOfElement(3);
}

// SETTER
void FEM::setSizeX(int sizeX)
        {       size_x = sizeX;                 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void FEM::setSizeY(int sizeY)
        {       size_y = sizeY;                         }
void FEM::setNodePos(int ndPos)
        {       nodePos = ndPos;                        }
void FEM::setNodeSize(int ndSize)
        {       nodeSize = ndSize;                      }
void FEM::setEPOL(int elem_per_one_line)
        {       epol = elem_per_one_line;       }
void FEM::setElemSize(int elemSize)
        {       elem_size = elemSize;           }
void FEM::setElem(int elements)
        {       elem = elements;                        }
void FEM::setNodes(int i,int j,int value)
        {       nodes[i][j] = value;            }
void FEM::setKs(int i,int j,float value)
        {       ks[i][j] = value;               }
void FEM::setKe(int i,int j,float value)
        {       ke[i][j] = value;               }
void FEM::setXYPos(int i,int j,float value)
        {       xyPos[i][j] = value;            }
void FEM::setFe(int i,float value)
        {       fe[i] = value;          }
void FEM::setFs(int i,float value)
        {       fs[i] = value;          }
void FEM::setAns(int i,float value)
        {       tempX[i] = value;       }
void FEM::setIsPoisson(bool b)
        {       isPoisson = b;  }
void FEM::setGConst(float g)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {       gconst = g;             }
void FEM::setIsX(bool b)
        {       isX = b;        }
void FEM::setIsY(bool b)
        {       isY = b;        }
void FEM::setIsXY(bool b)
        {       isXY = b;               }
void FEM::setDiffX(float x)
        {       diff_x = x;             }
void FEM::setDiffY(float y)
        {       diff_y = y;             }
void FEM::setNumberOfNodes(int n)
        {       number_of_nodes = n;
                setSizeOfK(number_of_nodes);    }
void FEM::setNumberOfElements(int e)
        {       number_of_elements = e;         }
void FEM::setSizeOfK(int k)
        {       size_of_k = k;                  }
void FEM::setSizeOfElement(int e)
        {       size_of_element = e;    }
void FEM::setSizeOfBCD(int d)
        {       size_of_bc_d = d;               }
void FEM::setSizeOfBCN(int n)
        {       size_of_bc_n = n;               }
void FEM::setBcdof(int index,int node)
        {       bcdof[index] = node;    }
void FEM::setBcval(int index,float data)
        {       bcval[index] = data;    }
void FEM::setBeginElementIndex(int begin)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {       begin_element_index = begin;    }
void FEM::setNumberOfSides(int ns)
        {       number_of_sides = ns;   }

// GETTER
int FEM::getSizeX()
        {       return size_x;          }
int FEM::getSizeY()
        {       return size_y;          }
int FEM::getNodePos()
        {       return nodePos;         }
int FEM::getNodeSize()
        {       return nodeSize; }
int FEM::getEPOL()
        {       return epol;            }
int FEM::getElemSize()
        {       return elem_size;       }
int FEM::getElem()
        {       return elem;            }
int FEM::getNodes(int i,int j)
        {       return nodes[i][j];     }
float FEM::getKs(int i,int j)
        {       return ks[i][j];    }
float FEM::getKe(int i,int j)
        {       return ke[i][j];    }
float FEM::getXYPos(int i,int j)
        {       return xyPos[i][j];     }
float FEM::getFe(int i)
        {       return fe[i];           }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float FEM::getFs(int i)
        {       return fs[i];                   }
float FEM::getAns(int i)
        {       return tempX[i]; }
bool FEM::getIsPoisson()
        {       return isPoisson; }
float FEM::getGConst()
        {       return gconst;                  }
bool FEM::getIsX()
        {       return isX;                     }
bool FEM::getIsY()
        {       return isY;                     }
bool FEM::getIsXY()
        {       return isXY;                    }
float FEM::getDiffX()
        {       return diff_x;                  }
float FEM::getDiffY()
        {       return diff_y;                  }
int FEM::getNumberOfNodes()
        {       return number_of_nodes; }
int FEM::getNumberOfElements()
        {       return number_of_elements;      }
int FEM::getSizeOfK()
        {       return size_of_k;}
int FEM::getSizeOfElement()
        {       return size_of_element;   }
int FEM::getSizeOfBCD()
        {       return size_of_bc_d;      }
int FEM::getSizeOfBCN()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {       return size_of_bc_n;    }
int FEM::getBcdof(int index)
        {       return bcdof[index];    }
float FEM::getBcval(int index)
        {       return bcval[index];    }
int FEM::getBeginElementIndex()
        {       return begin_element_index;     }
int FEM::getNumberOfSides()
        {       return number_of_sides; }

/*--------------------------------------------------------------------------*/
//                      Process
//
void FEM::calSize()
{
        float dfX = getDiffX();
        float dfY = getDiffY();
        float x[ARRAY_SIZE],y[ARRAY_SIZE];
        int i,j;
        int cnt = 1;

        // set data for X-axe and Y-axe
        for (i=1; i<=getSizeY(); i++)
        {
                x[cnt] = 0.0;
                for (j=1; j<=getSizeX(); j++)
                {
                        if (i==getSizeY() && j==1) j++;
                        x[cnt+1] = x[cnt] + dfX;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                cnt++;
            }
        }
        float sum = 0.0;
        cnt = 1;
        for (i=1; i<=getSizeY(); i++)
        {
            for (j=1; j<=getSizeX(); j++)
            {
                y[cnt] = sum;
                cnt++;
            }
            sum += dfY;
        }
        setNodeSize(getSizeX()*getSizeY());   // find number of node
        setNumberOfNodes(getNodeSize());
        fm.writeFile("X.txt",&x[1],getSizeOfK());
        fm.writeFile("Y.txt",&y[1],getSizeOfK());
        while (getNodePos() <= getNodeSize())  // show all of node by...
        {
            setNodePos(getNodePos()+1);
        }
        setEPOL((getSizeX()-1)*2); // number of Element Per One Line
        setElemSize(((getNodeSize()*getEPOL())/size_x)-getEPOL());

        // set number of nodes on boundary
        setSizeOfBCD(2*getSizeX() + 2*getSizeY() - 4);
        setNumberOfElements((getSizeX()-1)*(getSizeY()-1)*2);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void FEM::setNodeMember()
{
	int i=0;
	int me=1;
	int mo=0;
	int ly = 1;
	int m = 0;
	int n =1;
	int p =0;

	for (i=0; i<=number_of_elements; i++)
	{
		nodes[i][1] = 0;
		nodes[i][2] = 0;
		nodes[i][3] = 0;
	}

	while (ly < getSizeY())//1
	{
		while (n <= getSizeX())
		{
			for (i=0; i<getSizeX()-1; i++)
			{
				nodes[n+i+p][1] = 1+i+m;
			}
			n = n + getSizeX() - 1;
		}
		p = p + getEPOL();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        n = 1;
        m = m + getSizeX();
        ly++;
}// while outter 1

ly = 1;
m = 0;
n =1;
p=0;
i=0;

while (ly < getSizeY()) // 2
{
        if ((n+i+p) <= (getSizeX()-1))
        {
                for (i=0; i<getSizeX()-1; i++)
                {
                        setNodes(n+i+p,2,2+i);
                }
                n = 1;
                ly++;
        }

        while (n <= getSizeX())
        {
                n = n + getSizeX() - 1;
                if ((n+i+p) >= getSizeX() || (n+i+p) < (getElemSize()-(getSizeX()-1)))
                {
                        for (i=0; i<getSizeX()-1; i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    {
                        setNodes(n+i+p,2,2+getSizeX()+i+m);
                    }
                }
            }

            p = p + getEPOL();
            n = 1;
            m = m + getSizeX();
            ly++;

            if ((n+i+p) > (getElemSize()-(getSizeX()-1)))
            {
                n = n + getSizeX() - 1;
                for (i=0; i<getSizeX()-1; i++)
                {
                    setNodes(n+i+p,2,getNodeSize()-(getSizeX()-2)+i);
                }
            }
        }//while outter 2

        ly = 1;
        m = 0;
        n = 1;
        p = 0;

        while (ly < getSizeY())// 3
        {
            while (n <= getSizeX())
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            if (n == 1)
            {
                me = (me + getSizeX());
                for (i=0; i<getSizeX()-1; i++)
                {
                    setNodes(n+i+p,3,1+i+me);
                }
            }
            else if (n == getSizeX())
            {
                mo = mo + getSizeX();
                for (i=0; i<getSizeX()-1; i++)
                {
                    setNodes(n+i+p,3,1+i+mo);
                }
            }
            n = n + getSizeX() - 1;
        }
        p = p + getEPOL();
        n = 1;
        ly++;
    }//while outter 3
}

void FEM::setRectBC(float side_value[],int type_arg[])
{
    int i;
    int cnt = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// initial value for bcdof
for (i=1; i<=(2*size_x + 2*size_y - 4); i++) bcdof[i] = 0;
if (bcdof[cnt] <= size_x)  // edge of the bottom
{
        for (i=1; i<=size_x; i++)
        {
                bcdof[cnt] = i;
                bcval[cnt] = side_value[0];
                type[cnt] = type_arg[0];
                cnt++;
        }
}
int in = i;
int a;
if (bcdof[cnt-1] >= size_x)  // other edge
{
        for (a=2; a<=size_y; a++)  // right_edge
        {

                int l_val = size_x*a;
                bcdof[cnt] = l_val;
                bcval[cnt] = side_value[2];
                type[cnt] = type_arg[2];
                l_val = 0;
                in++;
                cnt++;
        }

        int ind=in;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int l_val1 = 0;
if (size_x == size_y)  // left_edge if size_x=size_y
{
        for (i=size_x+1; i<=size_x*size_y; i=i+size_x)
        {
                bcdof[cnt] = i;
                bcval[cnt] = side_value[1];
                type[cnt] = type_arg[1];
                l_val1 = 0;
                ind++;
                cnt++;
        }
}
else  // left_edge
{
        for (i=size_x+1; i<=size_x*size_y; i=i+size_x)
        {
                bcdof[cnt] = i;
                bcval[cnt] = side_value[1];
                type[cnt] = type_arg[1];
                l_val1 = 0;
                ind++;
                cnt++;
        }
}
int index = ind;
int b_val = 0;
if (bcdof[cnt-1] <= size_x*(size_y-1)+2)  // upper_edge
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                for (i=size_x*(size_y-1)+2; i<size_x*size_y; i++)
                {
                        bcdof[cnt] = i;
                        bcval[cnt] = side  value[3];
                        type[cnt] = type_arg[3];
                        b_val = 0;
                        index++;
                        cnt++;
                }
        }
    }
}

bool FEM::setXYPosition(char x_name[],char y_name[])
// increase array size as 1 because we want initial origin array index as 1
{
        int x = 1;
        int y = 2;
        bool pass = true;

        fm.loadPositionFile(x_name,&xyPos[1][x],getNumberOfNodes());
        fm.loadPositionFile(y_name,&xyPos[1][y],getNumberOfNodes());

        return pass;
}

void FEM::aplyDirichlet()
{
        int i,j;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        int c;
        for (i=1; i<=size_of_bc_d; i++)
        {
                if (type[i] == DIRICHLET)
                {
                        c = bcdof[i];
                        for (j=1; j<=size_of_k; j++)
                        {
                                ks[c][j] = 0.0;
                        }
                        ks[c][c] = 1.0;
                        fs[c] = bcval[i];
                }
        }
        printf("Apply Dirichlet boundary conditions success.\n\n");
}

void FEM::aplyNeuman()
{
        int i;
        int nd1,nd2;
        float length;
        for (i=1; i<size_of_bc_n; i++)
        {
                if (type[i] == NEUMAN)
                {
                        nd1 = bcdof[i];
                        nd2 = bcdof[i+1];
                        length = getXYPos(nd2,2) - getXYPos(nd1,2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                fs[nd1] += bcval[i]*length/2;
                fs[nd2] += bcval[i]*length/2;
            }
        }
        printf("Apply Neuman boundary conditions success.\n\n");
}

void FEM::felp2dt3(float x1,float y1,float x2,float y2,float x3,float y3)
//-----------------------------------------------------------------
// This function use for compute element matrix K.
//-----------------------------------------------------------------
{
        float area;
        area = (float)(0.5*(x2*y3+x1*y2+x3*y1-x2*y1-x1*y3-x3*y2));
        // set element matrix, k
        ke[1][1] = ((x3-x2)*(x3-x2)+(y2-y3)*(y2-y3))/(4*area);
        ke[1][2] = ((x3-x2)*(x1-x3)+(y2-y3)*(y3-y1))/(4*area);
        ke[1][3] = ((x3-x2)*(x2-x1)+(y2-y3)*(y1-y2))/(4*area);
        ke[2][1] = ke[1][2];
        ke[2][2] = ((x1-x3)*(x1-x3)+(y3-y1)*(y3-y1))/(4*area);
        ke[2][3] = ((x1-x3)*(x2-x1)+(y3-y1)*(y1-y2))/(4*area);
        ke[3][1] = ke[1][3];
        ke[3][2] = ke[2][3];
        ke[3][3] = ((x2-x1)*(x2-x1)+(y1-y2)*(y1-y2))/(4*area);
        // set element matrix, f
        fe[1] = getGConst()*area/3;
        fe[2] = getGConst()*area/3;
        fe[3] = getGConst()*area/3;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void FEM::assembly()
{
        int i=1,j=1,k=1;
        // initil array, Ks
        for (i=1; i<=size_of_k; i++)
        {
                for (j=1; j<=size_of_k; j++)
                {
                        ks[i][j] = 0.0;
                }
        }
        // process
        for (k=getBeginElementIndex(); k<=number_of_elements; k++)
        {
                float x1,y1,x2,y2,x3,y3;
                int nd1,nd2,nd3;
                nd1 = getNodes(k,1);
                nd2 = getNodes(k,2);
                nd3 = getNodes(k,3);
                x1 = getXYPos(nd1,1);
                y1 = getXYPos(nd1,2);
                x2 = getXYPos(nd2,1);
                y2 = getXYPos(nd2,2);
                x3 = getXYPos(nd3,1);
                y3 = getXYPos(nd3,2);
                felp2dt3(x1,y1,x2,y2,x3,y3);

                for (i=1; i<=size_of_element; i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        for (j=1; j<=size_of_element; j++)
        {
                ks[getNodes(k,i)][getNodes(k,j)] += ke[i][j];
        }
        if (getIsPoisson())
        {
                if (getIsX())
                {
                        if (i==1) fs[nodes[k][i]] += x1;
                        else if (i==2) fs[nodes[k][i]] += x2;
                        else if (i==3) fs[nodes[k][i]] += x3;
                }
                else if (getIsY())
                {
                        if (i==1) fs[nodes[k][i]] += y1;
                        else if (i==2) fs[nodes[k][i]] += y2;
                        else if (i==3) fs[nodes[k][i]] += y3;
                }
                else if (getIsXY())
                {
                        if (i==1) fs[nodes[k][i]] += x1*y1;
                        else if (i==2) fs[nodes[k][i]] += x2*y2;
                        else if (i==3) fs[nodes[k][i]] += x3*y3;
                }
                else
                {
                        fs[nodes[k][i]] += fe[i];
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                }
        }
        printf("Assembly K(sys) success.\n\n");
}

void FEM::triDecomp()
{
        const int SIZE = 200;

        float tempY[SIZE];
        float matrixL[SIZE][SIZE];
        float matrixU[SIZE][SIZE];
        float a[SIZE][SIZE];
        float b[SIZE];
        int i, j, k;

        // initialization
        for (i=0; i<size_of_k; i++)
        {
                for (j=0; j<size_of_k; j++)
                {
                        matrixL[i][j] = 0.0;
                        matrixU[i][j] = 0.0;
                }
                tempX[i] = 0.0;
                tempY[i] = 0.0;
        }
        // set values into 'a' matrix
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for (i=0; i<size_of_k; i++)
{
        for (j=0; j<size_of_k; j++)
        {
                a[i][j] = ks[i+1][j+1];
        }
}
// set values into 'b' matrix
for (i=0; i<size_of_k; i++)
{
        b[i] = fs[i+1];
}

// compute for 'A = LU'
for (j=0; j<size_of_k; j++)
{
        matrixL[j][j] = 1.0;
        matrixU[0][j] = a[0][j];
}
for (i=1; i<size_of_k; i++)
{
        for (j=0; j<size_of_k; j++)
        {
                float sum = 0.0;

                if (i > j)
                {
                        for (k=0; k<size_of_k; k++)
                        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        if (k != j)
                        {
                                sum += matrixL[i][k]*matrixU[k][j];
                        }
                }
                matrixL[i][j] = (a[i][j] - sum)/matrixU[j][j];
        }
        else if (i <= j)
        {
                for (k=0; k<size_of_k; k++)
                {
                        if (k != j)
                        {
                                sum += matrixL[i][k]*matrixU[k][j];
                        }
                }
                matrixU[i][j] = (a[i][j] - sum);
        }
    }
}

for (i=0; i<size_of_k; i++)
{
        float sum = 0.0;

        for (j=0; j<size_of_k; j++)
        {
                if (i != j)
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        sum += matrixL[i][j]*tempY[j];
                    }
                }
                tempY[i] = b[i] - sum;
        }
        for (i=(size_of_k-1); i>=0; i--)
        {
                float sum = 0.0;

                for (j=1; j<size_of_k; j++)
                {
                    if (i != j)
                    {
                        sum += matrixU[i][j]*tempX[j];
                    }
                }
                tempX[i] = (tempY[i] - sum)/matrixU[i][i];
        }
}

float FEM::findMaxAns()
{
        int i;
        float tmp = -9999999;
        for (i=1; i<=number_of_nodes; i++)
        {
            if (tmp < getAns(i)) tmp = getAns(i);
        }
        return tmp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

float FEM::findMinAns()
{
        int i;
        float tmp = 9999999;
        for (i=1; i<=number_of_nodes; i++)
        {
                if (tmp > getAns(i)) tmp = getAns(i);
        }
        return tmp;
}

float FEM::calAnsFromRectXY(float inputed_x,float inputed_y)
{
        float x1,y1,
                x2,y2,
                x3,y3,
                x4,y4;
        float dx = getDiffX();
        float dy = getDiffY();
        int nd1=0, nd2=0, nd3=0, nd4=0;
        int i,j;

        // find element area that inputed point inside
        // rectangle area, (x1,y1) (x2,y2) (x3,y3) (x4,y4)
        x1 = 0;
        x2 = 0;
        float x_distance = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for (i=1; i<=size_x; i++)
{
        if (inputed_x < x_distance) break;
        x1 = x_distance;
        x_distance += dx;
        x2 = x_distance;
}
y1 = 0;
y4 = 0;
float y_distance = 0;
for (j=1; j<=size_y; j++)
{
        if (inputed_y < y_distance) break;
        y1 = y_distance;
        y_distance += dy;
        y4 = y_distance;
}
x4 = x1;
x3 = x2;
y3 = y4;
y2 = y1;
nd1 = (size_x-(size_x-i)-1)*(j-1) + (size_x-i+1)*(j-2);
nd2 = i*(j-1) + (size_x-i)*(j-2);
nd3 = i*j + (size_x-i)*(j-1);
nd4 = (i-1)*j + (size_x-i+1)*(j-1);
// which triangle element area is inside
float m = (y3-y1)/(x3-x1);
bool bottom_elm = false;
float ys = 0, xs = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if ((inputed_y > y1) && (inputed_y < y3))
{
        if ((inputed_x < x2) && (inputed_x > x1))
        {
                ys = m*(inputed_x - x1) + y1;
                if (inputed_y < ys)
                {
                        xs = (inputed_y - y1)/m + x1;
                        if (inputed_x > xs)
                        {
                                bottom_elm = true;
                        }
                }
        }
}
// calculate weight
float xx1,yy1,
        xx2,yy2,
        xx3,yy3,
        xx4,yy4;
float m1,m2,m3,m4;
float long_line_1,short_line_1,
        long_line_2,short_line_2,
        long_line_3,short_line_3,
        long_line_4,short_line_4;
float w1,w2,w3,w4;
m1 = (inputed_y - y1)/(inputed_x - x1);
m3 = (inputed_y - y3)/(inputed_x - x3);
if (bottom_elm)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        m2 = (inputed_y - y2)/(inputed_x - x2);
        xx1 = x2;
        yy1 = m1*(x2 - x1) + y1;
        xx2 = (m2*x2 - m1*x1 + y1 - y2)/(m2 - m1);
        yy2 = (m1*m2*(x2 - x1) + m2*y1 - m1*y2)/(m2 - m1);
        xx3 = (y1 - y3)/m3 + x3;
        yy3 = y1;
        long_line_1 = sqrt(pow(xx1-x1,2) + pow(yy1-y1,2));
        short_line_1 = sqrt(pow(inputed_x-x1,2) + pow(inputed_y-y1,2));
        w1 = 1 - short_line_1/long_line_1;
        long_line_2 = sqrt(pow(xx2-x2,2) + pow(yy2-y2,2));
        short_line_2 = sqrt(pow(inputed_x-x2,2) + pow(inputed_y-y2,2));
        w2 = 1 - short_line_2/long_line_2;
        long_line_3 = sqrt(pow(xx3-x3,2) + pow(yy3-y3,2));
        short_line_3 = sqrt(pow(inputed_x-x3,2) + pow(inputed_y-y3,2));
        w3 = 1 - short_line_3/long_line_3;
}
else  // upper element
{
        m4 = (inputed_y - y4)/(inputed_x - x4);
        xx1 = (y4 - y1)*m1 + x1;
        yy1 = y4;
        xx3 = x1;
        yy3 = m3*(x1 - x3) + y3;
        xx4 = (m3*m4 - m3*x3 + y3 - y4)/(m4 - m1);
        yy4 = (m3*m4*(x4 - x3) + m4*y3 - m3*y4)/(m4 - m3);
        long_line_1 = sqrt(pow(xx1-x1,2) + pow(yy1-y1,2));
        short_line_1 = sqrt(pow(inputed_x-x1,2) + pow(inputed_y-y1,2));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                w1 = 1 - short_line_1/long_line_1;
                long_line_3 = sqrt(pow(xx3-x3,2) + pow(yy3-y3,2));
                short_line_3 = sqrt(pow(inputed_x-x3,2) + pow(inputed_y-y3,2));
                w3 = 1 - short_line_3/long_line_3;
                if (inputed_x == x4) {
                        long_line_4 = dy;
                }
                else {
                        long_line_4 = sqrt(pow(xx4-x4,2) + pow(yy4-y4,2));
                }
                short_line_4 = sqrt(pow(inputed_x-x4,2) + pow(inputed_y-y4,2));
                w4 = 1 - short_line_4/long_line_4;
        }
        // calculate phi, answer
        float ans = 0;
        if (bottom_elm)
        {
printf(""); // debug
                // convert to positive number
                float cvt_t1,cvt_t2,cvt_t3,
                        cvt_w1,cvt_w2,cvt_w3;
                cvt_t1 = (tempX[nd1-1]*w1);
                cvt_t2 = (tempX[nd2-1]*w2);
                cvt_t3 = (tempX[nd3-1]*w3);
                cvt_w1 = w1;
                cvt_w2 = w2;
                cvt_w3 = w3;
                if ((tempX[nd1-1]*w1)<0) cvt_t1 *= -1;
                if ((tempX[nd2-1]*w2)<0) cvt_t2 *= -1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if ((tempX[nd3-1]*w3)<0) cvt_t3 *= -1;
            if (w1<0) cvt_w1 *= -1;
            if (w2<0) cvt_w2 *= -1;
            if (w3<0) cvt_w3 *= -1;
            // answer
            ans = (cvt_t1+cvt_t2+cvt_t3)/(cvt_w1+cvt_w2+cvt_w3);
        }
        else
        {
printf("");  // debug
            // convert to positive number
            float cvt_t1,cvt_t3,cvt_t4,
                  cvt_w1,cvt_w3,cvt_w4;
            cvt_t1 = (tempX[nd1-1]*w1);
            cvt_t3 = (tempX[nd3-1]*w3);
            cvt_t4 = (tempX[nd4-1]*w4);
            cvt_w1 = w1;
            cvt_w3 = w3;
            cvt_w4 = w4;
            if ((tempX[nd1-1]*w1)<0) cvt_t1 *= -1;
            if ((tempX[nd3-1]*w3)<0) cvt_t3 *= -1;
            if ((tempX[nd4-1]*w4)<0) cvt_t4 *= -1;
            if (w1<0) cvt_w1 *= -1;
            if (w3<0) cvt_w3 *= -1;
            if (w4<0) cvt_w4 *= -1;
            // answer
            ans = (cvt_t1+cvt_t3+cvt_t4)/(cvt_w1+cvt_w3+cvt_w4);
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return ans;
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FileManager.cpp

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

#include "FileManager.h"
#include "Constant.h"
#include "FEM.h"

/* Constructor */
FileManager::FileManager()
{
}

void FileManager::loadPositionFile(char fileName[20],float *pl,int number_of_nodes)
{
        int i;
        FILE *fpn;
        fpn = fopen(fileName,"r");
        if (fpn == NULL)
        {
                printf("Cannot OPEN file, %s\n",fileName);
                exit(0);
        }
        else
        {
                char a[10];
                for (i=0; i<number_of_nodes; i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        if (fscanf(fpn,"%s",a) != EOF)
                        {
                                *pl = (float)atof(a);
                                pl += 3;
                        }
                }
                *pl = OUT_OF_POINTER * 10;
                printf("%s, Load data from file success.\n\n",fileName);
        }
        fclose(fpn);
}

void FileManager::writeFile(char fileName[20],float *pw,int size_of_k)
{
        int i;
        FILE *fpn;
        fpn = fopen(fileName,"w");
        if (fpn == NULL)
        {
                printf("Cannot OPEN file, %s\n",fileName);
                exit(0);
        }
        else
        {
                char str[20]  = "";
                for (i=0; i<size_of_k; i++)
                {
                        sprintf(str, "%f", *pw++);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                fprintf(fpn,"%s",str);
                fprintf(fpn,"\n");
            }
        }
        fclose(fpn);
}

void FileManager::loadPositionFile_gmsh(char fileName[20],float *pl,float nod,float elm)
{
        int i;
        FILE *fpn;
        fpn = fopen(fileName,"r");
        if (fpn == NULL)
        {
            printf("Cannot OPEN file, %s\n",fileName);
            exit(0);
        }
        else
        {
            char a[1000];
            for (i=0; i<((nod*4)+(elm*8)+5); i++)
            {
                if (fscanf(fpn,"%s",a) != EOF)
                {
                    *pl = (float)atof(a);
                    pl += 1;
                }
            }
            *pl = OUT_OF_POINTER * 10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            printf("%s, Load data from file success.\n\n",fileName);
        }
        fclose(fpn);
}

void FileManager::writeFile_gmsh(char fileName[20],float *pw,float nod)
{
        int i;
        FILE *fpn;
        fpn = fopen(fileName,"w");
        if (fpn == NULL)
        {
            printf("Cannot OPEN file, %s\n",fileName);
            exit(0);
        }
        else
        {
            for (i=0; i<nod; i++)
            {
                fprintf(fpn,"%f ",*pw++);
                fprintf(fpn,"\n");
            }
        }
        fclose(fpn);
}

void FileManager::showReadMe()
{
        FILE *fpn;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char fileName[] = "readme.txt";
fpn = fopen(fileName,"r");
if (fpn == NULL)
{
        printf("Cannot OPEN file, %s\n",fileName);
        exit(0);
}
else
{
        char a[100];
        while (true)
        {
                if (fscanf(fpn,"%s",a) != EOF)
                {
                        if (a[0] == '-' && a[1] == 'n') {
                                printf("\n");
                        }
                        else if (a[0] == '-' && a[1] == 't') {
                                printf(" ");
                        }
                        else {
                                printf("%s ",a);
                        }
                }
                else {
                        break;
                }
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        fclose(fpn);
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GLGraph.cpp

```
/*
 * GLGraph.cpp
 * Author Peeba_Merlin
*/

#include <math.h>

#include "Constant.h"
#include "FEM.h"

#include "GLGraph.h"

class FEM *finite;

HINSTANCE hInstance;
HWND hWnd;
HDC hDC;
HGLRC hRC;
bool keys[256];
const char appName[] = "OpenGLGraph";

VertexHandler* vertexhandler = NULL;

GLfloat angle = -45.0f;
GLfloat rotx = 1.0f;
GLfloat roty = 0.0f;
GLfloat rotz = 0.0f;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// translation
GLfloat transx = -30.0f;
GLfloat transy = -30.0f;
GLfloat transz = -60.0f;
bool isWireFrame = false;

GLfloat colorTable[64][3];

GLfloat txt_on_color[3] = {1.0f, 1.0f, 1.0f};
GLfloat txt_off_color[3] = {0.1f, 1.0f, 0.565f};
bool inputed_x = false;
bool inputed_y = false;

GLfloat button_cal_click = false;
GLfloat button_clear_click = false;

// data in text box
float x_value, y_value;
float z_value;
char x_str[10], y_str[10], z_str[10],
        x_dec[10], y_dec[10];
int cnt_input_x = 0,
        cnt_input_y = 0,
        cnt_dec_x   = 0,
        cnt_dec_y   = 0,
        cnt_z       = 0;
bool has_dot_x = false,
        has_dot_y = false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int size_x,size_y;
float max_ans,min_ans;


void Begin2D(void) {
        glMatrixMode(GL_PROJECTION);
        glPushMatrix();                                  // save projection settings
        glLoadIdentity();
        gluOrtho2D(0, 0, 800, 600);                      // left, right, bottom, top, near, far
        glMatrixMode(GL_MODELVIEW);
        glPushMatrix();                                  // save model view
settings
        glLoadIdentity();
        glDisable(GL_DEPTH_TEST);                        // disable depth test for 2D
}

void End2D(void) {
        glEnable(GL_DEPTH_TEST);                         // enable depth test for 3D
        glMatrixMode(GL_PROJECTION);
        glPopMatrix();                                   // load projection settings
        glMatrixMode(GL_MODELVIEW);
        glPopMatrix();                                   // load model view
settings
}

/* resize viewport when window size is changed */
GLvoid resizeGLWindow (GLsizei width, GLsizei height) {
        if (height == 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        height = 1;

    glViewport(0, 0, width, height);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    gluPerspective(90.0f, (GLfloat)width/(GLfloat)height, 0.01f, 400.0f);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

/* initialize opengl */
bool initOpenGL() {
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // background
    glClearDepth(1.0f); // clear depth
    glEnable(GL_DEPTH_TEST); // enable depth test
    glDepthFunc(GL_LEQUAL); // depth test function
    glFrontFace(GL_CW); //set front face to clockwise
    return true;
}

void initVertices() {
    vertexhandler = new VertexHandler();
    vertexhandler->test();
}

/* release all context & destroy window */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void destroyOpenGLWindow() {
        if (hRC) {
                if (!wglMakeCurrent(NULL, NULL)) {
                        errmsg("Error with release DC & RC");
                }
                if (!wglDeleteContext(hRC)) {
                        errmsg("Can't delete context");
                        hRC = NULL;
                }
        }

        if (!ReleaseDC(hWnd, hDC)) {
                errmsg("Can't release DC");
                hDC = NULL;
        }

        if (hWnd && !DestroyWindow(hWnd)) {
                errmsg("Can't Destroy Window");
                hWnd = 0;
        }

        if (!UnregisterClass(appName, hInstance)) {
                errmsg("Can't Unregister Class!");
                hInstance = 0;
        }
}

/* key checker function */
void checkKeys() {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (!(inputed_x || inputed_y)) {
        if (keys[VK_ESCAPE])
                destroyOpenGLWindow();

        if (keys[VK_UP])
                transy += 0.01f;

        if (keys[VK_DOWN])
                transy -= 0.01f;

        if (keys[VK_LEFT])
                transx -= 0.01f;

        if (keys[VK_RIGHT])
                transx += 0.01f;

        if (keys[VK_W]) {
                roty += 0.05f;
        }

        if (keys[VK_S]) {
                roty -= 0.05f;
        }

        if (keys[VK_A]) {
                rotx -= 0.05f;
        }

        if (keys[VK_D]) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            rotx += 0.05f;
        }

        if (keys[VK_Q]) {
            angle--;
        }

        if (keys[VK_E]) {
            angle++;
        }

        if (keys[VK_Z]) {
            transz -= 0.01f;
        }

        if (keys[VK_X]) {
            transz += 0.01f;
        }
}
else {
        int max_input = 8;
        if (keys[48]) {
            //printf("0\n");
            if (inputed_x && (cnt_input_x+cnt_dec_x)<=max_input) {
                if (!has_dot_x) {
                    if (cnt_input_x != 0) {
                        x_value *= 10;
                        x_str[cnt_input_x] = '0';
                        cnt_input_x++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
            }
            else {
                x_dec[cnt_dec_x++] = '0';
            }
        }
        if (inputed_y && (cnt_input_y+cnt_dec_y)<=max_input) {
            if (!has_dot_y) {
                if (cnt_input_y != 0) {
                    y_value *= 10;
                    y_str[cnt_input_y] = '0';
                    cnt_input_y++;
                }
            }
            else {
                y_dec[cnt_dec_y++] = '0';
            }
        }
    }
    if (keys[49]) {
        //printf("1\n");
        if (inputed_x && (cnt_input_x+cnt_dec_x)<=max_input) {
            if (!has_dot_x) {
                x_value *= 10;
                x_value += 1;
                x_str[cnt_input_x] = '1';
                cnt_input_x++;
            }
            else {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                x_dec[cnt_dec_x++] = '1';
            }
        }
        if (inputed_y && (cnt_input_y+cnt_dec_y)<=max_input) {
            if (!has_dot_y) {
                y_value *= 10;
                y_value += 1;
                y_str[cnt_input_y] = '1';
                cnt_input_y++;
            }
            else {
                y_dec[cnt_dec_y++] = '1';
            }
        }
    }
    if (keys[50]) {
        //printf("2\n");
        if (inputed_x && (cnt_input_x+cnt_dec_x)<=max_input) {
            if (!has_dot_x) {
                x_value *= 10;
                x_value += 2;
                x_str[cnt_input_x] = '2';
                cnt_input_x++;
            }
            else {
                x_dec[cnt_dec_x++] = '2';
            }
        }
        if (inputed_y && (cnt_input_y+cnt_dec_y)<=max_input) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if (!has_dot_y) {
                        y_value *= 10;
                        y_value += 2;
                        y_str[cnt_input_y] = '2';
                        cnt_input_y++;
                }
                else {
                        y_dec[cnt_dec_y++] = '2';
                }
        }
}
if (keys[51]) {
        //printf("3\n");
        if (inputed_x && (cnt_input_x+cnt_dec_x)<=max_input) {
                if (!has_dot_x) {
                        x_value *= 10;
                        x_value += 3;
                        x_str[cnt_input_x] = '3';
                        cnt_input_x++;
                }
                else {
                        x_dec[cnt_dec_x++] = '3';
                }
        }
        if (inputed_y && (cnt_input_y+cnt_dec_y)<=max_input) {
                if (!has_dot_y) {
                        y_value *= 10;
                        y_value += 3;
                        y_str[cnt_input_y] = '3';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                cnt_input_y++;
            }
            else {
                y_dec[cnt_dec_y++] = '3';
            }
        }
    }
    if (keys[52]) {
        //printf("4\n");
        if (inputed_x && (cnt_input_x+cnt_dec_x)<=max_input) {
            if (!has_dot_x) {
                x_value *= 10;
                x_value += 4;
                x_str[cnt_input_x] = '4';
                cnt_input_x++;
            }
            else {
                x_dec[cnt_dec_x++] = '4';
            }
        }
        if (inputed_y && (cnt_input_y+cnt_dec_y)<=max_input) {
            if (!has_dot_y) {
                y_value *= 10;
                y_value += 4;
                y_str[cnt_input_y] = '4';
                cnt_input_y++;
            }
            else {
                y_dec[cnt_dec_y++] = '4';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
        }
}
if (keys[53]) {
        //printf("5\n");
        if (inputed_x && (cnt_input_x+cnt_dec_x)<=max_input) {
                if (!has_dot_x) {
                        x_value *= 10;
                        x_value += 5;
                        x_str[cnt_input_x] = '5';
                        cnt_input_x++;
                }
                else {
                        x_dec[cnt_dec_x++] = '5';
                }
        }
        if (inputed_y && (cnt_input_y+cnt_dec_y)<=max_input) {
                if (!has_dot_y) {
                        y_value *= 10;
                        y_value += 5;
                        y_str[cnt_input_y] = '5';
                        cnt_input_y++;
                }
                else {
                        y_dec[cnt_dec_y++] = '5';
                }
        }
}
if (keys[54]) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        //printf("6\n");
        if (inputed_x && (cnt_input_x+cnt_dec_x)<=max_input) {
                if (!has_dot_x) {
                        x_value *= 10;
                        x_value += 6;
                        x_str[cnt_input_x] = '6';
                        cnt_input_x++;
                }
                else {
                        x_dec[cnt_dec_x++] = '6';
                }
        }
        if (inputed_y && (cnt_input_y+cnt_dec_y)<=max_input) {
                if (!has_dot_y) {
                        y_value *= 10;
                        y_value += 6;
                        y_str[cnt_input_y] = '6';
                        cnt_input_y++;
                }
                else {
                        y_dec[cnt_dec_y++] = '6';
                }
        }
}
if (keys[55]) {
        //printf("7\n");
        if (inputed_x && (cnt_input_x+cnt_dec_x)<=max_input) {
                if (!has_dot_x) {
                        x_value *= 10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                x_value += 7;
                x_str[cnt_input_x] = '7';
                cnt_input_x++;
            }
            else {
                x_dec[cnt_dec_x++] = '7';
            }
        }
        if (inputed_y && (cnt_input_y+cnt_dec_y)<=max_input) {
            if (!has_dot_y) {
                y_value *= 10;
                y_value += 7;
                y_str[cnt_input_y] = '7';
                cnt_input_y++;
            }
            else {
                y_dec[cnt_dec_y++] = '7';
            }
        }
    }
    if (keys[56]) {
        //printf("8\n");
        if (inputed_x && (cnt_input_x+cnt_dec_x)<=max_input) {
            if (!has_dot_x) {
                x_value *= 10;
                x_value += 8;
                x_str[cnt_input_x] = '8';
                cnt_input_x++;
            }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                else {
                        x_dec[cnt_dec_x++] = '8';
                }
        }
        if (inputed_y && (cnt_input_y+cnt_dec_y)<=max_input) {
                if (!has_dot_y) {
                        y_value *= 10;
                        y_value += 8;
                        y_str[cnt_input_y] = '8';
                        cnt_input_y++;
                }
                else {
                        y_dec[cnt_dec_y++] = '8';
                }
        }
}
if (keys[57]) {
        //printf("9\n");
        if (inputed_x && (cnt_input_x+cnt_dec_x)<=max_input) {
                if (!has_dot_x) {
                        x_value *= 10;
                        x_value += 9;
                        x_str[cnt_input_x] = '9';
                        cnt_input_x++;
                }
                else {
                        x_dec[cnt_dec_x++] = '9';
                }
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (inputed_y && (cnt_input_y+cnt_dec_y)<=max_input) {
                    if (!has_dot_y) {
                            y_value *= 10;
                            y_value += 9;
                            y_str[cnt_input_y] = '9';
                            cnt_input_y++;
                    }
                    else {
                            y_dec[cnt_dec_y++] = '9';
                    }
            }
    }
    if (keys[190]) {  // dot
            if (inputed_x && (cnt_input_x+cnt_dec_x)<=max_input) {
                    if (!has_dot_x) {
                            has_dot_x = true;
                            if (cnt_input_x == 0) {
                                    x_str[cnt_input_x++] = '0';
                            }
                            x_str[cnt_input_x++] = '.';
                    }
            }
            if (inputed_y && (cnt_input_y+cnt_dec_y)<=max_input) {
                    if (!has_dot_y) {
                            has_dot_y = true;
                            if (cnt_input_y == 0) {
                                    y_str[cnt_input_y++] = '0';
                            }
                            y_str[cnt_input_y++] = '.';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                }
            }
        }
}

void InitColorTable() {
        GLfloat r, g, b;
        GLfloat inc = 1.0f / 32.0f;
        r = 0.0f; g = 0.0f; b = 1.0f;
        for (int i=0; i<64; i++) {
            colorTable[i][0] = r;
            colorTable[i][1] = g;
            colorTable[i][2] = b;

            if (i < 32) {
                r = 0.0f;
                g += inc;
                b -= inc;
            } else if ( i < 64 ) {
                g -= inc;
                r += inc;
                b = 0.0f;
            }
        }
}

void GetColor64f(GLfloat* dest_color, GLfloat color, GLfloat vMin, GLfloat vMax)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        GLfloat vmin;
        GLfloat inc = (vMax - vMin) / 64;

        if ( vMin < 0) {
                vmin = -vMin;
        } else {
                vmin = vMin;
        }

        int index = (int)( (color + vmin)  / inc);

        if ( color >= vMax ) {
                index = 63;
        } else if ( color <= vMin ) {
                index = 0;
        }

        dest_color[0] = colorTable[index][0];
        dest_color[1] = colorTable[index][1];
        dest_color[2] = colorTable[index][2];
}

void DrawColor64Scale(void) {
        GLfloat y = -0.925f;
        Begin2D();
        glBegin(GL_QUAD_STRIP);
        for (int i=0; i<63; i++) {
                glColor3fv(colorTable[i]);
                glVertex2f(0.888f, y/2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            glVertex2f(0.988f, y/2);
            y += 0.03f;
        }
        glEnd();
        End2D();
}

void drawHeader() {
        GLfloat hdr_x1 = -1.0f,
                hdr_y1 = 1.0f,
                hdr_x2 = 1.0f,
                hdr_y2 = 0.6f;
        Begin2D();
        glBegin(GL_POLYGON);
            glColor3f(0.5f, 0.8f, 0.2f);
            glVertex2f(hdr_x1, hdr_y1);
            glVertex2f(hdr_x2, hdr_y1);
            glVertex2f(hdr_x2, hdr_y2);
            glVertex2f(hdr_x1, hdr_y2);
        glEnd();
        // Text
        int i;
        // Header
        char header[] = "POISSON SOLUTION GRAPH";
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glColor3f(0.0f, 0.0f, 0.0f);
        glRasterPos2f(-0.3f, 0.9f);
        for (i=0; i<sizeof(header); i++) {
            glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, header[i]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        // Body Message
        char body1[] = "There are Q,W,E,A,S,D,Z and X Buttons for control graph view";
        char body2[] = "and use Arrow Buttons to move graph position.";
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glColor3f(0.0f, 0.0f, 0.0f);
        glRasterPos2f(-0.45f, 0.8f);
        for (i=0; i<sizeof(body1); i++) {
                glutBitmapCharacter(GLUT_BITMAP_8_BY_13, body1[i]);
        }
        glRasterPos2f(-0.33f, 0.75f);
        for (i=0; i<sizeof(body2); i++) {
                glutBitmapCharacter(GLUT_BITMAP_8_BY_13, body2[i]);
        }
        End2D();
}

void drawCalButton() {
        GLfloat color_upper  = 0.9f,
                        color_bottom = 0.4f,
                        color_body   = 0.7f;
        GLfloat btnCal_x1 = -0.3f,
                        btnCal_y1 = -0.75f,
                        btnCal_x2 = -0.1f,
                        btnCal_y2 = -0.85f;
        if (button_cal_click) {
                color_upper  = 0.4f;
                color_bottom = 0.9f;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else {
        color_upper  = 0.9f,
        color_bottom = 0.4f,
        color_body   = 0.7f;
}
Begin2D();
glBegin(GL_POLYGON);
        glColor3f(color_upper, color_upper, color_upper);
        glVertex2f(btnCal_x1-0.01f, btnCal_y1+0.01f);
        glVertex2f(btnCal_x2, btnCal_y1+0.01f);
        glVertex2f(btnCal_x2, btnCal_y2-0.01f);
        glVertex2f(btnCal_x1-0.01f, btnCal_y2-0.01f);
glEnd();
glBegin(GL_POLYGON);
        glColor3f(color_body, color_body, color_body);
        glVertex2f(btnCal_x1, btnCal_y1);
        glVertex2f(btnCal_x2, btnCal_y1);
        glVertex2f(btnCal_x2, btnCal_y2);
        glVertex2f(btnCal_x1, btnCal_y2);
glEnd();
glBegin(GL_POLYGON);
        glColor3f(color_bottom, color_bottom, color_bottom);
        glVertex2f(btnCal_x1, btnCal_y2);
        glVertex2f(btnCal_x2, btnCal_y2);
        glVertex2f(btnCal_x2, btnCal_y2-0.01f);
        glVertex2f(btnCal_x1, btnCal_y2-0.01f);
glEnd();
glBegin(GL_POLYGON);
        glColor3f(color_bottom, color_bottom, color_bottom);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            glVertex2f(btnCal_x2, btnCal_y1+0.01f);
            glVertex2f(btnCal_x2+0.01f, btnCal_y1+0.01f);
            glVertex2f(btnCal_x2+0.01f, btnCal_y2-0.01f);
            glVertex2f(btnCal_x2, btnCal_y2-0.01f);
        glEnd();
        // Caption, "Cal"
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glColor3f(0.0f, 0.0f, 0.0f);
        glRasterPos2f(btnCal_x1+0.07f, btnCal_y1-0.06f);
        glutBitmapCharacter(GLUT_BITMAP_8_BY_13, 'C');
        glutBitmapCharacter(GLUT_BITMAP_8_BY_13, 'a');
        glutBitmapCharacter(GLUT_BITMAP_8_BY_13, 'l');
        End2D();
}

void drawClearButton() {
        GLfloat color_upper = 0.9f,
                        color_bottom = 0.4f,
                        color_body  = 0.7f;
        GLfloat btnClear_x1 = 0.0f,
                        btnClear_y1 = -0.75f,
                        btnClear_x2 = 0.2f,
                        btnClear_y2 = -0.85f;
        Begin2D();
        if (button_clear_click) {
                color_upper  = 0.4f;
                color_bottom = 0.9f;
        }
        else {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        color_upper  = 0.9f,
        color_bottom = 0.4f,
        color_body   = 0.7f;
}
glBegin(GL_POLYGON);
        glColor3f(color_upper, color_upper, color_upper);
        glVertex2f(btnClear_x1-0.01f, btnClear_y1+0.01f);
        glVertex2f(btnClear_x2, btnClear_y1+0.01f);
        glVertex2f(btnClear_x2, btnClear_y2-0.01f);
        glVertex2f(btnClear_x1-0.01f, btnClear_y2-0.01f);
glEnd();
glBegin(GL_POLYGON);
        glColor3f(color_body, color_body, color_body);
        glVertex2f(btnClear_x1, btnClear_y1);
        glVertex2f(btnClear_x2, btnClear_y1);
        glVertex2f(btnClear_x2, btnClear_y2);
        glVertex2f(btnClear_x1, btnClear_y2);
glEnd();
glBegin(GL_POLYGON);
        glColor3f(color_bottom, color_bottom, color_bottom);
        glVertex2f(btnClear_x1, btnClear_y2);
        glVertex2f(btnClear_x2, btnClear_y2);
        glVertex2f(btnClear_x2, btnClear_y2-0.01f);
        glVertex2f(btnClear_x1, btnClear_y2-0.01f);
glEnd();
glBegin(GL_POLYGON);
        glColor3f(color_bottom, color_bottom, color_bottom);
        glVertex2f(btnClear_x2, btnClear_y1+0.01f);
        glVertex2f(btnClear_x2+0.01f, btnClear_y1+0.01f);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            glVertex2f(btnClear_x2+0.01f, btnClear_y2-0.01f);
            glVertex2f(btnClear_x2, btnClear_y2-0.01f);
        glEnd();
        // Caption, "Clear"
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glColor3f(0.0f, 0.0f, 0.0f);
        glRasterPos2f(btnClear_x1+0.05f, btnClear_y1-0.06f);
        glutBitmapCharacter(GLUT_BITMAP_8_BY_13, 'C');
        glutBitmapCharacter(GLUT_BITMAP_8_BY_13, 'l');
        glutBitmapCharacter(GLUT_BITMAP_8_BY_13, 'e');
        glutBitmapCharacter(GLUT_BITMAP_8_BY_13, 'a');
        glutBitmapCharacter(GLUT_BITMAP_8_BY_13, 'r');
        End2D();
}

void drawInputPanel() {
        GLfloat black[] = {0.0f,0.0f,0.0f};
        GLfloat gray[]  = {0.8f,0.8f,0.8f};
        Begin2D();
        // show error
        if (finite->getAns(0)>pow(10,5)*9 || finite->getAns(0)<-pow(10,5)*9) {
                int i;
                char err_msg[] = "ERROR : No graph!!!";
                glClearColor(1.0, 1.0, 1.0, 1.0);
                glColor3f(0.0f, 0.0f, 0.0f);
                glRasterPos2f(-0.3f, 0.0f);
                for (i=0; i<sizeof(err_msg); i++) {
                        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, err_msg[i]);
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
// Background
glBegin(GL_POLYGON);
        glColor3f(0.5f, 0.8f, 0.2f);
        glVertex2f(-1.0f, -0.6f);
        glVertex2f(1.0f, -0.6f);
        glVertex2f(1.0f, -1.0f);
        glVertex2f(-1.0f, -1.0f);
glEnd();
// Text Box and Label
// inputting x
GLfloat txtX_x1 = -0.7,
                txtX_y1 = -0.7,
                txtX_x2 = -0.5,
                txtX_y2 = -0.78;
glClearColor(1.0, 1.0, 1.0, 1.0);
glColor3f(0.0f, 0.0f, 0.0f);
glRasterPos2f(txtX_x1-0.15f, txtX_y1-0.07f);
glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, 'X');
glRasterPos2f(txtX_x1-0.05f, txtX_y1-0.07f);
glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, ':');
glBegin(GL_POLYGON);
        if (inputed_x)
        {       glColor3fv(txt_on_color);  }
        else
        {       glColor3fv(txt_off_color); }
        glVertex2f(txtX_x1, txtX_y1);
        glVertex2f(txtX_x2, txtX_y1);
        glVertex2f(txtX_x2, txtX_y2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        glVertex2f(txtX_x1, txtX_y2);
glEnd();
glBegin(GL_LINES);
        glColor3fv(black);
        glVertex2f(txtX_x1, txtX_y1);
        glVertex2f(txtX_x2, txtX_y1);
        glVertex2f(txtX_x1, txtX_y2);
        glVertex2f(txtX_x1, txtX_y1);
        glColor3fv(gray);
        glVertex2f(txtX_x2, txtX_y1);
        glVertex2f(txtX_x2, txtX_y2);
        glVertex2f(txtX_x2, txtX_y2);
        glVertex2f(txtX_x1, txtX_y2);
glEnd();
// inputting y
GLfloat txtY_x1 = -0.7,
                txtY_y1 = -0.83,
                txtY_x2 = -0.5,
                txtY_y2 = -0.91;
glClearColor(1.0, 1.0, 1.0, 1.0);
glColor3f(0.0f, 0.0f, 0.0f);
glRasterPos2f(txtY_x1-0.15f, txtY_y1-0.07f);
glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, 'Y');
glRasterPos2f(txtY_x1-0.05f, txtY_y1-0.07f);
glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, ':');
glBegin(GL_POLYGON);
        if (inputed_y)
        {       glColor3fv(txt_on_color); }
        else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {       glColor3fv(txt_off_color); }
        glVertex2f(txtY_x1, txtY_y1);
        glVertex2f(txtY_x2, txtY_y1);
        glVertex2f(txtY_x2, txtY_y2);
        glVertex2f(txtY_x1, txtY_y2);
glEnd();
glBegin(GL_LINES);
        glColor3fv(black);
        glVertex2f(txtY_x1, txtY_y1);
        glVertex2f(txtY_x2, txtY_y1);
        glVertex2f(txtY_x1, txtY_y2);
        glVertex2f(txtY_x1, txtY_y1);
        glColor3fv(gray);
        glVertex2f(txtY_x2, txtY_y1);
        glVertex2f(txtY_x2, txtY_y2);
        glVertex2f(txtY_x2, txtY_y2);
        glVertex2f(txtY_x1, txtY_y2);
glEnd();
// output
GLfloat txtZ_x1 = 0.5,
                txtZ_y1 = -0.75,
                txtZ_x2 = 0.7,
                txtZ_y2 = -0.85;
glClearColor(1.0, 1.0, 1.0, 1.0);
glColor3f(0.0f, 0.0f, 0.0f);
glRasterPos2f(txtZ_x1-0.15f, txtZ_y1-0.07f);
glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, 'Z');
glRasterPos2f(txtZ_x1-0.05f, txtZ_y1-0.07f);
glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, ':');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
glBegin(GL_POLYGON);
        glColor3fv(gray);
        glVertex2f(txtZ_x1, txtZ_y1);
        glVertex2f(txtZ_x2, txtZ_y1);
        glVertex2f(txtZ_x2, txtZ_y2);
        glVertex2f(txtZ_x1, txtZ_y2);
glEnd();
glBegin(GL_LINES);
        glColor3fv(black);
        glVertex2f(txtZ_x1, txtZ_y1);
        glVertex2f(txtZ_x2, txtZ_y1);
        glVertex2f(txtZ_x2, txtZ_y1);
        glVertex2f(txtZ_x2, txtZ_y2);
        glVertex2f(txtZ_x2, txtZ_y2);
        glVertex2f(txtZ_x1, txtZ_y2);
        glVertex2f(txtZ_x1, txtZ_y2);
        glVertex2f(txtZ_x1, txtZ_y1);
glEnd();
End2D();
// Button
drawCalButton();
drawClearButton();
// data in Text Box
Begin2D();
GLfloat gap = 0.02f;
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glColor3f(0.0f, 0.0f, 0.0f);
for (int ix=0; ix<cnt_input_x; ix++) {
        // data in Text Box, X
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            glRasterPos2f(txtX_x1+0.01f+ix*gap, txtX_y1-0.06f);
            glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, x_str[ix]);
        }
        for (int ix_dec=0; ix_dec<cnt_dec_x; ix_dec++) {
            // data, decimal, in Text Box, X
            glRasterPos2f(txtX_x1+0.01f+ix_dec*gap+ix*gap, txtX_y1-0.06f);
            glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, x_dec[ix_dec]);
        }
        for (int iy=0; iy<cnt_input_y; iy++) {
            // data in Text Box, Y
            glRasterPos2f(txtY_x1+0.01f+iy*gap, txtY_y1-0.06f);
            glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, y_str[iy]);
        }
        for (int iy_dec=0; iy_dec<cnt_dec_y; iy_dec++) {
            // data, decimal, in Text Box, Y
            glRasterPos2f(txtY_x1+0.01f+iy_dec*gap+iy*gap, txtY_y1-0.06f);
            glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, y_dec[iy_dec]);
        }
        for (int iz=0; iz<cnt_z; iz++) {
            // data in Text Box, Z
            glRasterPos2f(txtZ_x1+0.01f+iz*gap, txtZ_y1-0.07f);
            glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, z_str[iz]);
        }
        End2D();
}

GLvoid renderScene() {
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        glLoadIdentity();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
glTranslatef(transx, transy, transz);
glRotatef(angle, rotx, roty, rotz);

DrawColor64Scale();

glScalef(10.0f, 10.0f, 0.05f);
//draw axes
glBegin(GL_LINES);
        glColor3f(1.0f, 0.0f, 0.0f);
        glVertex3f(-1.0f, 0.0f, 0.0f);
        glVertex3f(5.0f, 0.0f, 0.0f);
        glColor3f(0.0f, 1.0f, 0.0f);
        glVertex3f(0.0f, 5.0f, 0.0f);
        glVertex3f(0.0f, -0.5f, 0.0f);
        glColor3f(0.0f, 0.0f, 1.0f);
        glVertex3f(0.0f, 0.0f, -5.0f);
        glVertex3f(0.0f, 0.0f, 5.0f);

/* draw graph */
int i,j=0;
Vertex *v = NULL;
float color[3];
for (i=1; i<(size_y+1); i++) {
        for (; j < (size_x-1)*i + (i-1); j++) {
                v = &vertexhandler->getVetex(j);
                GetColor64f(color, v->z, min_ans, max_ans);
                glColor3fv(color);
                glVertex3f(v->x, v->y, v->z);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                //
                v = &vertexhandler->getVetex(j+1);
                GetColor64f(color, v->z, min_ans, max_ans);
                glColor3fv(color);
                glVertex3f(v->x, v->y, v->z);
        }
        j++;
}

for (i=0; i<(size_x+1); i++) {
        for (j = i; j < size_x * (size_y-1); j+= size_x) {
                v = &vertexhandler->getVetex(j);
                GetColor64f(color, v->z, min_ans, max_ans);
                glColor3fv(color);
                glVertex3f(v->x, v->y, v->z);
                //
                v = &vertexhandler->getVetex(j+size_x);
                GetColor64f(color, v->z, min_ans, max_ans);
                glColor3fv(color);
                glVertex3f(v->x, v->y, v->z );
        }
        j++;
}

glEnd();

// set point
glBegin(GL_POLYGON);
        glColor3f(0.0f, 0.0f, 0.0f);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            glVertex3f(x_value-0.1f, y_value+0.1f, z_value);
            glVertex3f(x_value+0.1f, y_value+0.1f, z_value);
            glVertex3f(x_value+0.1f, y_value-0.1f, z_value);
            glVertex3f(x_value-0.1f, y_value-0.1f, z_value);
        glEnd();

        // 2D
        drawHeader();
        drawInputPanel();
}

/* create window & initialize opengl */
bool createOpenGLWindow(char *title, int width, int height, int bits) {
        WNDCLASSEX wc;
        int nPixelFormat;

        hInstance = GetModuleHandle(NULL);
        wc.cbClsExtra = NULL;
        wc.cbWndExtra = NULL;
        wc.cbSize = sizeof(WNDCLASSEX);
        wc.hbrBackground = CreateSolidBrush(RGB(0,0,0));
        wc.hCursor = LoadCursor(NULL, IDC_ARROW);
        wc.hIcon = LoadIcon(NULL, IDI_WINLOGO);
        wc.hIconSm = LoadIcon(NULL, IDI_APPLICATION);
        wc.hInstance = hInstance;
        wc.lpfnWndProc = (WNDPROC)WndProc;
        wc.lpszClassName = appName;
        wc.lpszMenuName = NULL;
        wc.style = CS_VREDRAW | CS_HREDRAW;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (!RegisterClassEx(&wc)) {
        errmsg("Can't Register Class!");
        return false;
}

hWnd = CreateWindowEx(WS_EX_OVERLAPPEDWINDOW,
                      appName,
                      title,
                      WS_OVERLAPPEDWINDOW,
                      0, 0,
                      width, height,
                      NULL,
                      NULL,
                      hInstance,
                      NULL);

static  PIXELFORMATDESCRIPTOR pfd=      // pfd Tells Windows How We Want
                                        //Things To Be
{
        sizeof(PIXELFORMATDESCRIPTOR),  // Size Of This Pixel Format Descriptor
        1,                              // Version Number
        PFD_DRAW_TO_WINDOW |            // Format Must Support Window
        PFD_SUPPORT_OPENGL |            // Format Must Support OpenGL
        PFD_DOUBLEBUFFER,               // Must Support Double Buffering
        PFD_TYPE_RGBA,                  // Request An RGBA Format
        bits,                           // Select Our Color Depth
        0, 0, 0, 0, 0, 0,               // Color Bits Ignored
        0,                              // No Alpha Buffer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        0,                              // Shift Bit Ignored
        0,                              // No Accumulation Buffer
        0, 0, 0, 0,                     // Accumulation Bits Ignored
        16,                             // 16Bit Z-Buffer (Depth Buffer)
        0,                              // No Stencil Buffer
        0,                              // No Auxiliary Buffer
        PFD_MAIN_PLANE,                 // Main Drawing Layer
        0,                              // Reserved
        0, 0, 0                         // Layer Masks Ignored
};

if (!(hDC = GetDC(hWnd))) {
        errmsg("Can't Get Device Context!");
        return false;
}

if (!(nPixelFormat = ChoosePixelFormat(hDC, &pfd))) {
        errmsg("Can't Choose PixelFormat");
        return false;
}

if (!SetPixelFormat(hDC, nPixelFormat, &pfd)) {
        errmsg("Can't Set PixelFormat");
        return false;
}

if (!(hRC = wglCreateContext(hDC))) {
        errmsg("Can't Create Context");
        return false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }

        if (!wglMakeCurrent(hDC, hRC)) {
                errmsg("Error with wglMakeCurrent");
                return false;
        }

        if (hWnd == NULL) {
                errmsg("Can't Create Window!");
                return false;
        }

        if (!initOpenGL()) {
                errmsg("Can't Initialize OpenGL");
                return false;
        }

        ShowWindow(hWnd, SW_SHOW);
        SetForegroundWindow(hWnd);
        UpdateWindow(hWnd);
        SetFocus(hWnd);
        resizeGLWindow(width, height);
        RedrawWindow(hWnd, NULL, NULL, RDW_INTERNALPAINT);

        return true;
}

char chgToC(int iin) {
        switch (iin) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 0:
                return '0';
        case 1:
                return '1';
        case 2:
                return '2';
        case 3:
                return '3';
        case 4:
                return '4';
        case 5:
                return '5';
        case 6:
                return '6';
        case 7:
                return '7';
        case 8:
                return '8';
        case 9:
                return '9';
        }
        return '0';
}

int chgToI(char cch) {
        switch (cch) {
        case '0':
                return 0;
        case '1':
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            return 1;
        case '2':
            return 2;
        case '3':
            return 3;
        case '4':
            return 4;
        case '5':
            return 5;
        case '6':
            return 6;
        case '7':
            return 7;
        case '8':
            return 8;
        case '9':
            return 9;
        }
        return 0;
}

LRESULT CALLBACK WndProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM
lParam) {
        switch (uMsg) {
        case WM_CREATE:
            initVertices();
            break;
        case WM_PAINT:
            renderScene();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SwapBuffers(hDC);
        break;
case WM_LBUTTONDOWN:
        POINT point;
        GetCursorPos(&point);
        // text box of X
        if ((point.x>=150) && (point.x<=256) && (point.y>=620) && (point.y<=650)) {
                inputed_x = true;
        }
        else {
                inputed_x = false;
        }
        // text box of Y
        if ((point.x>=150) && (point.x<=256) && (point.y>=670) && (point.y<=700)) {
                inputed_y = true;
        }
        else {
                inputed_y = false;
        }
        // button 'Cal'
        if ((point.x>=354) && (point.x<=464) && (point.y>=640) && (point.y<=681)) {
                button_cal_click = true;
                cnt_z = 0;
                x_value = (int)x_value;
                y_value = (int)y_value;
                for (int ix_dec=0; ix_dec<cnt_dec_x; ix_dec++) {
                        x_value += chgToI(x_dec[ix_dec])*pow(10,-(ix_dec+1));
                }
                for (int iy_dec=0; iy_dec<cnt_dec_y; iy_dec++) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                y_value += chgToI(y_dec[iy_dec])*pow(10,-(iy_dec+1));
        }
// compute z_value
        float dx = finite->getDiffX(),
                dy = finite->getDiffY();
        float chk_x, chk_y;
        chk_x = x_value/dx;
        chk_y = y_value/dy;
        // keep decimal
        chk_x = chk_x - (int)chk_x;
        chk_y = chk_y - (int)chk_y;
        // check decimal is or not
        bool is_int_x, is_int_y;
        if ((int)(chk_x*pow(10,5)) == 0) {
                is_int_x = true;
        }
        else {
                is_int_x = false;
        }
        if ((int)(chk_y*pow(10,5)) == 0) {
                is_int_y = true;
        }
        else {
                is_int_y = false;
        }
        //
        if (is_int_y && is_int_x) {
                int pos = (x_value/dx+1)*(y_value/dy+1)
                        + (size_x-(x_value/dx+1))*y_value/dy;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                z_value = finite->getAns(pos-1);
        }
        else {
                z_value = finite->calAnsFromRectXY(x_value,y_value);
        }
// end compute z_value
        // get output integer into string data
        for (int i=0; i<5; i++) {
                if (z_value < 1 && i == 0) {
                        z_str[cnt_z] = '0';
                        cnt_z++;
                        break;
                }
                if (z_value < 10 && (int)z_value > 0) {
                        z_str[cnt_z] = chgToC((int)(z_value));
                        z_value -= (int)z_value;
                        cnt_z++;
                }
                else if (z_value < pow(10,2) && (int)z_value > 0) {
                        z_str[cnt_z] = chgToC((int)(z_value/pow(10,1)));
                        z_value -= ((int)(z_value/10))*10;
                        cnt_z++;
                        if (z_value < 1) {
                                z_str[cnt_z] = '0';
                                cnt_z++;
                        }
                }
                else if (z_value < pow(10,3) && (int)z_value > 0) {
                        z_str[cnt_z] = chgToC((int)(z_value/pow(10,2)));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                z_value -= ((int)(z_value/pow(10,2)))*pow(10,2);
                cnt_z++;
                if (z_value < 10) {
                        z_str[cnt_z] = '0';
                        cnt_z++;
                }
        }
        else if (z_value < pow(10,4) && (int)z_value > 0) {
                z_str[cnt_z] = chgToC((int)(z_value/pow(10,3)));
                z_value -= ((int)(z_value/pow(10,3)))*pow(10,3);
                cnt_z++;
        }
        else if (z_value < pow(10,5) && (int)z_value > 0) {
                z_str[cnt_z] = chgToC((int)(z_value/pow(10,4)));
                z_value -= ((int)(z_value/pow(10,4)))*pow(10,4);
                cnt_z++;
        }
}
// get output decimal into string data
if (is_int_y && is_int_x) {
        int pos = (x_value/dx+1)*(y_value/dy+1)
                + (size_x-(x_value/dx+1))*y_value/dy;
        z_value = finite->getAns(pos-1);
}
else {
        z_value = finite->calAnsFromRectXY(x_value,y_value);
}
z_str[cnt_z] = '.';
cnt_z++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            z_value -= (int)z_value;
            for (int id=0; id<3; id++) {
                z_str[cnt_z] = chgToC((int)(z_value*10));
                z_value *= 10;
                z_value -= (int)z_value;
                cnt_z++;
            }
        }
        else {
            button_cal_click = false;
        }
        // button 'Clear'
        if ((point.x>=507) && (point.x<=618) && (point.y>=640) && (point.y<=681)) {
            button_clear_click = true;
            has_dot_x = false;
            has_dot_y = false;
            x_value = 0;
            y_value = 0;
            for (int ix=0; ix<cnt_input_x; ix++) x_str[ix] = '\0';
            for (int ix_dec=0; ix_dec<cnt_dec_x; ix_dec++) x_dec[ix_dec] = '\0';
            for (int iy=0; iy<cnt_input_y; iy++) y_str[iy] = '\0';
            for (int iy_dec=0; iy_dec<cnt_dec_y; iy_dec++) y_dec[iy_dec] = '\0';
            for (int iz=0; iz<cnt_z; iz++) z_str[iz] = '\0';
            cnt_input_x = 0;
            cnt_input_y = 0;
            cnt_dec_x = 0;
            cnt_dec_y = 0;
            cnt_z = 0;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else {
                button_clear_click = false;
        }
        RedrawWindow(hWnd, NULL, NULL, RDW_INTERNALPAINT);
        break;
case WM_LBUTTONUP:
        button_cal_click = false;
        button_clear_click = false;
        RedrawWindow(hWnd, NULL, NULL, RDW_INTERNALPAINT);
        break;
case WM_KEYUP:
        keys[wParam] = false;
        break;
case WM_KEYDOWN:
        keys[wParam] = true;
        checkKeys();
        RedrawWindow(hWnd, NULL, NULL, RDW_INTERNALPAINT);
        break;
case WM_SIZE:
        resizeGLWindow(LOWORD(lParam), HIWORD(lParam));
        break;
case WM_CLOSE:
        destroyOpenGLWindow();
        delete vertexhandler;
case WM_DESTROY:
        // clear all data in graph
        has_dot_x = false;
        has_dot_y = false;
        x_value = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            y_value = 0;
            for (int ix=0; ix<cnt_input_x; ix++) x_str[ix] = '\0';
            for (int ix_dec=0; ix_dec<cnt_dec_x; ix_dec++) x_dec[ix_dec] = '\0';
            for (int iy=0; iy<cnt_input_y; iy++) y_str[iy] = '\0';
            for (int iy_dec=0; iy_dec<cnt_dec_y; iy_dec++) y_dec[iy_dec] = '\0';
            for (int iz=0; iz<cnt_z; iz++) z_str[iz] = '\0';
            cnt_input_x = 0;
            cnt_input_y = 0;
            cnt_dec_x = 0;
            cnt_dec_y = 0;
            cnt_z = 0;
            //
            PostQuitMessage(0);
            break;
        }
        return DefWindowProc(hWnd, uMsg, wParam, lParam);
}

int showGraph(int x,int y,float maxAns,float minAns,FEM *fini) {
        size_x = x;
        size_y = y;
        max_ans = maxAns;
        min_ans = minAns;

        finite = fini;

        MSG msg;

        InitColorTable();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        createOpenGLWindow("OpenGLGraph", 800, 600, 16);
        while (GetMessage(&msg, hWnd, 0, 0) > 0) {
                TranslateMessage(&msg);
                DispatchMessage(&msg);
        }

        return msg.wParam;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rectangle.cpp

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

#include "Constant.h"
#include "FEM.h"
#include "Rectangle.h"

// constructor
Rectangle::Rectangle()
{
}

void Rectangle::run(FEM *p_fem)
{
        int i,j;
        char ch1;
        int ch2;
        int size_x,size_y;
        float df_x,df_y;
        float g_value;
        float side_value[4];        // 0 = bottom
        int type[4];                        // 1 = left
                                                    // 2 = right
                                                    // 3 = upper
        char x_file[] = "X.txt",
             y_file[] = "Y.txt";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
p_fem->setBeginElementIndex(1);

printf("Set number of node on X-axis and Y-axis :\n");
printf("number on X-axis = ");
scanf("%d",&size_x);
p_fem->setSizeX(size_x);
printf("number on Y-axis = ");
scanf("%d",&size_y);
p_fem->setSizeY(size_y);
printf("\n");

printf("Set step size of grid on X-axis and Y-axis :\n");
printf("step size on X-axis = ");
scanf("%f",&df_x);
p_fem->setDiffX(df_x);
printf("step size on Y-axis = ");
scanf("%f",&df_y);
p_fem->setDiffY(df_y);

p_fem->calSize();

printf("\n----------\n");

// set boundary value
float inputed_type;
printf("\n");
printf("SET BOUNDARY VALUE...\n");
printf("\nNote : Don't input character!!\n\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
printf("Set bottom side = ");
scanf("%f",&side_value[0]);
do {
        printf("Dirichlet(0) or Neuman(1) = ");
        scanf("%f",&inputed_type);
        type[0] = (int)inputed_type;
        if (inputed_type>0 && inputed_type<1) {
                type[0] = -1;
        }
        if (inputed_type>1 && inputed_type<2) {
                type[0] = -1;
        }
        char debug;
        scanf("%c",&debug);
} while (type[0]<0 || type[0]>1);
printf("Set left side  = ");
scanf("%f",&side_value[1]);
do {
        printf("Dirichlet(0) or Neuman(1) = ");
        scanf("%f",&inputed_type);
        type[1] = (int)inputed_type;
        if (inputed_type>0 && inputed_type<1) {
                type[1] = -1;
        }
        if (inputed_type>1 && inputed_type<2) {
                type[1] = -1;
        }
        char debug;
        scanf("%c",&debug);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
} while (type[1]<0 || type[1]>1);
printf("Set right side  = ");
scanf("%f",&side_value[2]);
do {
        printf("Dirichlet(0) or Neuman(1) = ");
        scanf("%f",&inputed_type);
        type[2] = (int)inputed_type;
        if (inputed_type>0 && inputed_type<1) {
                type[2] = -1;
        }
        if (inputed_type>1 && inputed_type<2) {
                type[2] = -1;
        }
        char debug;
        scanf("%c",&debug);
} while (type[2]<0 || type[2]>1);
printf("Set upper side  = ");
scanf("%f",&side_value[3]);
do {
        printf("Dirichlet(0) or Neuman(1) = ");
        scanf("%f",&inputed_type);
        type[3] = (int)inputed_type;
        if (inputed_type>0 && inputed_type<1) {
                type[3] = -1;
        }
        if (inputed_type>1 && inputed_type<2) {
                type[3] = -1;
        }
        char debug;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        scanf("%c",&debug);
} while (type[3]<0 || type[3]>1);
printf("\n");

// initial
for (i=0; i<=p_fem->getNumberOfNodes(); i++)
{
        for (j=0; j<=p_fem->getNumberOfNodes(); j++)
        {
                p_fem->setKs(i,j,0.0);
        }
        p_fem->setFs(i,0.0);
        p_fem->setAns(i,0.0);
}
// initial for debug logical error when compute Laprace
p_fem->setIsPoisson(false);
p_fem->setIsX(false);
p_fem->setIsY(false);
p_fem->setIsXY(false);

printf("Please, check all before continue...");
scanf("%c",&ch1);  // debug
printf("\n");
printf("Laprace(l) or Poisson(p) : ");
scanf("%c",&ch1);
if (ch1 == 'p')
{
        p_fem->setIsPoisson(true);
        printf("\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
printf("set G :\n");
printf("  1. Constant\n");
printf("  2. G = x\n");
printf("  3. G = y\n");
printf("  4. G = xy\n");
printf(" ==> ");
scanf("%d",&ch2);
printf("\n");
if (ch2 == 1)
{
        printf("Value = ");
        scanf("%f",&g_value);
        p_fem->setGConst(g_value);
}
else if (ch2 == 2)
{
        p_fem->setIsX(true);
        p_fem->setIsY(false);
        p_fem->setIsXY(false);
        printf("Set G as function of X.\n");
}
else if (ch2 == 3)
{
        p_fem->setIsX(false);
        p_fem->setIsY(true);
        p_fem->setIsXY(false);
        printf("Set G as function of Y.\n");
}
else if (ch2 == 4)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                p_fem->setIsX(false);
                p_fem->setIsY(false);
                p_fem->setIsXY(true);
                printf("Set G as function of XY.\n");
        }
}
printf("\n");

p_fem->setRectBC(side_value,type);

// initial vector, Fs
for (i=1; i<=p_fem->getSizeOfK(); i++)
{
        p_fem->setFs(i,0.0);
}

if (p_fem->setXYPosition(x_file,y_file))
{
        p_fem->setNodeMember();
        p_fem->assembly();
        p_fem->aplyDirichlet();
        p_fem->aplyNeuman();
        p_fem->triDecomp();

        // show answer data
        printf("Answer :\n");
        printf("(x,y) = answer\n");
        for (i=0; i<p_fem->getSizeOfK(); i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                printf("(%.2f,%.2f) = %f\n",
                    p_fem->getXYPos(i+1,1),p_fem->getXYPos(i+1,2),p_fem-
>getAns(i));
            }

            // show error
            if (p_fem->getAns(0)>pow(10,5)*9 || p_fem->getAns(0)<-pow(10,5)*9) {
                printf("\n\nERROR : No graph!!!\n\n");
            }
        }
        printf("\n");

        // write matrix K into file
        FILE *fpn;
        char fileName[] = "K.txt";
        fpn = fopen(fileName,"w");
        if (fpn == NULL)
        {
            printf("Cannot OPEN file, %s\n",fileName);
            exit(0);
        }
        else
        {
            for (i=1; i<=p_fem->getSizeOfK(); i++)
            {
                for (j=1; j<=p_fem->getSizeOfK(); j++)
                {
                    if (p_fem->getKs(i,j) < 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        fprintf(fpn,"%.2f ",p_fem->getKs(i,j));
                }
                else
                {
                        fprintf(fpn," %.2f ",p_fem->getKs(i,j));
                }
        }
        fprintf(fpn,"\n");
    }
}
fclose(fpn);

// write matrix F into file
fpn = fopen("F.txt","w");
if (fpn == NULL)
{
    printf("Cannot OPEN file, %s\n","F.txt");
    exit(0);
}
else
{
    for (i=0; i<p_fem->getSizeOfK(); i++)
    {
        fprintf(fpn,"%f ",p_fem->getFs(i+1));
        fprintf(fpn,"\n");
    }
}
fclose(fpn);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // write answer into file
        fpn = fopen("Z.txt","w");
        if (fpn == NULL)
        {
                printf("Cannot OPEN file, %s\n","Z.txt");
                exit(0);
        }
        else
        {
                for (i=0; i<p_fem->getSizeOfK(); i++)
                {
                        fprintf(fpn,"%f ",p_fem->getAns(i));
                        fprintf(fpn,"\n");
                }
        }
        fclose(fpn);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

vertexhandler.cpp

```
/*
 * vertexhandler.cpp - vertex descriptor class
 * Author Peeba_Merlin
*/

#include "vertexhandler.h"

/* constructor */
VertexHandler::VertexHandler() {
        _size = 0;
        _loaded = false;
}

/* load all vertices in files */
bool VertexHandler::loadVertices(const char *filename) {
        _size = 0;
        _loaded = false;

        FILE* fp;
        if ((fp=fopen(filename, "r")) == NULL) {
                // do something with error here!
                return false;
        }

        Vertex* vertex;
        float x, y, z;
        /* get vertices */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        while ((fscanf(fp, "%f", &x) > 0) &&
                (fscanf(fp, "%f", &y) > 0) &&
                (fscanf(fp, "%f", &z) > 0)) {
            vertex = new Vertex(x, y, z);
            vertices.push_back(*vertex);
        }

        _size = vertices.size();
        _loaded = true;

        fclose(fp);
        return true;
}

/* load vetices from file for each axis */
bool VertexHandler::loadVertices(const char *fx,const char *fy,const char *fz) {
        _size = 0;
        _loaded = false;

        FILE* fpx;
        FILE* fpy;
        FILE* fpz;
        Vertex* vertex;
        float x,y,z;

        fpx = fopen(fx, "r");
        fpy = fopen(fy, "r");
        fpz = fopen(fz, "r");
        if ((fpx ||fpy || fpz) == NULL) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            // do something with error here!
            return false;
        }

        /* get vertices */
        while ((fscanf(fpx, "%f", &x) > 0) &&
                (fscanf(fpy, "%f", &y) > 0) &&
                (fscanf(fpz, "%f", &z) > 0)) {
            vertex = new Vertex(x, y, z);
            vertices.push_back(*vertex);
        }

        _size = vertices.size();
        _loaded = true;
        fclose(fpx);
        fclose(fpy);
        fclose(fpz);
        return true;
}

/* getVertex : return reference vertex class by index */
Vertex VertexHandler::getVetex(int index) const {
        return vertices[index];
}

/* test purpose only! */
void VertexHandler::test() {
        VertexHandler::loadVertices("x.txt", "y.txt", "z.txt");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* size : return amount of vertices */
int VertexHandler::size(void) {
        return _size;
}

bool VertexHandler::isLoaded(void) {
        return _loaded;
}

/* clear vertices from memory */
VertexHandler::~VertexHandler() {
        vertices.clear();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้