

ระเบียบวิธีไฟไนต์เอลิเมนต์แบบไร้จุดต่อที่ปรับขนาดได้
สำหรับการวิเคราะห์อุณหภูมิและโครงสร้างด้วยหลักการของฟลักซ์



นายสุธี ไตรวิวัฒนา

ศูนย์วิทยทรัพยากร
วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

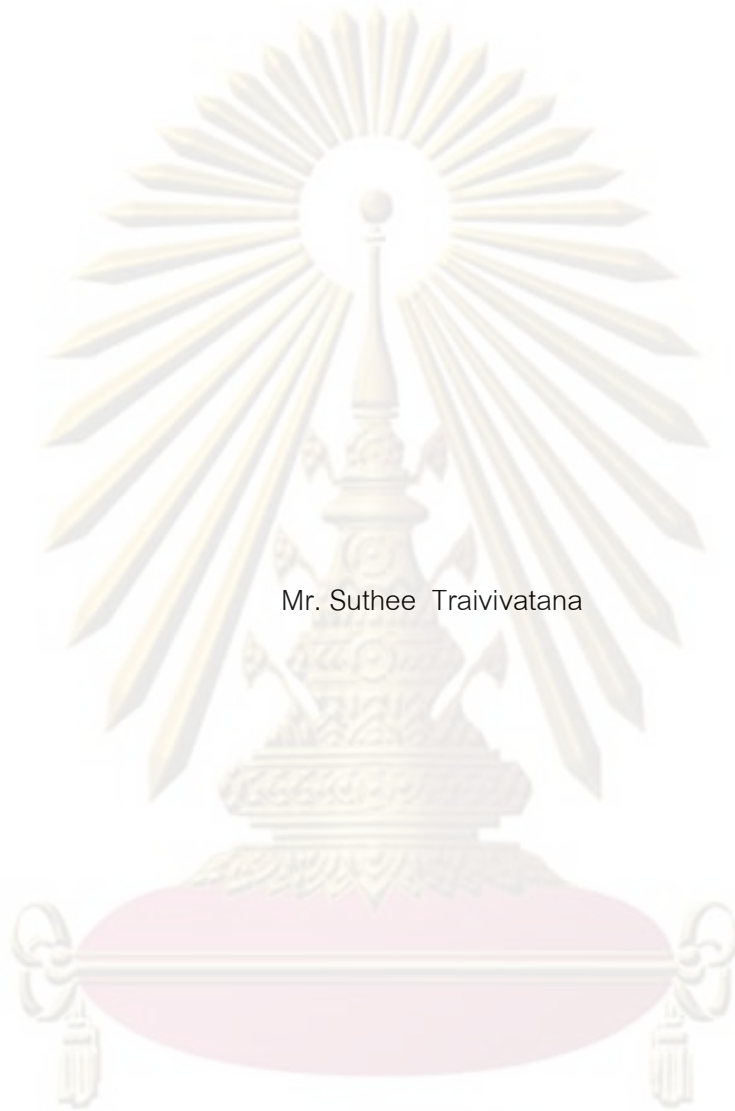
สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2552

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

ADAPTIVE NODELESS VARIABLE FINITE ELEMENT METHOD FOR
THERMAL-STRUCTURAL ANALYSIS USING FLUX-BASED FORMULATION



Mr. Suthee Traivivatana

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Mechanical Engineering

Department of Mechanical Engineering

Faculty of Engineering

Chulalongkorn University

Academic year 2009

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

ระเบียบวิธีไฟไนต์เอลิเมนต์แบบไร้จุดต่อที่ปรับขนาดได้สำหรับ
การวิเคราะห์อุณหภูมิและโครงสร้างด้วยหลักการของฟลักซ์

โดย

นายสุธี ไตรวิวัฒนา

สาขาวิชา

วิศวกรรมเครื่องกล

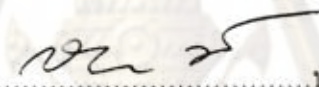
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

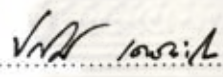
ศาสตราจารย์ ดร.ปราโมทย์ เตชะอำไพ

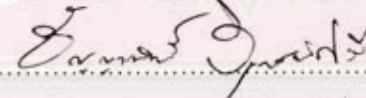
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาตรีบัณฑิต



..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศนันทวงศ์)

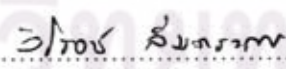
คณะกรรมการสอบวิทยานิพนธ์


..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.กฤษณี มณีรัตน์)


..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ศาสตราจารย์ ดร.ปราโมทย์ เตชะอำไพ)


..... กรรมการ
(อาจารย์ ดร.ชญานันท์ วิรุฬห์ศรี)


..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.วรางค์รัตน์ จันทสาโร)


..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.วิโรจน์ ลิ้มตระการ)

ศูนย์วิจัยทรัพยากรชีวภาพ

จุฬาลงกรณ์มหาวิทยาลัย

นายสุธี ไตรวิวัฒน์ : ระเบียบวิธีไฟไนต์เอลิเมนต์แบบไร้จุดต่อที่ปรับขนาดได้สำหรับการวิเคราะห์อุณหภูมิและโครงสร้างด้วยหลักการของฟลักซ์. (ADAPTIVE NODELESS VARIABLE FINITE ELEMENT METHOD FOR THERMAL-STRUCTURAL ANALYSIS USING FLUX-BASED FORMULATION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ศ. ดร.ปราโมทย์ เดชะอำไพ, 155 หน้า.

วิทยานิพนธ์ฉบับนี้นำเสนอระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อสำหรับการวิเคราะห์ปัญหาอุณหภูมิและโครงสร้าง ซึ่งระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อนั้นใช้การประมาณภายในเอลิเมนต์เป็นแบบควอดราติกโดยไม่จำเป็นต้องใช้จุดต่อแท้เพิ่มเติม และประยุกต์หลักการของฟลักซ์เข้ากับระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อเพื่อลดความซับซ้อนในการประดิษฐ์ไฟไนต์เอลิเมนต์เมทริกซ์ เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติช่วยเพิ่มความถูกต้องเที่ยงตรงของผลลัพธ์ให้สูงขึ้น ด้วยการสร้างวางเอลิเมนต์ขนาดเล็กในบริเวณที่มีการเปลี่ยนแปลงของผลลัพธ์สูงและวางเอลิเมนต์ขนาดใหญ่ในบริเวณที่มีการเปลี่ยนแปลงของผลลัพธ์ต่ำ การปรับขนาดเอลิเมนต์โดยอัตโนมัตินั้นถูกประยุกต์ใช้ร่วมกับการคะเนความคลาดเคลื่อนแบบโพสเทรือริที่เป็นการคะเนความคลาดเคลื่อนภายหลังจากที่ได้ผลเฉลยจากระเบียบวิธีไฟไนต์เอลิเมนต์ เพื่อเป็นดัชนีชี้วัดความคลาดเคลื่อนและนำไปสู่การยุติการปรับขนาดเอลิเมนต์ภายใต้ขอบเขตความคลาดเคลื่อนที่ยอมรับได้


การตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นทำโดยการนำผลลัพธ์ที่ได้จากการวิเคราะห์ไปเปรียบเทียบกับปัญหาที่มีผลเฉลยแม่นยำตรง ก่อนจะนำไปวิเคราะห์ปัญหาที่มีความซับซ้อนมากยิ่งขึ้น ผลลัพธ์ที่ได้จากการวิเคราะห์ปัญหาแสดงให้เห็นถึงประสิทธิภาพของการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติเข้ากับระเบียบวิธีไฟไนต์เอลิเมนต์แบบไร้จุดต่อด้วยหลักการของฟลักซ์ ในการวิเคราะห์ปัญหาและสร้างความเข้าใจถึงพฤติกรรมของอุณหภูมิและโครงสร้าง

ภาควิชา วิศวกรรมเครื่องกล

สาขาวิชา วิศวกรรมเครื่องกล

ปีการศึกษา 2552

ลายมือชื่อนิสิต 

ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก 

จุฬาลงกรณ์มหาวิทยาลัย

4771869521 : MAJOR MECHANICAL ENGINEERING

KEYWORDS : NODELESS VARIABLE FINITE ELEMENT METHOD / THERMAL-STRUCTURAL ANALYSIS / FLUX-BASED FORMULATION / ADAPTIVE MESH / ERROR ESTIMATION

SUTHEE TRAIWIVATANA : ADAPTIVE NODELESS VARIABLE FINITE ELEMENT METHOD FOR THERMAL-STRUCTURAL ANALYSIS USING FLUX-BASED FORMULATION. THESIS ADVISOR : PROF. PRAMOTE DECHAUMPHAI, Ph.D., 155 pp.

A nodeless variable finite element method has been developed for thermal-structural analyses. In a nodeless variable finite element, the quadratic interpolation functions is used to describe the solution distribution over the element without additional actual nodes. This dissertation also introduces and implements the flux-based formulation to derive the finite element matrices for such nodeless variable element. The solution accuracy is further improved by implementing an adaptive meshing technique to generate finite element meshes that can adapt and move with the solution behavior. A posteriori error estimation is also incorporated into the nodeless variable finite element method with the flux-based formulation.

A corresponding finite element computer program was developed and evaluated using simple examples that have exact solutions prior to solving more complex problems. The computational results from several problems demonstrate the effectiveness of the presented finite element method to accurately predict the thermal-structural phenomena.

Department : Mechanical Engineering... Student's Signature *Suthee Traiwivatana*
 Field of Study : Mechanical Engineering... Advisor's Signature *Pratek Ongk*
 Academic Year : 2009.....

กิตติกรรมประกาศ

ผู้วิจัยขอกราบขอบพระคุณ ศาสตราจารย์ ดร.ปราโมทย์ เดชะอำไพ อาจารย์ที่ปรึกษาวิทยานิพนธ์เป็นอย่างสูง ที่ท่านได้ให้ความรู้ คำแนะนำ ตลอดจนข้อคิดที่มีคุณค่ายิ่งในการทำวิจัย นอกจากนี้ท่านยังได้ถ่ายทอดข้อคิดหลายสิ่งหลายอย่างที่มีคุณค่ายิ่งเกี่ยวกับการทำงาน และการดำเนินชีวิตของผู้วิจัย

ขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.กฤษณี มณีรัตน์ ประธานกรรมการ รองศาสตราจารย์ ดร.วราภรณ์รัตน์ จันทสาโร ผู้ช่วยศาสตราจารย์ ดร.วิโรจน์ ลิ้มตระการ และ อาจารย์ ดร.ชญญาพันธ์ วิรุฬห์ศรี กรรมการ ที่ได้ให้คำแนะนำและถ่ายทอดความรู้ตลอดระยะเวลาในการทำงานวิจัยนี้ ซึ่งทำให้วิทยานิพนธ์ฉบับนี้มีความสมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณทุน 90 ปี จุฬาลงกรณ์มหาวิทยาลัย ที่ให้การสนับสนุนทางการเงินตลอดระยะเวลาที่ทำงานวิจัยฉบับนี้

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.นิพนธ์ วรรณโสภาคย์ และอาจารย์ ดร.สุทธิศักดิ์ พงศ์ธนาพานิช ที่ได้ถ่ายทอดความรู้ คอยให้คำปรึกษา และประสบการณ์ในทุก ๆ ด้าน ขอขอบคุณ คุณปริญา นุญมาเลิศ คุณอิทธิพงษ์ มาลาทิพย์ คุณพัชรี ธีระเอก และคุณพิชญนทร์ โพธิกุล ซึ่งเป็นผู้ร่วมงานในห้องปฏิบัติการวิจัยกลศาสตร์การคำนวณ สำหรับความช่วยเหลือและกำลังใจตลอดเวลาทำงานวิจัยนี้

ท้ายสุดนี้ผู้วิจัยขอกราบขอบพระคุณบิดามารดาที่เป็นกำลังใจและสนับสนุนการศึกษาของผู้วิจัยมาโดยตลอด อนึ่งประโยชน์และคุณค่าอันใดที่ได้รับจากวิทยานิพนธ์นี้ขอมอบเป็นกตัญญูตานุชาแต่บิดามารดา ครูอาจารย์ ตลอดจนผู้มีพระคุณทุกท่าน

ศูนย์วิทยทรัพยากร

จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ.....	ฎ
คำอธิบายสัญลักษณ์.....	ณ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	3
1.3 ขอบเขตของการวิจัย.....	4
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	4
1.5 วิธีดำเนินการวิจัย.....	5
1.6 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	5
บทที่ 2 สมการเชิงอนุพันธ์ย่อยสำหรับการถ่ายเทความร้อนและของแข็ง	8
2.1 สมการเชิงอนุพันธ์ย่อยสำหรับปัญหาการถ่ายเทความร้อน.....	8
2.2 สมการเชิงอนุพันธ์ย่อยสำหรับปัญหาของแข็ง.....	9
บทที่ 3 ระเบียบวิธีไฟไนต์เอลิเมนต์สำหรับการถ่ายเทความร้อนและของแข็ง	12
3.1 ขั้นตอนทั่วไปของระเบียบวิธีไฟไนต์เอลิเมนต์.....	12
3.2 ระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ.....	15
3.3 รูปแบบของระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์.....	19
3.4 สมการไฟไนต์เอลิเมนต์สำหรับปัญหาการถ่ายเทความร้อน.....	22
3.5 สมการไฟไนต์เอลิเมนต์สำหรับปัญหาของแข็ง.....	23
3.6 การประดิษฐ์ไฟไนต์เอลิเมนต์เมทริกซ์.....	27

3.6.1	เอลิเมนต์เมทริกซ์สำหรับปัญหาการถ่ายเทความร้อน.....	27
3.6.2	เอลิเมนต์เมทริกซ์สำหรับปัญหาความเค้นในโครงสร้าง.....	31
บทที่ 4	โปรแกรมคอมพิวเตอร์.....	32
4.1	โปรแกรมคอมพิวเตอร์สำหรับวิเคราะห์ปัญหาการถ่ายเทความร้อน.....	32
4.2	รายละเอียดของโปรแกรม.....	33
4.3	ลักษณะของไฟล์ข้อมูลที่โปรแกรมต้องการ.....	33
4.4	โปรแกรมคอมพิวเตอร์สำหรับวิเคราะห์ปัญหาความเค้นอันเนื่องมาจาก อุณหภูมิ.....	35
4.5	รายละเอียดของโปรแกรม.....	36
4.6	ลักษณะของไฟล์ข้อมูลที่โปรแกรมต้องการ.....	36
บทที่ 5	เทคนิคการปรับขนาดเอลิเมนต์.....	38
5.1	หลักการของเทคนิคการปรับขนาดเอลิเมนต์.....	38
5.2	โปรแกรมคอมพิวเตอร์สำหรับการปรับขนาดเอลิเมนต์โดยอัตโนมัติ.....	42
บทที่ 6	การตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์.....	44
6.1	ตรวจสอบโปรแกรมสำหรับวิเคราะห์ปัญหาการถ่ายเทความร้อน.....	44
6.1.1	ปัญหาแผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้เอง.....	44
6.2	ตรวจสอบโปรแกรมสำหรับวิเคราะห์ปัญหาของแข็ง.....	47
6.2.1	ปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง.....	47
6.3	ตรวจสอบโปรแกรมสำหรับวิเคราะห์ปัญหาความเค้นอันเนื่องมาจาก อุณหภูมิ.....	48
6.3.1	ปัญหาแผ่นสี่เหลี่ยมที่มีอุณหภูมิเปลี่ยนแปลงเชิงเส้นตามความ ยาว.....	48
6.3.2	ปัญหาแหวนที่มีอุณหภูมิเปลี่ยนแปลงตามแนวรัศมี.....	50
6.4	ตัวอย่างการวิเคราะห์ปัญหาด้วยเทคนิคการปรับเอลิเมนต์โดยอัตโนมัติ.....	54
6.4.1	ปัญหาแผ่นสี่เหลี่ยมจัตุรัสภายใต้กรวยของเกรเดียนต์ความร้อน.....	54
6.4.2	ปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง.....	60
6.4.3	ปัญหาแหวนที่มีอุณหภูมิเปลี่ยนแปลงตามแนวรัศมี.....	62

บทที่ 7	การคะเนความคลาดเคลื่อนแบบโพสเทอริออรี	64
7.1	ตัวชี้วัดความคลาดเคลื่อน	64
7.2	การคะเนความคลาดเคลื่อนแบบโพสเทอริออรี	67
7.3	การคะเนความคลาดเคลื่อนด้วยการใช้หลักการของฟลักซ์	68
7.3.1	ความคลาดเคลื่อนที่คำนวณจากผลต่างของฟลักซ์	69
7.3.2	ค่าแอลทูนอร์มสำหรับความคลาดเคลื่อนโดยประมาณ	70
7.3.3	ค่าแอลทูนอร์มสำหรับความคลาดเคลื่อนแม่นยำ	71
7.4	โปรแกรมสำหรับการคะเนความคลาดเคลื่อนที่คำนวณจากผลต่างของฟลักซ์	73
7.5	การทดสอบโปรแกรมย่อยสำหรับการคะเนความคลาดเคลื่อนที่คำนวณจากผลต่างของฟลักซ์	74
7.5.1	ปัญหาแผ่นสี่เหลี่ยมผืนผ้าซึ่งถูกกำหนดอุณหภูมิที่ขอบ	74
7.5.2	ปัญหาแผ่นสี่เหลี่ยมจัตุรัสพร้อมแหล่งกำเนิด	80
7.5.3	ปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมภายใต้แรงดึงทิศทางเดียว	85
7.5.4	ปัญหาคานปลายยื่นชนิดสั้นภายใต้แรงกด	90
7.5.5	ปัญหาแผ่นโลหะรูปตัวแอลภายใต้แรงดึง	94
บทที่ 8	บทสรุป ปัญหาที่พบและข้อเสนอแนะ	99
8.1	บทสรุป	99
8.2	ปัญหาที่พบในขณะทำวิทยานิพนธ์	101
8.3	ข้อเสนอแนะสำหรับงานวิจัยในอนาคต	101
	รายการอ้างอิง	102
	ภาคผนวก	105
	ภาคผนวก ก รายละเอียดของโปรแกรมวิเคราะห์ปัญหาการถ่ายเทความร้อน	107
	ภาคผนวก ข รายละเอียดของโปรแกรมวิเคราะห์ปัญหาความเค้นอันเนื่องมาจากอุณหภูมิ	128
	ประวัติผู้เขียนวิทยานิพนธ์	155

สารบัญญัตราสาร

ตารางที่		หน้า
6.1	ตารางแสดงการเปรียบเทียบค่าเปอร์เซ็นต์ความผิดพลาดจากโปรแกรม NLFB และโปรแกรม standard ในแนวแกน x ที่ตำแหน่ง $y = 0$ สำหรับปัญหาแผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้เอง.....	46
6.2	ตารางแสดงค่าความเค้นในแนวแกน x ที่จุด A และการเปรียบเทียบค่าเปอร์เซ็นต์ความผิดพลาดจากการปรับขนาดเอลิเมนต์ครั้งที่ 1 – 4 สำหรับปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง.....	61
6.3	ตารางแสดงการเปรียบเทียบค่าเปอร์เซ็นต์ความผิดพลาดของผลลัพธ์ (ค่าการเคลื่อนตัวตามทิศทางรัศมี ความเค้นในแนวรัศมี และความเค้นในทิศ θ ในแนวมุม θ เท่ากับ 0) จากรูปแบบจำลองไฟไนต์เอลิเมนต์เริ่มต้น และรูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 1 และ 2.....	63
7.1	ตารางแสดงการเปรียบเทียบค่าแอลลูทอนอร์มสำหรับความคลาดเคลื่อนและค่าเปอร์เซ็นต์ความคลาดเคลื่อนระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งสี่แบบ.....	79
7.2	ตารางแสดงการเปรียบเทียบค่าแอลลูทอนอร์มสำหรับความคลาดเคลื่อนและค่าเปอร์เซ็นต์ความคลาดเคลื่อนระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งสามแบบ...	84
7.3	ตารางแสดงการเปรียบเทียบค่าแอลลูทอนอร์มสำหรับความคลาดเคลื่อนและค่าเปอร์เซ็นต์ความคลาดเคลื่อนระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งหกแบบของปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมภายใต้แรงดึงทิศทางเดียว.....	89
7.4	ตารางแสดงการเปรียบเทียบจำนวนจุดต่อและค่าความคลาดเคลื่อนระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งแปดรูปแบบของปัญหาคานปลายยื่นชนิดสั้นภายใต้แรงกด.....	93
7.5	ตารางแสดงการเปรียบเทียบจำนวนจุดต่อและค่าความคลาดเคลื่อนระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งแปดรูปแบบของปัญหาแผ่นโลหะรูปตัวแอลภายใต้แรงดึง.....	97

สารบัญภาพ

ภาพที่		หน้า
2.1	ขอบเขตและเงื่อนไขขอบเขตของปัญหาการถ่ายเทความร้อน.....	8
2.2	ขอบเขตและเงื่อนไขขอบเขตของปัญหาของแข็ง.....	9
3.1	การแบ่งขอบเขตรูปร่างของปัญหาออกเป็นเอลิเมนต์ย่อย ๆ.....	12
3.2	เอลิเมนต์สามเหลี่ยมแบบ 3 จุดต่อและตัวไม่ทราบค่าที่จุดต่อ.....	13
3.3	ลักษณะการกระจายตัวของอุณหภูมิที่ตำแหน่งจุดต่อ.....	15
3.4	ลักษณะการกระจายตัวของอุณหภูมิที่ตำแหน่งไร้จุดต่อ.....	15
3.5	ลักษณะการกระจายตัวของอุณหภูมิรวม.....	15
3.6	เอลิเมนต์สามเหลี่ยมสามจุดต่อ และจุด p ที่อยู่ภายในเอลิเมนต์.....	16
3.7	การอ้างอิงหมายเลขจุดต่อด้วยตัวเลข α, β และ γ บนเอลิเมนต์สามเหลี่ยม หกจุดต่อ.....	18
5.1	แสดงการวางตัวของเอลิเมนต์ในแนวแกนหลักทั้งสอง (X, Y).....	40
5.2	ตัวอย่างการวางตัวของเอลิเมนต์สามเหลี่ยมล้อมจุดต่อ j ใด ๆ.....	40
6.1	แผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้เอง.....	45
6.2	รูปแบบไฟไนต์เอลิเมนต์สำหรับปัญหาแผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้ เอง.....	45
6.3	อุณหภูมิในแนวแกน x ที่ตำแหน่ง $y = 0$ สำหรับปัญหาแผ่นสี่เหลี่ยมที่มีการ ผลิตความร้อนได้เอง.....	46
6.4	แผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง.....	47
6.5	รูปแบบไฟไนต์เอลิเมนต์สำหรับปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง.....	48
6.6	แผ่นสี่เหลี่ยมที่มีอุณหภูมิเปลี่ยนแปลงเชิงเส้นตามความยาว.....	49
6.7	รูปแบบไฟไนต์เอลิเมนต์สำหรับปัญหาแผ่นสี่เหลี่ยมที่มีอุณหภูมิเปลี่ยนแปลง เชิงเส้นตามความยาว.....	49
6.8	ค่าการเคลื่อนตัวเนื่องจากความร้อนในแนวแกน x สำหรับปัญหาแผ่นสี่เหลี่ยม ที่มีอุณหภูมิเปลี่ยนแปลงเชิงเส้นตามความยาว.....	50
6.9	วงแหวนที่มีอุณหภูมิเปลี่ยนแปลงตามแนวรัศมี.....	51
6.10	รูปแบบไฟไนต์เอลิเมนต์สำหรับปัญหาวงแหวนที่มีอุณหภูมิเปลี่ยนแปลงตาม แนวรัศมี.....	52

ภาพที่	หน้า
6.11	การเปลี่ยนแปลงของค่าการเคลื่อนตัวตามทิศทางรัศมี ในแนวมุม θ เท่ากับ 0... 52
6.12	ความเค้นในแนวรัศมี σ_r ตามทิศทางรัศมีที่มีแนวมุม θ เท่ากับ 0..... 53
6.13	ความเค้นในทิศ θ, σ_θ ตามทิศทางรัศมีที่มีแนวมุม θ เท่ากับ 0..... 53
6.14	ปัญหาแผ่นสี่เหลี่ยมจัตุรัสภายใต้กรวยของเกรเดียนท์ความร้อน..... 54
6.15	ก) รูปแบบไฟไนต์เอลิเมนต์สำหรับปัญหาแผ่นสี่เหลี่ยมจัตุรัสภายใต้กรวยของเกรเดียนท์ความร้อน..... 55
	ข) การกระจายตัวของอุณหภูมิสำหรับแบบจำลองไฟไนต์เอลิเมนต์เริ่มต้น..... 55
6.16	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 1..... 56
	ข) การกระจายตัวของอุณหภูมิที่คำนวณได้..... 56
6.17	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 2..... 56
	ข) การกระจายตัวของอุณหภูมิที่คำนวณได้..... 56
6.18	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งสุดท้าย..... 57
	ข) การกระจายตัวของอุณหภูมิที่คำนวณได้..... 57
6.19	ก) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลองเริ่มต้น..... 58
	ข) การกระจายตัวของอุณหภูมิสำหรับแบบจำลองจากการปรับขนาดครั้งที่ 1.... 58
	ค) การกระจายตัวของอุณหภูมิสำหรับแบบจำลองจากการปรับขนาดครั้งที่ 2.... 59
	ง) การกระจายตัวของอุณหภูมิสำหรับแบบจำลองจากการปรับขนาดครั้งสุดท้าย..... 59
6.20	การเปรียบเทียบผลลัพธ์จากการคำนวณโดยใช้รูปแบบไฟไนต์เอลิเมนต์เริ่มต้นและรูปแบบไฟไนต์เอลิเมนต์ที่ทำการปรับขนาดกับผลเฉลยแม่นยำ..... 60
6.21	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 1..... 61
	ข) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 2..... 61
	ค) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 3..... 61
	ง) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 4..... 61
6.22	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 1..... 62
	ข) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 2..... 62
6.23	การเปรียบเทียบความเค้นในแนวรัศมี σ_r ที่มีแนวมุม θ เท่ากับ 0 ระหว่างผลเฉลยแม่นยำ รูปแบบไฟไนต์เอลิเมนต์เริ่มต้น รูปแบบปรับขนาดครั้งที่ 1 และรูปแบบปรับขนาดครั้งที่ 2..... 63

ภาพที่		หน้า
7.1	ความคลาดเคลื่อนจากการประมาณภายในสำหรับปัญหาการถ่ายเทความร้อน หนึ่งมิติ.....	65
7.2	ลักษณะการกระจายของฟลักซ์ที่ได้จากระเบียบวิธีไฟไนต์เอลิเมนต์แบบดั้งเดิม	69
7.3	ลักษณะการกระจายของฟลักซ์ที่ได้จากระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้ หลักการของฟลักซ์.....	69
7.4	ปัญหาแผ่นสี่เหลี่ยมผืนผ้าซึ่งถูกกำหนดอุณหภูมิที่ขอบ.....	74
7.5	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M1.....	75
	ข) การกระจายตัวของอุณหภูมิในรูปแบบสองมิติสำหรับแบบจำลอง T3NL-M1	75
	ค) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลอง T3NL-M1	75
7.6	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M2.....	76
	ข) การกระจายตัวของอุณหภูมิในรูปแบบสองมิติสำหรับแบบจำลอง T3NL-M2	76
	ค) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลอง T3NL-M2	76
7.7	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M3.....	77
	ข) การกระจายตัวของอุณหภูมิในรูปแบบสองมิติสำหรับแบบจำลอง T3NL-M3	77
	ค) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลอง T3NL-M3	77
7.8	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M4.....	78
	ข) การกระจายตัวของอุณหภูมิในรูปแบบสองมิติสำหรับแบบจำลอง T3NL-M4	78
	ค) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลอง T3NL-M4	78
7.9	ผลการเปรียบเทียบค่าอุณหภูมิซึ่งได้จากผลเฉลยแม่นยำและจากโปรแกรม NLFB ในแนวแกน AA.....	79
7.10	ปัญหาแผ่นสี่เหลี่ยมจัตุรัสพร้อมแหล่งกำเนิด.....	80
7.11	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M1.....	81
	ข) การกระจายตัวของอุณหภูมิในรูปแบบสองมิติสำหรับแบบจำลอง T3NL-M1	81
	ค) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลอง T3NL-M1	81
7.12	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M2.....	82
	ข) การกระจายตัวของอุณหภูมิในรูปแบบสองมิติสำหรับแบบจำลอง T3NL-M2	82
	ค) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลอง T3NL-M2	82
7.13	ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M3.....	83
	ข) การกระจายตัวของอุณหภูมิในรูปแบบสองมิติสำหรับแบบจำลอง T3NL-M3	83

ภาพที่	หน้า	
7.14	ค) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลอง T3NL-M3 ผลการเปรียบเทียบค่าตัวแปรปฐภูมิซึ่งได้จากผลเฉลยแม่นยำตรงและจาก โปรแกรม NLFB ตลอดแนว $x = 0.5$	83 84
7.15	ก) ปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมภายใต้แรงดึงทิศทางเดียว.....	85
	ข) ขอบเขตการคำนวณ.....	85
7.16	รูปแบบจำลองไฟไนต์เอลิเมนต์อย่างเป็นทางการระเบียบและรูปแบบจำลองไฟไนต์ เอลิเมนต์อย่างไร้ระเบียบ	
	ก) Mesh S1.....	87
	ข) Mesh S2.....	87
	ค) Mesh S3.....	87
	ง) Mesh U1.....	87
	จ) Mesh U2.....	87
	ฉ) Mesh U3.....	87
7.17	ผลการเปรียบเทียบผลเฉลยเชิงวิเคราะห์และผลเฉลยโดยประมาณของความ เค้นตั้งฉาก σ_x	88
7.18	ผลการเปรียบเทียบผลเฉลยเชิงวิเคราะห์และผลเฉลยโดยประมาณของความ เค้นตั้งฉาก σ_y	88
7.19	ผลการเปรียบเทียบผลเฉลยเชิงวิเคราะห์และผลเฉลยโดยประมาณของความ เค้นเฉือน τ_{xy}	89
7.20	ปัญหาคานปลายยื่นชนิดสั้นภายใต้แรงกด.....	90
7.21	รูปแบบจำลองไฟไนต์เอลิเมนต์อย่างเป็นทางการระเบียบและรูปแบบจำลองไฟไนต์ เอลิเมนต์อย่างไร้ระเบียบ	
	ก) Mesh S1 (U1).....	92
	ข) Mesh S2.....	92
	ค) Mesh S3.....	92
	ง) Mesh S4.....	92
	จ) Mesh S5.....	92
	ฉ) Mesh U2.....	92
	ช) Mesh U3.....	93

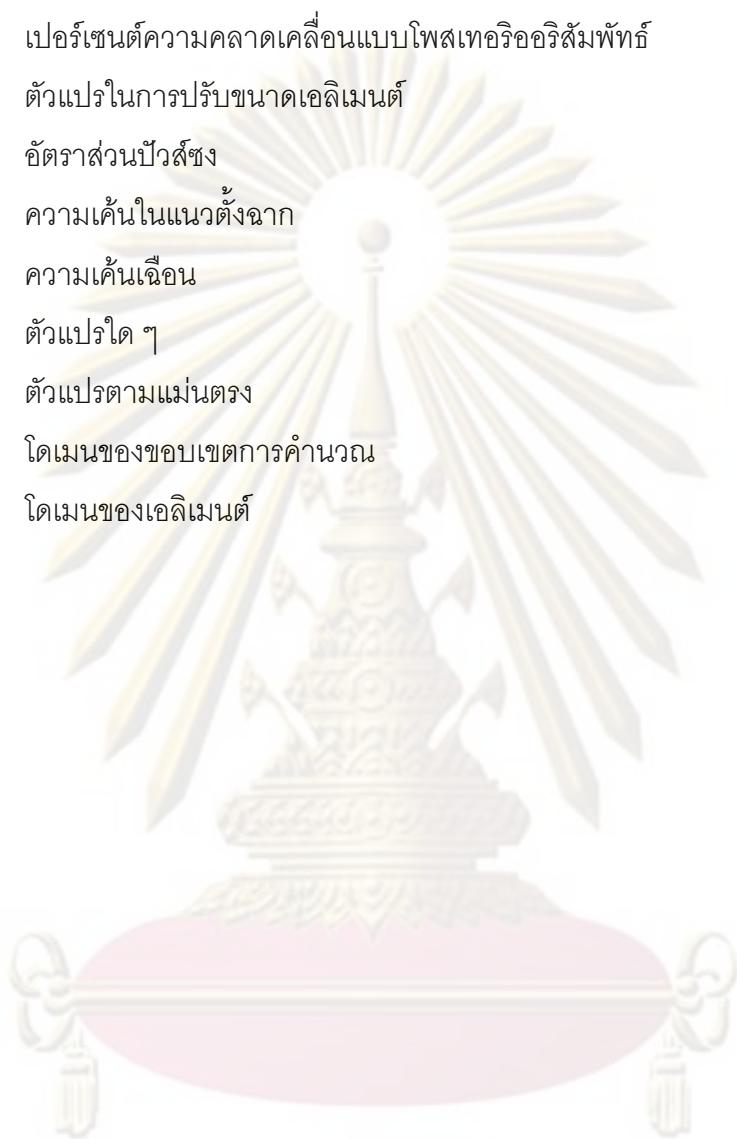
ภาพที่	หน้า
7.22 ผลการเปรียบเทียบการลู่เข้าของผลเฉลยโดยประมาณระหว่างรูปแบบจำลอง อย่างเป็นระเบียบและรูปแบบจำลองจากการปรับขนาดเอลิเมนต์สำหรับปัญหา คานปลายยื่นชนิดสั้นภายใต้แรงกด.....	94
7.23 ปัญหาแผ่นโลหะรูปตัวแอลภายใต้แรงดึง.....	94
7.24 รูปแบบจำลองไฟไนต์เอลิเมนต์อย่างเป็นระเบียบและรูปแบบจำลองไฟไนต์ เอลิเมนต์อย่างไร้ระเบียบ	
ก) Mesh S1.....	96
ข) Mesh S2.....	96
ค) Mesh S3.....	96
ง) Mesh S4.....	96
จ) Mesh U1.....	96
ฉ) Mesh U2.....	96
ช) Mesh U3.....	97
ฌ) Mesh U4.....	97
7.25 ผลการเปรียบเทียบการลู่เข้าของผลเฉลยโดยประมาณระหว่างรูปแบบจำลอง อย่างเป็นระเบียบและรูปแบบจำลองจากการปรับขนาดเอลิเมนต์สำหรับปัญหา แผ่นโลหะรูปตัวแอลภายใต้แรงดึง.....	98

คำอธิบายสัญลักษณ์

A	พื้นที่, เมทริกซ์สัมประสิทธิ์
A_d	พื้นที่โดเมนรวม
a_i	สัมประสิทธิ์ของฟังก์ชันการประมาณภายใน
B	เมทริกซ์สัมประสิทธิ์, โหลดเวกเตอร์ขอบเขต
b_i	สัมประสิทธิ์ของฟังก์ชันการประมาณภายใน
C	เมทริกซ์สัมประสิทธิ์
c	คุณสมบัติของวัสดุ
c_i	สัมประสิทธิ์ของฟังก์ชันการประมาณภายใน
D	ตัวดำเนินการเชิงอนุพันธ์
D_x	เมทริกซ์ส่วนประกอบของพลาทซ์ในแนวแกน x
D_y	เมทริกซ์ส่วนประกอบของพลาทซ์ในแนวแกน y
d	ตัวแปรทุติยภูมิ
E	โมดูลัสยืดหยุ่น, ส่วนประกอบของพลาทซ์ในแนวแกน x , ค่าความคลาดเคลื่อน
e	ค่าความคลาดเคลื่อน
$\ E\ $	นอร์มของความคลาดเคลื่อน
$\ E\ _{L_2}$	แอลทูนอร์มของความคลาดเคลื่อน
$^{\circ}\ E\ $	ค่าความคลาดเคลื่อนแบบโพสเทอริอริ
F	แรงที่กระทำกับขอบนอกของโครงสร้าง, โหลดเวกเตอร์, ส่วนประกอบของพลาทซ์ในแนวแกน y
f	ฟังก์ชันแหล่งกำเนิด
f_x	แรงวัตถุต่อหนึ่งหน่วยมวลในแนวแกน x
f_y	แรงวัตถุต่อหนึ่งหน่วยมวลในแนวแกน y
G	เมทริกซ์สัมประสิทธิ์
g	เวกเตอร์ของพลาทซ์
g^{EX}	พลาทซ์ซึ่งคำนวณจากผลเฉลยแม่นยำ
g^{FB}	พลาทซ์ซึ่งคำนวณด้วยการใช้หลักการของพลาทซ์
g^{ST}	พลาทซ์ซึ่งคำนวณด้วยระเบียบวิธีแบบดั้งเดิม
h	ขนาดโดยเฉลี่ยของเอลิเมนต์, มิติของเอลิเมนต์คุณลักษณะเฉพาะ

K	เมทริกซ์สัมประสิทธิ์
k	สัมประสิทธิ์การนำความร้อน
L	ตัวแปรระยะทาง
L_i	พิกัดธรรมชาติ
M	เมทริกซ์สัมประสิทธิ์
N_i	ฟังก์ชันการประมาณภายในเอลิเมนต์แบบตัวแปรไร้จุดต่อ
\bar{N}_i	ฟังก์ชันการประมาณภายในเอลิเมนต์แบบเชิงเส้น
\hat{n}	ทิศทางตั้งฉาก
n_x	เวกเตอร์หนึ่งหน่วยในทิศทาง x
n_y	เวกเตอร์หนึ่งหน่วยในทิศทาง y
Q	อัตราปริมาณความร้อนที่ผลิตได้เองต่อหนึ่งหน่วยมวล
q_s	ปริมาณความร้อนที่ผ่านขอบ
q_x	ปริมาณความร้อนในแนวแกน x
q_y	ปริมาณความร้อนในแนวแกน y
R	เศษตกค้าง, โหลดเวกเตอร์แหล่งกำเนิด
s	อัตราส่วนระหว่างความยาวเอลิเมนต์ทั้งสองด้าน
T	อุณหภูมิ
T_0	อุณหภูมิอ้างอิง
U	ตัวแปรปฐมภูมิ
U_∞	ตัวแปรทุติยภูมิ
u	ระยะการเคลื่อนตัวในแนวแกน x
v	ระยะการเคลื่อนตัวในแนวแกน y
W	ฟังก์ชันน้ำหนัก
x	ระยะในแนวราบ
y	ระยะในแนวตั้ง
α	สัมประสิทธิ์การขยายตัวเนื่องจากอุณหภูมิ
Γ	บริเวณขอบของขอบเขตการคำนวณ
$\Gamma^{(e)}$	บริเวณขอบของเอลิเมนต์
γ	ความเครียดเฉือน
δ	ระยะการเคลื่อนตัว

ε	ความเครียด
θ	ดัชนีแสดงผล
η	เปอร์เซ็นต์ความคลาดเคลื่อนแบบโพสเทอริออร์ริซึมพัทธ์
λ	ตัวแปรในการปรับขนาดเอลิเมนต์
ν	อัตราส่วนปัวส์ซอง
σ	ความเค้นในแนวตั้งฉาก
τ	ความเค้นเฉือน
ϕ	ตัวแปรใด ๆ
ϕ'	ตัวแปรตามแมนตรง
Ω	โดเมนของขอบเขตการคำนวณ
$\Omega^{(e)}$	โดเมนของเอลิเมนต์



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระเบียบวิธีไฟไนต์เอลิเมนต์เป็นระเบียบวิธีเชิงตัวเลขวิธีหนึ่งที่ได้รับการพัฒนาอย่างต่อเนื่อง [1] เพื่อใช้วิเคราะห์ปัญหาต่าง ๆ ทางวิศวกรรม เช่น ปัญหาความแข็งแรงของโครงสร้าง ปัญหาการถ่ายเทความร้อน ปัญหาลักษณะการไหลของของไหล รวมถึงปัญหาความเค้นเนื่องจากความร้อนที่เกิดขึ้นบนโครงสร้าง [2] และปัญหาด้านอื่น ๆ อีกเป็นจำนวนมากในปัจจุบัน ระเบียบวิธีไฟไนต์เอลิเมนต์ได้ถูกนำมาประยุกต์ใช้กันโดยแพร่หลายเนื่องจากสามารถจำลองรูปแบบของปัญหาที่มีลักษณะซับซ้อนได้โดยสะดวก หลักการโดยทั่วไปของระเบียบวิธีไฟไนต์เอลิเมนต์นี้ เริ่มต้นจากการแบ่งรูปแบบของปัญหา (discretization) ออกเป็นชิ้นส่วนย่อย ๆ ที่เรียกว่าเอลิเมนต์ (elements) จากนั้นจึงทำการสร้างสมการสำหรับแต่ละเอลิเมนต์ (element equations) ที่แบ่งขึ้นมาทำให้สอดคล้องกับชนิดของปัญหาที่ต้องการวิเคราะห์นั้น แล้วจึงนำสมการเหล่านี้มารวมเข้าด้วยกันซึ่งเปรียบเสมือนการนำเอลิเมนต์ย่อย ๆ นั้นมาประกอบรวมกันขึ้นเป็นรูปแบบจำลองของปัญหาที่แท้จริง การรวมตัวกันของสมการย่อยเหล่านี้ ก่อให้เกิดระบบสมการรวม (system of equations) ซึ่งหลังจากทำการกำหนดเงื่อนไขขอบเขต (boundary conditions) ที่เหมาะสมแล้ว จึงแก้ระบบสมการรวมนั้นเพื่อหาผลลัพธ์โดยประมาณตามตำแหน่งต่าง ๆ ของปัญหานั้น

เอลิเมนต์ที่ใช้ในระเบียบวิธีไฟไนต์เอลิเมนต์สำหรับปัญหาสองมิติมีทั้งเอลิเมนต์สามเหลี่ยมและเอลิเมนต์สี่เหลี่ยม [3] โดยเอลิเมนต์สามเหลี่ยมนั้นเป็นที่นิยมมาก เนื่องจากแบบจำลองไฟไนต์เอลิเมนต์ที่ประกอบด้วยเอลิเมนต์สามเหลี่ยมจำนวนมากสามารถสร้างขึ้นได้ง่ายและโดยอัตโนมัติ [4] อีกทั้งสามารถประยุกต์ใช้งานกับขอบเขตที่เป็นส่วนโค้งได้เป็นอย่างดี เอลิเมนต์สามเหลี่ยมทั่วไปมีหลายชนิด [5] ได้แก่ 1) เอลิเมนต์สามเหลี่ยมสามจุดต่อซึ่งมีการประมาณภายในเอลิเมนต์เป็นแบบเชิงเส้น (linear interpolation functions), 2) เอลิเมนต์สามเหลี่ยมหกจุดต่อซึ่งมีการประมาณภายในเอลิเมนต์เป็นแบบควอดราติก (quadratic interpolation functions) และ 3) เอลิเมนต์สามเหลี่ยมสิบจุดต่อซึ่งมีการประมาณภายในเอลิเมนต์เป็นแบบคิวบิก (cubic interpolation functions) อีกทั้งเอลิเมนต์สามเหลี่ยมยังมีจุดเด่นที่สามารถประดิษฐ์ไฟไนต์เอลิเมนต์เมทริกซ์ (finite element matrices) ให้อยู่ในรูปแบบปิด (closed-form) เพียงแต่การประดิษฐ์ไฟไนต์เอลิเมนต์เมทริกซ์สำหรับสามเหลี่ยมหกจุดต่อและสิบ

จุดต่อ นั้น มีความซับซ้อนมากกว่าสามเหลี่ยมสามจุดต่อ ในงานวิจัยชิ้นนี้จึงได้นำแนวคิดเกี่ยวกับตัวแปรไร้จุดต่อ [6] ซึ่งประสบความสำเร็จในการประยุกต์ใช้กับปัญหาความแข็งแรงของโครงสร้างโดยระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อนั้นใช้จุดต่อจำนวนสามจุดต่อเท่ากับเอลิเมนต์สามเหลี่ยมสามจุดต่อ หากแต่มีการประมาณภายในเอลิเมนต์เป็นแบบควอดราติกเช่นเดียวกับเอลิเมนต์สามเหลี่ยมหกจุดต่อ แล้วจึงประยุกต์ระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์ [7] เข้ากับระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อเพื่อลดความซับซ้อนในการประดิษฐ์ไฟไนต์เอลิเมนต์เมทริกซ์

ความถูกต้องเที่ยงตรงของผลลัพธ์ที่คำนวณได้โดยทั่วไปแล้ว จะขึ้นอยู่กับขนาดเอลิเมนต์ที่ใช้ กล่าวคือ หากต้องการค่าความถูกต้องเที่ยงตรงสูงก็จำเป็นต้องใช้เอลิเมนต์ขนาดเล็กเป็นจำนวนมาก ยิ่งใช้เอลิเมนต์ขนาดเล็กมากเท่าใดก็จะได้ผลลัพธ์ที่มีความถูกต้องมากยิ่งขึ้นเท่านั้น แต่ในขณะเดียวกัน จำนวนเอลิเมนต์ทั้งหมดที่ใช้ในการจำลองแบบของปัญหานั้นจะเพิ่มมากขึ้นตามไปด้วย ซึ่งอาจจะเกินขนาดของหน่วยความจำของเครื่องคอมพิวเตอร์ที่มีอยู่ หรืออาจใช้เวลาในการคำนวณมากจนเกินความจำเป็น วิธีหนึ่งที่สามารถนำมาแก้ปัญหาข้างต้นโดยยังคงรักษาค่าความถูกต้องเที่ยงตรงที่สูง ก็คือ วิธีการปรับขนาดเอลิเมนต์โดยอัตโนมัติ (adaptive remeshing) [3] ซึ่งจะวางเอลิเมนต์ขนาดเล็กในบริเวณที่มีการเปลี่ยนแปลงของผลลัพธ์สูง และวางเอลิเมนต์ขนาดใหญ่ในบริเวณที่มีการเปลี่ยนแปลงของผลลัพธ์ต่ำ หลักการของวิธีดังกล่าวก่อให้เกิดอุปสรรคในการสร้างรูปแบบของไฟไนต์เอลิเมนต์ กล่าวคือ การจัดเอลิเมนต์ซึ่งมีขนาดไม่เท่ากันแต่ต้องมีความต่อเนื่องอย่างค่อยเป็นค่อยไป และสำหรับปัญหาทั่วไปในทางปฏิบัติซึ่งลักษณะของผลลัพธ์นั้นจะไม่สามารถทราบล่วงหน้า อุปสรรคข้างต้นอาจก่อให้เกิดความไม่มั่นใจในวิธีดังกล่าว ทั้งในแง่ของความถูกต้องแม่นยำและจำนวนครั้งในการปรับเอลิเมนต์โดยอัตโนมัติ

การกระเนควมคลาดเคลื่อนถูกนำมาใช้โดยมีวัตถุประสงค์เพื่อเพิ่มความมั่นใจในวิธีการปรับขนาดเอลิเมนต์นั้นว่ามีความถูกต้องและแม่นยำมาก โดยเริ่มต้นจากการตรวจหาเอลิเมนต์ที่มีความคลาดเคลื่อนสูงที่สุด แล้วจึงแบ่งแยกความคลาดเคลื่อนนี้ให้น้อยลงจนอยู่ภายใต้ขอบเขตที่ยอมรับได้ ในการวิเคราะห์ปัญหาบางปัญหาอาจมีการปรับขนาดเอลิเมนต์หลายครั้ง ดังนั้นการกระเนควมคลาดเคลื่อนจึงสามารถประยุกต์ใช้เพื่อเป็นดัชนีชี้วัดการยุติการปรับขนาดเอลิเมนต์นั่นเอง เพียงแต่ การกระเนควมคลาดเคลื่อนแบบคลาสสิกนั้นใช้ทรัพยากรในการคำนวณจำนวนมาก จึงมีการพยายามที่จะสร้างและประดิษฐ์การกระเนควมคลาดเคลื่อนแบบใหม่นั้นคือ “การกระเนควมคลาดเคลื่อนแบบโพสเทอริออริ” [8, 9] ซึ่งการกระเนควมคลาดเคลื่อนแบบโพสเทอริออรินี้เป็นการกระเนควมคลาดเคลื่อนภายหลังจากที่ได้ผลเฉลยจากระเบียบวิธีไฟไนต์

เอลิเม้นต์ ซึ่งกระบวนการที่ว่่านี้จำเป็นที่จะสร้างค่าประมาณอันดับสูงซึ่งได้มาจากผลเฉลยของระเบียบวิธีไฟไนต์เอลิเม้นต์ เพื่อทำการคะเนความคลาดเคลื่อนโดยการแทนค่าดังกล่าวลงในผลเฉลยแม่นตรง

งานวิจัยนี้จะอธิบายถึงการสร้างและปรับขนาดของเอลิเม้นต์โดยอัตโนมัติสำหรับปัญหาการถ่ายเทความร้อน ปัญหาความแข็งแรงในโครงสร้าง และปัญหาการพิจารณาความเค้นและความเครียดในโครงสร้างขณะที่มีการถ่ายเทความร้อน โดยปัญหาดังกล่าวเป็นปัญหาที่ยังไม่ทราบลักษณะของผลลัพธ์มาก่อน งานวิจัยจะอธิบายอย่างเป็นขั้นตอนผ่านตัวอย่างต่าง ๆ และในช่วงหลังของงานวิจัยนี้จะอธิบายถึงหลักการการคะเนความคลาดเคลื่อน การคะเนความคลาดเคลื่อนแบบโพสเทอริออริ และดัชนีชี้วัดความคลาดเคลื่อน รวมถึงการนำเทคนิคการคะเนความคลาดเคลื่อนแบบโพสเทอริออริมาใช้ เพื่อเพิ่มความแม่นยำในการวิเคราะห์ฟลักซ์ความร้อน ความเค้น และความเครียด

1.2 วัตถุประสงค์ของการวิจัย

- 1.2.1 ศึกษาระเบียบวิธีไฟไนต์เอลิเม้นต์แบบตัวแปรไร้จุดต่อสำหรับวิเคราะห์ปัญหาการถ่ายเทความร้อน
- 1.2.2 ศึกษาระเบียบวิธีไฟไนต์เอลิเม้นต์แบบตัวแปรไร้จุดต่อสำหรับวิเคราะห์ความเค้นในปัญหาของแข็ง
- 1.2.3 ศึกษาลักษณะความเค้นเนื่องจากความร้อนที่เกิดขึ้นบนโครงสร้าง
- 1.2.4 ศึกษาระเบียบวิธีไฟไนต์เอลิเม้นต์ด้วยการใช้หลักการของฟลักซ์และประยุกต์เข้ากับระเบียบวิธีไฟไนต์เอลิเม้นต์แบบตัวแปรไร้จุดต่อ เพื่อใช้แก้ปัญหาการถ่ายเทความร้อนและปัญหาของแข็งภายใต้สภาวะคงตัว
- 1.2.5 ประดิษฐ์โปรแกรมคอมพิวเตอร์ที่สามารถทำนายลักษณะความเค้นและความเครียดของโครงสร้างขณะที่มีการถ่ายเทความร้อน
- 1.2.6 ศึกษาและประยุกต์ใช้เทคนิคการปรับขนาดเอลิเม้นต์ เพื่อเพิ่มประสิทธิภาพในการคำนวณ
- 1.2.7 ศึกษาและประยุกต์การคะเนความคลาดเคลื่อนแบบโพสเทอริออริ เพื่อสร้างเงื่อนไขในการยุติการปรับขนาดเอลิเม้นต์ซึ่งปกติจำเป็นต้องใช้ความชำนาญของผู้วิเคราะห์ปัญหา
- 1.2.8 พัฒนาเทคนิคการคะเนความคลาดเคลื่อนแบบโพสเทอริออริ เพื่อเพิ่มความแม่นยำในการวิเคราะห์ฟลักซ์ความร้อน ความเค้น และความเครียด

1.3 ขอบเขตของการวิจัย

ขอบเขตของการศึกษาที่สอดคล้องกับวัตถุประสงค์ข้างต้น มีดังต่อไปนี้

- 1.3.1 ศึกษาถึงพฤติกรรมการถ่ายเทความร้อนและการเสียรูปของโครงสร้างภายใต้สภาวะคงตัวในสองมิติ
- 1.3.2 ศึกษาถึงพฤติกรรมของความเค้นเนื่องจากความร้อนที่เกิดขึ้นบนโครงสร้างภายใต้สภาวะคงตัวในสองมิติ
- 1.3.3 ละทิ้งน้ำหนักของตัวเองในการศึกษาปัญหาของแข็ง
- 1.3.4 ความสัมพันธ์ของความเค้นและความเครียดของโครงสร้างมีลักษณะยืดหยุ่นเชิงเส้น (linear elastic) และเปลี่ยนรูปน้อย (small deformation)
- 1.3.5 พัฒนาระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์ให้สามารถประยุกต์เข้ากับระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ
- 1.3.6 ศึกษาและพัฒนาเทคนิคการปรับขนาดเอลิเมนต์ เพื่อเพิ่มประสิทธิภาพในการคำนวณ
- 1.3.7 ศึกษาการคะเนความคลาดเคลื่อนแบบโพสเทรืออริ เพื่อประยุกต์เข้ากับเทคนิคการปรับขนาดเอลิเมนต์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 มีความเข้าใจในระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อและระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์
- 1.4.2 ก่อให้เกิดความรู้ความเข้าใจถึงความสัมพันธ์ระหว่างการถ่ายเทความร้อน และกลศาสตร์ของแข็ง อันเป็นแนวทางพื้นฐานที่เป็นประโยชน์ต่องานวิจัยระดับสูงต่อไป
- 1.4.3 สามารถประดิษฐ์ไฟไนต์เอลิเมนต์เมทริกซ์ในรูปแบบปิดสำหรับเอลิเมนต์สามเหลี่ยมที่มีฟังก์ชันการประมาณภายในเป็นฟังก์ชันควอดราติก
- 1.4.4 โปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นสามารถทำนายพฤติกรรมการกระจายของอุณหภูมิ และลักษณะของความเค้นและความเครียดอันเนื่องจากความร้อนบนโครงสร้างได้
- 1.4.5 สามารถนำไปสู่การพัฒนาโปรแกรมคอมพิวเตอร์ที่สามารถนำไปใช้แก้ปัญหามีรูปร่างหรือเงื่อนไขขอบเขตที่แตกต่างกันได้

- 1.4.6 สามารถนำไปขยายผลเพื่อศึกษาปัญหาชนิดอื่น ๆ เช่น ปัญหาการถ่ายเทความร้อนภายใต้สภาวะไม่คงตัว หรือ การวิเคราะห์ลักษณะของความเค้นและความเครียดภายใต้ภาวะที่มีการเปลี่ยนแปลง หรือ ประยุกต์ระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์และแบบตัวแปรไร้จุดต่อเข้ากับปัญหาการไหลแบบราบเรียบชนิดอัดตัวไม่ได้ภายใต้สภาวะคงตัว เป็นต้น

1.5 วิธีดำเนินการวิจัย

- 1.5.1 ศึกษาหลักการและทฤษฎีด้านการถ่ายเทความร้อน กลศาสตร์ของแข็ง และระเบียบวิธีเชิงตัวเลขที่เกี่ยวข้องกับงานวิจัยนี้
- 1.5.2 ศึกษาและประดิษฐ์ระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์สำหรับเอลิเมนต์สามเหลี่ยมแบบตัวแปรไร้จุดต่อ
- 1.5.3 ศึกษาลักษณะการกระจายอุณหภูมิในปัญหาการถ่ายเทความร้อน และลักษณะการกระจายความเค้นในปัญหาของแข็ง
- 1.5.4 ศึกษาลักษณะความเค้นเนื่องมาจากความร้อนที่เกิดขึ้นบนโครงสร้าง
- 1.5.5 ประดิษฐ์โปรแกรมคอมพิวเตอร์ที่สอดคล้องกับปัญหาข้างต้น
- 1.5.6 ประยุกต์เทคนิคการปรับขนาดเอลิเมนต์
- 1.5.7 ทดสอบความถูกต้องของวิธีที่นำเสนอทั้งหมดข้างต้น ด้วยการเปรียบเทียบผลลัพธ์ที่ได้กับปัญหาที่มีผลเฉลยแม่นยำตรงและปัญหาที่มีความซับซ้อน
- 1.5.8 ประยุกต์การคะเนความคลาดเคลื่อนแบบโพสเทอริอริ
- 1.5.9 ทดสอบโปรแกรมที่พัฒนาขึ้นกับปัญหาต่าง ๆ
- 1.5.10 จัดทำรายงานเพื่อนำเสนอ และสรุปผล

1.6 เอกสารและงานวิจัยที่เกี่ยวข้อง

- 1.6.1 Dechaumphai, P. and Thornton, E.A. [6] พัฒนาระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ เพื่อวิเคราะห์ปัญหาความเค้นเนื่องจากความร้อนที่เกิดขึ้นบนโครงสร้าง โดยระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ นั้นใช้จุดต่อจำนวนสามจุดต่อเท่ากับเอลิเมนต์สามเหลี่ยมสามจุดต่อหากแต่มีการประมาณภายในเอลิเมนต์เป็นแบบควอดราติกเช่นเดียวกับเอลิเมนต์สามเหลี่ยมหกจุดต่อ
- 1.6.2 Pandey, A. K., Dechaumphai, P. and Wieting, A. R. [10] ประยุกต์ระเบียบวิธีลักษณะฟลักซ์เชิงเส้นเพื่อการวิเคราะห์ปัญหาความเค้นเนื่องจากความร้อนที่เกิดขึ้นบนโครงสร้างภายใต้สภาวะคงตัว โดยกำหนดฟลักซ์ของเอลิเมนต์ให้อยู่ใน

ลักษณะเชิงเส้นเช่นเดียวกับตัวแปรไม่ทราบค่าทั่วไป ทำให้สามารถสร้างไฟไนต์เอลิเมนต์เมทริกซ์ให้อยู่ในรูปแบบปิด ผลลัพธ์ที่ได้มีความถูกต้องแม่นยำมากขึ้น รวมถึงถึงความเร็วในการคำนวณที่เพิ่มมากขึ้นเช่นเดียวกัน

- 1.6.3 Polesky, S. P. [13] ประยุกต์ระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของพลักซ์เพื่อวิเคราะห์ปัญหาการถ่ายเทความร้อนของโครงสร้างใน 1 มิติ และ 2 มิติ โดยแสดงรายละเอียดการประดิษฐ์สมการไฟไนต์เอลิเมนต์และไฟไนต์เอลิเมนต์เมทริกซ์ที่อยู่ในรูปแบบปิด รวมถึงการเพิ่มองศาความอิสระในการประมาณการกระจายตัวของตัวแปรไม่ทราบค่าบนเอลิเมนต์โดยใช้ตัวแปรไร้จุดต่อ
- 1.6.4 Paweenawat, A. and Dechaumphai, P. [11] ประยุกต์ระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อเข้ากับการวิเคราะห์การไหลแบบหนึ่งมิติอัดตัวไม่ได้ โดยใช้ฟังก์ชันการประมาณภายในเอลิเมนต์แบบควอดราติกสำหรับพจน์ของความเร็ว และใช้ฟังก์ชันการประมาณภายในเอลิเมนต์แบบเชิงเส้นสำหรับพจน์ของความดัน
- 1.6.5 Phongthanapanich, S., Traivivatana, S., Boonmaruth, P. and Dechaumphai, P. [7] ประยุกต์ระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของพลักซ์เข้ากับระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ เพื่อประดิษฐ์ไฟไนต์เอลิเมนต์เมทริกซ์สำหรับตัวแปรไร้จุดต่อให้อยู่ในรูปแบบปิด และนำเทคนิคการปรับขนาดเอลิเมนต์มาใช้เพื่อเพิ่มความถูกต้องให้กับผลลัพธ์ โดยนำไปประยุกต์เข้ากับการวิเคราะห์การถ่ายเทความร้อนภายใต้สภาวะคงตัวและไม่คงตัว
- 1.6.6 Kelly, D. W., De S.R. Gago, J. P. Zienkiewicz, O. C. and Babuska, I. [14, 15] นำเสนอการวิเคราะห์ความคลาดเคลื่อนและขั้นตอนในการปรับขนาดเอลิเมนต์สำหรับระเบียบวิธีไฟไนต์เอลิเมนต์ โดยแบ่งออกเป็น 2 ส่วน ได้แก่ 1) ทฤษฎีและระเบียบวิธีในการคะเนความคลาดเคลื่อน และ 2) กลยุทธ์ในการปรับขนาดเอลิเมนต์ โดยเน้นไปที่ระเบียบวิธีปรับขนาดแบบพี (p - refinement)
- 1.6.7 Zienkiewicz, O. C. and Zhu, J. Z. [8] นำเสนอการคะเนความคลาดเคลื่อนแบบใหม่ ซึ่งมีความแม่นยำและสะดวกต่อการนำไปประยุกต์ใช้กับไฟไนต์เอลิเมนต์โปรแกรม โดยสามารถคะเนความคลาดเคลื่อนโดยรวม (global error) และความคลาดเคลื่อนเฉพาะที่ (local error) ได้ดี
- 1.6.8 Ainsworth, M., Zhu, J. Z., Craig, A. W. and Zienkiewicz, O. C. [12] วิเคราะห์ตัวคะเนความคลาดเคลื่อนแบบโพสเทอริออริสำหรับการวิเคราะห์ด้วย

ระเบียบวิธีไฟไนต์เอลิเมนต์ ซึ่ง Zienkiewicz และ Zhu ได้นำเสนอก่อนหน้านี้ และแสดงให้เห็นถึงประสิทธิภาพและการลู่เข้า (convergence) ของการคะเน ความคลาดเคลื่อนแบบโพสเทอริอริ

- 1.6.9 วิทยา วัฒนนุกูลชัย [26] นำเสนอเทคนิคการปรับปริมาตรควบคุมเฉพาะบริเวณ (local grid refinement) โดยใช้ค่าความผิดพลาดตกค้าง (residual error) เป็นเงื่อนไขในการปรับขนาดและเงื่อนไขในการหยุดการปรับขนาด สำหรับการแก้ปัญหาความเค้นเนื่องจากอุณหภูมิใน 2 มิติ ด้วยระเบียบวิธีไฟไนต์วอลุ่ม



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

สมการเชิงอนุพันธ์ย่อยสำหรับการถ่ายเทความร้อนและของแข็ง

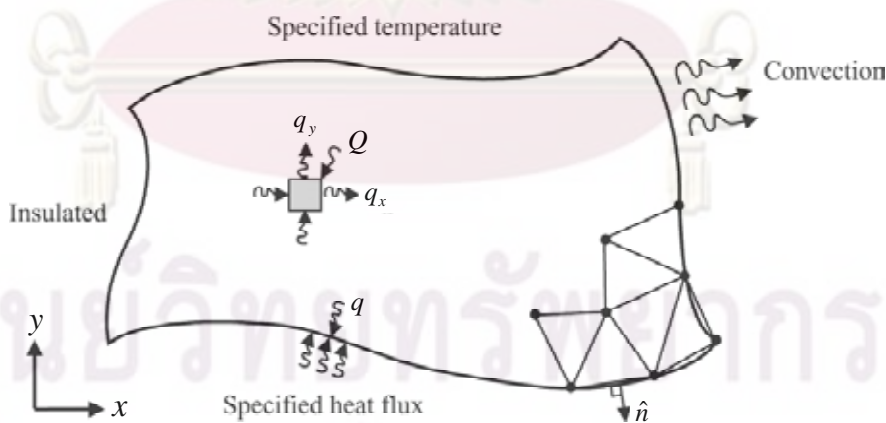
งานวิจัยนี้จะศึกษาปัญหาปฏิสัมพันธ์ระหว่างโครงสร้างและอุณหภูมิในสองมิติ โดยใช้ระเบียบวิธีไฟไนต์เอลิเมนต์ การทำความเข้าใจในการหาลักษณะความเค้นเนื่องจากความร้อนที่เกิดขึ้นในโครงสร้างจะต้องอาศัยทฤษฎีการถ่ายเทความร้อน และทฤษฎีกลศาสตร์ของแข็ง ดังนั้นในบทนี้จะได้กล่าวถึงทฤษฎีการถ่ายเทความร้อน [17] เพื่อหาสมการเชิงอนุพันธ์ย่อยสำหรับปัญหาการถ่ายเทความร้อน และทฤษฎีกลศาสตร์ของแข็ง [18] เพื่อหาสมการเชิงอนุพันธ์ย่อยสำหรับปัญหาของแข็ง ตามลำดับ

2.1 สมการเชิงอนุพันธ์ย่อยสำหรับปัญหาการถ่ายเทความร้อน

สมการเชิงอนุพันธ์ที่อธิบายความสมดุลของปริมาณพลักซ์ q_x และ q_y ซึ่งผ่านเข้าและออกในแนวแกน x และ y ของเอลิเมนต์เล็ก ๆ ที่วางตัวอยู่ ณ ตำแหน่งใด ๆ บนแผ่นระนาบดังแสดงในรูปที่ 2.1 คือ

$$\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} - Q = 0 \quad (2.1)$$

โดย Q แทนอัตราปริมาณความร้อนที่ผลิตได้เอง (internal heat generation) ต่อหนึ่งหน่วยปริมาตร



รูปที่ 2.1 ขอบเขตและเงื่อนไขขอบเขตของปัญหาการถ่ายเทความร้อน

ปริมาณพลักซ์เหล่านี้ขึ้นอยู่กับความชันของอุณหภูมิ (temperature gradients) T และค่าสัมประสิทธิ์การนำความร้อน (thermal conductivity) k ของแผ่นระนาบตามกฎฟูริเยร์ (Fourier's law) ดังนี้

$$q_x = -k \frac{\partial T}{\partial x} \quad (2.2ก)$$

$$q_y = -k \frac{\partial T}{\partial y} \quad (2.2ข)$$

แทนสมการ (2.2) ลงในสมการ (2.1) ก่อให้เกิดสมการเชิงอนุพันธ์สำหรับการถ่ายเทความร้อน ภายใต้สภาวะอยู่ตัว คือ

$$\frac{\partial}{\partial x} \left[k \frac{\partial T}{\partial x} \right] + \frac{\partial}{\partial y} \left[k \frac{\partial T}{\partial y} \right] + Q = 0 \quad (2.3)$$

ความซับซ้อนของลักษณะการกระจายของอุณหภูมิ $T(x, y)$ ที่เกิดขึ้นจากสมการ (2.3) นี้ ขึ้นอยู่กับรูปร่างลักษณะของแผ่นระนาบและเงื่อนไขขอบเขตที่กำหนดให้ เงื่อนไขขอบเขตตลอดขอบของแผ่นบางอาจประกอบด้วย

(ก) การกำหนดอุณหภูมิตลอดขอบ เช่น

$$T(x, y) = T_1(x, y) \quad (2.4ก)$$

(ข) การกำหนดปริมาณฟลักซ์ q_s ที่ผ่านเข้าสู่ขอบ

$$q_s = -q = k \frac{\partial T}{\partial x} n_x + k \frac{\partial T}{\partial y} n_y \quad (2.4ข)$$

โดย n_x และ n_y แทนทิศทางโคไซน์ (cosine directions) ของเวกเตอร์หนึ่งหน่วย \hat{n} ที่ตั้งฉากกับขอบนั้น

(ค) การกำหนดว่าขอบเป็นฉนวน ไม่มีปริมาณฟลักซ์ผ่านเข้าหรือออกได้

$$q_s = k \frac{\partial T}{\partial x} n_x + k \frac{\partial T}{\partial y} n_y = 0 \quad (2.4ค)$$

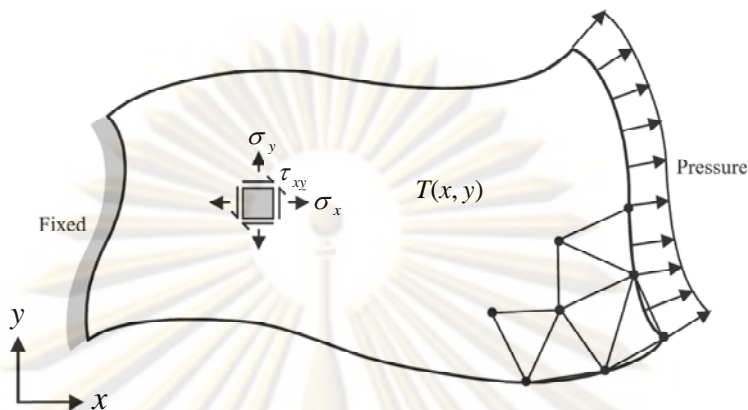
2.2 สมการเชิงอนุพันธ์ย่อยสำหรับปัญหาของแข็ง

สมการเชิงอนุพันธ์ย่อยซึ่งแสดงความสมดุลของแรงในแนวแกน x และ y บนแผ่นระนาบ เมื่อไม่คิดน้ำหนักของตัวเอง ดังแสดงในรูปที่ 2.2 คือ

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} = 0 \quad (2.5ก)$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} = 0 \quad (2.5ข)$$

โดย σ_x , σ_y แทนค่าความเค้นฉาก (normal stress) ในแนวแกน x และ y ตามลำดับ ส่วน τ_{xy} แทนค่าความเค้นเฉือน (shearing stress)



รูปที่ 2.2 ขอบเขตและเงื่อนไขขอบเขตของปัญหาของแข็ง

ในกรณีของแผ่นบางซึ่งเป็นปัญหาความเค้นในระนาบ (plane stress) ค่าความเค้นในแนวแกน z จะถูกสมมติให้มีค่าเป็นศูนย์ โดยค่าความเค้นย่อยต่าง ๆ สามารถเขียนให้อยู่ในรูปแบบของค่าความเครียดย่อย (strain components) ได้ดังนี้

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x - \alpha(T-T_0) \\ \varepsilon_y - \alpha(T-T_0) \\ \gamma_{xy} \end{Bmatrix} \quad (2.6)$$

โดย E แทนค่าโมดูลัสของยังส์ (Young's modulus) และ ν แทนค่าอัตราส่วนของปัวส์ซง (Poisson's ratio) ส่วน ε_x , ε_y แทนค่าความเครียดฉาก (normal strain) ในแนวแกน x และแกน y ตามลำดับ, γ_{xy} แทนค่าความเครียดเฉือน, α แทนสัมประสิทธิ์การขยายตัวเนื่องจากอุณหภูมิ (coefficient of thermal expansion) และ T_0 แทนอุณหภูมิอ้างอิงที่วัสดุไม่เกิดความเค้น (reference temperature for zero stress) เช่นอุณหภูมิห้อง

ค่าความเครียดย่อยเหล่านี้เขียนให้อยู่ในรูปของระยะการเคลื่อนตัว u และ v ในแนวแกน x และ y ได้คือ

$$\varepsilon_x = \frac{\partial u}{\partial x} ; \varepsilon_y = \frac{\partial v}{\partial y} \quad (2.7)$$

และ

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

ซึ่งหมายความว่า ตัวไม่รู้ค่าในปัญหาแผ่นระนาบมีเพียง 2 ค่า คือ u และ v เมื่อทราบค่า u และ v แล้วจึงสามารถนำไปคำนวณหาค่าความเครียดและความเค้นย่อยต่าง ๆ ได้ตามลำดับ

กรณีปัญหาความเครียดในระนาบ (plane strain) ค่าความเครียดในแนวแกน z ถูกสมมติให้มีค่าเป็นศูนย์ สมการเชิงอนุพันธ์ย่อย (2.5) และสมการ (2.7) ซึ่งแสดงความสัมพันธ์ระหว่างความเครียดย่อยและค่าการเสียรูป ยังคงใช้ได้เช่นเดิม ยกเว้นความสัมพันธ์ระหว่างค่าความเค้นย่อยและความเครียดย่อยดังแสดงในสมการ (2.6) ต้องเปลี่ยนมาเป็น

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x - \alpha(T-T_0) \\ \varepsilon_y - \alpha(T-T_0) \\ \gamma_{xy} \end{Bmatrix} \quad (2.8)$$

สำหรับเงื่อนไขขอบเขตโดยทั่วไปดังแสดงในรูปที่ 2.2 ประกอบด้วย การยึดแน่น (fixed) หรือปล่อยให้อิสระ (free) ตลอดขอบต่าง ๆ รวมทั้งอาจกำหนดแรงดัน (traction) ซึ่งแทนแรงที่กระทำต่อพื้นที่ตลอดขอบนั้น ๆ อีกทั้งยังต้องการผลการกระจายตัวของอุณหภูมิ $T = T(x, y)$ ที่สามารถคำนวณได้จากสมการ (2.3) ก่อนด้วย ซึ่งผลจากอุณหภูมิ T นี้เองแฝงตัวอยู่ในสมการ (2.6) และ (2.8) ซึ่งอธิบายลักษณะของความเค้นย่อยนั่นเอง

บทที่ 3

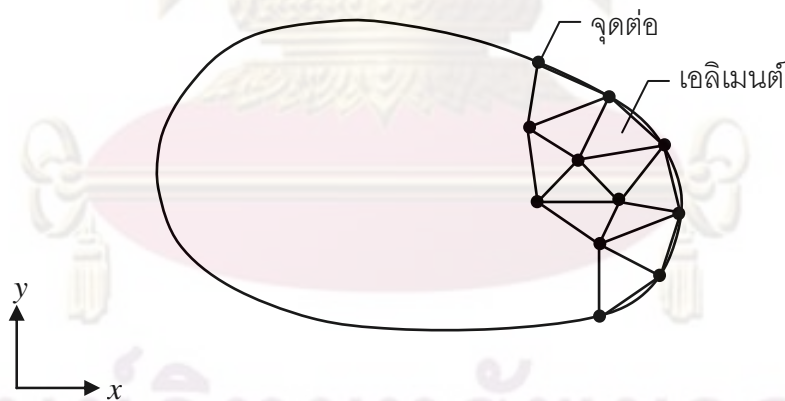
ระเบียบวิธีไฟไนต์เอลิเมนต์สำหรับการถ่ายเทความร้อนและของแข็ง

ในบทนี้จะเป็นการนำระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ (nodeless variables) มาประยุกต์ใช้ในการวิเคราะห์ปัญหาการถ่ายเทความร้อน และของแข็ง โดยจะเริ่มจากขั้นตอนทั่วไปของระเบียบวิธีไฟไนต์เอลิเมนต์ จากนั้นจะอธิบายรายละเอียดของระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ แล้วจึงอธิบายรูปแบบของระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์ และสุดท้ายจะเป็นการประดิษฐ์สมการไฟไนต์เอลิเมนต์เพื่อการวิเคราะห์การถ่ายเทความร้อน และความเค้นในของแข็ง

3.1 ขั้นตอนทั่วไปของระเบียบวิธีไฟไนต์เอลิเมนต์

การแก้ปัญหาไฟไนต์เอลิเมนต์โดยวิธีการถ่วงน้ำหนักเศษตค่างประกอบด้วยขั้นตอนที่สำคัญ 6 ขั้นตอน [1] คือ

ขั้นตอนที่ 1 แบ่งขอบเขตรูปร่างของปัญหาออกเป็นเอลิเมนต์ย่อย ๆ เช่น แบ่งออกเป็นเอลิเมนต์สามเหลี่ยมย่อยๆ สำหรับปัญหาในสองมิติ ดังรูปที่ 3.1



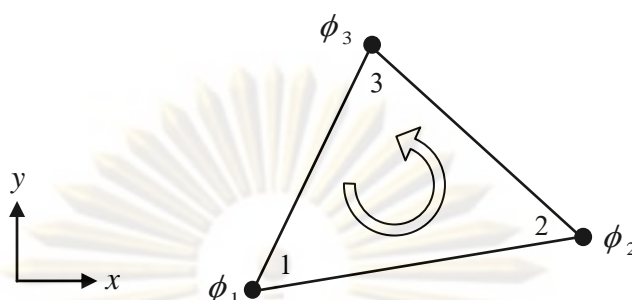
รูปที่ 3.1 การแบ่งขอบเขตรูปร่างของปัญหาออกเป็นเอลิเมนต์ย่อย ๆ

จากนั้นก็จึงหาสมการเชิงอนุพันธ์ที่สอดคล้องกับปัญหาที่ต้องการแก้ นั้น โดยสมการเชิงอนุพันธ์ทั่วไปสามารถเขียนให้อยู่ในรูป

$$D(\phi') = 0 \quad (3.1)$$

โดยที่ D คือตัวดำเนินการเชิงอนุพันธ์ (differential operator) และ ϕ' คือตัวแปรตามแน่นอนตรง

ขั้นตอนที่ 2 เลือกลักษณะการกระจายของผลเฉลยโดยประมาณบนเอลิเมนต์



รูปที่ 3.2 เอลิเมนต์สามเหลี่ยมแบบ 3 จุดต่อและตัวไม่ทราบค่าที่จุดต่อ

ยกตัวอย่างเช่น สำหรับเอลิเมนต์ที่ประกอบด้วยสามจุดต่อดังแสดงในรูปที่ 3.2 โดยที่จุดต่อนี้เป็นตำแหน่งของตัวไม่รู้ค่า ϕ_1 , ϕ_2 และ ϕ_3 ตัวไม่รู้ค่าเหล่านี้เป็นคุณสมบัติต่าง ๆ ของปัญหา ซึ่งสามารถสร้างสมการอธิบายลักษณะการกระจายของตัวไม่ทราบค่าที่จุดต่อได้ดังนี้

$$\phi(x, y) = N_1(x, y)\phi_1 + N_2(x, y)\phi_2 + N_3(x, y)\phi_3 \quad (3.2)$$

โดย $N_i(x, y)$; $i=1, 2, 3$ แทนฟังก์ชันการประมาณภายในเอลิเมนต์ สมการ (3.2) นี้สามารถเขียนให้อยู่ในรูปของเมทริกซ์ได้ดังนี้

$$\begin{aligned} \phi = \phi(x, y) &= [N_1 \quad N_2 \quad N_3] \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix} \\ &= [N(x, y)]_{(1 \times 3)} \{\phi\}_{(3 \times 1)} \end{aligned} \quad (3.3)$$

โดยที่ $[N]$ คือ เมทริกซ์ของฟังก์ชันการประมาณภายในเอลิเมนต์

$\{\phi\}$ คือ เวกเตอร์เมทริกซ์ที่ประกอบไปด้วยตัวไม่ทราบค่าที่จุดต่อของเอลิเมนต์นั้น

ขั้นตอนที่ 3 ประยุกต์ระเบียบวิธีถ่วงน้ำหนักเศษตกค้าง (Method of Weighted Residual; MWR) เข้ากับสมการเชิงอนุพันธ์ย่อยเพื่อให้ผลลัพธ์โดยประมาณนั้นมีความคลาดเคลื่อนน้อยที่สุด ซึ่งจะก่อให้เกิดสมการไฟไนต์เอลิเมนต์ (finite element equations) ที่สอดคล้องกัน ซึ่งสามารถเขียนในรูปเมทริกซ์ได้ดังนี้

$$D(\phi) \text{ จะไม่เท่ากับ } 0 \text{ แต่จะเท่ากับ } R$$

โดยที่ R คือเศษตกค้าง (residual) นั้นหมายถึง

$$R = D(\phi) = D\left(\sum_{i=1}^m N_i \phi_i\right) \quad (3.4)$$

โดย m คือจำนวนจุดต่อของเอลิเมนต์นั้น

จากวิธีการเกอริคิน (Galerkin) ซึ่งมีขั้นตอนโดยเริ่มจากการคูณเศษตกค้าง R ด้วยฟังก์ชันน้ำหนัก (weighting function; W) จากนั้นจึงอินทิเกรตตลอดทั้งโดเมนของเอลิเมนต์แล้วกำหนดผลที่ได้ให้เท่ากับศูนย์นั้นคือ

$$\int_{\Omega} W_i R d\Omega = 0 \quad i = 1, 2, \dots, m \quad (3.5)$$

ปัญหาการถ่ายเทความร้อนในของแข็งเราจะเลือกใช้ฟังก์ชันน้ำหนักเป็น $W_i = N_i$ ซึ่งเรียกโดยทั่วไปว่าบับโนฟ-กาเลอร์คิน (Bubnov-Galerkin)

ขั้นตอนที่ 4 อินทิเกรตทีละส่วน (integrate by part) เพื่อก่อให้เกิดพจน์ที่เกี่ยวข้องกับขอบเขตของเอลิเมนต์ ซึ่งหากแทนสมการ (3.4) ลงในสมการ (3.5) แล้วอินทิเกรตทีละส่วนจะได้

$$\begin{aligned} \int_{\Omega^{(e)}} W_i R d\Omega &= \int_{\Omega^{(e)}} W_i D\left(\sum_{i=1}^m N_i \phi_i\right) d\Omega \\ &= \underbrace{\int_{\Omega^{(e)}} (W_i, N_i, \phi_i) d\Omega}_{\text{พจน์ที่เกี่ยวข้องกับโดเมนของเอลิเมนต์ } \Omega^{(e)}} + \underbrace{\int_{\Gamma^{(e)}} (W_i, N_i, \phi_i) d\Gamma}_{\text{พจน์ที่เกี่ยวข้องกับขอบเขตของเอลิเมนต์ } \Gamma^{(e)}} = 0 \end{aligned} \quad (3.6)$$

ขั้นตอนที่ 5 แทนพจน์ที่เกี่ยวข้องกับขอบเขตของเอลิเมนต์ $\Gamma^{(e)}$ ด้วยภาวะขอบเขตอื่น ๆ ที่เกี่ยวข้อง ซึ่งจะก่อให้เกิดสมการของเอลิเมนต์ที่สมบูรณ์สำหรับปัญหาที่พิจารณา

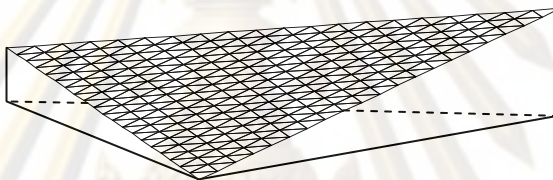
ขั้นตอนที่ 6 จากนั้นเขียนสมการของเอลิเมนต์ ซึ่งมีทั้งหมด m สมการให้อยู่ในรูปของเมทริกซ์ นั่นคือ

$$\begin{bmatrix} K \end{bmatrix}_{(m \times m)} \begin{Bmatrix} \phi \end{Bmatrix}_{(m \times 1)} = \begin{Bmatrix} F \end{Bmatrix}_{(m \times 1)} \quad (3.7)$$

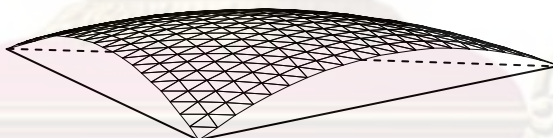
โดย $[K]$ คือ เอลิเมนต์เมทริกซ์ของความแข็งเกร็ง (element stiffness matrix) $\{\phi\}$ คือ เวกเตอร์ซึ่งประกอบด้วยตัวไม่รู้ค่าที่จุดต่อต่าง ๆ ของเอลิเมนต์ และ $\{F\}$ คือ โหลดเวกเตอร์ของเอลิเมนต์นั้น เมื่อได้สมการไฟไนต์เอลิเมนต์ดังเช่นแสดงในสมการ (3.7) แล้วลำดับขั้นตอนต่อไปก็จะทำการรวมสมการของเอลิเมนต์ย่อยเข้าด้วยกันก่อให้เกิดระบบสมการรวม จากนั้นกำหนดค่าที่ขอบเขตแล้วจึงแก้ระบบสมการรวมเพื่อหาค่าผลลัพธ์ที่จุดต่อต่าง ๆ

3.2 ระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ

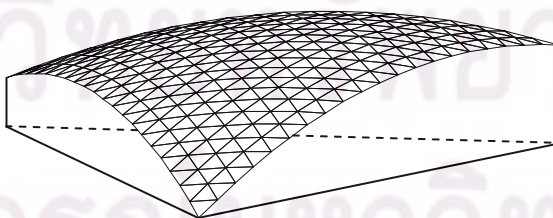
เอลิเมนต์สามเหลี่ยมเป็นเอลิเมนต์ที่นิยมใช้กันมากเนื่องจากแบบจำลองไฟไนต์เอลิเมนต์ที่ประกอบด้วยเอลิเมนต์สามเหลี่ยมจำนวนมากสามารถสร้างขึ้นมาโดยง่ายและโดยอัตโนมัติ [4] ซึ่งเอลิเมนต์สามเหลี่ยมแบบสามจุดต่อจะให้ลักษณะการกระจายตัวของค่าต่าง ๆ (เช่น อุณหภูมิสำหรับปัญหาการถ่ายเทความร้อน ระยะการเคลื่อนตัวสำหรับปัญหาของแข็ง เป็นต้น โดยจะใช้อุณหภูมิเป็นตัวอย่างเพื่ออธิบายต่อไป) บนเอลิเมนต์เป็นแบบเชิงเส้นดังแสดงในรูปที่ 3.3 และโดยการเพิ่มพจน์ที่เกี่ยวข้องกับตัวแปรไร้จุดต่อซึ่งมีลักษณะการกระจายตัวดังแสดงในรูปที่ 3.4 เพื่อที่จะสามารถประมาณการกระจายตัวของอุณหภูมิบนเอลิเมนต์ด้วยฟังก์ชันควอดราติกดังแสดงรูปที่ 3.5 ได้



รูปที่ 3.3 ลักษณะการกระจายตัวของอุณหภูมิที่ตำแหน่งจุดต่อ



รูปที่ 3.4 ลักษณะการกระจายตัวของอุณหภูมิที่ตำแหน่งไร้จุดต่อ



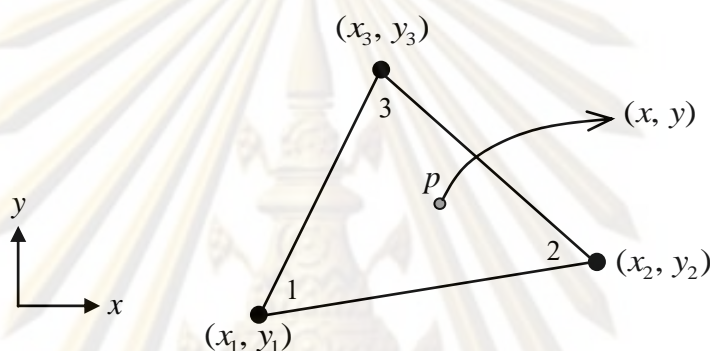
รูปที่ 3.5 ลักษณะการกระจายตัวของอุณหภูมิรวม

ดังนั้นในหัวข้อนี้จะแสดงการสร้างฟังก์ชันภายในเอลิเมนต์รูปสามเหลี่ยมแบบตัวแปรไร้จุดต่อโดยใช้พิกัดธรรมชาติ (natural coordinates) ซึ่งการใช้พิกัดดังกล่าวมีข้อดีก็คือจะได้

ฟังก์ชันการประมาณภายในเอลิเมนต์อยู่ในรูปแบบปิด และสามารถที่จะแสดงจุดพิกัดของจุดใด ๆ ที่อยู่ในเอลิเมนต์ให้อยู่ในรูปของจุดต่อของเอลิเมนต์นั้นได้ ซึ่งจะอยู่ในรูปแบบดังต่อไปนี้

$$T(x, y) = \underbrace{N_1T_1 + N_2T_2 + N_3T_3}_{\text{อุณหภูมิที่ตำแหน่งจุดต่อ}} + \underbrace{N_4T_4 + N_5T_5 + N_6T_6}_{\text{อุณหภูมิที่ตำแหน่งไร้จุดต่อ}} \quad (3.8)$$

รูปที่ 3.6 แสดงรายละเอียดของเอลิเมนต์สามเหลี่ยมสามจุดต่อในระบบแกนพิกัดฉาก โดยมีจุด p เป็นจุดที่อยู่ภายในเอลิเมนต์สามเหลี่ยม



รูปที่ 3.6 เอลิเมนต์สามเหลี่ยมสามจุดต่อ และจุด p ที่อยู่ภายในเอลิเมนต์

ถ้ากำหนดให้ L_i ($i = 1, 2, \dots, n$) เป็นฟังก์ชันธรรมชาติ โดยที่ n คือจำนวนจุดต่อของเอลิเมนต์ (ในที่นี้ $n = 3$) เราก็สามารถเขียนจุดพิกัดของจุด p ให้อยู่ในรูปของฟังก์ชันธรรมชาติ L_1, L_2 , และ L_3 ได้ดังนี้

$$\begin{aligned} x &= L_1x_1 + L_2x_2 + L_3x_3 \\ y &= L_1y_1 + L_2y_2 + L_3y_3 \end{aligned} \quad (3.9)$$

โดยฟังก์ชันธรรมชาติ L_1, L_2 , และ L_3 นี้สามารถเรียกใหม่ว่าฟังก์ชันถ่วงน้ำหนัก ซึ่งคุณสมบัติของฟังก์ชันถ่วงน้ำหนักก็คือผลรวมของฟังก์ชันจะมีค่าเท่ากับหนึ่ง ดังนี้

$$L_1 + L_2 + L_3 = 1 \quad (3.10)$$

เมื่อแก้สมการ (3.9) กับสมการ (3.10) พร้อม ๆ กันเพื่อเขียน L_1, L_2 , และ L_3 ให้อยู่ในระบบพิกัดแกน x และแกน y จะได้

$$L_i(x, y) = \frac{1}{2A} (a_i + b_i x + c_i y) \quad i = 1, 2, 3 \quad (3.11)$$

โดย A คือ พื้นที่ของเอลิเมนต์สามเหลี่ยม สามารถคำนวณได้จากพิกัดของจุดต่อทั้งสามจุด ดังนี้

$$A = \frac{1}{2}[x_2(y_3 - y_1) + x_1(y_2 - y_3) + x_3(y_1 - y_2)] \quad (3.12)$$

$$\begin{aligned} a_1 &= x_2 y_3 - x_3 y_2 & b_1 &= y_2 - y_3 & c_1 &= x_3 - x_2 \\ a_2 &= x_3 y_1 - x_1 y_3 & b_2 &= y_3 - y_1 & c_2 &= x_1 - x_3 \\ a_3 &= x_1 y_2 - x_2 y_1 & b_3 &= y_1 - y_2 & c_3 &= x_2 - x_1 \end{aligned} \quad (3.13)$$

สำหรับเอลิเมนต์สามเหลี่ยมสามจุดต่อหรือเอลิเมนต์สามเหลี่ยมที่มีอันดับสูง ๆ ขึ้นไป ฟังก์ชันการประมาณภายในของเอลิเมนต์สามเหลี่ยมนั้น ๆ สามารถเขียนให้อยู่ในรูปของพิกัดธรรมชาติ L_1, L_2 , และ L_3 ของเอลิเมนต์สามเหลี่ยมสามจุดต่อได้ ซึ่งนักวิจัยที่ชื่อ Silvester [19] ได้ให้สูตรสำหรับสร้างฟังก์ชันการประมาณภายในของเอลิเมนต์สามเหลี่ยมอันดับ n ใด ๆ ไว้ดังนี้

$$N_{\alpha\beta\gamma}(L_1, L_2, L_3) = N_\alpha(L_1) N_\beta(L_2) N_\gamma(L_3) \quad (3.14)$$

$$\text{โดยที่} \quad N_\alpha(L_1) = \prod_{i=1}^{\alpha} \left(\frac{nL_1 - i + 1}{i} \right), \quad \alpha \geq 1 \quad (3.15ก)$$

$$= 1, \quad \alpha = 0 \quad (3.15ข)$$

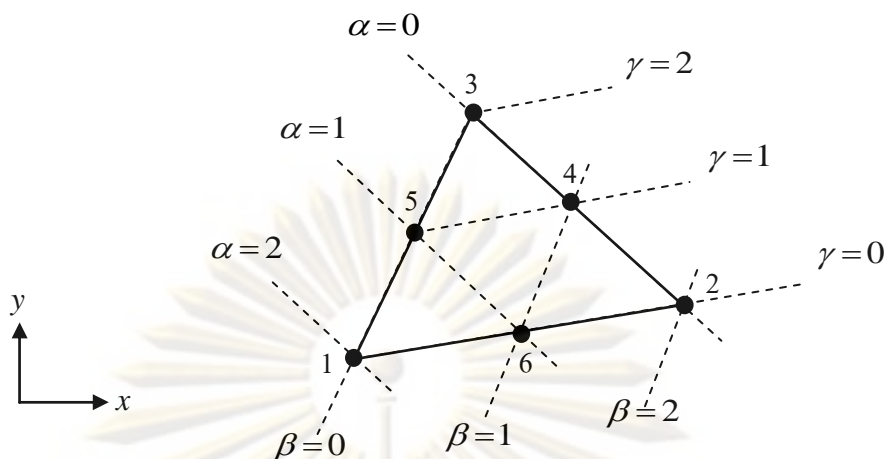
สำหรับ $N_\beta(L_2)$ และ $N_\gamma(L_3)$ ก็สามารถเขียนอยู่ในรูปแบบเดียวกับสมการ (3.15ก) และ (3.15ข) ส่วน α, β และ γ นั้นเป็นเลขจำนวนเต็มที่ใช้สำหรับอ้างอิงจุดต่อบนเอลิเมนต์ ซึ่งมีคุณสมบัติว่า $\alpha + \beta + \gamma = n$ โดยที่ n คืออันดับของเอลิเมนต์สามเหลี่ยม เช่น เอลิเมนต์สามเหลี่ยมหกจุดต่อเป็นเอลิเมนต์สามเหลี่ยมอันดับที่ 2 เป็นต้น ส่วนการใช้งานของเลขจำนวนเต็มชุดนี้ได้แสดงในรูปที่ 3.5 ยกตัวอย่างเช่นที่จุดต่อหมายเลข 1 จะมีหมายเลขอ้างอิงคือ $\alpha = 0, \beta = 0, \gamma = 0$ ซึ่งฟังก์ชันการประมาณภายในเอลิเมนต์ก็คือ N_{200} และจากสมการ (3.15ก) และ (3.15ข) จะได้ว่า

$$\begin{aligned} N_\alpha &= N_2 = \prod_{i=1}^2 \left(\frac{2L_1 - i + 1}{i} \right) \\ &= \left(\frac{2L_1 - 1 + 1}{1} \right) \left(\frac{2L_1 - 2 + 1}{2} \right) = L_1(2L_1 - 1) \end{aligned} \quad (3.16ก)$$

$$N_\beta = N_0 = 1 \quad (3.16ข)$$

$$N_\gamma = N_0 = 1 \quad (3.16ค)$$

$$\text{ดังนั้น} \quad N_{200} = N_2(L_1) N_0(L_2) N_0(L_3) = L_1(2L_1 - 1) \quad (3.17)$$



รูปที่ 3.7 การอ้างอิงหมายเลขจุดต่อด้วยตัวเลข α , β และ γ บนเอลิเมนต์สามเหลี่ยมหกจุดต่อ สำหรับจุดต่ออื่น ๆ ก็จะสามารถสร้างฟังก์ชันการประมาณภายในได้ด้วยวิธีเดียวกัน ดังนั้นสำหรับเอลิเมนต์สามเหลี่ยมหกจุดต่อจะมีฟังก์ชันการประมาณภายในเอลิเมนต์ของแต่ละจุดตอดังนี้

$$\begin{aligned} N_1 &= L_i(2L_i - 1), \quad i=1, 2, 3 \\ N_4 &= 4L_2L_3 \\ N_5 &= 4L_1L_3 \\ N_6 &= 4L_1L_2 \end{aligned} \quad (3.18)$$

ด้วยหลักเกณฑ์และขั้นตอนที่ได้กล่าวข้างต้น ฟังก์ชันการประมาณภายในสำหรับตำแหน่งจุดต่อและฟังก์ชันการประมาณภายในสำหรับตำแหน่งไร้จุดต่อจึงถูกประดิษฐ์ขึ้น [10, 20] จากนั้นมีการปรับฟังก์ชันการประมาณภายในสำหรับตำแหน่งไร้จุดต่อไปใช้ในงานวิจัย [7] และสำหรับฟังก์ชันการประมาณภายในทั้งสองที่ใช้ในวิทยานิพนธ์นี้จะอยู่ในรูปดังนี้

$$\text{ฟังก์ชันการประมาณภายในสำหรับตำแหน่งจุดต่อ} \quad N_1 = L_1 \quad (3.19ก)$$

$$N_2 = L_2 \quad (3.19ข)$$

$$N_3 = L_3 \quad (3.19ค)$$

$$\text{ฟังก์ชันการประมาณภายในสำหรับตำแหน่งไร้จุดต่อ} \quad N_4 = L_2L_3 \quad (3.19ง)$$

$$N_5 = L_1L_3 \quad (3.19จ)$$

$$N_6 = L_1L_2 \quad (3.19ฉ)$$

3.3 รูปแบบของระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์ (Flux-Based Finite Element Formulation)

รูปแบบของระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์ถูกประดิษฐ์ขึ้นเพื่อประยุกต์ใช้กับระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ โดยมีเป้าหมายเพื่อลดความซับซ้อนในการคำนวณเมื่อเปรียบเทียบกับระเบียบวิธีไฟไนต์เอลิเมนต์แบบดั้งเดิม

สำหรับปัญหาสองมิติในระบบแกนพิกัดฉาก สมการปัวร์ซง (Poisson's equation) สามารถเขียนอยู่ในรูปแบบอนุพันธ์ดังนี้

$$\frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = f(x, y) \quad (3.20)$$

โดย $f(x, y)$ แทนฟังก์ชันแหล่งกำเนิด (source function) E และ F แทนส่วนประกอบของฟลักซ์ในแนวแกน x และ y ตามลำดับ ซึ่งสามารถเขียนให้อยู่ในรูปแบบของตัวแปรปฐมภูมิ (primary variable) ได้ดังนี้

$$E = -c \frac{\partial U}{\partial x} \quad \text{และ} \quad F = -c \frac{\partial U}{\partial y} \quad (3.21)$$

โดย U แทนตัวแปรปฐมภูมิ และ c แทนคุณสมบัติของวัสดุซึ่งแตกต่างกันไปตามชนิดของปัญหา ความซับซ้อนของปัญหาที่เกิดขึ้นจากสมการ (3.20) นี้ ขึ้นอยู่กับรูปร่างลักษณะทางกายภาพและเงื่อนไขขอบเขตที่กำหนดให้ เงื่อนไขขอบเขตตลอดขอบเขตอาจประกอบด้วย

(ก) การกำหนดค่าตัวแปรปฐมภูมิตลอดขอบ เช่น

$$U(x, y) = U_1(x, y)$$

(ข) การกำหนดปริมาณฟลักซ์ q ที่ผ่านเข้าสู่ขอบ

$$c \nabla U \cdot \hat{n} + d(U - U_\infty) = q$$

โดย \hat{n} แทนเวกเตอร์หนึ่งหน่วยที่ตั้งฉากกับขอบนั้น ในขณะที่ d และ U_∞ แทนตัวแปรทุติยภูมิ (secondary variable)

ขั้นตอนที่สำคัญของระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์ประกอบด้วย 6 ขั้นตอนคือ

ขั้นตอนที่ 1 สมมุติลักษณะการกระจายของตัวแปรปฐมภูมิตลอดขอบให้อยู่ในรูป

$$U(x, y) = \sum_{i=1}^6 N_i(x, y) U_i = [N(x, y)] \{U\} \quad (3.22)$$

โดย $[N(x, y)]$ ประกอบด้วยฟังก์ชันการประมาณภายในเอลิเมนต์ที่ตำแหน่งจุดต่อและที่ตำแหน่งไร้จุดต่อตามสมการ (3.19ก-ค) และ $\{U\}$ แทนเวกเตอร์ของตัวแปรปฐมภูมิที่ตำแหน่งจุดต่อ U_1, U_2, U_3 และที่ตำแหน่งไร้จุดต่อ U_4, U_5, U_6

ขั้นตอนที่ 2 ประยุกต์ระเบียบวิธีถ่วงน้ำหนักเศษตค่างเข้ากับสมการ (3.20) ดังเช่นในหัวข้อ 3.1 โดยเลือกใช้ฟังก์ชันน้ำหนักตามฟังก์ชันการประมาณภายในสำหรับตำแหน่งจุดต่อและตำแหน่งไร้จุดต่อดังแสดงในสมการ (3.19ก-ค) ซึ่งสมการจะอยู่ในรูปดังนี้

$$\int_{\Omega} N \left(\frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} \right) d\Omega - \int_{\Omega} N f(x, y) d\Omega = 0 \quad (3.23)$$

ขั้นตอนที่ 3 อินทิเกรตทีละส่วนโดยใช้ทฤษฎีบทของเกาส์ (Gauss' theorem) เพื่อก่อให้เกิดพจน์ที่เกี่ยวข้องกับขอบเขตของเอลิเมนต์ ซึ่งทฤษฎีบทของเกาส์ว่าดังนี้

$$\int_{\Omega} u (\nabla \cdot \vec{V}) d\Omega = \int_{\Gamma} u (\vec{V} \cdot \hat{n}) d\Gamma - \int_{\Omega} (\nabla u \cdot \vec{V}) d\Omega \quad (3.24)$$

เมื่อทำการเปรียบเทียบสัญลักษณ์ส่วนประกอบของพจน์ทางด้านซ้ายมือของสมการ (3.23) กับพจน์ทางด้านซ้ายมือของสมการ (3.24) จะพบว่า

$$\left. \begin{aligned} u &= N \\ \nabla &= \frac{\partial}{\partial x} \hat{i} + \frac{\partial}{\partial y} \hat{j} \\ \vec{V} &= \frac{\partial E}{\partial x} \hat{i} + \frac{\partial F}{\partial y} \hat{j} \end{aligned} \right\} (\nabla \cdot \vec{V}) = \left(\frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} \right)$$

และเนื่องจาก $\hat{n} = n_x \hat{i} + n_y \hat{j}$ ดังนั้น

$$\begin{aligned} \vec{V} \cdot \hat{n} &= \frac{\partial E}{\partial x} n_x + \frac{\partial F}{\partial y} n_y \\ u (\vec{V} \cdot \hat{n}) &= N \left(\frac{\partial E}{\partial x} n_x + \frac{\partial F}{\partial y} n_y \right) \end{aligned}$$

$$\begin{aligned} \nabla u &= \frac{\partial N}{\partial x} \hat{i} + \frac{\partial N}{\partial y} \hat{j} \\ \nabla u \cdot \vec{V} &= \frac{\partial N}{\partial x} \frac{\partial E}{\partial x} + \frac{\partial N}{\partial y} \frac{\partial F}{\partial y} \end{aligned}$$

พจน์ทางด้านซ้ายมือของสมการ (3.23) จะกลายเป็น

$$\int_{\Omega} N \frac{\partial E}{\partial x} d\Omega = \int_{\Gamma} N E n_x d\Gamma - \int_{\Omega} \frac{\partial N}{\partial x} E d\Omega \quad (3.25ก)$$

$$\int_{\Omega} N \frac{\partial F}{\partial y} d\Omega = \int_{\Gamma} N F n_y d\Gamma - \int_{\Omega} \frac{\partial N}{\partial y} F d\Omega \quad (3.25ข)$$

จากนั้น แทนสมการ (3.25ก-ข) ลงในสมการ (3.23) จะได้

$$\begin{aligned} \int_{\Omega} \frac{\partial N}{\partial x} E d\Omega + \int_{\Omega} \frac{\partial N}{\partial y} F d\Omega = \\ \int_{\Gamma} N E n_x d\Gamma + \int_{\Gamma} N F n_y d\Gamma - \int_{\Omega} N f(x, y) d\Omega \end{aligned} \quad (3.26)$$

ขั้นตอนที่ 4 สมมุติลักษณะการกระจายของฟลักซ์สำหรับเอลิเมนต์ (element flux distributions) สามารถคำนวณได้จากฟลักซ์ที่ตำแหน่งจุดต่อซึ่งอยู่ในรูป

$$E = \sum_{i=1}^3 \bar{N}_i E_i = [\bar{N}] \{E\} \quad \text{และ} \quad F = \sum_{i=1}^3 \bar{N}_i F_i = [\bar{N}] \{F\} \quad (3.27)$$

โดย $[\bar{N}]$ แทนฟังก์ชันการประมาณภายในแบบเชิงเส้น $[L_1 \ L_2 \ L_3]$ สำหรับเวกเตอร์ $\{E\}$ และ $\{F\}$ แทนเวกเตอร์ของฟลักซ์ที่ตำแหน่งจุดต่อในแนวแกน x และ y ตามลำดับ โดยเวกเตอร์ทั้งสองมีรายละเอียดดังนี้

$$\{E\} = \begin{Bmatrix} -c \left[\frac{\partial N}{\partial x} \right] \{U\} \Big|_1 \\ -c \left[\frac{\partial N}{\partial x} \right] \{U\} \Big|_2 \\ -c \left[\frac{\partial N}{\partial x} \right] \{U\} \Big|_3 \end{Bmatrix} \quad \text{และ} \quad \{F\} = \begin{Bmatrix} -c \left[\frac{\partial N}{\partial y} \right] \{U\} \Big|_1 \\ -c \left[\frac{\partial N}{\partial y} \right] \{U\} \Big|_2 \\ -c \left[\frac{\partial N}{\partial y} \right] \{U\} \Big|_3 \end{Bmatrix} \quad (3.28)$$

ขั้นตอนที่ 5 แทนสมการ (3.27) ลงในสมการ (3.26) จะได้สมการไฟไนต์เอลิเมนต์ที่อยู่ในรูปฟลักซ์ที่ตำแหน่งจุดต่อดังแสดง

$$\begin{aligned} \int_{\Omega} \left\{ \frac{\partial N}{\partial x} \right\} [\bar{N}] d\Omega \{E\} + \int_{\Omega} \left\{ \frac{\partial N}{\partial y} \right\} [\bar{N}] d\Omega \{F\} = \\ \int_{\Gamma} \{N\} [\bar{N}] d\Gamma \{E\} n_x + \int_{\Gamma} \{N\} [\bar{N}] d\Gamma \{F\} n_y - \int_{\Omega} \{N\} f(x, y) d\Omega \end{aligned} \quad (3.29)$$

3.4 สมการไฟไนต์เอลิเมนต์สำหรับปัญหาการถ่ายเทความร้อน

ในส่วนนี้จะอธิบายการประดิษฐ์สมการไฟไนต์เอลิเมนต์สำหรับปัญหาการถ่ายเทความร้อน โดยระเบียบวิธีถ่วงน้ำหนักเศษตักด้วยระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อซึ่งมีขั้นตอนเช่นเดียวกับที่ได้อธิบายไว้ในส่วนที่ 3.1 โดยเริ่มจากสมการเชิงอนุพันธ์สำหรับการถ่ายเทความร้อนภายใต้สภาวะอยู่ตัว

$$\frac{\partial}{\partial x} \left[k \frac{\partial T}{\partial x} \right] + \frac{\partial}{\partial y} \left[k \frac{\partial T}{\partial y} \right] + Q = 0 \quad (2.3)$$

เมื่อทำการเปรียบเทียบสัญลักษณ์ของสมการ (2.3) กับสมการ (3.20) และ (3.21) จะพบว่า

$$\begin{aligned} U &= T \\ c &= k \\ f(x, y) &= Q \\ E &= q_x = -k \frac{\partial T}{\partial x} \\ F &= q_y = -k \frac{\partial T}{\partial y} \end{aligned}$$

สมการ (3.29) สำหรับปัญหาการถ่ายเทความร้อนภายใต้สภาวะอยู่ตัวจะกลายมาเป็น

$$\int_{\Omega} \left\{ \frac{\partial N}{\partial x} \right\} [N] d\Omega \{E\} + \int_{\Omega} \left\{ \frac{\partial N}{\partial y} \right\} [N] d\Omega \{F\} = \int_{\Gamma} \{N\} [N] d\Gamma (\{E\} n_x + \{F\} n_y) - \int_{\Omega} \{N\} Q d\Omega \quad (3.30)$$

และสมการ (3.28) สำหรับปัญหาการถ่ายเทความร้อนภายใต้สภาวะอยู่ตัวก็จะกลายมาเป็น

$$\{E\} = \begin{Bmatrix} -k \left[\frac{\partial N}{\partial x} \right] \{T\} \Big|_1 \\ -k \left[\frac{\partial N}{\partial x} \right] \{T\} \Big|_2 \\ -k \left[\frac{\partial N}{\partial x} \right] \{T\} \Big|_3 \end{Bmatrix} \quad \text{และ} \quad \{F\} = \begin{Bmatrix} -k \left[\frac{\partial N}{\partial y} \right] \{T\} \Big|_1 \\ -k \left[\frac{\partial N}{\partial y} \right] \{T\} \Big|_2 \\ -k \left[\frac{\partial N}{\partial y} \right] \{T\} \Big|_3 \end{Bmatrix} \quad (3.31)$$

สมการ (3.30) และ (3.31) ที่ได้สามารถนำไปใช้ในการประดิษฐ์เอลิเมนต์เมทริกซ์และโปรแกรมคอมพิวเตอร์ได้โดยตรง หรืออาจเขียนให้อยู่ในรูปอย่างง่ายของสมการไฟไนต์เอลิเมนต์ได้ดังนี้

$$[D_x] \{E\} + [D_y] \{F\} = \{B\} - \{R\} \quad (3.32)$$

โดย $[D_x], [D_y]$ แทน เมทริกซ์ส่วนประกอบของฟลักซ์

$\{B\}$ แทน โหลดเวกเตอร์ขอบเขต

$\{R\}$ แทน โหลดเวกเตอร์แหล่งกำเนิด

$$[D_x] = \int_{\Omega} \left\{ \frac{\partial N}{\partial x} \right\} [N] d\Omega \quad (3.33ก)$$

$$[D_y] = \int_{\Omega} \left\{ \frac{\partial N}{\partial y} \right\} [N] d\Omega \quad (3.33ข)$$

$$\{B\} = \int_{\Gamma} \{N\} [N] d\Gamma (\{E\} n_x + \{F\} n_y) \quad (3.33ค)$$

$$\{R\} = \int_{\Omega} \{N\} Q d\Omega \quad (3.33ง)$$

3.5 สมการไฟไนต์เอลิเมนต์สำหรับปัญหาของแข็ง

การประดิษฐ์สมการไฟไนต์เอลิเมนต์สำหรับปัญหาของแข็ง โดยระเบียบวิธีถ่วงน้ำหนักเศษตักด้วยระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อซึ่งมีขั้นตอนดังที่ได้อธิบายข้างต้น โดยเริ่มจากสมการเชิงอนุพันธ์แสดงความสมดุลของแรงในแนวแกน x และ y บนแผ่นระนาบ เมื่อไม่คิดน้ำหนักของตัวเอง

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} = 0 \quad (2.5ก)$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} = 0 \quad (2.5ข)$$

เมื่อทำการเปรียบเทียบสัญลักษณ์ของสมการ (2.5ก-ข) กับสมการ (3.21) จะได้สมการ (3.29) สำหรับปัญหาของแข็งเมื่อไม่คิดน้ำหนักของตัวเอง คือ

$$\int_{\Omega} \left\{ \frac{\partial N}{\partial x} \right\} [N] d\Omega \{E\} + \int_{\Omega} \left\{ \frac{\partial N}{\partial y} \right\} [N] d\Omega \{F\} = \int_{\Gamma} \{N\} [N] d\Gamma (\{E\} n_x + \{F\} n_y) \quad (3.34)$$

โดยพจน์ขวามือของสมการ (3.34) ที่เกี่ยวข้องกับขอบ จะเป็นแรงภายนอก F_x และ F_y ที่กระทำต่อโครงสร้างในทิศทาง x และ y ตามลำดับ และ

$$\{E\} = \begin{Bmatrix} E_1 \\ E_2 \end{Bmatrix} = \begin{Bmatrix} \sigma_x \\ \tau_{xy} \end{Bmatrix} \quad (3.35ก)$$

$$\{F\} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \begin{Bmatrix} \tau_{xy} \\ \sigma_y \end{Bmatrix} \quad (3.35ข)$$

ดังนั้นสมการ (3.34) จะสามารถเขียนอยู่ในรูป

$$\int_{\Omega} \begin{bmatrix} \frac{\partial N}{\partial x} & \frac{\partial N}{\partial x} \\ \frac{\partial N}{\partial y} & \frac{\partial N}{\partial y} \end{bmatrix} \begin{bmatrix} \bar{N} & 0 \\ 0 & \bar{N} \end{bmatrix} d\Omega \begin{Bmatrix} E_1 \\ E_2 \end{Bmatrix} + \int_{\Omega} \begin{bmatrix} \frac{\partial N}{\partial y} & \frac{\partial N}{\partial y} \\ \frac{\partial N}{\partial x} & \frac{\partial N}{\partial x} \end{bmatrix} \begin{bmatrix} \bar{N} & 0 \\ 0 & \bar{N} \end{bmatrix} d\Omega \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} \quad (3.36)$$

ความสัมพันธ์ระหว่างค่าความเค้นย่อยและความเครียดย่อยจากสมการ (2.6) และ (2.8) สามารถแสดงในรูปอย่างง่ายคือ

$$\sigma_x = C_1 \varepsilon_x + C_2 \varepsilon_y - (C_1 + C_2) \alpha_1 (T - T_0) \quad (3.37ก)$$

$$\sigma_y = C_2 \varepsilon_x + C_1 \varepsilon_y - (C_1 + C_2) \alpha_1 (T - T_0) \quad (3.37ข)$$

$$\tau_{xy} = C_3 \gamma_{xy} \quad (3.37ค)$$

สำหรับปัญหาความเค้นในระนาบ $C_1 = \frac{E}{1-\nu^2} \quad ; \quad C_2 = \frac{\nu E}{1-\nu^2}$

$$C_3 = \frac{E}{2(1+\nu)} \quad ; \quad \alpha_1 = \alpha$$

สำหรับปัญหาความเครียดในระนาบ $C_1 = \frac{(1-\nu)E}{(1+\nu)(1-2\nu)} \quad ; \quad C_2 = \frac{\nu E}{(1+\nu)(1-2\nu)}$

$$C_3 = \frac{E}{2(1+\nu)} \quad ; \quad \alpha_1 = \alpha(1+\nu)$$

โดยพจน์สุดท้ายของสมการ (3.37ก-ข) คือค่าความเค้นอันเนื่องมาจากอุณหภูมิ ซึ่งค่าความเครียดย่อยในสมการ (3.37ก-ค) สามารถเขียนให้อยู่ในรูปของระยะการเคลื่อนตัว u และ v ในแนวแกน x และ y ได้คือ

$$\varepsilon_x = \partial u / \partial x \quad ; \quad \varepsilon_y = \partial u / \partial y \quad (2.7)$$

และ $\gamma_{xy} = \partial u / \partial y + \partial v / \partial x$

เมื่อแทนค่าความเครียดย่อยในรูปของระยะการเคลื่อนตัวสมการ (2.7) ลงในสมการ (3.37ก-ค) จะได้

$$\sigma_x = C_1 \frac{\partial u}{\partial x} + C_2 \frac{\partial v}{\partial y} - (C_1 + C_2)\alpha_1(T - T_0) \quad (3.38ก)$$

$$\sigma_y = C_2 \frac{\partial u}{\partial x} + C_1 \frac{\partial v}{\partial y} - (C_1 + C_2)\alpha_1(T - T_0) \quad (3.38ข)$$

$$\tau_{xy} = C_3 \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad (3.38ค)$$

นำความสัมพันธ์ในสมการ (3.38ก-ค) แทนลงในสมการ (3.35) เวกเตอร์ทั้งสองจะกลายเป็น

$$\begin{Bmatrix} E_1 \\ E_2 \end{Bmatrix} = \begin{Bmatrix} C_1 \frac{\partial u}{\partial x} + C_2 \frac{\partial v}{\partial y} - (C_1 + C_2)\alpha_1(T - T_0) \\ C_3 \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \end{Bmatrix} \quad (3.39ก)$$

$$\begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \begin{Bmatrix} C_3 \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ C_2 \frac{\partial u}{\partial x} + C_1 \frac{\partial v}{\partial y} - (C_1 + C_2)\alpha_1(T - T_0) \end{Bmatrix} \quad (3.39ข)$$

รายละเอียดของเวกเตอร์ทั้งสองมีดังนี้

$$\{E_1\} = \begin{Bmatrix} C_1 \left[\frac{\partial N}{\partial x} \right] \{u\} \Big|_1 + C_2 \left[\frac{\partial N}{\partial y} \right] \{v\} \Big|_1 - (C_1 + C_2)\alpha_1 [N] \{T - T_0\} \Big|_1 \\ C_1 \left[\frac{\partial N}{\partial x} \right] \{u\} \Big|_2 + C_2 \left[\frac{\partial N}{\partial y} \right] \{v\} \Big|_2 - (C_1 + C_2)\alpha_1 [N] \{T - T_0\} \Big|_2 \\ C_1 \left[\frac{\partial N}{\partial x} \right] \{u\} \Big|_3 + C_2 \left[\frac{\partial N}{\partial y} \right] \{v\} \Big|_3 - (C_1 + C_2)\alpha_1 [N] \{T - T_0\} \Big|_3 \end{Bmatrix} \quad (3.40ก)$$

$$\{E_2\} = \begin{Bmatrix} C_3 \left[\frac{\partial N}{\partial y} \right] \{u\} \Big|_1 + C_2 \left[\frac{\partial N}{\partial x} \right] \{v\} \Big|_1 \\ C_3 \left[\frac{\partial N}{\partial y} \right] \{u\} \Big|_2 + C_2 \left[\frac{\partial N}{\partial x} \right] \{v\} \Big|_2 \\ C_3 \left[\frac{\partial N}{\partial y} \right] \{u\} \Big|_3 + C_2 \left[\frac{\partial N}{\partial x} \right] \{v\} \Big|_3 \end{Bmatrix} \quad (3.40ข)$$

$$\{F_1\} = \left\{ \begin{array}{l} C_3 \left[\frac{\partial N}{\partial y} \right] \{u\} \Big|_1 + C_2 \left[\frac{\partial N}{\partial x} \right] \{v\} \Big|_1 \\ C_3 \left[\frac{\partial N}{\partial y} \right] \{u\} \Big|_2 + C_2 \left[\frac{\partial N}{\partial x} \right] \{v\} \Big|_2 \\ C_3 \left[\frac{\partial N}{\partial y} \right] \{u\} \Big|_3 + C_2 \left[\frac{\partial N}{\partial x} \right] \{v\} \Big|_3 \end{array} \right\} \quad (3.40ค)$$

$$\{F_2\} = \left\{ \begin{array}{l} C_2 \left[\frac{\partial N}{\partial x} \right] \{u\} \Big|_1 + C_1 \left[\frac{\partial N}{\partial y} \right] \{v\} \Big|_1 - (C_1 + C_2) \alpha_1 [N] \{T - T_0\} \Big|_1 \\ C_2 \left[\frac{\partial N}{\partial x} \right] \{u\} \Big|_2 + C_1 \left[\frac{\partial N}{\partial y} \right] \{v\} \Big|_2 - (C_1 + C_2) \alpha_1 [N] \{T - T_0\} \Big|_2 \\ C_2 \left[\frac{\partial N}{\partial x} \right] \{u\} \Big|_3 + C_1 \left[\frac{\partial N}{\partial y} \right] \{v\} \Big|_3 - (C_1 + C_2) \alpha_1 [N] \{T - T_0\} \Big|_3 \end{array} \right\} \quad (3.40ง)$$

สมการ (3.36) และ (3.40ก-ง) ที่ได้สามารถนำไปใช้ในการประดิษฐ์เอลิเมนต์เมทริกซ์และโปรแกรมคอมพิวเตอร์ได้โดยตรง หรืออาจเขียนให้อยู่ในรูปอย่างง่ายของสมการไฟไนต์เอลิเมนต์ได้ดังนี้

$$[D_x \quad D_x] \begin{Bmatrix} E_1 \\ E_2 \end{Bmatrix} + [D_y \quad D_y] \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} \quad (3.41)$$

โดย $[D_x], [D_y]$ แทน เมทริกซ์ส่วนประกอบของพลาทซ์
 $\begin{Bmatrix} F_x \\ F_y \end{Bmatrix}$ แทน โหลดเวกเตอร์เนื่องจากแรงภายนอก

$$\text{และ} \quad [D_x] = \int_{\Omega} \left\{ \frac{\partial N}{\partial x} \right\} [\bar{N}] d\Omega \quad (3.42ก)$$

$$[D_y] = \int_{\Omega} \left\{ \frac{\partial N}{\partial y} \right\} [\bar{N}] d\Omega \quad (3.42ข)$$

จะเห็นได้ว่าเมทริกซ์ส่วนประกอบของพลาทซ์ในปัญหาของแข็งเหมือนกับเมทริกซ์ส่วนประกอบของพลาทซ์ในปัญหาการถ่ายเทความร้อน ดังนั้นการประยุกต์ใช้ระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของพลาทซ์เข้ากับปัญหาอื่น ๆ ก็จะได้เมทริกซ์ส่วนประกอบของพลาทซ์ลักษณะเดียวกัน ดังแสดงข้างต้น อีกทั้งเมทริกซ์ส่วนประกอบของพลาทซ์นั้นสามารถประดิษฐ์ขึ้นให้อยู่ในรูปแบบปิด (closed-form) เพื่อนำไปใช้ประดิษฐ์โปรแกรมคอมพิวเตอร์ได้โดยตรง

3.6 การประดิษฐ์ไฟไนต์เอลิเมนต์เมทริกซ์

สมการไฟไนต์เอลิเมนต์ และค่าสัมประสิทธิ์ต่างๆ ดังในสมการที่ได้แสดงข้างต้น นั้นสามารถประดิษฐ์ขึ้นได้โดยง่าย วิธีการดังกล่าวจะได้แสดงในหัวข้อนี้

3.6.1. เอลิเมนต์เมทริกซ์สำหรับปัญหาการถ่ายเทความร้อน

$$[D_x] \{E\} + [D_y] \{F\} = \{B\} - \{R\} \quad (3.32)$$

จากสมการ (3.32) นำค่าเวกเตอร์ของฟลักซ์ $\{E\}$ และ $\{F\}$ มาจัดรูปใหม่โดยเริ่มจากการแทนค่า ฟังก์ชันการประมาณภายในเอลิเมนต์สามเหลี่ยมแบบตัวแปรไร้จุดต่อ โดยมีรายละเอียดดังนี้

$$\begin{aligned} \{E\} &= \begin{Bmatrix} -k \left[\frac{\partial N}{\partial x} \right] \{T\} \Big|_1 \\ -k \left[\frac{\partial N}{\partial x} \right] \{T\} \Big|_2 \\ -k \left[\frac{\partial N}{\partial x} \right] \{T\} \Big|_3 \end{Bmatrix} \\ &= -k \begin{Bmatrix} \left[\frac{\partial N_1}{\partial x} \quad \frac{\partial N_2}{\partial x} \quad \frac{\partial N_3}{\partial x} \quad \frac{\partial N_4}{\partial x} \quad \frac{\partial N_5}{\partial x} \quad \frac{\partial N_6}{\partial x} \right] \Big|_1 \\ \left[\frac{\partial N_1}{\partial x} \quad \frac{\partial N_2}{\partial x} \quad \frac{\partial N_3}{\partial x} \quad \frac{\partial N_4}{\partial x} \quad \frac{\partial N_5}{\partial x} \quad \frac{\partial N_6}{\partial x} \right] \Big|_2 \\ \left[\frac{\partial N_1}{\partial x} \quad \frac{\partial N_2}{\partial x} \quad \frac{\partial N_3}{\partial x} \quad \frac{\partial N_4}{\partial x} \quad \frac{\partial N_5}{\partial x} \quad \frac{\partial N_6}{\partial x} \right] \Big|_3 \end{Bmatrix} \{T\} \\ &= -k \begin{Bmatrix} \left[\frac{\partial L_1}{\partial x} \quad \frac{\partial L_2}{\partial x} \quad \frac{\partial L_3}{\partial x} \quad \frac{\partial(L_2L_3)}{\partial x} \quad \frac{\partial(L_1L_3)}{\partial x} \quad \frac{\partial(L_1L_2)}{\partial x} \right] \Big|_1 \\ \left[\frac{\partial L_1}{\partial x} \quad \frac{\partial L_2}{\partial x} \quad \frac{\partial L_3}{\partial x} \quad \frac{\partial(L_2L_3)}{\partial x} \quad \frac{\partial(L_1L_3)}{\partial x} \quad \frac{\partial(L_1L_2)}{\partial x} \right] \Big|_2 \\ \left[\frac{\partial L_1}{\partial x} \quad \frac{\partial L_2}{\partial x} \quad \frac{\partial L_3}{\partial x} \quad \frac{\partial(L_2L_3)}{\partial x} \quad \frac{\partial(L_1L_3)}{\partial x} \quad \frac{\partial(L_1L_2)}{\partial x} \right] \Big|_3 \end{Bmatrix} \{T\} \\ &= \frac{-k}{2A} \begin{bmatrix} b_1 & b_2 & b_3 & 0 & b_3 & b_2 \\ b_1 & b_2 & b_3 & b_3 & 0 & b_1 \\ b_1 & b_2 & b_3 & b_2 & b_1 & 0 \end{bmatrix} \{T\} \quad (3.43ก) \end{aligned}$$

$$\begin{aligned}
 \{F\} &= \left\{ \begin{array}{l} -k \left[\frac{\partial N}{\partial y} \right] \{T\} \Big|_1 \\ -k \left[\frac{\partial N}{\partial y} \right] \{T\} \Big|_2 \\ -k \left[\frac{\partial N}{\partial y} \right] \{T\} \Big|_3 \end{array} \right\} \\
 &= \frac{-k}{2A} \begin{bmatrix} c_1 & c_2 & c_3 & 0 & c_3 & c_2 \\ c_1 & c_2 & c_3 & c_3 & 0 & c_1 \\ c_1 & c_2 & c_3 & c_2 & c_1 & 0 \end{bmatrix} \{T\} \quad (3.43\text{ข})
 \end{aligned}$$

จากนั้นแทนค่าลักษณะการกระจายของฟังก์ชันสำหรับเอลิเมนต์ซึ่งสมมุติให้มีลักษณะเชิงเส้น $[\bar{N}]$ และฟังก์ชันการประมาณภายในเอลิเมนต์สามเหลี่ยมแบบตัวแปรไร้จุดต่อลงในสมการ (3.33ก-ง) จะได้เมทริกซ์และเวกเตอร์ต่าง ๆ ดังนี้

$$[D_x] = \int_{\Omega} \left\{ \frac{\partial N}{\partial x} \right\} [\bar{N}] d\Omega \quad (3.33\text{ก})$$

$$= \int_{\Omega} \left\{ \begin{array}{l} \partial N_1 / \partial x \\ \partial N_2 / \partial x \\ \partial N_3 / \partial x \\ \partial N_4 / \partial x \\ \partial N_5 / \partial x \\ \partial N_6 / \partial x \end{array} \right\} [L_1 \quad L_2 \quad L_3] d\Omega$$

$$= \int_{\Omega} \frac{1}{2A} \left\{ \begin{array}{l} b_1 \\ b_2 \\ b_3 \\ b_2 L_3 + b_3 L_2 \\ b_1 L_3 + b_3 L_1 \\ b_1 L_2 + b_2 L_1 \end{array} \right\} [L_1 \quad L_2 \quad L_3] d\Omega$$

$$= \frac{1}{2A} \int_{\Omega} \begin{bmatrix} b_1 L_1 & b_1 L_2 & b_1 L_3 \\ b_2 L_1 & b_2 L_2 & b_2 L_3 \\ b_3 L_1 & b_3 L_2 & b_3 L_3 \\ b_2 L_1 L_3 + b_3 L_1 L_2 & b_2 L_2 L_3 + b_3 L_2 L_2 & b_2 L_3 L_3 + b_3 L_2 L_3 \\ b_1 L_1 L_3 + b_3 L_1 L_1 & b_1 L_2 L_3 + b_3 L_1 L_2 & b_1 L_3 L_3 + b_3 L_1 L_3 \\ b_1 L_1 L_2 + b_2 L_1 L_1 & b_1 L_2 L_2 + b_2 L_1 L_2 & b_1 L_2 L_3 + b_2 L_1 L_3 \end{bmatrix} d\Omega$$

$$[D_x] = \frac{1}{24A} \begin{bmatrix} 4b_1 & 4b_1 & 4b_1 \\ 4b_2 & 4b_2 & 4b_2 \\ 4b_3 & 4b_3 & 4b_3 \\ b_2 + b_3 & b_2 + 2b_3 & 2b_2 + b_3 \\ b_1 + 2b_3 & b_1 + b_3 & 2b_1 + b_3 \\ b_1 + 2b_2 & 2b_1 + b_2 & b_1 + b_2 \end{bmatrix} \quad (3.44\text{ก})$$

$$[D_y] = \int_{\Omega} \left\{ \frac{\partial N}{\partial y} \right\} [\bar{N}] d\Omega \quad (3.33\text{ข})$$

$$= \int_{\Omega} \begin{Bmatrix} \partial N_1 / \partial y \\ \partial N_2 / \partial y \\ \partial N_3 / \partial y \\ \partial N_4 / \partial y \\ \partial N_5 / \partial y \\ \partial N_6 / \partial y \end{Bmatrix} [L_1 \quad L_2 \quad L_3] d\Omega$$

$$= \int_{\Omega} \frac{1}{2A} \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ c_2 L_3 + c_3 L_2 \\ c_1 L_3 + c_3 L_1 \\ c_1 L_2 + c_2 L_1 \end{Bmatrix} [L_1 \quad L_2 \quad L_3] d\Omega$$

$$= \frac{1}{2A} \int_{\Omega} \begin{bmatrix} c_1 L_1 & c_1 L_2 & c_1 L_3 \\ c_2 L_1 & c_2 L_2 & c_2 L_3 \\ c_3 L_1 & c_3 L_2 & c_3 L_3 \\ c_2 L_1 L_3 + c_3 L_1 L_2 & c_2 L_2 L_3 + c_3 L_2 L_2 & c_2 L_3 L_3 + c_3 L_2 L_3 \\ c_1 L_1 L_3 + c_3 L_1 L_1 & c_1 L_2 L_3 + c_3 L_1 L_2 & c_1 L_3 L_3 + c_3 L_1 L_3 \\ c_1 L_1 L_2 + c_2 L_1 L_1 & c_1 L_2 L_2 + c_2 L_1 L_2 & c_1 L_2 L_3 + c_2 L_1 L_3 \end{bmatrix} d\Omega$$

$$[D_y] = \frac{1}{24A} \begin{bmatrix} 4c_1 & 4c_1 & 4c_1 \\ 4c_2 & 4c_2 & 4c_2 \\ 4c_3 & 4c_3 & 4c_3 \\ c_2 + c_3 & c_2 + 2c_3 & 2c_2 + c_3 \\ c_1 + 2c_3 & c_1 + c_3 & 2c_1 + c_3 \\ c_1 + 2c_2 & 2c_1 + c_2 & c_1 + c_2 \end{bmatrix} \quad (3.44\text{ข})$$

ในส่วนของเวกเตอร์ $\{B\}$ นั้นหมายถึงเวกเตอร์ที่เกี่ยวข้องกับฟลักซ์ความร้อนที่ไหลผ่านขอบของเอลิเมนต์ ซึ่งพจน์ดังกล่าวจะตัดกันหมดไปภายในขอบเขตของปัญหาจะคงเหลือเฉพาะที่ขอบด้านนอกของขอบเขตที่พิจารณาเท่านั้น โดยถ้าขอบดังกล่าวถูกกำหนดให้มีอุณหภูมิคงที่หรือถูกกำหนดให้เป็นฉนวน (ฟลักซ์ความร้อนที่ผ่านขอบเป็นศูนย์) เวกเตอร์ $\{B\}$ ก็ไม่ต้องทำการคำนวณหรือกำหนดให้มีค่าเป็นศูนย์ตามลำดับ แต่หากมีฟลักซ์ความร้อนผ่านเข้า-ออกในลักษณะตั้งฉากกับบริเวณขอบดังกล่าว โดยสมมติให้ฟลักซ์ความร้อนผ่านขอบเอลิเมนต์ที่มีความยาว L อยู่ระหว่างจุดต่อหมายเลข 2 และ 3 โดยมีตำแหน่งไว้จุดต่อ 4 อยู่บริเวณกึ่งกลางของขอบนั้น เวกเตอร์ $\{B\}$ สำหรับฟลักซ์ความร้อนที่มีลักษณะข้างต้นจะสามารถคำนวณได้ดังนี้

$$\{B\} = \int_{\Gamma} \{N\} [\bar{N}] d\Gamma (\{E\}n_x + \{F\}n_y) \quad (3.33ค)$$

$$= \int_{\Gamma} \begin{Bmatrix} N_2 \\ N_3 \\ 0 \end{Bmatrix} [L_2 \quad L_3] d\Gamma \begin{Bmatrix} q_2 \\ q_3 \end{Bmatrix}$$

$$= \int_{\Gamma} \begin{Bmatrix} L_2 \\ L_3 \\ 0 \end{Bmatrix} [L_2 \quad L_3] d\Gamma \begin{Bmatrix} q_2 \\ q_3 \end{Bmatrix}$$

$$= \int_{\Gamma} \begin{bmatrix} L_2 L_2 & L_2 L_3 \\ L_2 L_3 & L_3 L_3 \\ 0 & 0 \end{bmatrix} d\Gamma \begin{Bmatrix} q_2 \\ q_3 \end{Bmatrix}$$

$$= \frac{L}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} q_2 \\ q_3 \end{Bmatrix} \quad (3.44ค)$$

และสุดท้ายคือเวกเตอร์ $\{R\}$ โดยมีความหมายถึงเวกเตอร์ที่แทนค่าการผลิตความร้อนภายในเอลิเมนต์ สมมติให้ค่าการผลิตความร้อนภายในเอลิเมนต์มีค่าคงที่ Q ซึ่งสามารถคำนวณได้ดังนี้

$$\{R\} = \int_{\Omega} \{N\} Q d\Omega \quad (3.33ง)$$

$$= Q \int_{\Omega} \{N\} d\Omega$$

$$= \frac{QA}{3} [1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0]^T \quad (3.44ง)$$

3.6.2. เอลิเมนต์เมทริกซ์สำหรับปัญหาความเค้นในโครงสร้าง

จากสมการ (3.41)

$$\begin{bmatrix} D_x & D_x \\ 0 & D_y \end{bmatrix} \begin{Bmatrix} E_1 \\ E_2 \end{Bmatrix} + \begin{bmatrix} D_y & D_y \\ 0 & D_x \end{bmatrix} \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} \quad (3.41)$$

สามารถจัดรูปใหม่ในรูปของความเค้นย่อยที่จุดต่อดังนี้

$$\begin{bmatrix} D_x & 0 & D_y \\ 0 & D_y & D_x \end{bmatrix} \begin{Bmatrix} \{\sigma_x\}_{1,2,3} \\ \{\sigma_y\}_{1,2,3} \\ \{\tau_{xy}\}_{1,2,3} \end{Bmatrix} = \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} \quad (3.45)$$

นำความสัมพันธ์ระหว่างค่าความเค้นย่อยและความเครียดย่อย และความสัมพันธ์ระหว่างค่าความเครียดย่อยและระยะการเคลื่อนตัว มาแทนในสมการ (3.45) จะได้

$$\begin{bmatrix} D_x & 0 & D_y \\ 0 & D_y & D_x \end{bmatrix} \begin{bmatrix} C_1 \left[\frac{\partial N}{\partial x} \right]_{1,2,3} & C_2 \left[\frac{\partial N}{\partial y} \right]_{1,2,3} \\ C_2 \left[\frac{\partial N}{\partial x} \right]_{1,2,3} & C_1 \left[\frac{\partial N}{\partial y} \right]_{1,2,3} \\ C_3 \left[\frac{\partial N}{\partial y} \right]_{1,2,3} & C_3 \left[\frac{\partial N}{\partial x} \right]_{1,2,3} \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} = \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} \quad (3.46)$$

เมทริกซ์ส่วนประกอบของพลาซึนในปัญหาความเค้นในโครงสร้างอันเนื่องมาจากอุณหภูมิ D_x , D_y สามารถคำนวณได้เช่นเดียวกับปัญหาการถ่ายเทความร้อนในหัวข้อย่อย 3.6.1 ข้างต้น และในส่วนของเวกเตอร์ $[\partial N/\partial x]$ และ $[\partial N/\partial y]$ ก็สามารถคำนวณได้ดังเช่นในสมการ (3.43ก-ข) โดยมีรายละเอียดดังนี้

$$\left[\frac{\partial N}{\partial x} \right]_{1,2,3} = \begin{bmatrix} b_1 & b_2 & b_3 & 0 & b_3 & b_2 \\ b_1 & b_2 & b_3 & b_3 & 0 & b_1 \\ b_1 & b_2 & b_3 & b_2 & b_1 & 0 \end{bmatrix} \quad (3.47ก)$$

$$\left[\frac{\partial N}{\partial y} \right]_{1,2,3} = \begin{bmatrix} c_1 & c_2 & c_3 & 0 & c_3 & c_2 \\ c_1 & c_2 & c_3 & c_3 & 0 & c_1 \\ c_1 & c_2 & c_3 & c_2 & c_1 & 0 \end{bmatrix} \quad (3.47ข)$$

บทที่ 4

โปรแกรมคอมพิวเตอร์

บทนี้จะกล่าวถึงรายละเอียดและขั้นตอนการคำนวณภายในโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นให้สอดคล้องกับสมการไฟไนต์เอลิเมนต์ที่ได้แสดงไว้ในบทที่ 3 โดยโปรแกรมดังกล่าวได้ถูกเขียนขึ้นมาด้วยภาษาฟอร์แทรน 90 (FORTRAN 90) ซึ่งประกอบไปด้วยโปรแกรมที่สามารถนำไปวิเคราะห์ปัญหาการถ่ายเทความร้อน ปัญหาความเค้นในของแข็ง และโปรแกรมวิเคราะห์ปัญหาความเค้นอันเนื่องมาจากอุณหภูมิ โดยรายละเอียดต่าง ๆ ของโปรแกรมมีดังนี้

4.1 โปรแกรมคอมพิวเตอร์สำหรับวิเคราะห์ปัญหาการถ่ายเทความร้อน

โปรแกรมคอมพิวเตอร์สำหรับวิเคราะห์ปัญหาการถ่ายเทความร้อน ประกอบไปด้วยโปรแกรมหลัก (main program) และ 9 โปรแกรมย่อย (subroutines) โดยมีขั้นตอนการทำงานดังนี้

- 4.1.1 เริ่มต้นการทำงานภายในโปรแกรมหลัก [MAIN PROGRAM] โดยจะเรียกโปรแกรมย่อยแรก [READ_INPUT] โดยในโปรแกรมย่อยดังกล่าวก็จะทำการอ่านข้อมูลของปัญหา เช่น จำนวนจุดต่อและจำนวนเอลิเมนต์ของปัญหา ค่าคุณสมบัติต่าง ๆ ของวัสดุ และตำแหน่งของจุดต่อต่าง ๆ
- 4.1.2 เรียกโปรแกรมย่อย [GEN_NL] เพื่อทำการสร้างตำแหน่งไร้จุดต่อ รวมไปถึงการกำหนดเงื่อนไขขอบเขตให้กับตำแหน่งไร้จุดต่อที่สร้างขึ้น และประกอบหมายเลขของตำแหน่งไร้จุดต่อทั้งสามในทิศทางแนวนอนเพิ่มเข้าไปในเอลิเมนต์นั้น ๆ
- 4.1.3 จากนั้นจึงเรียกโปรแกรมย่อย [CST] เพื่อทำการสร้างเอลิเมนต์เมทริกซ์ระบบรวมสำหรับระบบสมการใหญ่ โดยทำการเรียกโปรแกรมย่อย [CnK_ELE] และ [ALL_LOAD] เพื่อสร้างเอลิเมนต์เมทริกซ์และโหลดเวกเตอร์สำหรับแต่ละเอลิเมนต์ และดำเนินการประกอบเมทริกซ์จากทุกเอลิเมนต์ให้เป็นเมทริกซ์ระบบรวมโดยเรียกโปรแกรมย่อย [ASMP] เมื่อสร้างเมทริกซ์ระบบรวมเสร็จสิ้นแล้วก็ทำการประยุกต์เงื่อนไขขอบเขตด้วยโปรแกรมย่อย [APPLBC]

4.1.4 เมื่อได้ระบบสมการรวมและกำหนดเงื่อนไขขอบเขตเสร็จสิ้น โปรแกรมย่อย [SYMPCG] จะทำการแก้ระบบสมการรวมด้วยระเบียบวิธีคอนจูเกตเกรเดียนท์เพื่อหาผลลัพธ์ของอุณหภูมิที่จุดต่อ

4.1.5 ขั้นตอนสุดท้ายจะเรียกโปรแกรมย่อย [WRITE_OUTPUT] เพื่อทำการเขียนผลลัพธ์ที่คำนวณได้ลงไปไฟล์ที่กำหนดไว้

4.2 รายละเอียดของโปรแกรม

รายละเอียดของโปรแกรมแสดงไว้ในภาคผนวก ก

4.3 ลักษณะของไฟล์ข้อมูลที่โปรแกรมต้องการ

ลักษณะของข้อมูลที่โปรแกรมต้องการ สามารถแบ่งออกเป็น 5 ส่วนย่อยได้ดังนี้

ส่วนที่ 1

ประโยคอธิบายกำกับลักษณะของไฟล์

บรรทัดแรก	ตัวเลขระบุจำนวนบรรทัดที่เป็นตัวอักษร
บรรทัดต่อไป	ประโยคต่าง ๆ ที่มีจำนวนบรรทัดเท่าที่ระบุไว้
ตัวอย่างเช่น:	1 FINITE ELEMENT DATA FOR HEAT TRANSFER PROBLEM

ส่วนที่ 2

ขนาดของปัญหา พร้อมค่าที่จะใช้ในการคำนวณ

บรรทัดแรก	ค่าระบุจำนวนจุดต่อ จำนวนเอลิเมนต์ จำนวนขอบโดเมน และจำนวนขอบโดเมนที่มีฟลักซ์ความร้อนผ่านเข้า-ออก								
บรรทัดต่อไป	ตัวเลขจำนวนจุดต่อ จำนวนเอลิเมนต์ จำนวนขอบโดเมน และจำนวนขอบโดเมนที่มีฟลักซ์ความร้อนผ่านเข้า-ออก								
ตัวอย่างเช่น:	<table> <thead> <tr> <th>NPOIN</th> <th>NELEM</th> <th>NBOUN</th> <th>NFLUX</th> </tr> </thead> <tbody> <tr> <td>1521</td> <td>2888</td> <td>6</td> <td>0</td> </tr> </tbody> </table>	NPOIN	NELEM	NBOUN	NFLUX	1521	2888	6	0
NPOIN	NELEM	NBOUN	NFLUX						
1521	2888	6	0						

ส่วนที่ 3

คุณสมบัติต่าง ๆ ของวัสดุ

บรรทัดแรก	ค่าระบุคุณสมบัติต่าง ๆ										
บรรทัดต่อไป	ตัวเลขแสดงค่าสัมประสิทธิ์การนำความร้อน ค่าความหนาแน่น ค่าความร้อนจำเพาะ ความหนา และผลเฉลยแม่นยำตรงตามลำดับ										
ตัวอย่างเช่น:	<table> <thead> <tr> <th>TH-Con</th> <th>Den</th> <th>Spec-heat</th> <th>Thick</th> <th>Exact</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>1.</td> <td>0.</td> <td>1.</td> <td>1</td> </tr> </tbody> </table>	TH-Con	Den	Spec-heat	Thick	Exact	1.	1.	0.	1.	1
TH-Con	Den	Spec-heat	Thick	Exact							
1.	1.	0.	1.	1							

หมายเหตุ: ผลเฉลยแม่นยำตรง Exact คือการมีอยู่ของผลเฉลยแม่นยำตรง หมายถึง

Exact = 1 ปัญหาที่วิเคราะห์ห้อยู่ในนั้นมีผลเฉลยแม่นยำตรง

Exact = 0 ปัญหาที่วิเคราะห์ห้อยู่ในนั้นไม่มีผลเฉลยแม่นยำตรง

ส่วนที่ 4 ลักษณะของจุดต่อ

บรรทัดแรก คำระบุลักษณะของจุดต่อ

บรรทัดต่อ ๆ ไป ตัวเลขแสดงหมายเลขจุดต่อ เงื่อนไขขอบเขตของอุณหภูมิ ตำแหน่งจุดต่อในแกน x และ y ค่าอุณหภูมิ และค่าฟลักซ์ความร้อนที่จุดต่อ

ตัวอย่างเช่น:

NODE	IBCT	X-CO	Y-CO	TEMP	FLUX
1	1	0.0	0.0	0.0	0.0
2	0	1.0	0.0	0.0	0.5
3	0	1.0	0.5	0.0	0.0
⋮	⋮	⋮	⋮	⋮	⋮
51	1	1.0	1.0	0.0	0.0
52	1	0.0	1.0	0.0	0.0

หมายเหตุ: เงื่อนไขขอบเขต IBCT คือเงื่อนไขขอบเขตของอุณหภูมิ หมายถึง

IBC = 1 จุดต่อนั้นถูกกำหนดให้มีค่าตามที่ให้ไว้ และไม่ต้องทำการคำนวณหาค่าที่จุดตอดังกล่าว

IBC = 0 ให้ทำการคำนวณหาค่าที่จุดตอดังกล่าว

ส่วนที่ 5 ลักษณะของเอลิเมนต์

บรรทัดแรก คำระบุลักษณะของเอลิเมนต์

บรรทัดต่อ ๆ ไป หมายเลขเอลิเมนต์ หมายเลขของจุดต่อทั้งสามในทิศทวนเข็มนาฬิกาที่ประกอบขึ้นเป็นเอลิเมนต์ และค่าความร้อนผลิตเองภายในเอลิเมนต์

ตัวอย่างเช่น:

ELEM	I	J	K	Q
1	19	20	81	1.0
2	81	20	151	1.0
3	119	181	155	1.0
⋮	⋮	⋮	⋮	⋮
51	124	126	186	1.0
52	117	63	120	1.0

4.4 โปรแกรมคอมพิวเตอร์สำหรับวิเคราะห์ปัญหาความเค้นอันเนื่องมาจากอุณหภูมิ

โปรแกรมคอมพิวเตอร์สำหรับวิเคราะห์ปัญหาความเค้นอันเนื่องมาจากอุณหภูมิ ประกอบไปด้วยโปรแกรมหลัก (main program) และ 10 โปรแกรมย่อย (subroutines) โดยมี ขั้นตอนการทำงานดังนี้

- 4.4.1 เริ่มต้นการทำงานภายในโปรแกรมหลัก [MAIN PROGRAM] โดยจะ เรียกโปรแกรมย่อยแรก [READ_INPUT] โดยในโปรแกรมย่อยดังกล่าวก็จะทำการอ่านข้อมูลของปัญหา เช่น จำนวนจุดต่อและจำนวนเอลิเมนต์ของปัญหา ค่าคุณสมบัติต่าง ๆ ของวัสดุ และตำแหน่งของจุดต่อต่าง ๆ
- 4.4.2 เรียกโปรแกรมย่อย [GEN_NL] เพื่อทำการสร้างตำแหน่งไร้จุดต่อ รวมไปถึงการกำหนดเงื่อนไขขอบเขตให้กับตำแหน่งไร้จุดต่อที่สร้างขึ้น และประกอบหมายเลขของตำแหน่งไร้จุดต่อทั้งสามในทิศทางแนวนอนเพิ่มเข้าไปในเอลิเมนต์นั้น ๆ เมื่อสร้างตำแหน่งไร้จุดต่อเสร็จสิ้นโปรแกรมย่อย [ARRANGE] จะทำการจัดเรียงตัวแปรทั้งหมดใหม่ โดยเริ่มจากตัวแปรค่าระยะการเคลื่อนตัว u สำหรับทุกจุดต่อแล้วจึงค่อยดำเนินการจัดเรียงตัวแปรค่าระยะการเคลื่อนตัว v สำหรับทุกจุดต่อตามลำดับ
- 4.4.3 จากนั้นจึงเรียกโปรแกรมย่อย [CST] เพื่อทำการสร้างเอลิเมนต์เมทริกซ์ระบบรวมสำหรับระบบสมการใหญ่ โดยทำการเรียกโปรแกรมย่อย [CnK_ELE] และ [ALL_LOAD] เพื่อสร้างเอลิเมนต์เมทริกซ์และโหลดเวกเตอร์สำหรับแต่ละเอลิเมนต์ และดำเนินการประกอบเมทริกซ์จากทุกเอลิเมนต์ให้เป็นเมทริกซ์ระบบรวมโดยเรียกโปรแกรมย่อย [ASMP] เมื่อสร้างเมทริกซ์ระบบรวมเสร็จสิ้นแล้วก็ทำการประยุกต์เงื่อนไขขอบเขตด้วยโปรแกรมย่อย [APPLBC]
- 4.4.4 เมื่อได้ระบบสมการรวมและกำหนดเงื่อนไขขอบเขตเสร็จสิ้น โปรแกรมย่อย [SYMPCG] จะทำการแก้ระบบสมการรวมด้วยระเบียบวิธีคอนจูเกตเกรเดียนต์เพื่อหาผลลัพธ์ของค่าระยะการเคลื่อนตัว u และ v ที่จุดต่อ
- 4.4.5 คำนวณค่าความเค้น σ_x , σ_y และ τ_{xy} ที่ทุกจุดต่อโดยเรียกโปรแกรมย่อย [STRESS]

4.4.6 ขั้นตอนสุดท้ายจะเรียกโปรแกรมย่อย [WRITE_OUTPUT] เพื่อทำการเขียนผลลัพธ์ที่คำนวณได้ลงไปไฟล์ที่กำหนดไว้

4.5 รายละเอียดของโปรแกรม

รายละเอียดของโปรแกรมแสดงไว้ในภาคผนวก ข

4.6 ลักษณะของไฟล์ข้อมูลที่โปรแกรมต้องการ

ลักษณะของข้อมูลที่โปรแกรมต้องการ สามารถแบ่งออกเป็น 5 ส่วนย่อยได้ดังนี้

ส่วนที่ 1

ประโยคอธิบายกำกับลักษณะของไฟล์

บรรทัดแรก ตัวเลขระบุจำนวนบรรทัดที่เป็นตัวอักษร
 บรรทัดต่อไป ประโยคต่าง ๆ ที่มีจำนวนบรรทัดเท่าที่ระบุไว้
 ตัวอย่างเช่น: 1
 FINITE ELEMENT DATA FOR THERMAL STRESS ANALYSIS

ส่วนที่ 2

ขนาดของปัญหา พร้อมค่าที่จะใช้ในการคำนวณ

บรรทัดแรก คำระบุจำนวนจุดต่อ จำนวนเอลิเมนต์ จำนวนขอบโดเมน และชนิดของปัญหา
 บรรทัดต่อไป ตัวเลขจำนวนจุดต่อ จำนวนเอลิเมนต์ จำนวนขอบโดเมน ชนิดของปัญหา และผลเฉลยแม่นยำตรงตามลำดับ
 ตัวอย่างเช่น:

NPOIN	NELEM	NBOUN	STYPE	Exact
1521	2888	6	0	1

หมายเหตุ: ชนิดของปัญหา STYPE คือสมมุติฐานลักษณะของปัญหา ได้แก่

STYPE = 0 ปัญหาความเค้นในระนาบ (plane stress)

STYPE = 1 ปัญหาความเครียดในระนาบ (plane strain)

ผลเฉลยแม่นยำตรง Exact คือการมีอยู่ของผลเฉลยแม่นยำตรง หมายถึง

Exact = 1 ปัญหาที่วิเคราะห์ห้อยู่มีผลเฉลยแม่นยำตรง

Exact = 0 ปัญหาที่วิเคราะห์ห้อยู่ไม่มีผลเฉลยแม่นยำตรง

ส่วนที่ 3

คุณสมบัติต่าง ๆ ของวัสดุ

บรรทัดแรก คำระบุคุณสมบัติต่าง ๆ

บรทัดต่อไป ตัวเลขแสดงค่าโมดูลัสของความยืดหยุ่น อัตราส่วนของปัวส์ซง ระดับ
อุณหภูมิที่ไม่มีความเค้น สัมประสิทธิ์การขยายตัวเนื่องจากอุณหภูมิ และ
ความหนาของของโครงสร้าง

ตัวอย่างเช่น:

ELAS	PR	Tref	Alpha	Thick
1.e+5	0.3	0.	0.	0.01

ส่วนที่ 4 ลักษณะของจุดต่อ

บรทัดแรก คำระบุลักษณะของจุดต่อ

บรทัดต่อไป ตัวเลขแสดงหมายเลขจุดต่อ เงื่อนไขขอบเขตในทิศทาง x และ y
ตำแหน่งจุดต่อในแกน x และ y ค่าระยะการเคลื่อนตัวในแกน x และ y
ค่าแรงดันจากภายนอกในแกน x และ y และค่าอุณหภูมิของจุดต่อนั้น

ตัวอย่างเช่น:

NODE	IBCX	IBCY	X-CO	Y-CO	DelX	DelY	Sxx	Syy	Temp
1	1	0	0.0	0.0	0.0	0.0	0.0	0.0	25.0
2	0	0	1.0	0.0	0.0	0.0	0.0	0.0	25.0
3	1	0	1.0	0.5	0.0	0.0	0.0	0.0	25.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
51	0	1	1.0	1.0	0.0	0.0	0.0	0.0	50.0
52	1	0	0.0	1.0	0.0	0.0	0.0	0.0	50.0

หมายเหตุ: เงื่อนไขขอบเขต IBC ในทิศทาง x และ y หมายถึง

IBC = 1 จุดต่อนั้นถูกตรึงไม่ให้เคลื่อนที่ในทิศทางนั้น

IBC = 0 คำนวณหาการเคลื่อนตัวที่จุดต่อดังกล่าวในทิศทางนั้น

ส่วนที่ 5 ลักษณะของเอลิเมนต์

บรทัดแรก คำระบุลักษณะของเอลิเมนต์

บรทัดต่อไป หมายเลขเอลิเมนต์ และหมายเลขของจุดต่อทั้งสามในทิศทางแนวนอน
นาฬิกาที่ประกอบขึ้นเป็นเอลิเมนต์

ตัวอย่างเช่น:

ELEM	I	J	K
1	19	20	81
2	81	20	151
3	119	181	155
⋮	⋮	⋮	⋮
51	124	126	186
52	117	63	120

บทที่ 5

เทคนิคการปรับขนาดเอลิเมนต์

ในการวิเคราะห์ปัญหาด้วยระเบียบวิธีไฟไนต์เอลิเมนต์จะเริ่มต้นจากการแบ่งพื้นที่ภายในโดเมนของปัญหาออกเป็นเอลิเมนต์ย่อย ๆ แล้วทำการคำนวณหาค่าตัวไม่รู้ค่าที่จุดต่อของเอลิเมนต์นั้น ๆ ดังนั้นถ้าเราใช้เอลิเมนต์ที่มีขนาดเล็กมากเท่าไร จุดต่อต่าง ๆ ที่เป็นตัวแทนของตำแหน่งที่จะทำการคำนวณก็จะอยู่ใกล้กันมากขึ้นเท่านั้น ซึ่งจะเป็นผลทำให้คำตอบที่ได้มีความถูกต้องมากยิ่งขึ้น แต่การใช้เอลิเมนต์ขนาดเล็กจำนวนมากตลอดภายในโดเมนของปัญหาจะทำให้ต้องใช้เวลาในการคำนวณและจำนวนหน่วยความจำ (RAM) จำนวนมาก และอาจทำให้เครื่องคอมพิวเตอร์ที่มีจำนวนหน่วยความจำที่จำกัดนั้นไม่สามารถใช้วิเคราะห์ปัญหาได้ ดังนั้นหากเราสามารถเลือกใช้เอลิเมนต์ขนาดเล็กเฉพาะในบริเวณที่เหมาะสมก็จะช่วยลดจำนวนจุดต่อที่ต้องทำการคำนวณลง แต่ในการวิเคราะห์ปัญหาโดยทั่ว ๆ ไปนั้นเราไม่สามารถที่จะทราบผลเฉลยแม่นยำได้ จึงไม่อาจทราบว่าในบริเวณใดควรใช้เอลิเมนต์ขนาดเล็ก และในบริเวณใดที่สามารถใช้เอลิเมนต์ที่มีขนาดใหญ่ได้ จึงต้องอาศัยโปรแกรมคอมพิวเตอร์สำหรับปรับขนาดของเอลิเมนต์ให้มีขนาดที่เหมาะสม [21] โดยในบทนี้จะได้กล่าวถึงหลักการของเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติ และขั้นตอนในการนำไปประยุกต์ใช้กับไฟไนต์เอลิเมนต์โปรแกรมคอมพิวเตอร์ที่ได้ประดิษฐ์ขึ้นเพื่อวิเคราะห์ปัญหาต่าง ๆ

5.1 หลักการของเทคนิคการปรับขนาดเอลิเมนต์

หลักการของเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติ คือ จะทำการปรับใช้เอลิเมนต์ที่มีขนาดเล็กในบริเวณที่มีการเปลี่ยนแปลงของความชันของคำตอบสูง และปรับใช้เอลิเมนต์ขนาดใหญ่ในบริเวณอื่น โดยหลักการหาขนาดเอลิเมนต์ที่เหมาะสมตามตำแหน่งต่าง ๆ ดังกล่าวจะอาศัยแนวคิดที่กำหนดให้ค่าความคลาดเคลื่อนจากผลเฉลยแม่นยำมีค่าเท่ากันตลอดทั้งโดเมน [3] เช่นถ้าเราสนใจค่าของตัวแปรในโดเมนการไหลใน 1 มิติ ค่าความคลาดเคลื่อนที่ตำแหน่งใด ๆ จะมีค่าดังนี้

$$\text{ค่าความคลาดเคลื่อน} = h^2 \left| \frac{\partial^2 \phi}{\partial x^2} \right| = \text{ค่าคงที่} \quad (5.1)$$

สำหรับการวิเคราะห์ปัญหาใน 2 มิติ จะมีพจน์อนุพันธ์อันดับ 2 หลายพจน์ดังนี้

$$\begin{bmatrix} \frac{\partial^2 \phi}{\partial x^2} & \frac{\partial^2 \phi}{\partial x \partial y} \\ \frac{\partial^2 \phi}{\partial y \partial x} & \frac{\partial^2 \phi}{\partial y^2} \end{bmatrix}$$

ก่อนนำค่าอนุพันธ์อันดับ 2 มาใช้จะต้องปรับค่าดังกล่าว โดยอาศัยหลักการหาค่าความเค้นในแนวแกนหลักในวิชากลศาสตร์ของแข็ง [18]

$$\begin{bmatrix} \sigma_x & \tau_{yx} \\ \tau_{xy} & \sigma_y \end{bmatrix} \Rightarrow \begin{bmatrix} \sigma_X & 0 \\ 0 & \sigma_Y \end{bmatrix} \quad (5.2)$$

โดย σ_x คือความเค้นในแนวแกน x , σ_y คือความเค้นในแนวแกน y , τ_{xy} คือความเค้นเฉือนที่มีทิศทางในแนวแกน y , τ_{yx} คือความเค้นเฉือนที่มีทิศทางในแนวแกน x , σ_X คือความเค้นในแนวแกนหลัก X และ σ_Y คือความเค้นในแนวแกนหลัก Y ซึ่งหากประยุกต์เข้ากับปัญหาใน 2 มิติ จะได้ว่า

$$\begin{bmatrix} \frac{\partial^2 \phi}{\partial x^2} & \frac{\partial^2 \phi}{\partial x \partial y} \\ \frac{\partial^2 \phi}{\partial y \partial x} & \frac{\partial^2 \phi}{\partial y^2} \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{\partial^2 \phi}{\partial X^2} & 0 \\ 0 & \frac{\partial^2 \phi}{\partial Y^2} \end{bmatrix}$$

โดย

$$\begin{aligned} \frac{\partial^2 \phi}{\partial X^2} &= \frac{1}{2} \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) + \sqrt{\left(\frac{1}{2} \left(\frac{\partial^2 \phi}{\partial x^2} - \frac{\partial^2 \phi}{\partial y^2} \right) \right)^2 + \left(\frac{\partial^2 \phi}{\partial x \partial y} \right)^2} \\ \frac{\partial^2 \phi}{\partial Y^2} &= \frac{1}{2} \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) - \sqrt{\left(\frac{1}{2} \left(\frac{\partial^2 \phi}{\partial x^2} - \frac{\partial^2 \phi}{\partial y^2} \right) \right)^2 + \left(\frac{\partial^2 \phi}{\partial x \partial y} \right)^2} \end{aligned}$$

และมุมระหว่างแนวแกนหลัก $X-Y$ กับแนวแกน $x-y$ หาได้จาก

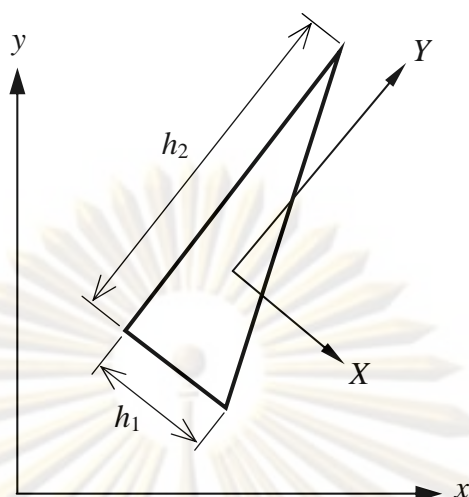
$$\alpha = \frac{1}{2} \tan^{-1} \left(\frac{\frac{\partial^2 \phi}{\partial x \partial y}}{\frac{1}{2} \left(\frac{\partial^2 \phi}{\partial x^2} - \frac{\partial^2 \phi}{\partial y^2} \right)} \right)$$

ดังนั้นค่าความคลาดเคลื่อนของปัญหาใน 2 มิติ คือ

$$h_1^2 |\lambda_1| = h_2^2 |\lambda_2| = \text{ค่าคงที่} \quad (5.3)$$

โดย h_1 คือความยาวของเอลิเมนต์ในแนวแกนหลัก X , h_2 คือความยาวเอลิเมนต์ในแนวแกนหลัก Y ดังแสดงในรูปที่ 5.1 และ

$$\lambda_1 = \frac{\partial^2 \phi}{\partial X^2}, \quad \lambda_2 = \frac{\partial^2 \phi}{\partial Y^2}$$



รูปที่ 5.1 แสดงการวางตัวของเอลิเมนต์ในแนวแกนหลักทั้งสอง (X, Y)

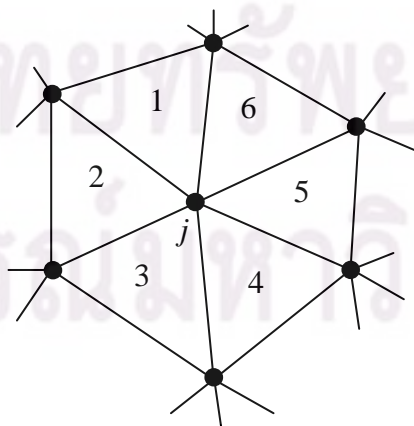
การหาค่าอนุพันธ์อันดับที่ 1 ของจุดต่อใด ๆ เช่น $\partial\phi/\partial x$ เป็นต้น จะหาได้จากการรวมค่าอนุพันธ์อันดับที่ 1 ของเอลิเมนต์ที่ล้อมรอบจุดต่อนั้น ดังแสดงในรูปที่ 5.2 การหาค่าอนุพันธ์อันดับที่ 1 ของจุดต่อ j จะรวมค่าอนุพันธ์อันดับที่ 1 ของเอลิเมนต์สามเหลี่ยม 6 เอลิเมนต์ [3] ดังนี้

$$\int_{\Omega} N_i N_j d\Omega \left(\frac{\partial\phi}{\partial X} \right)_j = \int_{\Omega} N_i d\Omega \left(\frac{\partial\phi}{\partial X} \right)_e \quad (5.4)$$

โดย N_i และ N_j คือฟังก์ชันการประมาณภายในของเอลิเมนต์

$\left(\frac{\partial\phi}{\partial X} \right)_j$ คือ ค่าอนุพันธ์อันดับที่ 1 ของค่าตัวแปรที่พิจารณาที่จุดต่อ j

$$\left(\frac{\partial\phi}{\partial X} \right)_e = \sum_{j=1}^{np} \frac{\partial N_j}{\partial X} \phi_j$$



รูปที่ 5.2 ตัวอย่างการวางตัวของเอลิเมนต์สามเหลี่ยมล้อมจุดต่อ j ใด ๆ

หากเราประยุกต์วิธีโดยชัดแจ้ง (explicit algorithm) เข้ากับสมการ (5.4) ทำให้ $\int_{\Omega} N_i N_j d\Omega$ อยู่ในรูปแบบรวมตัวที่จุดต่อ (lumped mass matrix) ดังนั้น

$$M_{ij} \left(\frac{\partial \phi}{\partial X} \right)_j = \int_{\Omega} N_i d\Omega \left(\frac{\partial \phi}{\partial X} \right)_e \quad (5.5)$$

$$\begin{aligned} \text{โดย } M_{ij} &= \int_{\Omega} N_i N_j d\Omega && \text{เมื่อ } i=j \\ &= 0 && \text{เมื่อ } i \neq j \end{aligned}$$

เราสามารถประยุกต์ในทำนองเดียวกันกับค่าอนุพันธ์อันดับที่สอง ได้ดังนี้

$$M_{ij} \left(\frac{\partial^2 \phi}{\partial X^2} \right)_j = \int_{\Omega} N_i d\Omega \left(\frac{\partial^2 \phi}{\partial X^2} \right)_e \quad (5.6)$$

โดย $\left(\frac{\partial^2 \phi}{\partial X^2} \right)_j$ คือ ค่าอนุพันธ์อันดับที่สอง ของค่าตัวแปรที่พิจารณาที่จุดต่อ j

$$\left(\frac{\partial^2 \phi}{\partial X^2} \right)_e = \sum_{j=1}^{mp} \frac{\partial N_j}{\partial X} \left(\frac{\partial \phi}{\partial x} \right)_j$$

ดังนั้นเมื่อใช้สมการ (5.5) และ (5.6) เราจะสามารถหาค่า λ_1 ในสมการ (5.3) ได้ และในทำนองเดียวกันก็จะสามารถหาค่า λ_2 ได้เช่นกัน

$$M_{ij} \left(\frac{\partial \phi}{\partial Y} \right)_j = \int_{\Omega} N_i d\Omega \left(\frac{\partial \phi}{\partial Y} \right)_e \quad (5.7)$$

$$M_{ij} \left(\frac{\partial^2 \phi}{\partial Y^2} \right)_j = \int_{\Omega} N_i d\Omega \left(\frac{\partial^2 \phi}{\partial Y^2} \right)_e \quad (5.8)$$

ค่าอนุพันธ์อันดับที่สองของค่าความหนาแน่นที่หาได้ในสมการ (5.7) และ (5.8) จากการจัดแบ่งเอลิเมนต์ครั้งล่าสุด จะถูกนำไปใช้ต่อไปเพื่อทำการแบ่งเอลิเมนต์ครั้งใหม่โดยอัตโนมัติ เราจะกำหนดขนาดเอลิเมนต์ที่น้อยที่สุด (h_{\min}) สำหรับการสร้างเอลิเมนต์ใหม่ ซึ่งเป็นตำแหน่งที่เกิดค่าอนุพันธ์อันดับสองสูงสุดด้วยตามข้อกำหนดที่จะให้ค่าความคลาดเคลื่อนเท่ากันตลอดทั้งโดเมนการไหล ดังแสดงในสมการ (5.3) ดังนั้น

$$h_{\min}^2 |\lambda_{\max}| = h_1^2 |\lambda_1| = h_2^2 |\lambda_2| = \text{ค่าคงที่} \quad (5.9)$$

$$\text{โดย } |\lambda_{\max}| = \max \left[\left| \frac{\partial^2 \phi}{\partial X^2} \right|, \left| \frac{\partial^2 \phi}{\partial Y^2} \right| \right]$$

และเราสามารถกำหนดขนาดเอลิเมนต์ที่ยาวที่สุด (h_{\max}) ได้เช่นกัน หากตำแหน่งใดมีการคำนวณตามสมการ (5.9) แล้วได้ค่าความยาวเอลิเมนต์ที่ยาวกว่า h_{\max} ก็จะปรับให้ค่าความยาวเอลิเมนต์มีค่าเท่ากับ h_{\max} แทน

บางครั้งในการแบ่งเอลิเมนต์เราจะพบว่าเอลิเมนต์บางตำแหน่งจะได้ค่าอัตราส่วนระหว่างความยาวเอลิเมนต์ทั้งสองด้าน (stretching parameter, s) มีค่ามากจนเกินไป จึงควรกำหนดค่าอัตราส่วนสูงสุดไว้ที่เหมาะสม (S_{\max}) หากตำแหน่งใดมีค่า s_i มากกว่า (S_{\max}) ก็จะถูกปรับค่าให้เท่ากับ S_{\max} แทน

$$s_i = \frac{h_2}{h_1} = \sqrt{\left| \frac{\lambda_1}{\lambda_2} \right|} \quad (5.10)$$

5.2 โปรแกรมคอมพิวเตอร์สำหรับการปรับขนาดเอลิเมนต์โดยอัตโนมัติ

ในวิทยานิพนธ์นี้ได้ใช้โปรแกรม FEMESH ซึ่งได้พัฒนาขึ้นโดย อ.ดร.สุทธิศักดิ์ พงศ์ธนาพานิช [21] ในการปรับขนาดเอลิเมนต์โดยอัตโนมัติ โปรแกรม FEMESH เป็นโปรแกรมที่ทำงานในโหมดกราฟิกสามารถที่จะสร้างรูปแบบของปัญหาและสร้างเอลิเมนต์สามเหลี่ยมได้ทันที และยังสามารถส่งต่อข้อมูลให้กับโปรแกรมไฟไนต์เอลิเมนต์อื่น ๆ เพื่อทำการวิเคราะห์ปัญหาได้ในหลายรูปแบบ รวมถึงการแสดงผลลัพธ์ที่เกิดขึ้นจากการคำนวณในรูปแบบกราฟิกได้ โดยประสิทธิภาพและความถูกต้องของโปรแกรม FEMESH ได้รับการตรวจสอบด้วยปัญหาในเอกสารอ้างอิง [22]

ภาพโดยรวมของโครงสร้างการทำงานของโปรแกรม FEMESH สามารถที่จะแบ่งออกได้เป็นสามส่วนหลัก ๆ ดังนี้

1. ส่วนทำงานก่อนการประมวลผล (Pre-processing) เป็นส่วนของโปรแกรมที่ช่วยในการสร้างรูปร่างของปัญหาต่างๆ เช่น การวาดเส้นตรงหรือเส้นโค้ง การสร้างเอลิเมนต์สามเหลี่ยม หรือการกำหนดเงื่อนไขที่ขอบ (boundary constraint) เป็นต้น งานหลักของส่วนทำงานก่อนการประมวลผลก็คือ การทำงานด้านคอมพิวเตอร์ช่วยการออกแบบ (CAD) ซึ่งประกอบด้วยคำสั่งต่าง ๆ มากมายที่ช่วยในการสร้างรูปทรงเรขาคณิต

2. ส่วนการประมวลผล (Processing) ซึ่งหมายถึง ส่วนของโปรแกรมที่ทำการคำนวณด้วยระเบียบวิธีไฟไนต์เอลิเมนต์สำหรับปัญหาต่าง ๆ

3. ส่วนทำงานหลังการประมวลผล (Post-processing) เป็นส่วนของโปรแกรมที่ช่วยในการแสดงผลที่ได้จากการคำนวณในรูปแบบกราฟิก เช่น การแสดงเส้นชั้น (contour) หรือการแสดงเวกเตอร์ของความเร็ว (velocity vector) เป็นต้น นอกจากนี้การประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติ (adaptive meshing technique) เข้ากับปัญหาเพื่อต้องการให้ได้ผลลัพธ์ที่มีความแม่นยำมากยิ่งขึ้น ก็จะเป็นการทำงานในส่วนนี้ด้วยเช่นกัน โดยจะนำผลลัพธ์ที่ได้มาทำการคำนวณขนาดของเอลิเมนต์ที่เหมาะสมและทำการสร้างเอลิเมนต์สามเหลี่ยมขึ้นมาใหม่ต่อไป



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6

การตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์

โปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นเพื่อวิเคราะห์ปัญหาการถ่ายเทความร้อน ปัญหาความแข็งแรงในโครงสร้าง และปัญหาความเค้นอันเนื่องมาจากอุณหภูมิ ด้วยระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อด้วยการใช้หลักการของฟลักซ์ จะถูกนำมาตรวจสอบความถูกต้องกับปัญหาที่มีผลเฉลยแม่นยำ และผลการวิเคราะห์ด้วยวิธีอื่น ๆ ดังรายละเอียดต่อไปนี้

6.1 ตรวจสอบโปรแกรมสำหรับวิเคราะห์ปัญหาการถ่ายเทความร้อน

ในหัวข้อนี้จะตรวจสอบความถูกต้องของโปรแกรมวิเคราะห์ปัญหาการถ่ายเทความร้อน โดยจะทำการทดสอบด้วยปัญหาที่มีผลเฉลยแม่นยำคือ

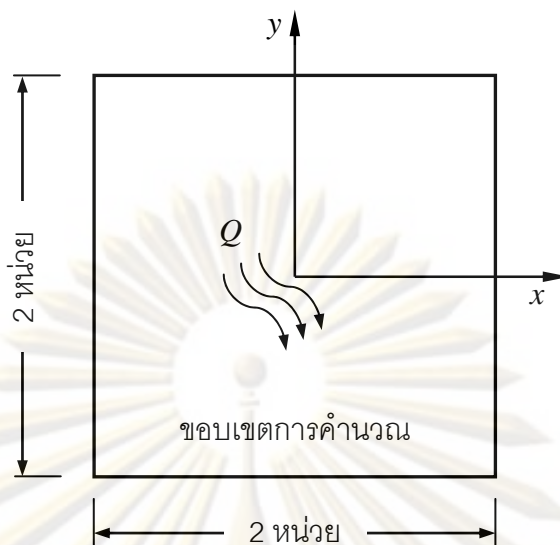
6.1.1 ปัญหาแผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้เอง

ตัวอย่างแรกเป็นปัญหาที่มีการผลิตความร้อนได้เองเพียงอย่างเดียว โดยมีแผ่นโลหะสี่เหลี่ยมจัตุรัสขนาด 2×2 หน่วย หนา 0.1 หน่วย และกำหนดให้สัมประสิทธิ์การนำความร้อน k มีค่าเท่ากับ 1 โดยมีการผลิตความร้อนได้เอง $Q = 1$ และอุณหภูมิตลอดขอบทั้งสี่ด้าน $T = 0$ ดังแสดงในรูปที่ 6.1 ซึ่งปัญหาดังกล่าวมีผลเฉลยแม่นยำของการกระจายอุณหภูมิบนแผ่นโลหะ [20] คือ

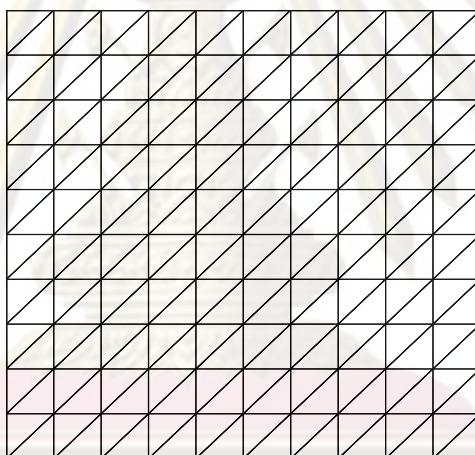
$$T(x, y) = \frac{1-x^2}{2} - \frac{16}{\pi^3} \sum_{n=0}^{\infty} \left(\frac{(-1)^n \cos((2n+1)\pi x/2) \cosh((2n+1)\pi y/2)}{(2n+1)^3 \cosh((2n+1)\pi/2)} \right) \quad (6.1)$$

ในการวิเคราะห์ปัญหาด้วยโปรแกรมที่ประดิษฐ์ขึ้น จะเริ่มต้นจากการสร้างรูปแบบจำลองไฟไนต์เอลิเมนต์ซึ่งได้แสดงไว้ในรูปที่ 6.2 โดยประกอบไปด้วย 121 จุดต่อ และ 200 เอลิเมนต์ จากนั้นทำการประยุกต์เงื่อนไขขอบเขตตามที่กล่าวข้างต้น

รูปที่ 6.3 เป็นการเปรียบเทียบอุณหภูมิในแนวแกน x ที่ตำแหน่ง $y = 0$ ที่คำนวณได้จากผลเฉลยแม่นยำ และจากโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นด้วยระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อด้วยการใช้หลักการของฟลักซ์ (NLFB) และประดิษฐ์ขึ้นจากเอลิเมนต์สามเหลี่ยมสามจุดต่อ (standard) สำหรับปัญหาแผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้เอง

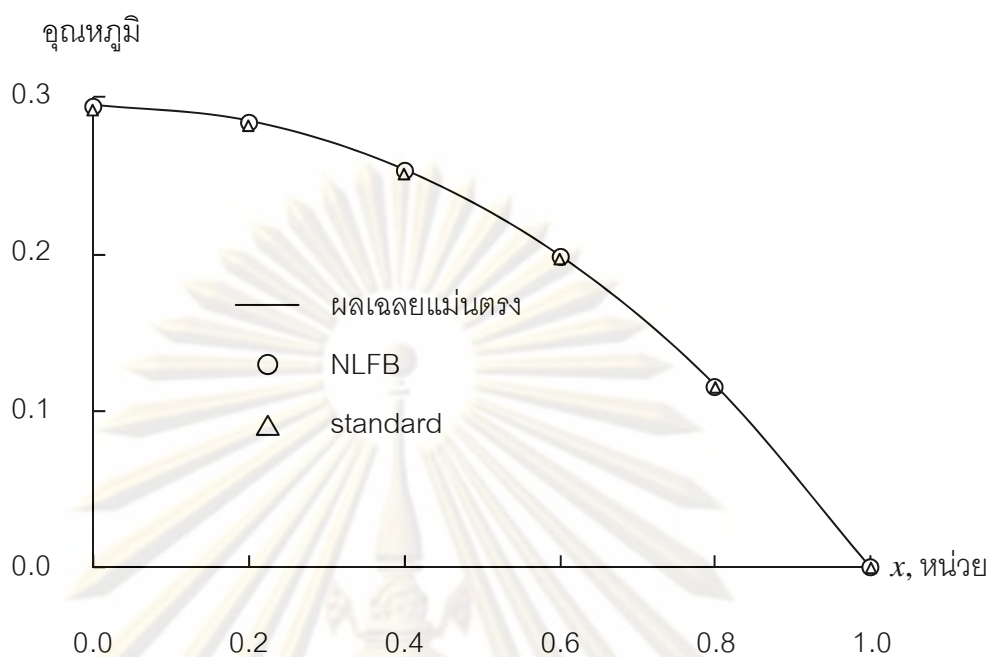


รูปที่ 6.1 แผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้เอง



รูปที่ 6.2 รูปแบบไฟไนต์เอลิเมนต์สำหรับปัญหาแผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้เอง

ตารางที่ 6.1 แสดงการเปรียบเทียบค่าเปอร์เซ็นต์ความผิดพลาดจากโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นด้วยระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อด้วยการใช้หลักการของฟลักซ์ (NLFB) และโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นจากเอลิเมนต์สามเหลี่ยมสามจุดต่อ (standard) ในแนวแกน x ที่ตำแหน่ง $y = 0$ สำหรับปัญหาแผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้เอง จากตารางดังกล่าวพบว่าค่าอุณหภูมิที่คำนวณจากโปรแกรม NLFB มีค่าเปอร์เซ็นต์ความผิดพลาดน้อยกว่าค่าอุณหภูมิที่คำนวณจากโปรแกรม standard ประมาณ 4-5 เท่า



รูปที่ 6.3 อุณหภูมิในแนวแกน x ที่ตำแหน่ง $y = 0$ สำหรับปัญหาแผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้เอง

ตารางที่ 6.1 ตารางแสดงการเปรียบเทียบค่าเปอร์เซ็นต์ความผิดพลาดจากโปรแกรม NLFB และโปรแกรม standard ในแนวแกน x ที่ตำแหน่ง $y = 0$ สำหรับปัญหาแผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้เอง

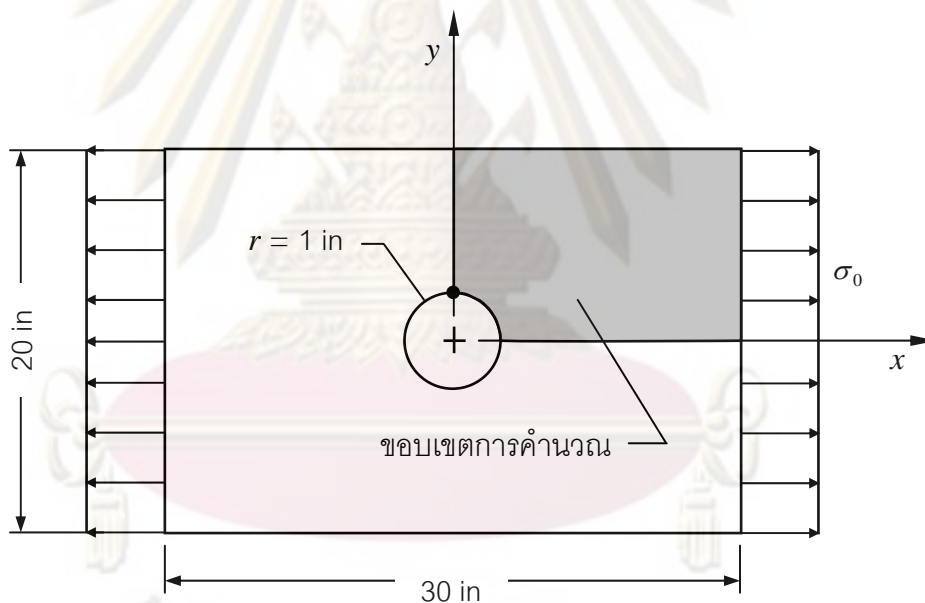
ตำแหน่ง x	เปอร์เซ็นต์ความผิดพลาด (%) จากโปรแกรม NLFB	เปอร์เซ็นต์ความผิดพลาด (%) จากโปรแกรม standard
0.0	0.131	0.778
0.2	0.145	0.779
0.4	0.165	0.781
0.6	0.147	0.772
0.8	0.144	0.736
1.0	0.000	0.000

6.2 ตรวจสอบโปรแกรมสำหรับวิเคราะห์ปัญหาของแข็ง

ในหัวข้อนี้จะตรวจสอบความถูกต้องของโปรแกรมวิเคราะห์ปัญหาของแข็ง โดยจะทำการทดสอบด้วยปัญหาที่มีผลเฉลยแน่นอนตรงคือ

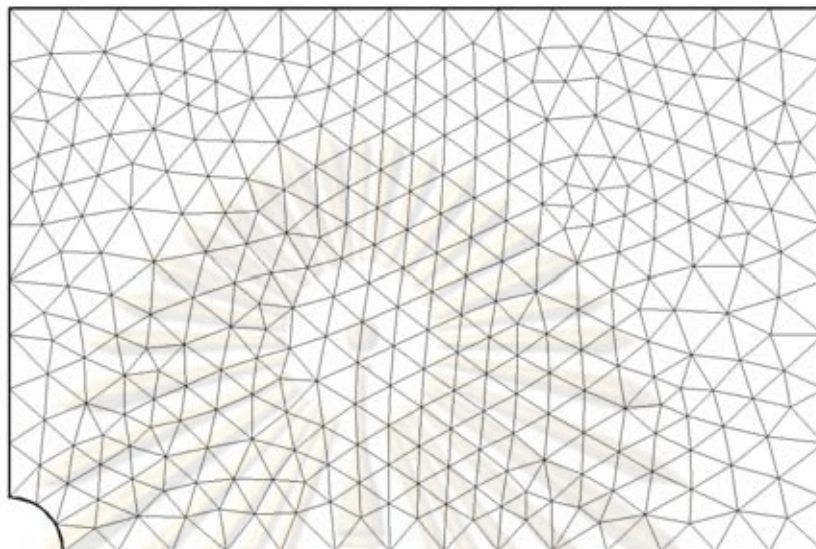
6.2.1 ปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง

ตัวอย่างนี้เป็นแผ่นโลหะสี่เหลี่ยมผืนผ้าขนาด 30×20 in หนา 1 in มีรูกลมตรงกลางขนาดรัศมี $r = 1$ in ถูกดึงตลอดขอบทั้งสองด้านด้วย $\sigma_0 = 1$ psi ดังรูปที่ 6.4 โดยมีคุณสมบัติทางกลของแผ่นโลหะดังนี้ ค่าโมดูลัสของความยืดหยุ่น $E = 30 \times 10^6$ psi และค่าอัตราส่วนปัวส์ซง $\nu = 0.25$ ซึ่งปัญหานี้สามารถนำมาใช้ตรวจสอบความถูกต้องของผลลัพธ์จากการคำนวณได้ เนื่องจาก ณ ตำแหน่งเหนือรูกลม (จุด A) จะมีค่าความเค้นในแนวแกน x สูงสุด [23] เท่ากับ $\sigma_x = 3\sigma_0$



รูปที่ 6.4 แผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง

เนื่องจากลักษณะของปัญหามีความสมมาตร ดังนั้นสามารถใช้ขอบเขตในการคำนวณเพียง 1 ใน 4 ของแผ่นโลหะสี่เหลี่ยมได้ ดังแสดงในรูปที่ 6.4 การวิเคราะห์ปัญหาจะเริ่มจากการแบ่งขอบเขตของปัญหาออกเป็นเอลิเมนต์ย่อย ๆ ซึ่งประกอบไปด้วย 351 จุดต่อ 649 เอลิเมนต์ ดังแสดงในรูปที่ 6.5 จากนั้นประยุกต์เงื่อนไขขอบเขตโดยกำหนดให้ตลอดขอบทางด้านซ้ายไม่มีการเคลื่อนตัวในแกน x และกำหนดให้ตลอดขอบด้านล่างไม่มีการเคลื่อนตัวในแนวแกน y แล้วคำนวณหาค่าการเคลื่อนตัวที่เกิดขึ้น



รูปที่ 6.5 รูปแบบไฟไนต์เอลิเมนต์สำหรับปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง

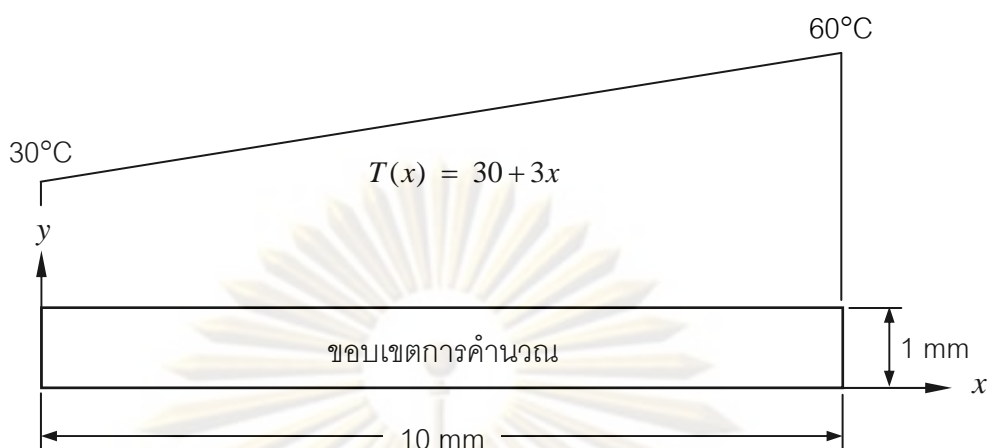
ผลเฉลยแม่นยำตรงของค่าความเค้นในแนวแกน x ที่จุด A สำหรับปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลางเท่ากับ 3 psi ในขณะที่ผลการคำนวณจากโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นด้วยระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อด้วยการใช้หลักการของฟลักซ์ (NLFB) มีค่า 1.977 psi และจากโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นจากเอลิเมนต์สามเหลี่ยมสามจุดต่อ (standard) มีค่า 1.801 psi

6.3 ตรวจสอบโปรแกรมสำหรับวิเคราะห์ปัญหาความเค้นอันเนื่องมาจากอุณหภูมิ

การตรวจสอบความถูกต้องของโปรแกรมวิเคราะห์ปัญหาความเค้นอันเนื่องมาจากอุณหภูมิ จะทำการทดสอบด้วยปัญหาที่มีผลเฉลยแม่นยำตรง 2 ปัญหาดังนี้

6.3.1 ปัญหาแผ่นสี่เหลี่ยมที่มีอุณหภูมิเปลี่ยนแปลงเชิงเส้นตามความยาว [24]

ลักษณะของปัญหาเป็นแผ่นสี่เหลี่ยมขนาด 10×1 mm มีอุณหภูมิที่ขึ้นกับระยะทาง x ด้วยความสัมพันธ์เชิงเส้น $T(x) = 30 + 3x$ โดยตลอดขอบทางซ้ายจะมีอุณหภูมิเท่ากับ 30°C ส่วนตลอดขอบทางด้านขวามีอุณหภูมิเท่ากับ 60°C ดังแสดงในรูปที่ 6.6 ส่วนขอบทางด้านซ้ายจะไม่มี การเคลื่อนที่ทางแกน x ส่วนคุณสมบัติทางกลของแผ่นสี่เหลี่ยมนี้คือ ค่าโมดูลัสของความยืดหยุ่น $E = 71,705.5 \text{ N/mm}^2$ ค่าอัตราส่วนปัวส์ซง $\nu = 0.29$ และสัมประสิทธิ์การขยายตัว $\alpha = 1.27 \times 10^{-5} \text{ mm/mm}^\circ\text{C}$



รูปที่ 6.6 แผ่นสี่เหลี่ยมที่มีอุณหภูมิเปลี่ยนแปลงเชิงเส้นตามความยาว

ปัญหานี้มีผลเฉลยแม่นยำดังนี้

$$u(x) = \alpha \left(30x + \frac{3x^2}{2} \right) \quad (6.2)$$

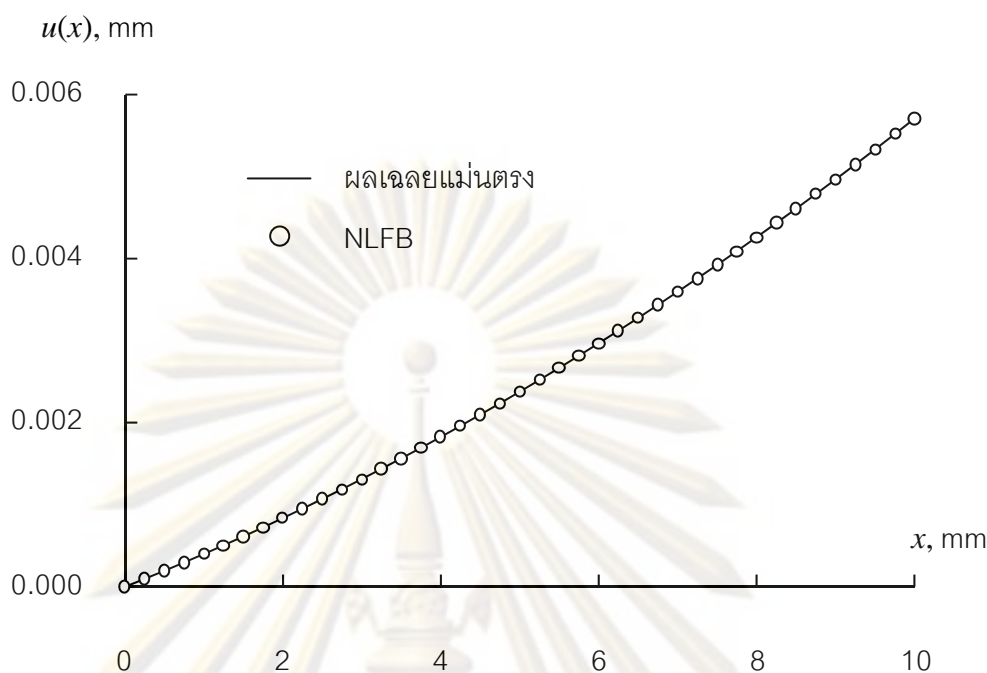
โดย $u(x)$ คือค่าการเคลื่อนตัวในแนวแกน x

การคำนวณค่าการเคลื่อนตัวบนแผ่นสี่เหลี่ยมดังที่แสดงข้างต้นโดยใช้โปรแกรมที่ได้ประดิษฐ์ขึ้นนี้ จะเริ่มต้นจากการแบ่งขอบเขตของปัญหาออกเป็นเอลิเมนต์ย่อย ๆ จำนวน 205 จุดต่อ 320 เอลิเมนต์ ดังแสดงในรูปที่ 6.7 จากนั้นทำการประยุกต์เงื่อนไขขอบเขตตามที่กล่าวข้างต้น และกำหนดอุณหภูมิขณะที่ยึดไม่มีความเค้น T_0 เท่ากับ 0°C แล้วทำการคำนวณหาค่าการเคลื่อนตัวที่เกิดขึ้น



รูปที่ 6.7 รูปแบบไฟไนต์เอลิเมนต์สำหรับปัญหาแผ่นสี่เหลี่ยมที่มีอุณหภูมิเปลี่ยนแปลงเชิงเส้นตามความยาว

รูปที่ 6.8 เป็นการเปรียบเทียบระยะการเคลื่อนตัวในแนวแกน x ที่คำนวณได้จากโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ด้วยระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อด้วยการใช้หลักการของฟลักซ์ (NLFB) กับผลเฉลยแม่นยำ พบว่ามีค่าความผิดพลาดเฉลี่ยเท่ากับ 0.123%



รูปที่ 6.8 ค่าการเคลื่อนตัวเนื่องจากความร้อนในแนวแกน x สำหรับปัญหาแผ่นสี่เหลี่ยมที่มีอุณหภูมิเปลี่ยนแปลงเชิงเส้นตามความยาว

6.3.2 ปัญหาวงแหวนที่มีอุณหภูมิเปลี่ยนแปลงตามแนวรัศมี

ลักษณะของปัญหาเป็นวงแหวนที่มีรัศมีภายใน a เท่ากับ 100 mm รัศมีภายนอก b เท่ากับ 200 mm และวงแหวนมีอุณหภูมิเป็นความสัมพันธ์เชิงเส้นตรงกับรัศมี r ดังนี้

$$T(r) = 110 - 0.4r \quad (6.3)$$

โดยอุณหภูมิตลอดขอบด้านนอกเท่ากับ 30°C และขอบด้านในเท่ากับ 70°C ดังแสดงในรูปที่ 6.9 โดยที่ค่าสัมประสิทธิ์ทางกลของวงแหวนคือ ค่าโมดูลัสของความยืดหยุ่น $E = 71,705.5 \text{ N/mm}^2$ ค่าอัตราส่วนปัวส์ซง $\nu = 0.29$ สัมประสิทธิ์การขยายตัว $\alpha = 1.27 \times 10^{-5} \text{ mm/mm}^{\circ}\text{C}$ และมีอุณหภูมิที่วัสดุไม่มีความเค้น T_0 เท่ากับ 0°C

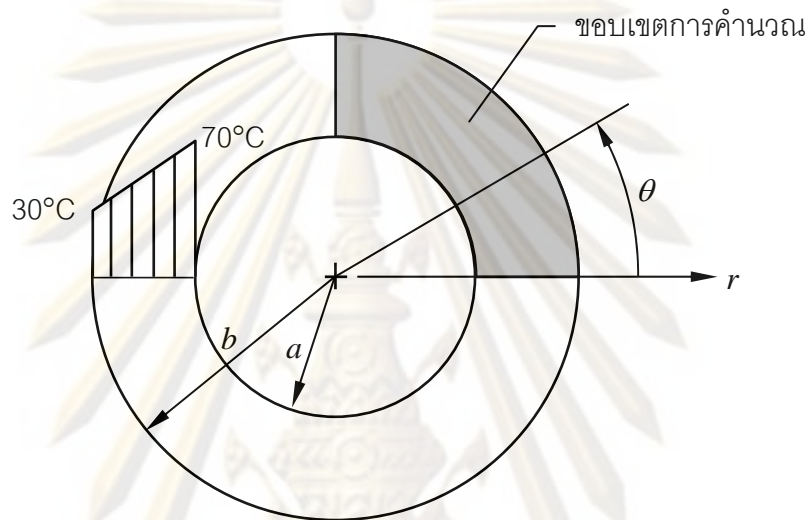
ปัญหานี้สามารถหาผลเฉลยแม่นยำตรงสำหรับกรณีความเค้นในระนาบดังนี้ [24]

$$u(r) = \frac{\alpha}{r} \left(110r^2 - \frac{0.4}{3} \left((1+\nu)(r^3 - a^3) + \frac{b^3 - a^3}{b^2 - a^2} ((1-\nu)r^2 + (1+\nu)a^2) \right) \right) \quad (6.4)$$

$$\sigma_r(r) = \frac{E\alpha}{r^2} \left(\frac{0.4}{3} \left(r^3 - a^3 - \frac{b^3 - a^3}{b^2 - a^2} (r^2 + a^2) \right) + 110a^2 \right) \quad (6.5)$$

$$\sigma_\theta(r) = \frac{0.4}{3} \frac{E\alpha}{r^2} \left(2r^3 + a^3 - \frac{b^3 - a^3}{b^2 - a^2} (r^2 + a^2) \right) \quad (6.6)$$

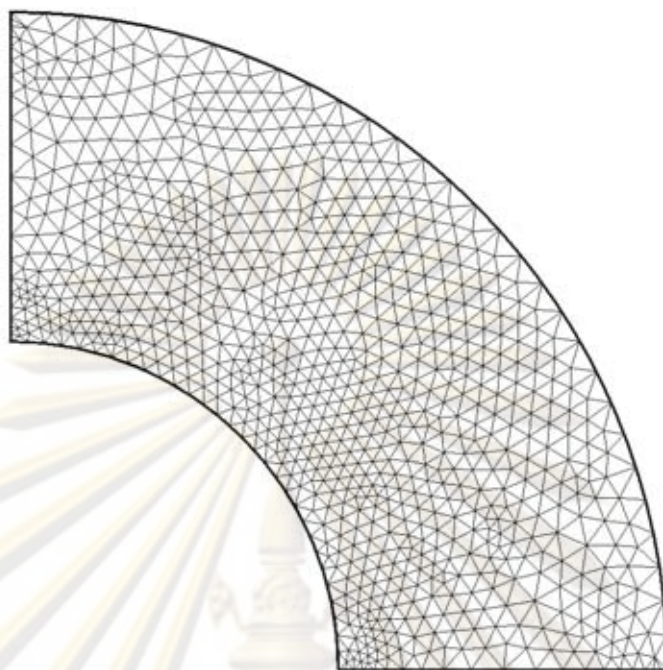
โดย $u(r)$ คือค่าการเคลื่อนตัวในแนวรัศมี



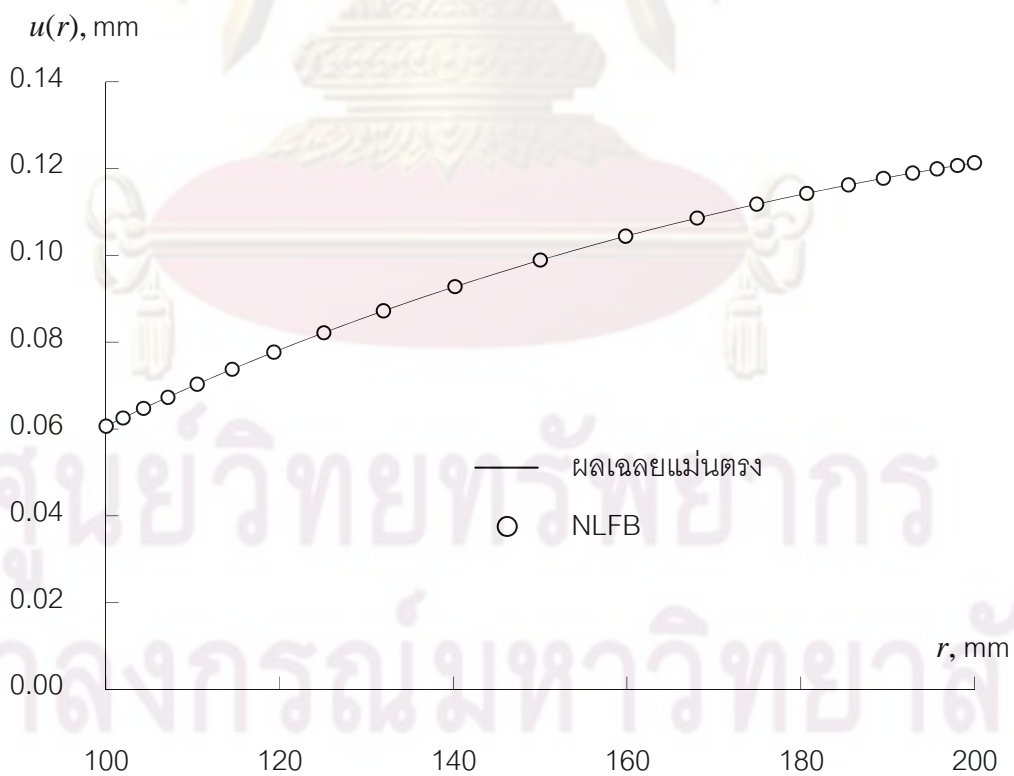
รูปที่ 6.9 วงแหวนที่มีอุณหภูมิเปลี่ยนแปลงตามแนวรัศมี

เนื่องจากลักษณะของปัญหามีความสมมาตร ดังนั้นสามารถใช้ขอบเขตในการคำนวณเพียง 1 ใน 4 ของวงแหวนได้ ดังแสดงในรูปที่ 6.9 การวิเคราะห์ปัญหาเริ่มจากการแบ่งขอบเขตของปัญหาออกเป็นเอลิเมนต์ย่อย ๆ ซึ่งประกอบไปด้วย 884 จุดต่อ 1,666 เอลิเมนต์ ดังแสดงในรูปที่ 6.10 จากนั้นประยุกต์เงื่อนไขขอบเขตโดยกำหนดให้ตลอดขอบด้านล่างไม่มีการเคลื่อนตัวในแนวแกน y และตลอดขอบทางด้านซ้ายกำหนดให้ไม่มีการเคลื่อนตัวในแกน x รวมทั้งกำหนดอุณหภูมิ $T(r)$ ตลอดทั้งขอบเขตการคำนวณ แล้วทำการคำนวณหาค่าการเคลื่อนตัวที่เกิดขึ้น

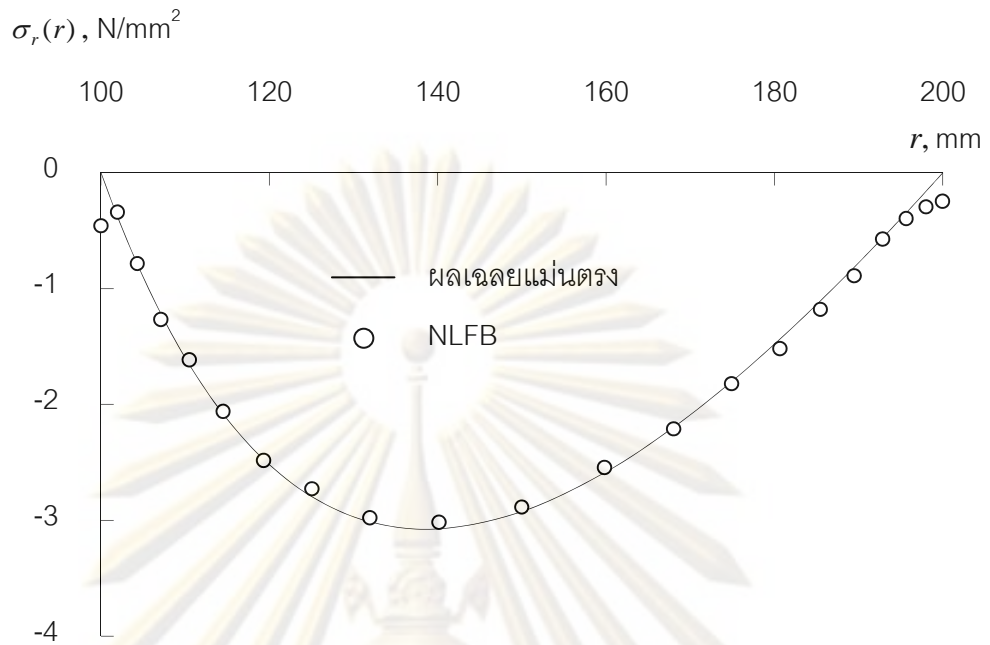
ค่าการเคลื่อนตัวในแนวรัศมี ค่าความเค้นในแนวรัศมี และค่าความเค้นในแนว θ ที่คำนวณได้จากโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ด้วยระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้มวล โดยการใช้หลักการของฟลักซ์ (NLFB) ได้นำมาแสดงเปรียบเทียบกับผลเฉลยแม่นยำตรงจากสมการ (6.4) – (6.6) ดังแสดงในรูปที่ 6.11 – 6.13 ซึ่งมีค่าความผิดพลาดเฉลี่ยเท่ากับ 0.018%, 8.089% และ 2.263% ตามลำดับ



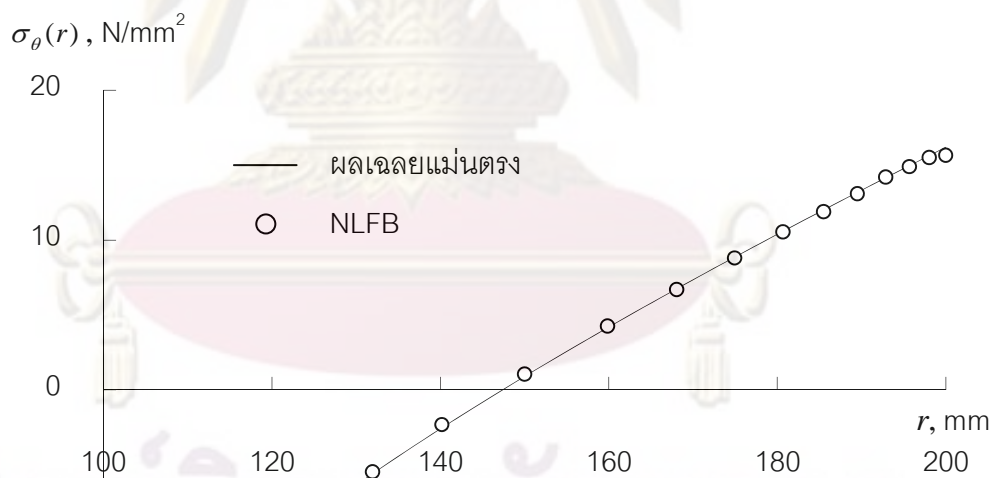
รูปที่ 6.10 รูปแบบไฟไนต์เอลิเมนต์สำหรับปัญหาวงแหวนที่มีอุณหภูมิเปลี่ยนแปลงตามแนวรัศมี



รูปที่ 6.11 การเปลี่ยนแปลงของค่าการเคลื่อนตัวตามทิศทางรัศมี ในแนวมุม θ เท่ากับ 0



รูปที่ 6.12 ความเค้นในแนวรัศมี σ_r ตามทิศทางรัศมีที่มีแนวมุม θ เท่ากับ 0



รูปที่ 6.13 ความเค้นในทิศ θ , σ_θ ตามทิศทางรัศมีที่มีแนวมุม θ เท่ากับ 0

6.4 ตัวอย่างการวิเคราะห์ปัญหาด้วยเทคนิคการปรับเอลิเมนต์โดยอัตโนมัติ

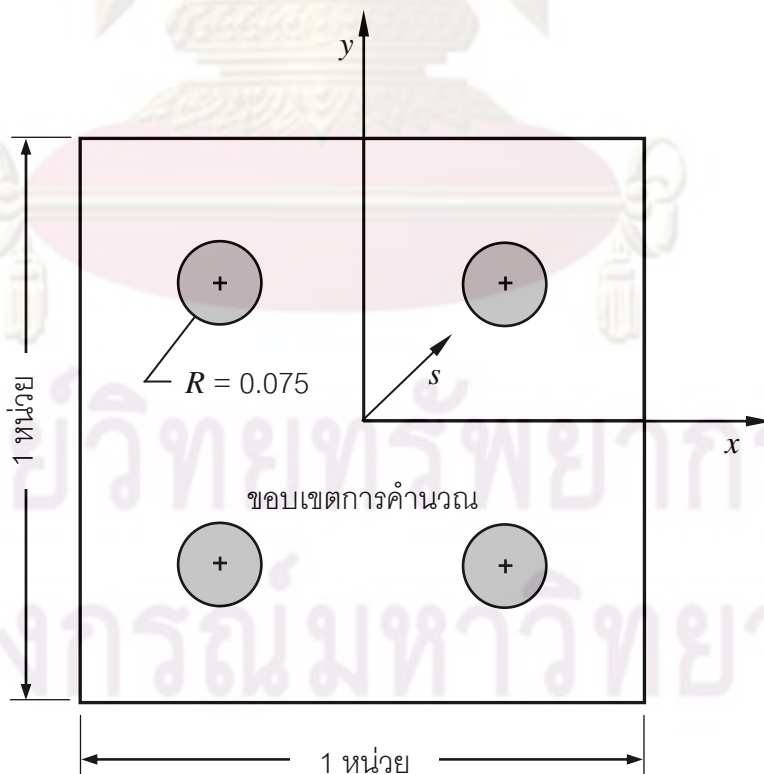
ในหัวข้อนี้จะแสดงตัวอย่างการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติกับปัญหาต่าง ๆ ซึ่งสามารถกระทำได้โดยการผนวกเทคนิคดังกล่าวเข้ากับโปรแกรม NLFB วิธีการทดสอบจะเลือกใช้ปัญหาการถ่ายเทความร้อน ปัญหาของแข็ง และปัญหาความเค้น อันเนื่องมาจากอุณหภูมิ รวมทั้งสิ้น 3 ปัญหา โดยมีรายละเอียดดังนี้

6.4.1 ปัญหาแผ่นสี่เหลี่ยมจัตุรัสภายใต้กรวยของเกรเดียนต์ความร้อน

ตัวอย่างนี้เป็นปัญหาที่มีการผลิตความร้อนในรูปของเกรเดียนต์ความร้อนสี่กรวยภายในพื้นที่รูปสี่เหลี่ยมจัตุรัสขนาด 1×1 หน่วย ดังแสดงรายละเอียดในรูปที่ 6.14 โดยมีอุณหภูมิตลอดขอบทั้งสิ้นที่ $T = 0$ และมีการผลิตความร้อนในรูปของเกรเดียนต์ความร้อน Q เป็นฟังก์ชันของพิกัด x และ y คือ

$$Q(x, y) = \begin{cases} 0 & ; B > 0.075 \\ 10A^2 \cos(AB) + 10AB^{-1} \sin(AB) & ; B \leq 0.075 \end{cases} \quad (6.7)$$

โดย $A = \pi/0.075$ และ $B = \sqrt{(x-0.25)^2 + (y-0.25)^2}$

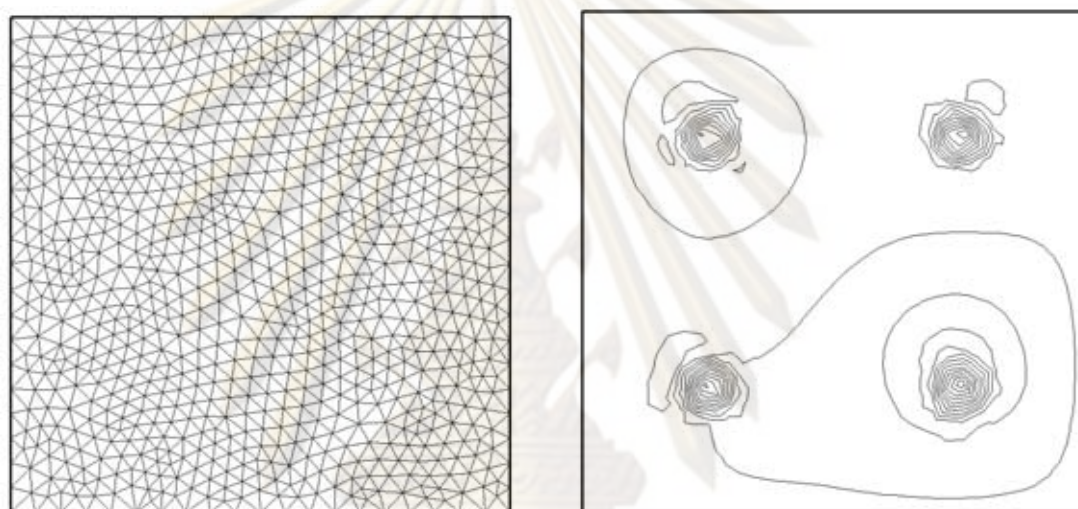


รูปที่ 6.14 ปัญหาแผ่นสี่เหลี่ยมจัตุรัสภายใต้กรวยของเกรเดียนต์ความร้อน

ปัญหานี้มีผลเฉลยแม่นยำตรงของลักษณะการกระจายของอุณหภูมิ [25] ดังนี้

$$T(x, y) = \begin{cases} 0 & ; B > 0.075 \\ 10[1 + \cos(\pi B / 0.075)] & ; B \leq 0.075 \end{cases} \quad (6.8)$$

ในการวิเคราะห์ปัญหา จะเริ่มต้นจากการสร้างรูปแบบจำลองไฟไนต์เอลิเมนต์ซึ่งได้แสดงไว้ในรูปที่ 6.15ก โดยประกอบไปด้วย 865 จุดต่อ และ 1,648 เอลิเมนต์ จากนั้นทำการประยุกต์เงื่อนไขขอบเขตตามที่กล่าวข้างต้น จากนั้นทำการวิเคราะห์ปัญหาโดยใช้รูปแบบจำลองไฟไนต์เอลิเมนต์ในรูปที่ 6.15ก ซึ่งผลการกระจายตัวของอุณหภูมิโดยใช้แบบจำลองดังกล่าวได้แสดงในรูปที่ 6.15ข



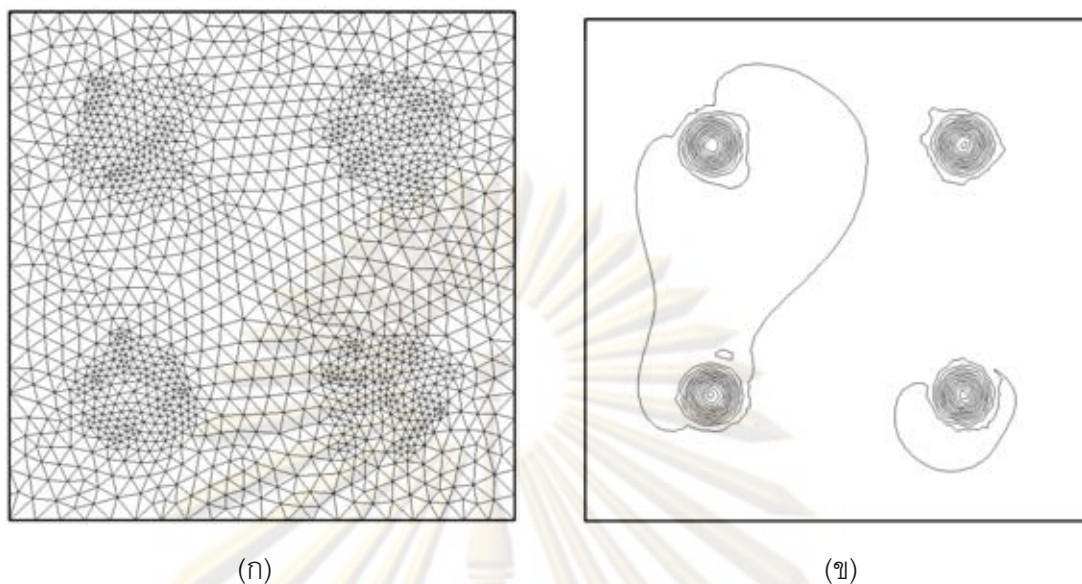
(ก)

(ข)

รูปที่ 6.15 ก) รูปแบบไฟไนต์เอลิเมนต์สำหรับปัญหาแผ่นสี่เหลี่ยมจัตุรัสภายใต้กรวยของเกรเดียนต์ความร้อน

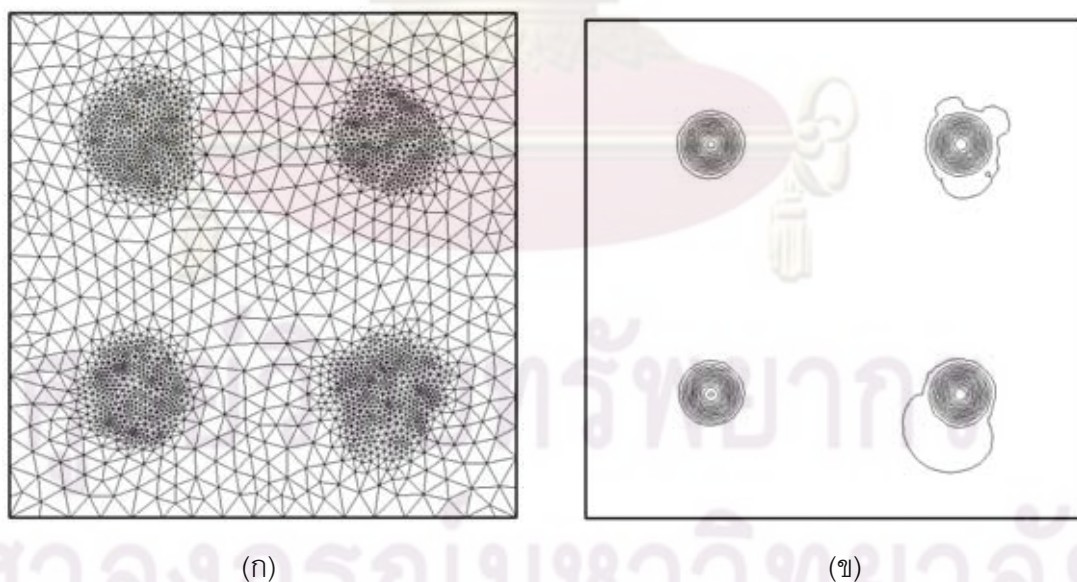
ข) การกระจายตัวของอุณหภูมิสำหรับแบบจำลองไฟไนต์เอลิเมนต์เริ่มต้น

จากนั้นนำผลลัพธ์ที่ได้ไปทำการปรับขนาดเอลิเมนต์ โดยแบบจำลองไฟไนต์เอลิเมนต์ของการปรับขนาดเอลิเมนต์ครั้งที่ 1 นี้ได้แสดงในรูปที่ 6.16ก ซึ่งประกอบไปด้วย 1,495 จุดต่อ 2,908 เอลิเมนต์ และผลลัพธ์จากรูปแบบจำลองนี้ได้แสดงในรูปที่ 6.16ข ซึ่งจะเห็นได้ว่าการวางตัวของเอลิเมนต์ในบริเวณกรวยของเกรเดียนต์ความร้อนทั้งสิ้น



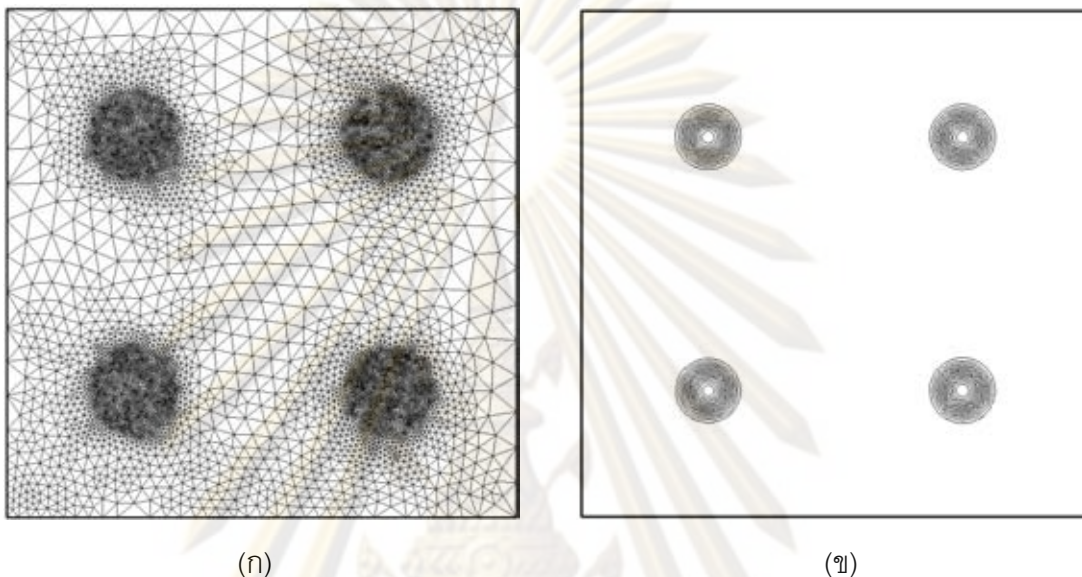
รูปที่ 6.16 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 1
ข) การกระจายตัวของอนุหภูมิที่คำนวณได้

และเมื่อทำการปรับขนาดเอลิเมนต์ครั้งที่ 2 จะได้รูปแบบจำลองไฟไนต์เอลิเมนต์ดังแสดงในรูปที่ 6.17ก ซึ่งประกอบไปด้วย 2,389 จุดต่อ 4,708 เอลิเมนต์ และผลลัพธ์จากรูปแบบจำลองนี้ได้แสดงในรูปที่ 6.17ข ซึ่งได้ผลลัพธ์ที่มีความถูกต้องมากยิ่งขึ้น



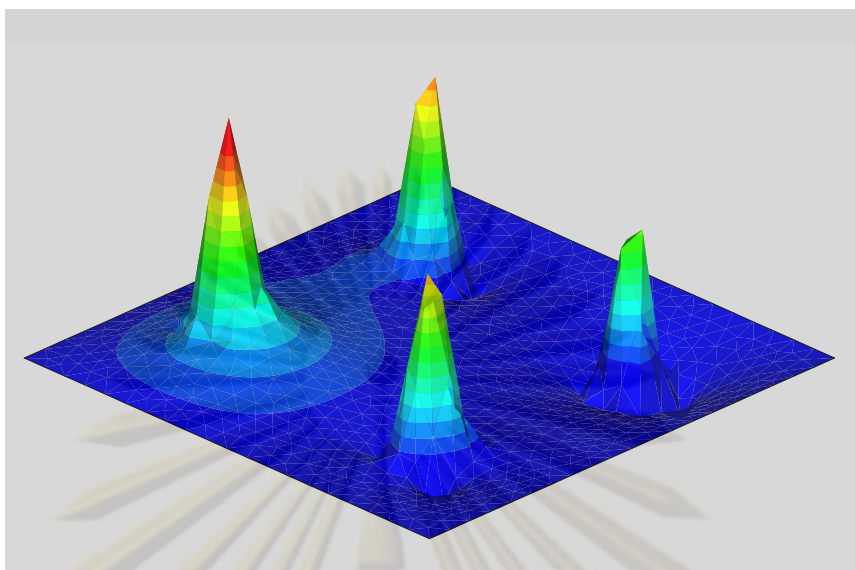
รูปที่ 6.17 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 2
ข) การกระจายตัวของอนุหภูมิที่คำนวณได้

จากนั้นจึงปรับขนาดเอลิเมนต์ครั้งสุดท้าย ซึ่งทำให้ได้รูปแบบจำลองไฟไนต์เอลิเมนต์ดังแสดงในรูปที่ 6.18ก ซึ่งประกอบไปด้วย 3,733 จุดต่อ 7,375 เอลิเมนต์ และผลลัพธ์จากรูปแบบจำลองนี้ได้แสดงในรูปที่ 6.18ข ผลลัพธ์ที่ได้มีความถูกต้องอย่างมากเมื่อเทียบกับการใช้เอลิเมนต์เริ่มต้น

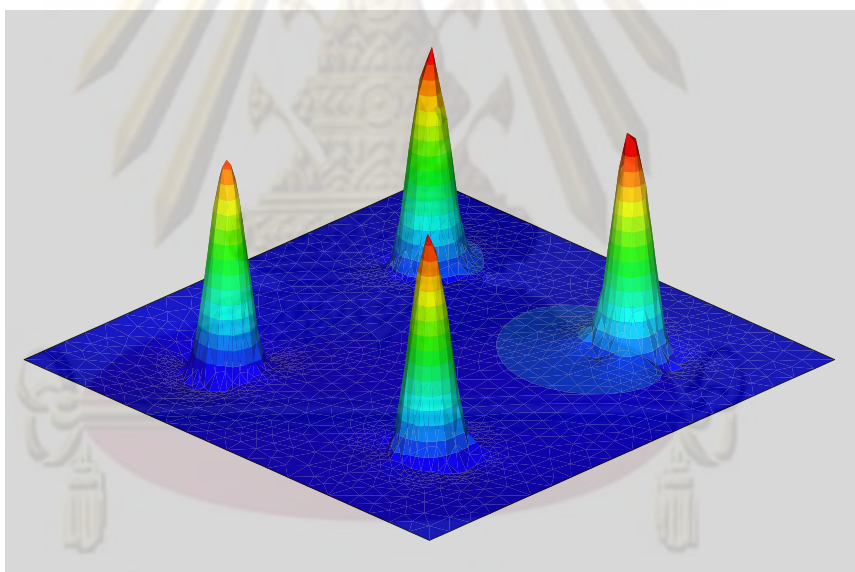


รูปที่ 6.18 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งสุดท้าย
ข) การกระจายตัวของอุณหภูมิที่คำนวณได้

นำผลเฉลยที่คำนวณได้จากรูปแบบจำลองไฟไนต์เอลิเมนต์ทั้งสี่รูปแบบมาพล็อตอีกครั้งในรูปแบบสามมิติดังแสดงในรูปที่ 6.19 โดยเรียงลำดับจากรูปแบบจำลองเริ่มต้น รูปแบบจำลองจากการปรับขนาดครั้งที่ 1 รูปแบบจำลองจากการปรับขนาดครั้งที่ 2 และรูปแบบจำลองจากการปรับขนาดครั้งสุดท้าย จากแบบจำลองสามมิติดังกล่าวพบว่า การกระจายตัวของอุณหภูมิบริเวณฐานของกรวยมีลักษณะการสั้นของผลเฉลยและมีการหักของยอดกรวยทั้งสี่ โดยเฉพาะอย่างยิ่งจากรูปแบบจำลองเริ่มต้นในรูปที่ 6.19ก และจากแบบจำลองจากการปรับขนาดครั้งที่ 1 จากนั้นการสั้นของผลเฉลยบริเวณฐานของกรวยมีแนวโน้มลดลงและความสมบูรณ์ของยอดกรวยเพิ่มขึ้นเมื่อจำนวนเอลิเมนต์ที่อยู่โดยรอบของกรวยมีจำนวนเพิ่มขึ้นจากการปรับขนาดเอลิเมนต์โดยอัตโนมัติ

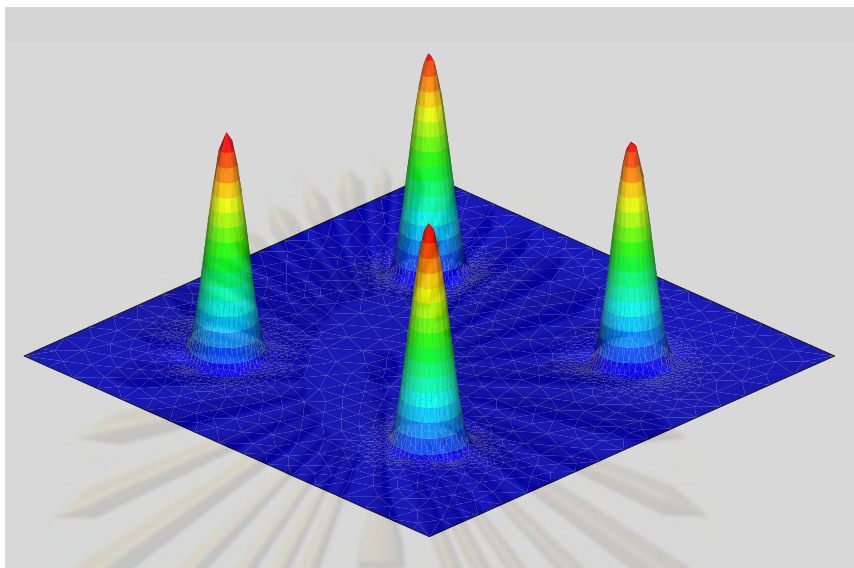


(ก)

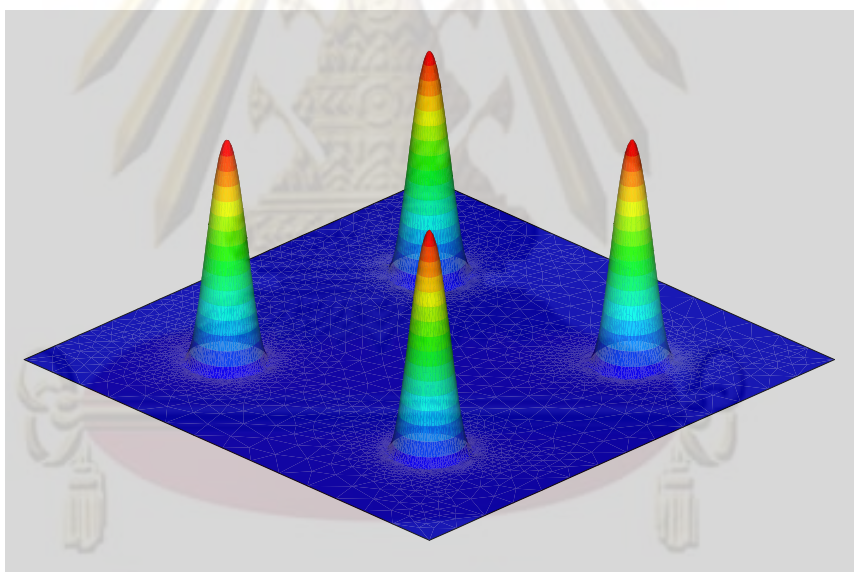


(ข)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



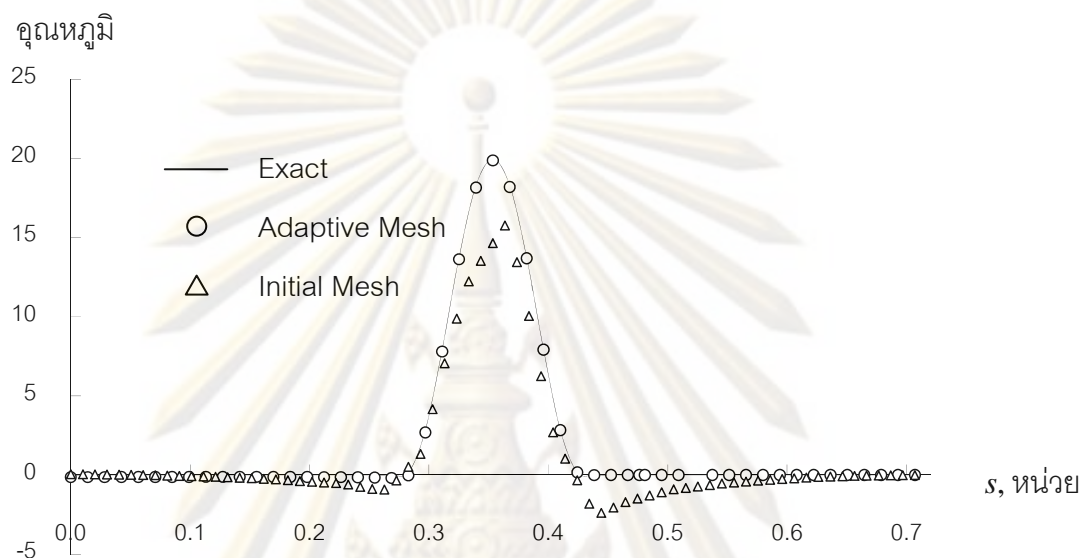
(ค)



(ง)

- รูปที่ 6.19 ก) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลองเริ่มต้น
 ข) การกระจายตัวของอุณหภูมิสำหรับแบบจำลองจากการปรับขนาดครั้งที่ 1
 ค) การกระจายตัวของอุณหภูมิสำหรับแบบจำลองจากการปรับขนาดครั้งที่ 2
 ง) การกระจายตัวของอุณหภูมิสำหรับแบบจำลองจากการปรับขนาดครั้งสุดท้าย

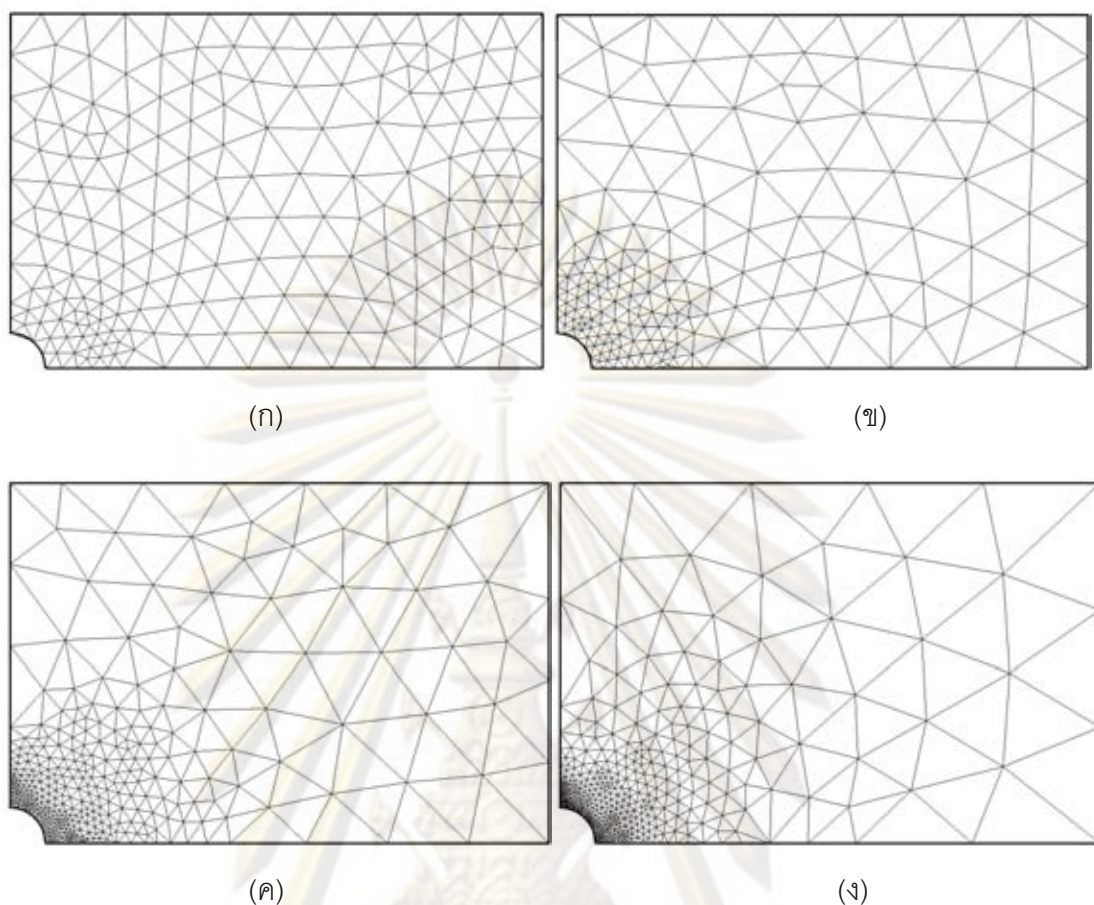
รูปที่ 6.20 แสดงผลการเปรียบเทียบค่าอุณหภูมิในแนวแกน r ที่คำนวณได้จากโปรแกรม NLFB ซึ่งจะเห็นได้ว่าผลลัพธ์ที่ได้หลังจากทำการปรับขนาดเอลิเมนต์แล้วมีค่าใกล้เคียงกับผลเฉลยแม่นยำตรงมาก



รูปที่ 6.20 การเปรียบเทียบผลลัพธ์จากการคำนวณโดยใช้รูปแบบไฟไนต์เอลิเมนต์เริ่มต้นและรูปแบบไฟไนต์เอลิเมนต์ที่ทำการปรับขนาดกับผลเฉลยแม่นยำตรง

6.4.2 ปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง

ในตัวอย่างนี้เป็นการทดสอบเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติกับปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง โดยมีรายละเอียดและลักษณะของปัญหาแสดงในหัวข้อ 6.2.1 ซึ่งการปรับขนาดเอลิเมนต์เริ่มต้นจากการนำผลลัพธ์ที่ได้จากหัวข้อ 6.2.1 ไปทำการปรับขนาด โดยแบบจำลองของการปรับขนาดเอลิเมนต์ครั้งที่ 1 นี้ประกอบด้วย 209 จุดต่อ 368 เอลิเมนต์ แบบจำลองของการปรับขนาดเอลิเมนต์ครั้งที่ 2 ประกอบด้วย 187 จุดต่อ 236 เอลิเมนต์ แบบจำลองของการปรับขนาดเอลิเมนต์ครั้งที่ 3 ประกอบด้วย 412 จุดต่อ 755 เอลิเมนต์ และแบบจำลองของการปรับขนาดเอลิเมนต์ครั้งที่ 4 ประกอบด้วย 490 จุดต่อ 901 เอลิเมนต์ ได้แสดงในรูปที่ 6.21ก – 6.21ง ผลลัพธ์จากการคำนวณ (ค่าความเค้นในแนวแกน x ที่จุด A) โดยการปรับขนาดเอลิเมนต์ในแต่ละครั้งจะถูกเปรียบเทียบค่าเปอร์เซ็นต์ความผิดพลาดดังแสดงในตารางที่ 6.2



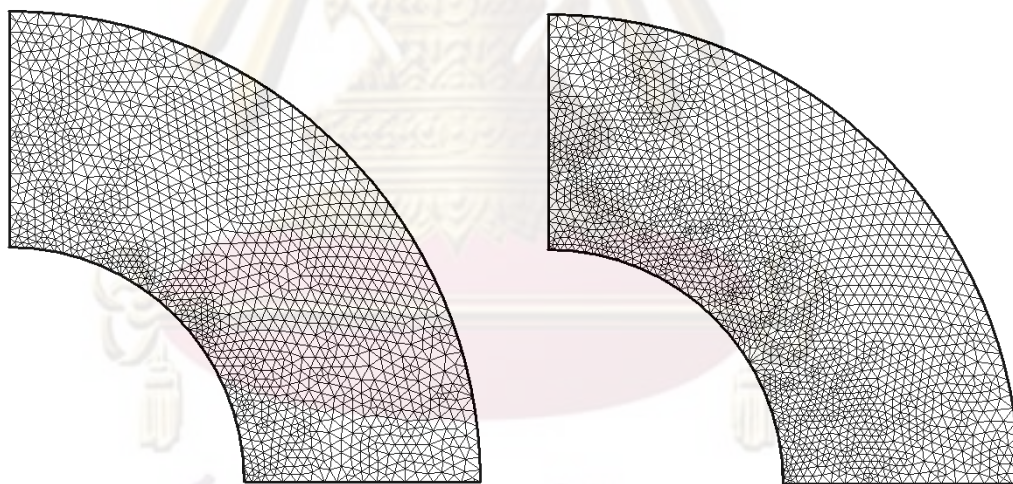
รูปที่ 6.21 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 1
 ข) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 2
 ค) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 3
 ง) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 4

ตารางที่ 6.2 ตารางแสดงค่าความเค้นในแนวแกน x ที่จุด A และการเปรียบเทียบค่าเปอร์เซ็นต์ความผิดพลาดจากการปรับขนาดเอลิเมนต์ครั้งที่ 1 – 4 สำหรับปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง

จุด A	ค่าความเค้นในแนวแกน x (psi)	เปอร์เซ็นต์ความผิดพลาด (%)
ปรับขนาดครั้งที่ 1	1.966	34.47
ปรับขนาดครั้งที่ 2	2.378	20.73
ปรับขนาดครั้งที่ 3	2.708	9.73
ปรับขนาดครั้งที่ 4	2.898	3.40

6.4.3 ปัญหาวงแหวนที่มีอุณหภูมิเปลี่ยนแปลงตามแนวรัศมี

ตัวอย่างสุดท้ายนี้เป็นการทดสอบเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติกับปัญหาความเค้นอันเนื่องมาจากอุณหภูมิ โดยเลือกใช้ปัญหาวงแหวนที่มีอุณหภูมิเปลี่ยนแปลงตามแนวรัศมี ซึ่งมีรายละเอียดและลักษณะของปัญหาแสดงในหัวข้อ 6.3.2 การปรับขนาดเอลิเมนต์เริ่มต้นจากการนำผลลัพธ์ที่ได้จากหัวข้อ 6.3.2 ไปทำการปรับขนาดเอลิเมนต์ โดยแบบจำลองของการปรับขนาดเอลิเมนต์ครั้งที่ 1 นี้ประกอบด้วย 1,470 จุดต่อ 2,815 เอลิเมนต์ แบบจำลองของการปรับขนาดเอลิเมนต์ครั้งที่ 2 ประกอบด้วย 2,008 จุดต่อ 3,878 เอลิเมนต์ ได้แสดงในรูปที่ 6.22ก - 6.22ข ภายหลังจากการคำนวณผลลัพธ์ (ค่าการเคลื่อนตัวตามทิศทางรัศมี ความเค้นในแนวรัศมี และความเค้นในทิศ θ ในแนวมุม θ เท่ากับ 0) และค่าเปอร์เซ็นต์ความผิดพลาดเฉลี่ยโดยการปรับขนาดเอลิเมนต์ทั้งสองครั้ง ผลทั้งหมดที่ได้จะถูกนำไปเปรียบเทียบกับผลลัพธ์ที่ได้จากแบบจำลองไฟไนต์เอลิเมนต์เริ่มต้น โดยจะนำความเค้นในแนวรัศมีแสดงในรูปที่ 6.23 และค่าเปอร์เซ็นต์ความผิดพลาดเฉลี่ยของผลลัพธ์ทั้งหมดจะแสดงในตารางที่ 6.3



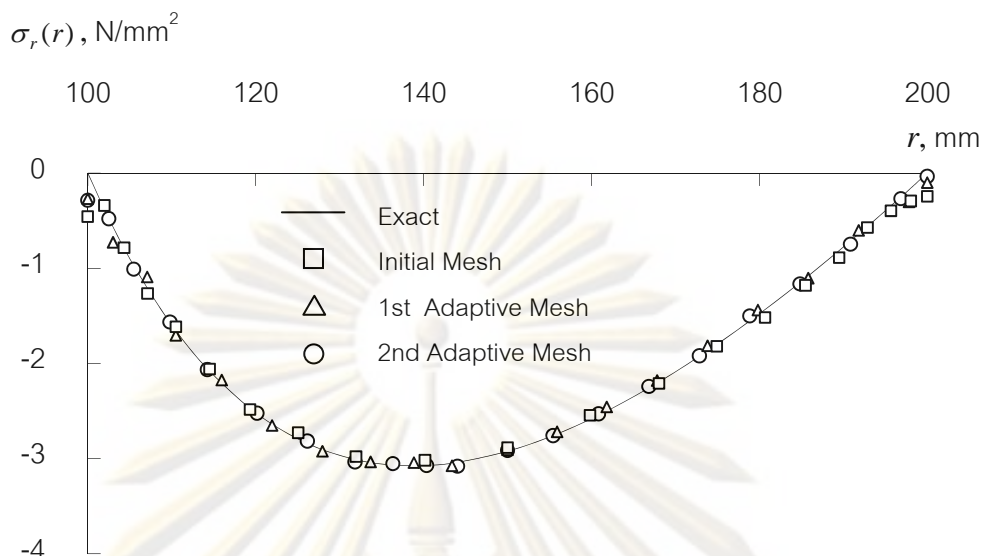
(ก)

(ข)

รูปที่ 6.22 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 1

ข) รูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาดเอลิเมนต์ครั้งที่ 2

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.23 การเปรียบเทียบความเค้นในแนวรัศมี σ_r ที่มีแนวมุม θ เท่ากับ 0 ระหว่าง ผลเฉลยแม่นยำตรง รูปแบบไฟไนต์เอลิเมนต์เริ่มต้น รูปแบบปรับขนาดครั้งที่ 1 และรูปแบบปรับขนาดครั้งที่ 2

ตารางที่ 6.3 ตารางแสดงการเปรียบเทียบค่าเปอร์เซ็นต์ความผิดพลาดของผลลัพธ์ (ค่าการเคลื่อนตัวตามทิศทางรัศมี ความเค้นในแนวรัศมี และความเค้นในทิศ θ ในแนวมุม θ เท่ากับ 0) จาก รูปแบบจำลองไฟไนต์เอลิเมนต์เริ่มต้น และรูปแบบจำลองไฟไนต์เอลิเมนต์จากการปรับขนาด เอลิเมนต์ครั้งที่ 1 และ 2

แนวมุม θ เท่ากับ 0	แบบจำลอง เริ่มต้น	แบบจำลอง การปรับครั้งที่ 1	แบบจำลอง การปรับครั้งที่ 2
เปอร์เซ็นต์ความผิดพลาด ของค่าการเคลื่อนตัว (%)	0.018	0.018	0.023
เปอร์เซ็นต์ความผิดพลาด ของความเค้นในแนวรัศมี (%)	8.089	7.604	1.398
เปอร์เซ็นต์ความผิดพลาด ของความเค้นในทิศ θ (%)	2.263	3.167	2.410

บทที่ 7

การคะเนความคลาดเคลื่อนแบบโพสเทอริออริ

การคะเนความคลาดเคลื่อนถูกนำมาใช้โดยมีวัตถุประสงค์เพื่อให้วิธีการปรับขนาดเอลิเมนต์นั้นมีความถูกต้องแม่นยำมากยิ่งขึ้น โดยเริ่มต้นจากการตรวจหาเอลิเมนต์ที่มีความคลาดเคลื่อนสูงที่สุด แล้วจึงแบ่งแยกความคลาดเคลื่อนนี้ให้น้อยลงจนอยู่ภายใต้ขอบเขตที่ยอมรับได้ ในการวิเคราะห์ปัญหาบางปัญหาอาจมีการปรับขนาดเอลิเมนต์หลายครั้งซึ่งจำเป็นต้องใช้ความชำนาญของผู้วิเคราะห์ปัญหาเพื่อยุติการปรับขนาดเอลิเมนต์ ดังนั้นการคะเนความคลาดเคลื่อนจึงสามารถนำมาใช้ในอีกวัตถุประสงค์เพื่อเป็นดัชนีชี้วัดการยุติการปรับขนาดเอลิเมนต์นั่นเอง ในบทนี้จึงเริ่มต้นจากการอธิบายความหมาย นิยาม และที่มาการคะเนของความคลาดเคลื่อน จากนั้นจะอธิบายรายละเอียดของการคะเนความคลาดเคลื่อนแบบโพสเทอริออริ และสุดท้ายจะนำไปสู่การประดิษฐ์สมการไฟไนต์เอลิเมนต์สำหรับการคะเนความคลาดเคลื่อนแบบโพสเทอริออริ และดัชนีชี้วัดความคลาดเคลื่อน

7.1 ดัชนีชี้วัดความคลาดเคลื่อน (Error Indicator)

ค่าความคลาดเคลื่อนในระเบียบวิธีไฟไนต์เอลิเมนต์ [3] ถูกนิยามว่า “ผลต่างระหว่างผลเฉลยแม่นยำและผลเฉลยโดยประมาณจากระเบียบวิธีไฟไนต์เอลิเมนต์”

$$\text{ค่าความคลาดเคลื่อน} = E(x, y) = T(x, y) - T_{FE}(x, y) \quad (7.1)$$

โดย $T(x, y)$ แทน ผลเฉลยแม่นยำ

$T_{FE}(x, y)$ แทน ผลเฉลยโดยประมาณจากระเบียบวิธีไฟไนต์เอลิเมนต์

ซึ่งค่าความคลาดเคลื่อนนี้จะแปรเปลี่ยนไปบนโดเมนของปัญหา ดังนั้นการที่จะวัดความคลาดเคลื่อนโดยรวม (global error) ที่ลดลงเนื่องจากการปรับขนาดเอลิเมนต์ จึงใช้นอร์มของความคลาดเคลื่อน (norm of the error), $\|E\|$, เป็นตัวชี้วัด ซึ่งนอร์มที่ใช้อย่างกว้างขวางก็คือ แอลทู, L_2 , โดยนอร์มดังกล่าวจะทำการวัดค่าความคลาดเคลื่อนรากที่สองของกำลังสองเฉลี่ย (root-mean-square error) บนโดเมน

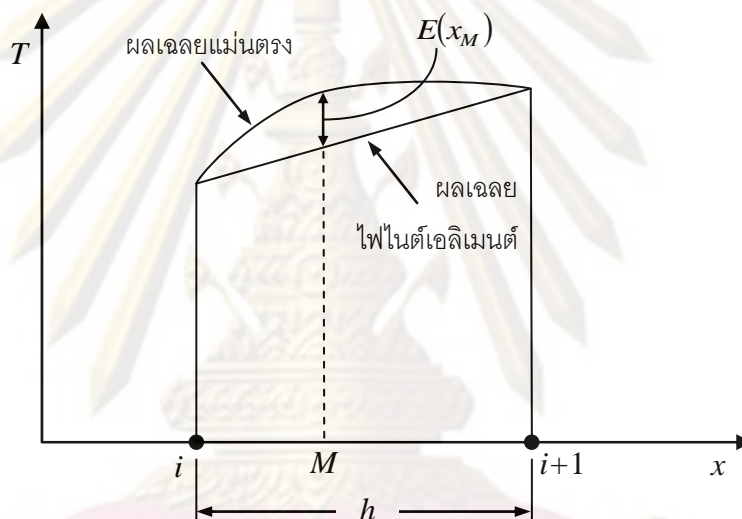
$$\text{แอลทูนอร์ม} = \|E\|_{L_2} = \left[\int_{\Omega} E^2 d\Omega \right]^{1/2} \quad (7.2)$$

แต่ค่าความคลาดเคลื่อนที่วัดได้จากแอลทูนอร์มนั้นจำเป็นที่จะต้องทราบค่าผลเฉลยแม่นยำซึ่งเห็นได้จากสมการ (7.1) ซึ่งอันที่จริงแล้วสำหรับปัญหาทั่ว ๆ ไปผลเฉลยแม่นยำนั้นไม่สามารถ

ทราบค่าได้ อีกทั้งความคลาดเคลื่อนที่วัดได้นั้นเป็นความคลาดเคลื่อนโดยรวมซึ่งในความเป็นจริงแล้วความคลาดเคลื่อนที่ต้องการก็คือความคลาดเคลื่อนเฉพาะที่ (local error) ที่อยู่ในระดับของเอลิเมนต์ เพราะสามารถแสดงให้เห็นถึงบริเวณที่มีการเปลี่ยนแปลงเกรเดียนท์อย่างสูงได้

หากพิจารณาเอลิเมนต์หนึ่งมิติที่มีความยาว h ซึ่งประกอบด้วยจุดต่อ i และ $i+1$ สำหรับปัญหาการถ่ายเทความร้อนในหนึ่งมิติดังแสดงในรูปที่ 7.1 โดยใช้อนุกรมเทเลอร์ (Taylor series) ค่าความคลาดเคลื่อนที่จุด \bar{x} ใด ๆ บนเอลิเมนต์สามารถเขียนอยู่ในรูป

$$E(x) = E(\bar{x}) + \left. \frac{dE}{dx} \right|_{\bar{x}} (x - \bar{x}) + \frac{1}{2} \left. \frac{d^2E}{dx^2} \right|_{\bar{x}} (x - \bar{x})^2 + \dots \quad (7.3)$$



รูปที่ 7.1 ความคลาดเคลื่อนจากการประมาณภายในสำหรับปัญหาการถ่ายเทความร้อนหนึ่งมิติ จากสมการ (7.3) แทนจุด \bar{x} ด้วยจุด x_M ซึ่งเป็นจุดที่มีค่าความคลาดเคลื่อนสูงสุด จะได้

$$\left. \frac{dE}{dx} \right|_{x_M} = 0 \quad (7.4)$$

เมื่อละทิ้งพจน์อนุพันธ์ที่มีอันดับสูง จะก่อให้เกิดค่าของพจน์อนุพันธ์อันดับสองโดยประมาณของสมการ (7.3) คือ

$$E(x) = E(x_M) + \frac{1}{2} \left. \frac{d^2E}{dx^2} \right|_{x_M} (x - x_M)^2 \quad (7.5)$$

สมมติให้ผลเฉลยจากระเบียบวิธีไฟไนต์เอลิเมนต์มีค่าแม่นยำที่จุดต่อทั้งสอง ซึ่งจะได้ $E(x_i) = E(x_{i+1}) = 0$ ดังนั้นที่จุดต่อ x_i สามารถเขียนสมการ (7.5) ได้เป็น

$$0 = E(x_M) + \frac{1}{2} \frac{d^2 E}{dx^2} \Big|_{x_M} (x_i - x_M)^2$$

$$\text{หรือ} \quad E(x_M) = -\frac{1}{2} \frac{d^2 E}{dx^2} \Big|_{x_M} (x_i - x_M)^2 \quad (7.6)$$

เช่นเดียวกับที่จุดต่อ x_{i+1} สามารถเขียนสมการ (7.5) ได้เป็น

$$E(x_M) = -\frac{1}{2} \frac{d^2 E}{dx^2} \Big|_{x_M} (x_{i+1} - x_M)^2 \quad (7.7)$$

จากการแก้สมการ (7.6) และ (7.7) จะได้

$$|x_{i+1} - x_M| < \frac{h}{2} \quad \text{หรือ} \quad |x_i - x_M| < \frac{h}{2} \quad (7.8)$$

ดังนั้นค่าความคลาดเคลื่อนจะมีค่า

$$|E| \leq \frac{1}{8} \left| \frac{d^2 E}{dx^2} \Big|_{x_M} \right| h^2 \quad (7.9)$$

และเนื่องจากค่าความคลาดเคลื่อนนั้นขึ้นอยู่กับผลเฉลยแม่นยำตรงที่ไม่ทราบค่า ดังนั้นค่าอนุพันธ์อันดับที่สองจึงไม่สามารถคำนวณได้เช่นกัน แต่อย่างไรก็ตามค่าอนุพันธ์นั้นสามารถประมาณด้วย [3]

$$\frac{d^2 E}{dx^2} \Big|_{x_M} = \frac{d^2 T_{FE}}{dx^2} \Big|_{(e)} \quad (7.10)$$

ค่าความคลาดเคลื่อนตามสมการ (7.9) จึงเขียนอยู่ในรูปแบบของเอลิเมนต์ได้ดังนี้

$$|E|_{(e)} \leq \frac{1}{8} \left| \frac{d^2 T_{FE}}{dx^2} \Big|_{(e)} \right| h^2 \quad (7.11)$$

ซึ่งสมการ (7.11) นี้ก็คือตัวชี้วัดความคลาดเคลื่อน และในทำนองเดียวกันกับปัญหาสองมิติ ตัวชี้วัดความคลาดเคลื่อนจะอยู่ในรูป

$$|E|_{(e)} \leq \frac{1}{8} \left(\left| \frac{d^2 T_{FE}}{dx^2} \right| + 2 \left| \frac{d^2 T_{FE}}{dx dy} \right| + \left| \frac{d^2 T_{FE}}{dy^2} \right| \right) h^2 \quad (7.12)$$

โดย h แทน มิติของเอลิเมนต์คุณลักษณะเฉพาะ (characteristic element dimension)

หากเปรียบเทียบสมการ (5.1) และ (7.11) จะพบว่าสมการทั้งสองมีลักษณะเดียวกันก็เพราะว่าหลักการของเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติก็คือ การกระจายค่าความคลาดเคลื่อนเฉพาะที่ให้มีค่าสม่ำเสมอตลอดโดเมนดังอธิบายในบทที่ 5

การพิจารณาค่าความคลาดเคลื่อนในปัญหาของแข็งก็มีหลักการเช่นเดียวกัน นั่นคือ

$$\text{ค่าความคลาดเคลื่อน} = e = u - \hat{u} \quad (7.13)$$

โดย u แทน ระยะเวลาเคลื่อนตัวแม่นยำตรง

\hat{u} แทน ระยะเวลาเคลื่อนตัวโดยประมาณ

ในทำนองเดียวกัน หากสนใจค่าความคลาดเคลื่อนในพจน์ของความเครียดหรือความเค้น ก็สามารถแสดงอยู่ในรูปดังนี้

$$e_\varepsilon = \varepsilon - \hat{\varepsilon} \quad (7.14)$$

$$e_\sigma = \sigma - \hat{\sigma} \quad (7.15)$$

โดย ε แทน ความเครียดแม่นยำตรง

$\hat{\varepsilon}$ แทน ความเครียดโดยประมาณ

σ แทน ความเค้นแม่นยำตรง

$\hat{\sigma}$ แทน ความเค้นโดยประมาณ

7.2 การคะเนความคลาดเคลื่อนแบบโพสเทอริออริ

การคะเนความคลาดเคลื่อนแบบคลาสสิก (a-priori error estimation) นั้นใช้กระบวนการทางคณิตศาสตร์เพื่อคำนวณหาความคลาดเคลื่อน ซึ่งกระบวนการทางคณิตศาสตร์นี้จำเป็นต้องกำหนดลักษณะเฉพาะของปัญหา เช่น ค่าสัมประสิทธิ์ในสมการเชิงอนุพันธ์ เงื่อนไขขอบเขต เงื่อนไขเริ่มต้น ลักษณะและขนาดของเมช (mesh) ความราบเรียบของผลเฉลย (smoothness) ดีกรีโพลิโนเมียลที่มีมิติ (space) เป็นฟังก์ชันต่อเนื่อง เป็นต้น แล้วจึงคำนวณด้วยวิธีทางคณิตศาสตร์ทำให้สามารถที่จะคะเนความคลาดเคลื่อนก่อนที่จะได้ผลเฉลยจากระเบียบวิธีไฟไนต์เอลิเมนต์ แต่เมื่อลักษณะเฉพาะของปัญหามีการเปลี่ยนแปลงก็จำเป็นที่จะต้องวิเคราะห์ปัญหานั้นใหม่ โดยเฉพาะอย่างยิ่งการวิเคราะห์ปัญหาบนเมชที่สามารถปรับขนาดเอลิเมนต์ได้ก็จำเป็นที่จะวิเคราะห์ซ้ำหลายครั้ง ทำให้สิ้นเปลืองเวลาและทรัพยากรการคำนวณจำนวนมาก จึงมีความจำเป็นต้องหากระบวนการใหม่ที่สะดวกกว่าการคะเนความคลาดเคลื่อนแบบคลาสสิก กระบวนการใหม่นี้คือการคะเนความคลาดเคลื่อนแบบโพสเทอริออริ (a-posteriori error estimation) ซึ่งเป็นการคะเนความคลาดเคลื่อนภายหลังจากที่ได้ผลเฉลยจากระเบียบวิธีไฟไนต์เอลิเมนต์ ด้วยเหตุที่ลักษณะเฉพาะของปัญหาถูกกำหนดไว้แล้วในระเบียบวิธีไฟไนต์เอลิเมนต์ทำให้การคำนวณความคลาดเคลื่อนหลังจากที่ได้ผลเฉลยนั้นมีความสอดคล้องกับระเบียบวิธีไฟไนต์เอลิเมนต์ กระบวนการนี้จะสร้างค่าประมาณอันดับสูงซึ่งได้มาจากผลเฉลยของระเบียบวิธีไฟไนต์

เอลิเมนต์ เพื่อทำการคะเนความคลาดเคลื่อนโดยการแทนค่าดังกล่าวลงในผลเฉลยแม่นยำตรง [8, 9] ดังเช่นการแทนค่าประมาณอันดับสูงลงในสมการ (7.1) ค่าความคลาดเคลื่อนแบบโพสเทอริออริสามารถเขียนได้เป็น

$$E(x, y) \approx T^*(x, y) - T_{FE}(x, y) \quad (7.16)$$

โดย $T^*(x, y)$ แทน ผลเฉลยโดยประมาณอันดับสูงจากระเบียบวิธีไฟไนต์เอลิเมนต์

ประสิทธิภาพของตัวชี้วัดความคลาดเคลื่อนแบบโพสเทอริออริสามารถพิจารณาได้จากดัชนีชี้วัดความคลาดเคลื่อน 2 ค่า ซึ่งได้แก่

(ก) ดัชนีแสดงผล (effectivity index) ซึ่งเป็นสัดส่วนระหว่างค่าความคลาดเคลื่อนแบบโพสเทอริออริกับค่าความคลาดเคลื่อนแท้จริง นั่นคือ

$$\theta = \frac{\|E\|}{\|E\|} \quad (7.17)$$

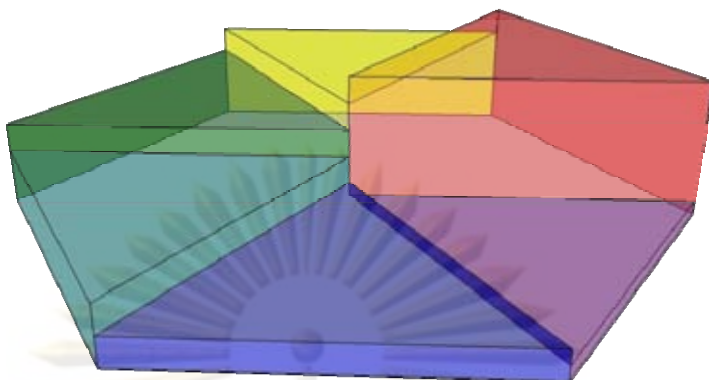
(ข) เปอร์เซ็นต์ความคลาดเคลื่อนแบบโพสเทอริออริสัมพัทธ์ (relative percentage of posteriori error) สามารถเขียนอยู่ในรูป

$${}^\circ\eta = \frac{\|E\|^2}{(\|E\|^2 + \|T_{FE}\|^2)^{1/2}} \times 100\% \quad (7.18)$$

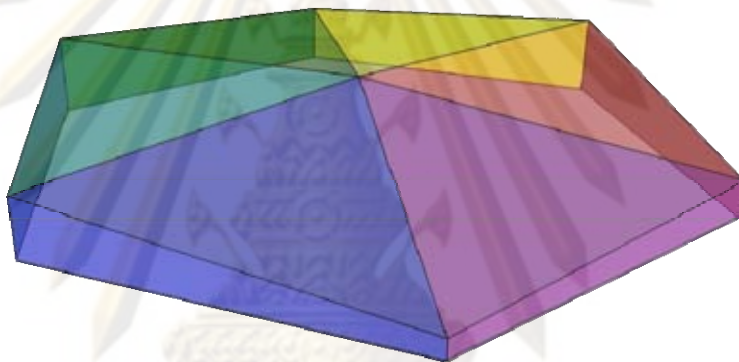
โดย $\|E\|$ แทน ค่าความคลาดเคลื่อนแบบโพสเทอริออริ

7.3 การคะเนความคลาดเคลื่อนด้วยการใช้หลักการของฟลักซ์

ตัวชี้วัดความคลาดเคลื่อนด้วยการคะเนความคลาดเคลื่อนแบบโพสเทอริออริซึ่งอธิบายไว้ในหัวข้อ 7.1 และ 7.2 นั้นได้จากการหาผลต่างระหว่างผลเฉลยแม่นยำตรงและผลเฉลยโดยประมาณอันดับสูงจากระเบียบวิธีไฟไนต์เอลิเมนต์ แต่ในวิทยานิพนธ์เล่มนี้จะนำเสนอตัวชี้วัดความคลาดเคลื่อนซึ่งคำนวณจากผลต่างระหว่างผลเฉลยโดยประมาณด้วยการใช้หลักการของฟลักซ์และผลเฉลยโดยประมาณแบบดั้งเดิม ทั้งนี้เนื่องมาจากการศึกษาระเบียบวิธีไฟไนต์เอลิเมนต์แบบดั้งเดิมและระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์ดังที่อธิบายในบทที่ 3 พบว่า ลักษณะการกระจายของฟลักซ์ที่ได้จากระเบียบวิธีไฟไนต์เอลิเมนต์แบบดั้งเดิมจะมีลักษณะไม่ต่อเนื่องระหว่างเอลิเมนต์ดังแสดงในรูปที่ 7.2 แต่จากระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์นั้นจะให้ลักษณะการกระจายของฟลักซ์ที่มีลักษณะต่อเนื่องซึ่งแสดงด้วยรูปที่ 7.3



รูปที่ 7.2 ลักษณะการกระจายของฟลักซ์ที่ได้จากระเบียบวิธีไฟไนต์เอลิเมนต์แบบดั้งเดิม



รูปที่ 7.3 ลักษณะการกระจายของฟลักซ์ที่ได้จากระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์

7.3.1 ความคลาดเคลื่อนที่คำนวณจากผลต่างของฟลักซ์

ความคลาดเคลื่อนโดยประมาณดังที่กล่าวมาข้างต้นสามารถเขียนในรูปสมการได้ดังนี้

$$e_g = g^{FB} - g^{ST} \quad (7.19)$$

โดย e_g แทน ความคลาดเคลื่อนโดยประมาณที่คำนวณจากผลต่างของฟลักซ์

g^{FB} แทน ฟลักซ์ซึ่งคำนวณด้วยการใช้หลักการของฟลักซ์

g^{ST} แทน ฟลักซ์ซึ่งคำนวณด้วยระเบียบวิธีแบบดั้งเดิม

จากนั้นจึงประยุกต์ระเบียบวิธีถ่วงน้ำหนักเศษตค่างเข้ากับสมการ (7.19) สมการจะอยู่ในรูปดังนี้

$$\int_{\Omega} \{\bar{N}\} (g^{FB} - g^{ST}) d\Omega = 0 \quad (7.20ก)$$

$$\text{หรือ} \quad \int_{\Omega} \{\bar{N}\} (g^{FB}) d\Omega = \int_{\Omega} \{\bar{N}\} (g^{ST}) d\Omega \quad (7.20\text{ข})$$

แทนลักษณะการกระจายของฟังก์ชันสำหรับเอลิเมนต์ดังสมการ (3.27) ลงในพจน์ทางด้านซ้ายของสมการ (7.20ข) จะได้

$$\int_{\Omega} \{\bar{N}\} ([\bar{N}] \{\bar{g}^{FB}\}) d\Omega = \int_{\Omega} \{\bar{N}\} \left(-c \left[\frac{\partial N}{\partial x_i} \right] \{U\} \right) d\Omega \quad (7.21)$$

โดย $\{\bar{g}^{FB}\}$ แทน เวกเตอร์ของฟังก์ชันที่ตำแหน่งจุดต่อซึ่งคำนวณด้วยการใช้หลักการของฟังก์ชัน
 $\{U\}$ แทน ตัวแปรปฐภูมิ
 c แทน คุณสมบัติของวัสดุ

จัดรูปใหม่จะได้สมการไฟไนต์เอลิเมนต์ที่อยู่ในรูปของฟังก์ชันที่ตำแหน่งจุดต่อดังแสดง

$$[M] \{\bar{g}^{FB}\} = -c [G] \{U\} \quad (7.22)$$

$$\text{โดย} \quad [M] = \int_{\Omega} \{\bar{N}\} [\bar{N}] d\Omega \quad (7.23\text{ก})$$

$$[G] = \int_{\Omega} \{\bar{N}\} \left[\frac{\partial N}{\partial x_i} \right] d\Omega \quad (7.23\text{ข})$$

สมการไฟไนต์เอลิเมนต์ในสมการ (7.22) สำหรับการวิเคราะห์ปัญหาการถ่ายเทความร้อนและปัญหาความแข็งแรงของโครงสร้าง มีลักษณะดังนี้

$$[M] \{\bar{q}^{FB}\} = -k [G] \{\bar{T}\} \quad \text{สำหรับปัญหาการถ่ายเทความร้อน} \quad (7.24\text{ก})$$

$$[M] \{\bar{\sigma}^{FB}\} = [G] \{\delta\} \quad \text{สำหรับปัญหาความแข็งแรงของโครงสร้าง} \quad (7.24\text{ข})$$

7.3.2 ค่าแอลทูนอร์ม $\|e_g^{FE}\|_{L_2}$ สำหรับความคลาดเคลื่อนโดยประมาณ

ค่ากำลังสองของแอลทูนอร์มสำหรับความคลาดเคลื่อนโดยประมาณที่คำนวณจากผลต่างของฟังก์ชันดังสมการ (7.19) จะอยู่ในรูปสมการดังนี้

$$\|e_g^{FE}\|_{L_2}^2 = \int_{\Omega} (g^{FB} - g^{ST})^2 d\Omega \quad (7.25)$$

แทนลักษณะการกระจายของฟังก์ชันสำหรับเอลิเมนต์เข้ากับสมการ (7.25) จะได้สมการไฟไนต์เอลิเมนต์สำหรับค่ากำลังสองของแอลทูนอร์มสำหรับความคลาดเคลื่อนโดยประมาณดังนี้

$$\|e_g^{FE}\|_{L_2}^2 = \int_{\Omega} \left([\bar{N}] \{\bar{g}^{FB}\} - \left(-c \left[\frac{\partial N}{\partial x_i} \right] \{U\} \right) \right)^2 d\Omega$$

$$\begin{aligned}\|e_g^{FE}\|_{L_2}^2 &= \int_{\Omega} \left([\bar{N}] \{\bar{g}^{FB}\} + c \left[\frac{\partial N}{\partial x_i} \right] \{U\} \right)^2 d\Omega \\ \|e_g^{FE}\|_{L_2}^2 &= \int_{\Omega} \left([\bar{N}] \{\bar{g}^{FB}\} + c \left[\frac{\partial N}{\partial x_i} \right] \{U\} \right)^T \left([\bar{N}] \{\bar{g}^{FB}\} + c \left[\frac{\partial N}{\partial x_i} \right] \{U\} \right) d\Omega \\ \|e_g^{FE}\|_{L_2}^2 &= [\bar{g}^{FB}] [M] \{\bar{g}^{FB}\} + 2c [\bar{g}^{FB}] [G] \{U\} + c^2 [U] [K] \{U\} \quad (7.26)\end{aligned}$$

โดย $[K] = \int_{\Omega} \left\{ \frac{\partial N}{\partial x_i} \right\} \left[\frac{\partial N}{\partial x_i} \right] d\Omega \quad (7.27)$

ดังนั้นค่ากำลังสองของแอลทูนอร์มสำหรับความคลาดเคลื่อนโดยประมาณในสมการ (7.26) สำหรับการวิเคราะห์ปัญหาการถ่ายเทความร้อนและปัญหาความแข็งแรงของโครงสร้าง มีลักษณะดังนี้

สำหรับปัญหาการถ่ายเทความร้อน

$$\|e_g^{FE}\|_{L_2}^2 = [\bar{q}^{FB}] [M] \{\bar{q}^{FB}\} + 2k [\bar{q}^{FB}] [G] \{\bar{T}\} + k^2 [\bar{T}] [K] \{\bar{T}\} \quad (7.28ก)$$

สำหรับปัญหาความแข็งแรงของโครงสร้าง

$$\|e_g^{FE}\|_{L_2}^2 = [\bar{\sigma}^{FB}] [M] \{\bar{\sigma}^{FB}\} - 2[\bar{\sigma}^{FB}] [G] \{\delta\} + [\delta] [K] \{\delta\} \quad (7.28ข)$$

ค่าแอลทูนอร์มสำหรับความคลาดเคลื่อนโดยประมาณจึงคำนวณได้จาก

$$\|e_g^{FE}\|_g = \left[\sum_{element} \|e_g^{FE}\|_{L_2}^2 / A_d \right]^{1/2} \quad (7.29)$$

โดย A_d แทน พื้นที่โดเมนรวม

7.3.3 ค่าแอลทูนอร์มสำหรับความคลาดเคลื่อนแม่นยำ

ในกรณีที่มีปัญหานั้น ๆ มีผลเฉลยแม่นยำ ค่าความคลาดเคลื่อนก็สามารถพิจารณาได้จากผลต่างระหว่างผลเฉลยแม่นยำและผลเฉลยโดยประมาณ ทั้งจากระเบียบวิธีไฟไนต์เอลิเมนต์แบบดั้งเดิมและระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์ นั่นคือ

$$e_g^{FB} = g^{EX} - g^{FB} \quad (7.30ก)$$

และ $e_g^{ST} = g^{EX} - g^{ST} \quad (7.30ข)$

- โดย e_g^{FB} แทน ความคลาดเคลื่อนที่คำนวณจากผลต่างระหว่างผลเฉลยแม่นยำและผลเฉลยโดยประมาณจากระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์
- e_g^{ST} แทน ความคลาดเคลื่อนที่คำนวณจากผลต่างระหว่างผลเฉลยแม่นยำและผลเฉลยโดยประมาณจากระเบียบวิธีไฟไนต์เอลิเมนต์แบบดั้งเดิม
- g^{EX} แทน ฟลักซ์ซึ่งคำนวณจากผลเฉลยแม่นยำ

ค่าแอสทอนอร์มสำหรับความคลาดเคลื่อนที่คำนวณจากผลต่างระหว่างผลเฉลยแม่นยำและผลเฉลยโดยประมาณทั้งสองวิธีสามารถคำนวณได้เช่นเดียวกับขั้นตอนในหัวข้อ 7.3.2 โดยเริ่มจากการแทนค่าสมการ (7.30ก) และ (7.30ข) ลงในพจน์ทางขวามือของสมการ (7.25) จะได้

$$\|e_g^{FB}\|_{L_2}^2 = \int_{\Omega} (g^{EX} - g^{FB})^2 d\Omega \quad (7.31ก)$$

และ $\|e_g^{ST}\|_{L_2}^2 = \int_{\Omega} (g^{EX} - g^{ST})^2 d\Omega \quad (7.31ข)$

โดย $\|e_g^{FB}\|_{L_2}^2$ แทน ค่ากำลังสองของแอสทอนอร์มสำหรับความคลาดเคลื่อนที่คำนวณจากผลต่างระหว่างผลเฉลยแม่นยำและผลเฉลยโดยประมาณจากระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์

$\|e_g^{ST}\|_{L_2}^2$ แทน ค่ากำลังสองของแอสทอนอร์มสำหรับความคลาดเคลื่อนที่คำนวณจากผลต่างระหว่างผลเฉลยแม่นยำและผลเฉลยโดยประมาณจากระเบียบวิธีไฟไนต์เอลิเมนต์แบบดั้งเดิม

จากนั้นจึงแทนลักษณะการกระจายของฟลักซ์สำหรับเอลิเมนต์เข้ากับสมการ (7.31ก) และ (7.31ข) จะได้สมการไฟไนต์เอลิเมนต์สำหรับค่ากำลังสองของแอสทอนอร์มสำหรับความคลาดเคลื่อนทั้งสองดังนี้

$$\|e_g^{FB}\|_{L_2}^2 = [\bar{g}^{EX}] [M] \{\bar{g}^{EX}\} - 2[\bar{g}^{EX}] [M] \{\bar{g}^{FB}\} + [\bar{g}^{FB}] [M] \{\bar{g}^{FB}\} \quad (7.32ก)$$

$$\|e_g^{ST}\|_{L_2}^2 = [\bar{g}^{EX}] [M] \{\bar{g}^{EX}\} + 2c[\bar{g}^{EX}] [G] \{U\} + c^2 [U] [K] \{U\} \quad (7.32ข)$$

ดังนั้นค่ากำลังสองของแอสทอนอร์มสำหรับความคลาดเคลื่อนที่คำนวณจากผลต่างระหว่างผลเฉลยแม่นยำและผลเฉลยโดยประมาณในสมการ (7.32ก) และ (7.32ข) สำหรับการวิเคราะห์ปัญหาการถ่ายเทความร้อนและปัญหาความแข็งแรงของโครงสร้างมีลักษณะดังนี้

สำหรับปัญหาการถ่ายเทความร้อน

$$\|e_g^{FB}\|_{L_2}^2 = [\bar{q}^{EX}] [M] \{\bar{q}^{EX}\} - 2[\bar{q}^{EX}] [M] \{\bar{q}^{FB}\} + [\bar{q}^{FB}] [M] \{\bar{q}^{FB}\} \quad (7.33ก)$$

$$\|e_g^{ST}\|_{L_2}^2 = [\bar{q}^{EX}] [M] \{\bar{q}^{EX}\} + 2k[\bar{q}^{EX}] [G] \{\bar{T}\} + k^2 [\bar{T}] [K] \{\bar{T}\} \quad (7.33ข)$$

สำหรับปัญหาความแข็งแรงของโครงสร้าง

$$\|e_g^{FB}\|_{L_2}^2 = [\bar{\sigma}^{EX}] [M] \{\bar{\sigma}^{EX}\} - 2[\bar{\sigma}^{EX}] [M] \{\bar{\sigma}^{FB}\} + [\bar{\sigma}^{FB}] [M] \{\bar{\sigma}^{FB}\} \quad (7.34ก)$$

$$\|e_g^{ST}\|_{L_2}^2 = [\bar{\sigma}^{EX}] [M] \{\bar{\sigma}^{EX}\} - 2[\bar{\sigma}^{EX}] [G] \{\delta\} + [\delta] [K] \{\delta\} \quad (7.34ข)$$

และค่าแอดทูนอร์มสำหรับความคลาดเคลื่อนที่คำนวณจากผลต่างระหว่างผลเฉลยแม่นยำและผลเฉลยโดยประมาณจากทั้งสองระเบียบวิธี จะคำนวณได้จาก

$$\|e_g^{FB}\|_g = \left[\sum_{element} \|e_g^{FB}\|_{L_2}^2 / A_d \right]^{1/2} \quad (7.35ก)$$

และ

$$\|e_g^{ST}\|_g = \left[\sum_{element} \|e_g^{ST}\|_{L_2}^2 / A_d \right]^{1/2} \quad (7.35ข)$$

7.4 โปรแกรมสำหรับการคะแนนความคลาดเคลื่อนที่คำนวณจากผลต่างของผลลัพธ์

ในการประยุกต์การคะแนนความคลาดเคลื่อนที่คำนวณจากผลต่างของผลลัพธ์เข้ากับการวิเคราะห์ปัญหาต่าง ๆ นั้น สามารถกระทำได้อย่างสะดวกโดยเริ่มจากการนำโปรแกรมวิเคราะห์ปัญหาต่าง ๆ ที่อธิบายอย่างละเอียดในบทที่ 4 และทำการเรียกโปรแกรมย่อย [ERR_EST] ซึ่งจะอยู่ก่อนโปรแกรมย่อยสุดท้าย [WRITE_OUTPUT] โดยโปรแกรมย่อย [ERR_EST] จะทำการคำนวณความคลาดเคลื่อนโดยประมาณที่คำนวณจากผลต่างของผลลัพธ์และค่าแอดทูนอร์มสำหรับความคลาดเคลื่อนโดยประมาณ หากปัญหานั้น ๆ มีผลเฉลยแม่นยำ โปรแกรมย่อย [ERR_EST] ก็จะทำให้การคำนวณความคลาดเคลื่อนก็สามารถพิจารณาได้จากผลต่างระหว่างผลเฉลยแม่นยำและผลเฉลยโดยประมาณ ทั้งจากระเบียบวิธีไฟไนต์เอลิเมนต์แบบดั้งเดิมและระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของผลลัพธ์ และค่าแอดทูนอร์มสำหรับความคลาดเคลื่อนที่คำนวณจากผลต่างระหว่างผลเฉลยแม่นยำและผลเฉลยโดยประมาณจากทั้งสองระเบียบวิธีด้วยเช่นกัน

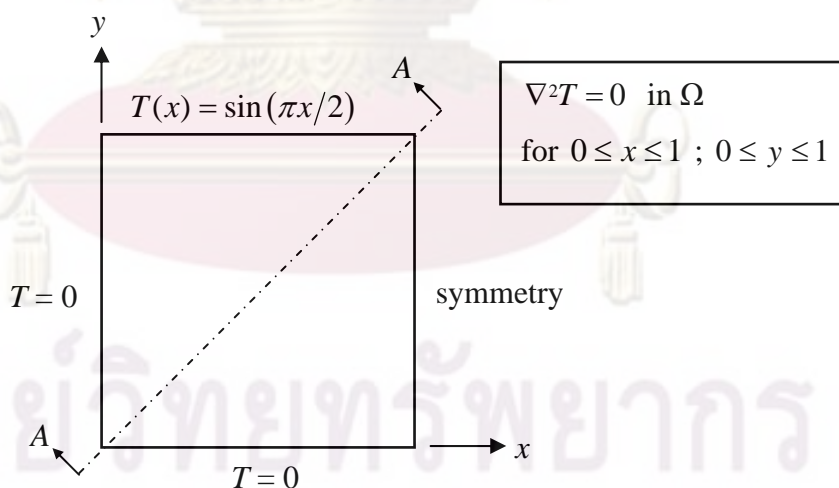
7.5 การทดสอบโปรแกรมย่อยสำหรับการคะแนนความคลาดเคลื่อนที่คำนวณจากผลต่างของฟังก์ชัน

ในหัวข้อนี้จะทดสอบโปรแกรมย่อยสำหรับการคะแนนความคลาดเคลื่อนที่คำนวณจากผลต่างของฟังก์ชัน โดยจะทำการทดสอบด้วยปัญหาที่มีผลเฉลยแน่นอนตรง ผลเฉลยเชิงวิเคราะห์ และปัญหาที่มีความซับซ้อน ดังนี้

7.5.1 ปัญหาแผ่นสี่เหลี่ยมผืนผ้าซึ่งถูกกำหนดอุณหภูมิที่ขอบ

ตัวอย่างแรกเป็นปัญหาแผ่นสี่เหลี่ยมผืนผ้าซึ่งถูกกำหนดอุณหภูมิที่ขอบ โดยแผ่นโลหะสี่เหลี่ยมผืนผ้ามีขนาด 2×1 หน่วย หนา 0.1 หน่วย และกำหนดให้สัมประสิทธิ์การนำความร้อน k มีค่าเท่ากับ 1 และอุณหภูมิตลอดขอบด้านซ้าย ขอบด้านขวา และขอบด้านล่าง $T = 0$ ส่วนอุณหภูมิตลอดขอบด้านบนแปรผันกับระยะในรูปแบบของ $\sin(\pi x/2)$ แต่เนื่องจากลักษณะของปัญหาที่มีความสมมาตร ดังนั้นสามารถใช้ขอบเขตในการคำนวณเพียงครึ่งซ้ายของแผ่นโลหะดังแสดงในรูปที่ 7.4 ซึ่งปัญหาดังกล่าวมีผลเฉลยแน่นอนตรงของการกระจายอุณหภูมิตนบนแผ่นโลหะ [26] คือ

$$T(x, y) = \frac{\sin(\pi x/2) \sinh(\pi y/2)}{\sinh(\pi/2)} \quad (7.36)$$

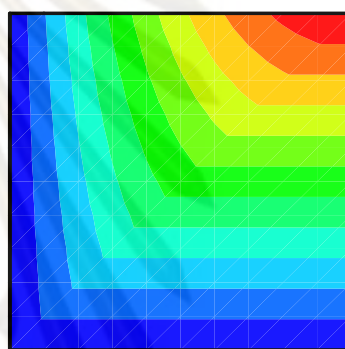


รูปที่ 7.4 ปัญหาแผ่นสี่เหลี่ยมผืนผ้าซึ่งถูกกำหนดอุณหภูมิที่ขอบ

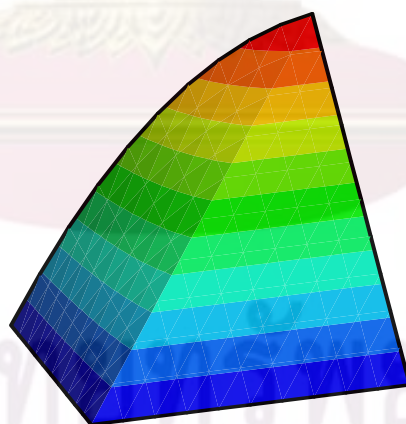
ในการวิเคราะห์ปัญหา จะเริ่มต้นจากการสร้างรูปแบบจำลองไฟไนต์เอลิเมนต์ (T3NL-M1) ซึ่งได้แสดงไว้ในรูปที่ 7.5ก โดยประกอบไปด้วย 4 จุดต่อ และ 2 เอลิเมนต์ จากนั้นทำการประยุกต์เงื่อนไขขอบเขตตามที่กล่าวข้างต้น จากนั้นทำการวิเคราะห์ปัญหาโดยใช้รูปแบบจำลองไฟไนต์เอลิเมนต์ในรูปที่ 7.5ก ซึ่งผลการกระจายตัวของอุณหภูมิโดยใช้แบบจำลองดังกล่าวได้แสดงในรูปที่ 7.5ข แล้วจึงนำผลการกระจายตัวของอุณหภูมิมาพล็อตอีกครั้งในรูปแบบสามมิติดังแสดงในรูปที่ 7.5ค



(ก)



(ข)



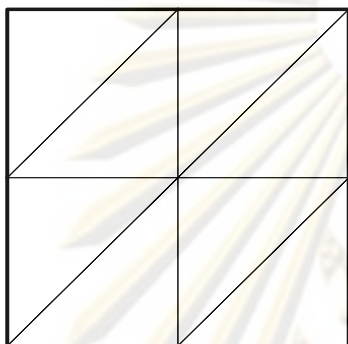
(ค)

รูปที่ 7.5 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M1

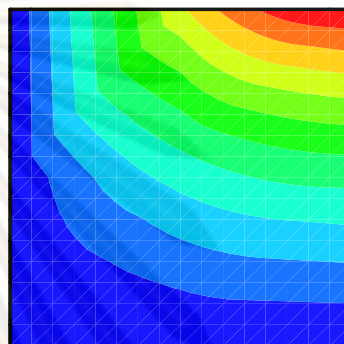
ข) การกระจายตัวของอุณหภูมิในรูปแบบสองมิติสำหรับแบบจำลอง T3NL-M1

ค) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลอง T3NL-M1

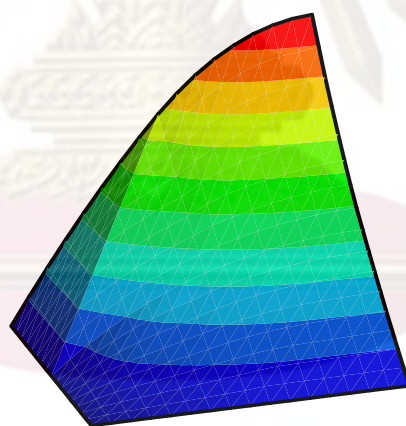
จากนั้นนำผลลัพธ์ที่ได้ไปทำการปรับขนาดเอลิเมนต์ โดยแบบจำลองไฟไนต์เอลิเมนต์ของการปรับขนาดเอลิเมนต์ครั้งที่ 1 นี้ (T3NL-M2) แสดงในรูปที่ 7.6 ซึ่งประกอบไปด้วย 9 จุดต่อ 8 เอลิเมนต์ และผลลัพธ์จากรูปแบบจำลองนี้ได้แสดงในรูปแบบสองมิติและสามมิติดังแสดงในรูปที่ 7.6ข และรูปที่ 7.6ค ตามลำดับ



(ก)



(ข)



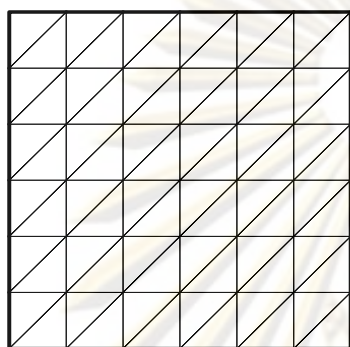
(ค)

รูปที่ 7.6 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M2

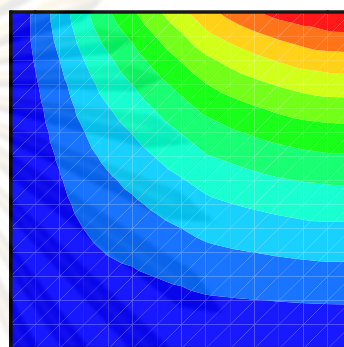
ข) การกระจายตัวของอุณหภูมิในรูปแบบสองมิติสำหรับแบบจำลอง T3NL-M2

ค) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลอง T3NL-M2

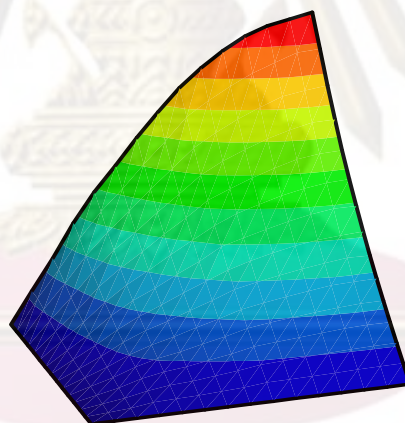
และเมื่อทำการปรับขนาดเอลิเมนต์ครั้งที่ 2 จะได้รูปแบบจำลองไฟไนต์เอลิเมนต์ (T3NL-M3) ซึ่งประกอบไปด้วย 49 จุดต่อ 72 เอลิเมนต์ ดังแสดงในรูปที่ 7.7ก ผลลัพธ์จากรูปแบบจำลองนี้มีความถูกต้องมากยิ่งขึ้นดังแสดงในแบบสองมิติและสามมิติในรูปที่ 7.7ข และรูปที่ 7.7ค



(ก)



(ข)



(ค)

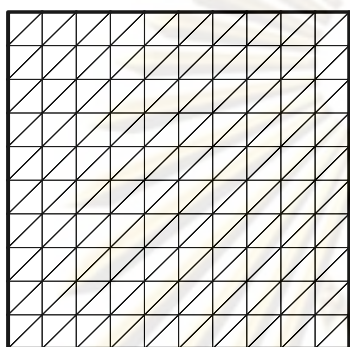
รูปที่ 7.7 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M3

ข) การกระจายตัวของอุณหภูมิในรูปแบบสองมิติสำหรับแบบจำลอง T3NL-M3

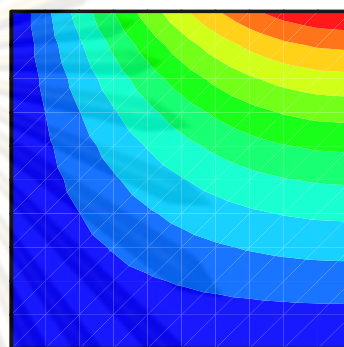
ค) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลอง T3NL-M3

จุฬาลงกรณ์มหาวิทยาลัย

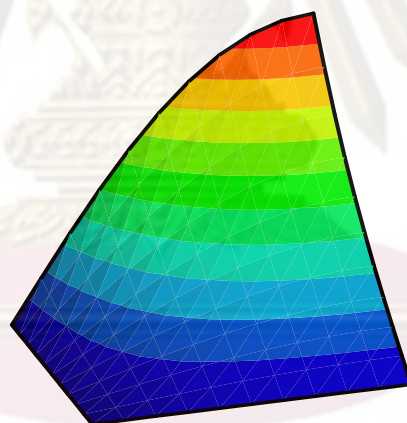
จากนั้นจึงปรับขนาดเอลิเมนต์ครั้งสุดท้ายทำให้ได้รูปแบบจำลองไฟไนต์เอลิเมนต์ (T3NL-M4) ซึ่งประกอบไปด้วย 121 จุดต่อ 200 เอลิเมนต์ ดังแสดงในรูปที่ 7.8ก และ จากผลลัพธ์ที่ได้แสดงในรูปแบบสองมิติและสามมิติในรูปที่ 7.8ข และรูปที่ 7.8ค พบว่ามีความถูกต้องอย่างมากเมื่อเทียบกับการใช้เอลิเมนต์เริ่มต้น



(ก)



(ข)



(ค)

รูปที่ 7.8 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M4

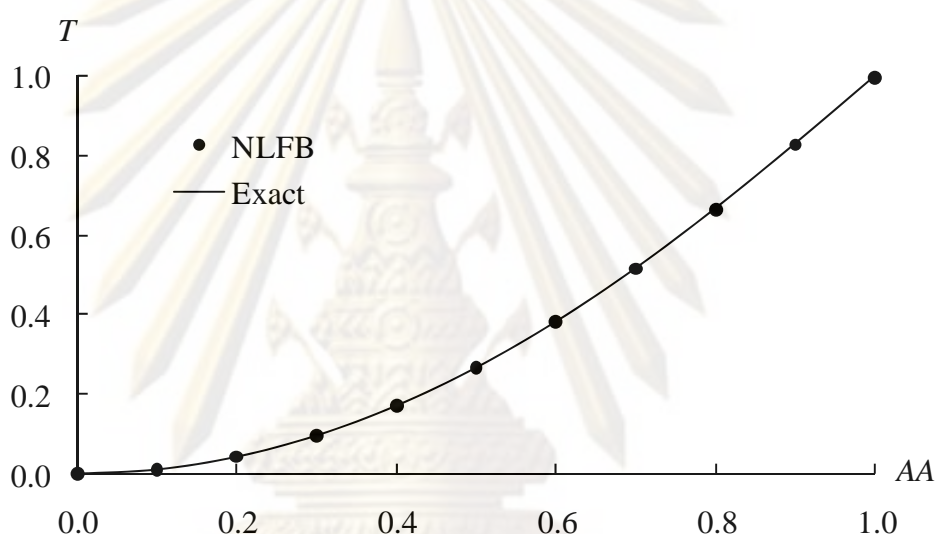
ข) การกระจายตัวของอุณหภูมิในรูปแบบสองมิติสำหรับแบบจำลอง T3NL-M4

ค) การกระจายตัวของอุณหภูมิในรูปแบบสามมิติสำหรับแบบจำลอง T3NL-M4

รูปที่ 7.9 แสดงผลการเปรียบเทียบค่าอุณหภูมิซึ่งได้จากผลเฉลยแม่นยำและได้จากโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้น (NLFB) ในแนวแกน AA ซึ่งจะเห็นได้ว่าผลลัพธ์ที่ได้หลังจากทำการปรับขนาดเอลิเมนต์แล้วมีค่าใกล้เคียงกับผลเฉลยแม่นยำมาก จากนั้นจึงเปรียบเทียบค่าเอลูทอนอร์มสำหรับความคลาดเคลื่อนและค่าเปอร์เซ็นต์ความคลาดเคลื่อน (ค่าใน

วงเล็บ) ระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งสี่แบบดังแสดงในตารางที่ 7.1 โดยเริ่มจากความคลาดเคลื่อนแม่นยำตรงจากระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์ ความคลาดเคลื่อนแม่นยำตรงจากระเบียบวิธีไฟไนต์เอลิเมนต์แบบดั้งเดิม และความคลาดเคลื่อนโดยประมาณจากระเบียบวิธีไฟไนต์เอลิเมนต์ทั้งสองแบบ ตามลำดับ

จากตารางที่ 7.1 พบว่าค่าความคลาดเคลื่อนโดยประมาณที่น่าเสนอนั้น สอดคล้องกับค่าความคลาดเคลื่อนแม่นยำตรงทั้งสองแบบ และรวมไปถึงค่าแอลทูนอร์มสำหรับความคลาดเคลื่อนเช่นเดียวกัน



รูปที่ 7.9 ผลการเปรียบเทียบค่าคุณภูมิซึ่งได้จากผลเฉลยแม่นยำตรงและจากโปรแกรม NLFB ในแนวแกน AA

ตารางที่ 7.1 ตารางแสดงการเปรียบเทียบค่าแอลทูนอร์มสำหรับความคลาดเคลื่อนและค่าเปอร์เซ็นต์ความคลาดเคลื่อนระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งสี่แบบ

Element	$\ e_g^{FB}\ _g, (\eta_{FB})$	$\ e_g^{ST}\ _g, (\eta_{ST})$	$\ e_g^{FE}\ _g, (\eta_{FE})$
T3NL-M1	0.455 (33.28%)	0.527 (36.66%)	0.267 (20.75%)
T3NL-M2	0.182 (16.44%)	0.242 (20.77%)	0.160 (14.55%)
T3NL-M3	0.027 (2.83%)	0.038 (3.90%)	0.026 (2.75%)
T3NL-M4	0.010 (1.02%)	0.013 (1.41%)	0.009 (0.98%)

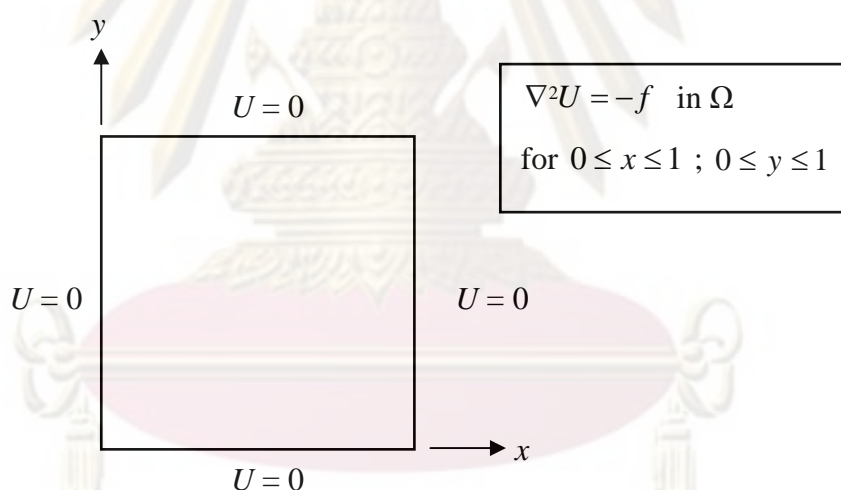
7.5.2 ปัญหาแผ่นสี่เหลี่ยมจัตุรัสพร้อมแหล่งกำเนิด

ตัวอย่างนี้เริ่มต้นด้วยการแก้สมการปัวซองพร้อมแหล่งกำเนิดเจาะจง โดยแผ่นโลหะสี่เหลี่ยมจัตุรัสมีขนาด 1×1 หน่วย หนา 0.1 หน่วย และกำหนดตัวแปรปฐุมภูมิตลอดขอบทั้งสี่ด้านมีค่าเท่ากับศูนย์ $U = 0$ โดยแหล่งกำเนิดเจาะจงบนแผ่นโลหะมีค่า

$$f(x, y) = -14x(1-x)(1-2y) - 4y(1-y)(1-2x) + 2(1+2x+7y)[x(1-x) + y(1-y)] \quad (7.37)$$

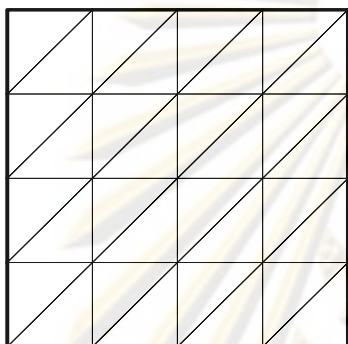
รายละเอียดและลักษณะของปัญหาแสดงในรูปที่ 7.10 ซึ่งปัญหานี้มีผลเฉลยแม่นยำตรงของการกระจายตัวแปรปฐุมภูมิตลอดขอบแผ่นโลหะ [27] ดังนี้

$$U(x, y) = x(1-x)y(1-y)(1+2x+7y) \quad (7.38)$$

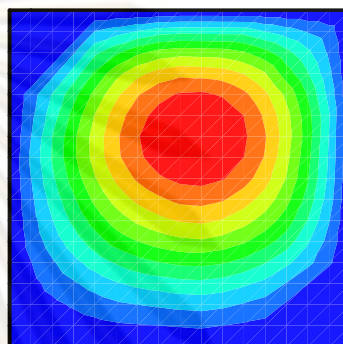


รูปที่ 7.10 ปัญหาแผ่นสี่เหลี่ยมจัตุรัสพร้อมแหล่งกำเนิด

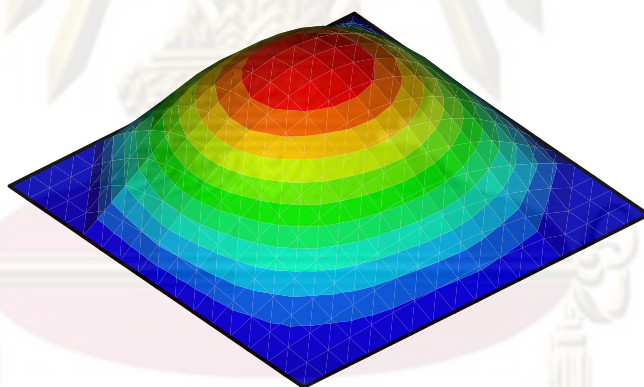
เริ่มต้นในการวิเคราะห์ปัญหาด้วยการสร้างรูปแบบจำลองไฟไนต์เอลิเมนต์ (T3NL-M1) ประกอบไปด้วย 25 จุดต่อ 32 เอลิเมนต์ ดังแสดงไว้ในรูปที่ 7.11ก จากนั้นทำการวิเคราะห์ปัญหาเพื่อหาผลลัพธ์ โดยผลการกระจายตัวของตัวแปรปฐมภูมิบนแผ่นโลหะแสดงไว้ในรูปที่ 7.11ข และนำผลการกระจายตัวที่ได้มาพล็อตในรูปแบบสามมิติดังแสดงในรูปที่ 7.11ค



(ก)



(ข)



(ค)

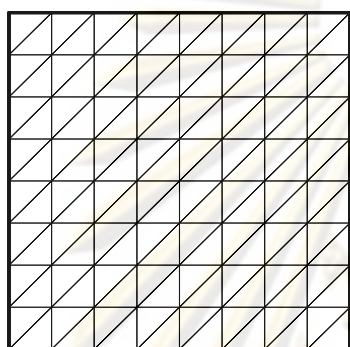
รูปที่ 7.11 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M1

ข) การกระจายตัวของตัวแปรปฐมภูมิในรูปแบบสองมิติของแบบจำลอง T3NL-M1

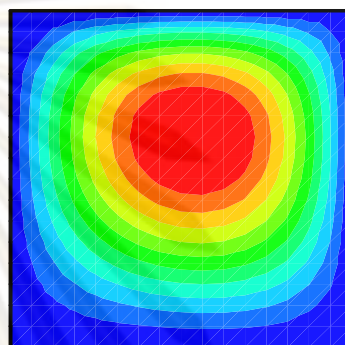
ค) การกระจายตัวของตัวแปรปฐมภูมิในรูปแบบสามมิติของแบบจำลอง T3NL-M1

จุฬาลงกรณ์มหาวิทยาลัย

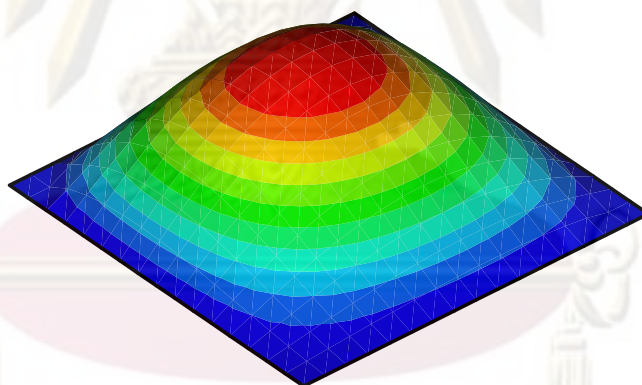
รูปที่ 7.12ก แสดงแบบจำลองไฟไนต์เอลิเมนต์ของการปรับขนาดเอลิเมนต์ครั้งที่ 1 นี้ (T3NL-M2) ซึ่งได้จากการนำผลลัพธ์ที่ได้มาปรับขนาดเอลิเมนต์ประกอบด้วย 81 จุดต่อ 128 เอลิเมนต์ รูปที่ 7.12ข และรูปที่ 7.12ค แสดงผลการกระจายตัวของตัวแปรปฏิสัมพันธ์บนแผ่นโลหะใน รูปแบบสองมิติและสามมิติตามลำดับ



(ก)



(ข)



(ค)

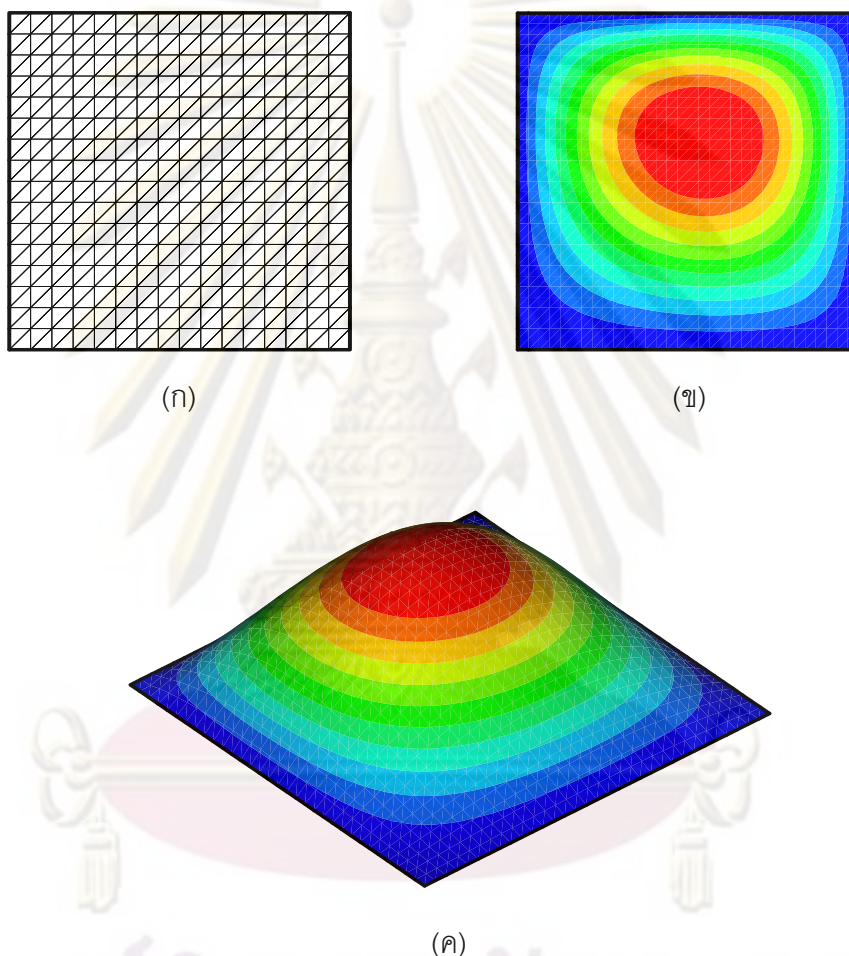
รูปที่ 7.12 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M2

ข) การกระจายตัวของตัวแปรปฏิสัมพันธ์ในรูปแบบสองมิติของแบบจำลอง T3NL-M2

ค) การกระจายตัวของตัวแปรปฏิสัมพันธ์ในรูปแบบสามมิติของแบบจำลอง T3NL-M2

จุฬาลงกรณ์มหาวิทยาลัย

และดำเนินการปรับขนาดเอลิเมนต์ซ้ำอีกครั้งจะได้แบบจำลองไฟไนต์เอลิเมนต์ของการปรับขนาดเอลิเมนต์ครั้งที่ 2 (T3NL-M3) ซึ่งประกอบไปด้วย 289 จุดต่อ 512 เอลิเมนต์ ดังแสดงไว้ในรูปที่ 7.13ก และผลการกระจายตัวของตัวแปรปฏิสัมพันธ์บนแผ่นโลหะในรูปแบบสองมิติและสามมิติได้ถูกแสดงไว้ในรูปที่ 7.13ข และรูปที่ 7.13ค ตามลำดับ



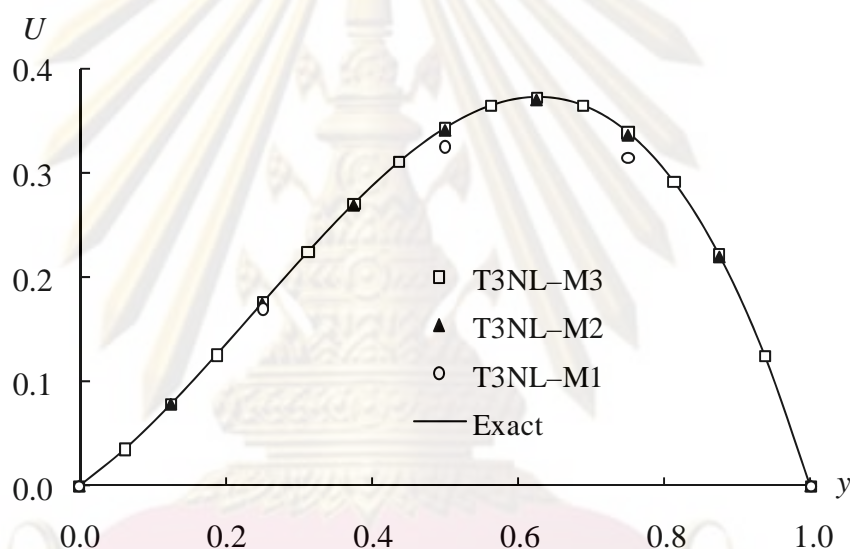
รูปที่ 7.13 ก) รูปแบบจำลองไฟไนต์เอลิเมนต์ T3NL-M3

ข) การกระจายตัวของตัวแปรปฏิสัมพันธ์ในรูปแบบสองมิติของแบบจำลอง T3NL-M3

ค) การกระจายตัวของตัวแปรปฏิสัมพันธ์ในรูปแบบสามมิติของแบบจำลอง T3NL-M3

จุฬาลงกรณ์มหาวิทยาลัย

ผลการเปรียบเทียบค่าตัวแปรปฐมภูมิซึ่งได้จากผลเฉลยแม่นยำและได้จากโปรแกรม NLFB ตลอดแนว $x = 0.5$ พบว่าภายหลังการปรับขนาดเอลิเมนต์เพียงสองครั้งผลลัพธ์ที่ได้มีค่าใกล้เคียงกับผลเฉลยแม่นยำเป็นอย่างมากดังแสดงในรูปที่ 7.14 และในตารางที่ 7.2 ได้แสดงการเปรียบเทียบค่าแวลูทูลูอรัมสำหรับความคลาดเคลื่อนและค่าเปอร์เซ็นต์ความคลาดเคลื่อน (ค่าในวงเล็บ) ระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งสามแบบ จากตารางดังกล่าวพบว่าค่าความคลาดเคลื่อนและค่าแวลูทูลูอรัมสำหรับความคลาดเคลื่อนโดยประมาณที่นำเสนอมีความสอดคล้องกับค่าความคลาดเคลื่อนและค่าแวลูทูลูอรัมสำหรับความคลาดเคลื่อนแม่นยำทั้งสองแบบ เช่นเดียวกับตัวอย่าง 7.5.1 ข้างต้น



รูปที่ 7.14 ผลการเปรียบเทียบค่าตัวแปรปฐมภูมิซึ่งได้จากผลเฉลยแม่นยำและจากโปรแกรม NLFB ตลอดแนว $x = 0.5$

ตารางที่ 7.2 ตารางแสดงการเปรียบเทียบค่าแวลูทูลูอรัมสำหรับความคลาดเคลื่อนและค่าเปอร์เซ็นต์ความคลาดเคลื่อนระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งสามแบบ

Element	$\ e_g^{FB}\ _g, (\eta_{FB})$	$\ e_g^{ST}\ _g, (\eta_{ST})$	$\ e_g^{FE}\ _g, (\eta_{FE})$
T3NL-M1	0.210 (19.98%)	0.283 (25.20%)	0.190 (19.22%)
T3NL-M2	0.065 (6.96%)	0.089 (9.29%)	0.061 (6.52%)
T3NL-M3	0.018 (1.99%)	0.025 (2.72%)	0.017 (1.88%)

7.5.3 ปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมภายใต้แรงดึงทิศทางเดียว

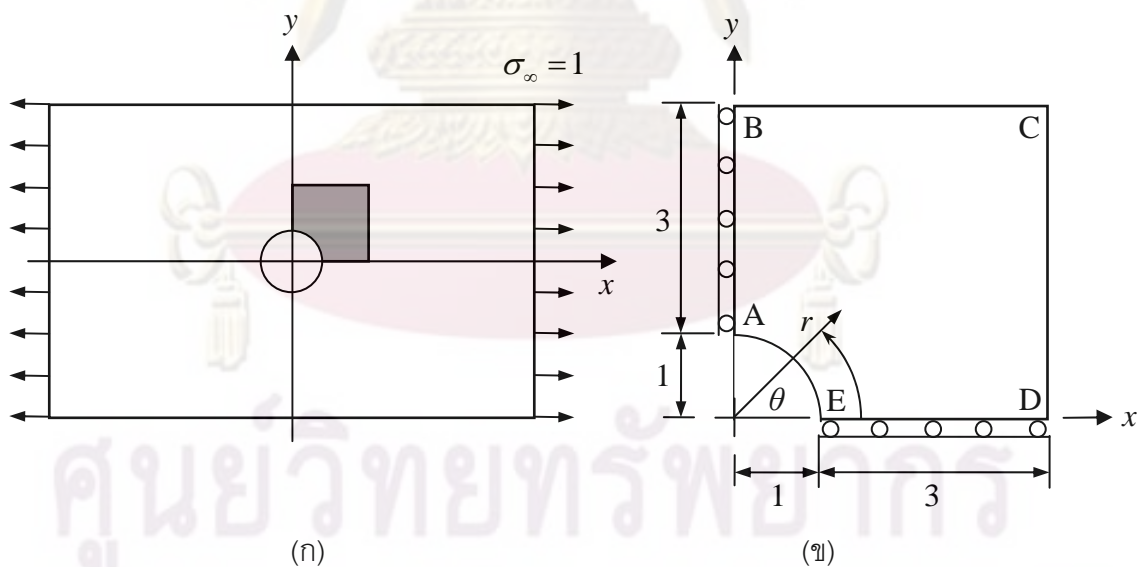
ตัวอย่างนี้เป็นแผ่นโลหะสี่เหลี่ยมผืนผ้าขนาด 30×20 in หนา 1 in มีรูกลมตรงกลางขนาดรัศมี $r = 1$ in ถูกดึงตลอดขอบทั้งสองด้านด้วย $\sigma_0 = 1$ psi ดังแสดงในรูปที่ 7.15ก แผ่นโลหะอยู่ภายใต้เงื่อนไขความเครียดในระนาบและมีคุณสมบัติทางกลดังนี้ ค่าโมดูลัสของความยืดหยุ่น $E = 1000$ psi และค่าอัตราส่วนปัวส์ซง $\nu = 0.3$ ซึ่งขอบเขตในการคำนวณของปัญหานี้แสดงอยู่ในรูปที่ 7.15ข โดยมีเงื่อนไขขอบเขตแบบสมมาตรตลอดขอบ AB และ ED ส่วนขอบ BC และ DC ของแผ่นโลหะนั้นถูกกำหนดแรงที่ขอบจากผลเฉลยเชิงวิเคราะห์ [28] ดังนี้

$$\sigma_x = 1 - \frac{a^2}{r^2} \left(\frac{3}{2} \cos 2\theta + \cos 4\theta \right) + \frac{3}{2} \frac{a^4}{r^4} \cos 4\theta \quad (7.39ก)$$

$$\sigma_y = -\frac{a^2}{r^2} \left(\frac{1}{2} \cos 2\theta - \cos 4\theta \right) - \frac{3}{2} \frac{a^4}{r^4} \cos 4\theta \quad (7.39ข)$$

$$\tau_{xy} = -\frac{a^2}{r^2} \left(\frac{1}{2} \sin 2\theta + \sin 4\theta \right) + \frac{3}{2} \frac{a^4}{r^4} \sin 4\theta \quad (7.39ค)$$

โดย a แทน รัศมีของรูกลม และ (r, θ) แทน พิกัดเชิงขั้ว (polar coordinates)



รูปที่ 7.15 ก) ปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมภายใต้แรงดึงทิศทางเดียว

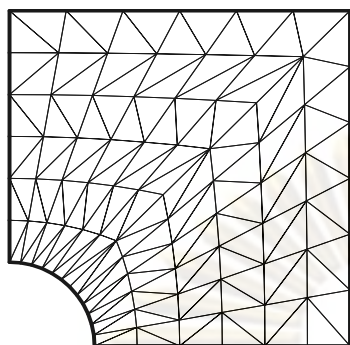
ข) ขอบเขตการคำนวณ

การคำนวณค่าการเคลื่อนตัวบนแผ่นสี่เหลี่ยมดังที่แสดงข้างต้นโดยใช้โปรแกรมที่ได้ประดิษฐ์ขึ้น จะเริ่มต้นจากการแบ่งขอบเขตของปัญหาออกเป็นเอลิเมนต์ย่อย ๆ ด้วยการสร้างรูปแบบจำลอง ไฟไนต์เอลิเมนต์ซึ่งประกอบไปด้วยรูปแบบจำลองอย่างเป็นระเบียบ (structured mesh) และรูปแบบจำลองอย่างไร้ระเบียบ (unstructured mesh) แตกต่างกันทั้งสิ้น 6 รูปแบบ ดังนี้

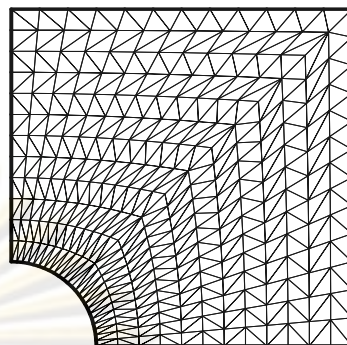
- 1) รูปแบบจำลองอย่างเป็นระเบียบ 1 (Mesh S1) ประกอบด้วยจำนวนจุดต่อ 91 จุดต่อและจำนวนเอลิเมนต์ 144 เอลิเมนต์
- 2) รูปแบบจำลองอย่างเป็นระเบียบ 2 (Mesh S2) ประกอบด้วยจำนวนจุดต่อ 325 จุดต่อและจำนวนเอลิเมนต์ 576 เอลิเมนต์
- 3) รูปแบบจำลองอย่างเป็นระเบียบ 3 (Mesh S3) ประกอบด้วยจำนวนจุดต่อ 1225 จุดต่อและจำนวนเอลิเมนต์ 2304 เอลิเมนต์
- 4) รูปแบบจำลองอย่างไร้ระเบียบ 1 (Mesh U1) ประกอบด้วยจำนวนจุดต่อ 51 จุดต่อและจำนวนเอลิเมนต์ 83 เอลิเมนต์
- 5) รูปแบบจำลองอย่างไร้ระเบียบ 2 (Mesh U2) ประกอบด้วยจำนวนจุดต่อ 180 จุดต่อและจำนวนเอลิเมนต์ 308 เอลิเมนต์
- 6) รูปแบบจำลองอย่างไร้ระเบียบ 3 (Mesh U3) ประกอบด้วยจำนวนจุดต่อ 180 จุดต่อและจำนวนเอลิเมนต์ 299 เอลิเมนต์

โดยรายละเอียดรูปแบบจำลองไฟไนต์เอลิเมนต์ข้างต้นแสดงในรูปที่ 7.16ก - 7.16ง

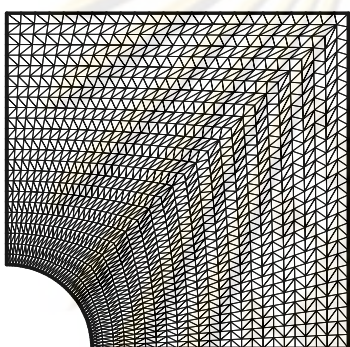
รูปที่ 7.17 – 7.19 แสดงการเปรียบเทียบผลเฉลยเชิงวิเคราะห์และผลเฉลยโดยประมาณของค่าความเค้นตั้งฉาก σ_x , σ_y และความเค้นเฉือน τ_{xy} ตลอดแนวความโค้งของ รูกลม จากรูปดังกล่าวพบว่าการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติเข้ากับ ระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์ ช่วยให้ผลเฉลยโดยประมาณมีความถูกต้องแม่นยำอย่างมากด้วยการใช้จำนวนจุดต่อที่ไม่ต่างจากแบบจำลองเริ่มต้น (Mesh U1) อีกทั้ง ลักษณะการกระจายตัวที่สม่ำเสมอของค่าความเค้นทั้งสามโดยเป็นผลมาจากระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ อันมีลักษณะแตกต่างจากระเบียบวิธีไฟไนต์เอลิเมนต์มาตรฐานที่ใช้ สามเหลี่ยม CST (Constant Strain Triangle) ซึ่งปกติแล้วจะเกิดการสั้นของการกระจายตัวของ ความเค้น [29]



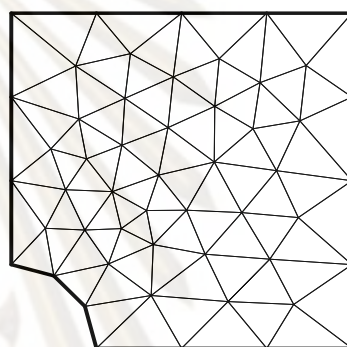
ก) Mesh S1



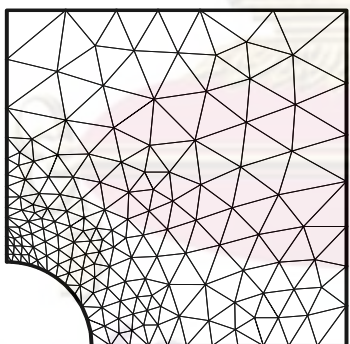
ข) Mesh S2



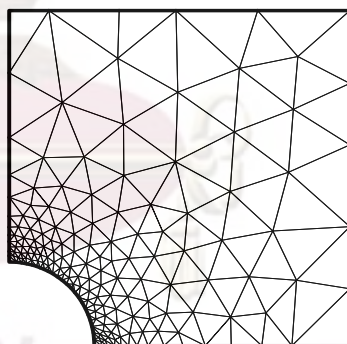
ค) Mesh S3



ง) Mesh U1



จ) Mesh U2

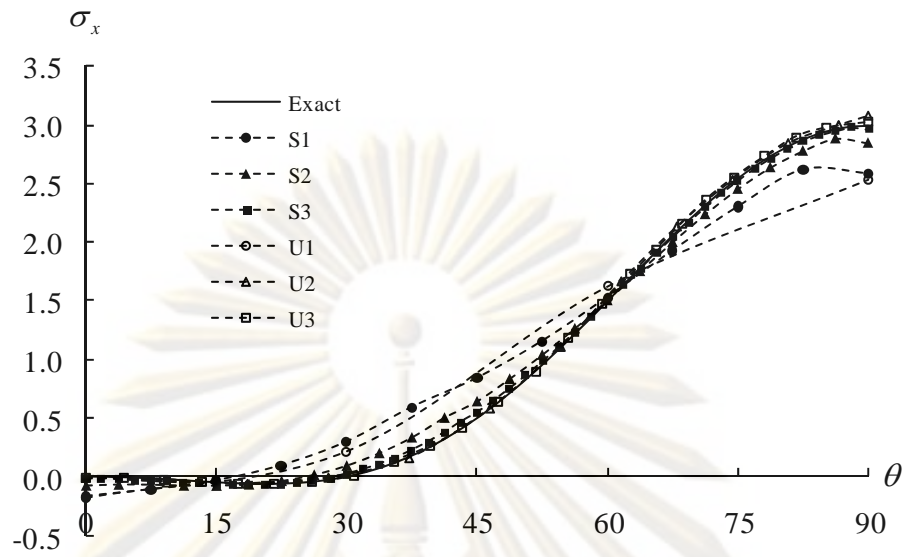


ฉ) Mesh U3

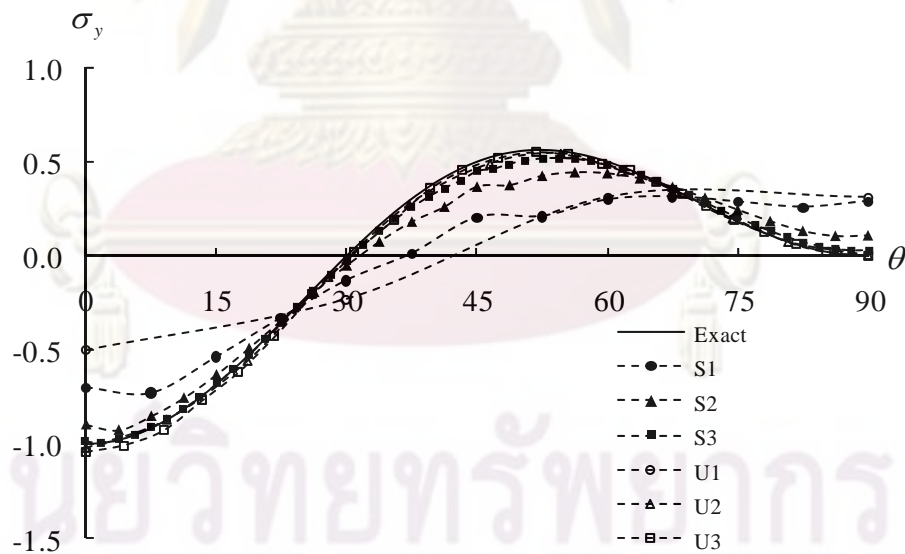
รูปที่ 7.16 รูปแบบจำลองไฟไนต์เอลิเมนต์อย่างเป็นระเบียบ ก) Mesh S1, ข) Mesh S2,

ค) Mesh S3 และรูปแบบจำลองไฟไนต์เอลิเมนต์อย่างไร้ระเบียบ ง) Mesh U1,

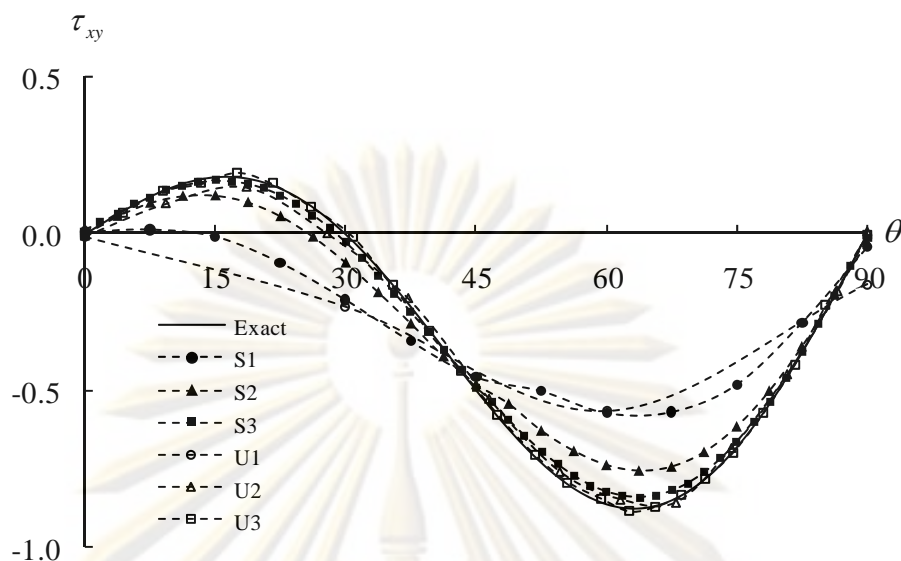
จ) Mesh U2, ฉ) Mesh U3



รูปที่ 7.17 ผลการเปรียบเทียบผลเฉลยเชิงวิเคราะห์และผลเฉลยโดยประมาณ
ของความเค้นตั้งฉาก σ_x



รูปที่ 7.18 ผลการเปรียบเทียบผลเฉลยเชิงวิเคราะห์และผลเฉลยโดยประมาณ
ของความเค้นตั้งฉาก σ_y



รูปที่ 7.19 ผลการเปรียบเทียบผลเฉลยเชิงวิเคราะห์และผลเฉลยโดยประมาณ
ของความเค้นเฉือน τ_{xy}

ตารางที่ 7.3 แสดงการเปรียบเทียบค่าแอลทูนอร์มสำหรับความคลาดเคลื่อนและค่าเปอร์เซ็นต์ความคลาดเคลื่อน (ค่าในวงเล็บ) ระหว่างรูปแบบจำลองอย่างเป็นระเบียบ และรูปแบบจำลองอย่างไร้ระเบียบรวมทั้งสิ้นหกแบบจำลอง โดยผลจากรายการแสดงให้เห็นว่าค่าความคลาดเคลื่อนที่เกิดขึ้นในรูปแบบจำลองอย่างเป็นระเบียบนั้นมีค่าลดลงเมื่อจำนวนเอลิเมนต์เพิ่มขึ้น แต่สำหรับแบบจำลองอย่างไร้ระเบียบมีค่าความคลาดเคลื่อนลดลงโดยใช้จำนวนเอลิเมนต์ใกล้เคียงกันอันเป็นผลมาจากเทคนิคการปรับขนาดเอลิเมนต์ รวมไปถึงค่าความคลาดเคลื่อนและค่าแอลทูนอร์มโดยประมาณที่น่าเสถียรมีความสอดคล้องไปในทางเดียวกันกับค่าความคลาดเคลื่อนและค่าแอลทูนอร์มแน่นอนตรง จากตัวอย่างนี้จึงชี้ให้เห็นว่าค่าความคลาดเคลื่อนและค่าแอลทูนอร์มโดยประมาณที่น่าเสถียรสามารถประยุกต์ใช้กับเทคนิคการปรับขนาดเอลิเมนต์ได้เป็นอย่างดี

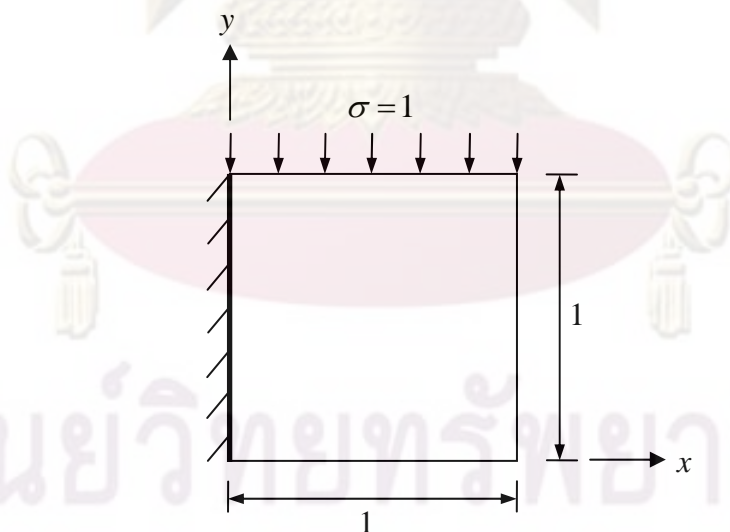
ตารางที่ 7.3 ตารางแสดงการเปรียบเทียบค่าแอลทูนอร์มสำหรับความคลาดเคลื่อนและค่าเปอร์เซ็นต์ความคลาดเคลื่อนระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งหกแบบของปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมภายใต้แรงดึงทิศทางเดียว

Element	$\ e_g^{FB}\ _g, (\eta_{FB})$	$\ e_g^{ST}\ _g, (\eta_{ST})$	$\ e_g^{FE}\ _g, (\eta_{FE})$
Mesh S1	0.068 (6.07%)	0.083 (7.27%)	0.047 (4.29%)

Element	$\ e_g^{FB}\ _g, (\eta_{FB})$	$\ e_g^{ST}\ _g, (\eta_{ST})$	$\ e_g^{FE}\ _g, (\eta_{FE})$
Mesh S2	0.026 (2.39%)	0.031 (2.84%)	0.017 (1.58%)
Mesh S3	0.016 (1.53%)	0.017 (1.59%)	0.005 (0.44%)
Mesh U1	0.076 (6.69%)	0.094 (8.17%)	0.056 (5.09%)
Mesh U2	0.018 (1.69%)	0.019 (1.79%)	0.006 (0.60%)
Mesh U3	0.020 (1.93%)	0.021 (2.01%)	0.006 (0.59%)

7.5.4 ปัญหาคานปลายยื่นชนิดสั้น (short cantilever beam) ภายใต้แรงกด

ตัวอย่างนี้เป็นปัญหาคานปลายยื่นชนิดสั้นซึ่งมีขนาด 1×1 หน่วย อยู่ภายใต้แรงกดขนาด $\sigma = 1$ [29] โดยตลอดขอบด้านซ้ายของคานยึดแน่นดังแสดงอยู่ในรูปที่ 7.20 ปัญหาคานปลายยื่นชนิดสั้นถูกสมมุติให้อยู่ภายใต้เงื่อนไขความเครียดในระนาบและมีคุณสมบัติทางกลดังนี้ ค่าโมดูลัสของความยืดหยุ่น $E = 10,000$ psi และค่าอัตราส่วนปัวส์ซง $\nu = 0.3$



รูปที่ 7.20 ปัญหาคานปลายยื่นชนิดสั้นภายใต้แรงกด

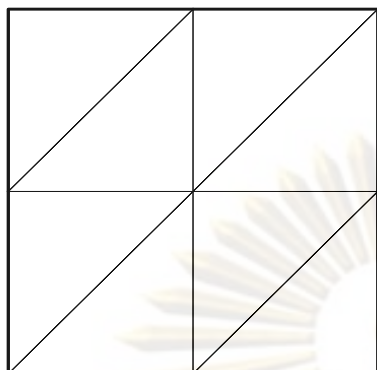
รูปแบบจำลองไฟไนต์เอลิเมนต์สำหรับปัญหาคานปลายยื่นชนิดสั้นภายใต้แรงกด ประกอบด้วย รูปแบบจำลองอย่างเป็นระเบียบจำนวน 5 รูปแบบ (S1 ถึง S5) และรูปแบบจำลองอย่างไร้ระเบียบจำนวน 3 รูปแบบ (U1 ถึง U3) โดยรูปแบบจำลองเริ่มต้น (U1) เป็นรูปแบบ

เดียวกันกับรูปแบบจำลองอย่างเป็นทางการเป็นระเบียบ S1 และรูปแบบจำลองจากการปรับขนาดเอลิเมนต์อีก 2 รูปแบบ (U2 และ U3) รายละเอียดของรูปแบบจำลองทั้ง 8 รูปแบบมีดังนี้

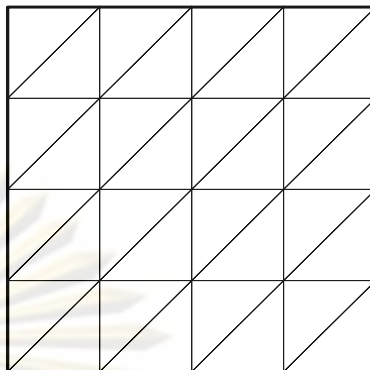
- 1) รูปแบบจำลองอย่างเป็นทางการเป็นระเบียบ 1 (Mesh S1) ประกอบด้วยจำนวนจุดต่อ 9 จุดต่อและจำนวนเอลิเมนต์ 8 เอลิเมนต์
- 2) รูปแบบจำลองอย่างเป็นทางการเป็นระเบียบ 2 (Mesh S2) ประกอบด้วยจำนวนจุดต่อ 25 จุดต่อและจำนวนเอลิเมนต์ 32 เอลิเมนต์
- 3) รูปแบบจำลองอย่างเป็นทางการเป็นระเบียบ 3 (Mesh S3) ประกอบด้วยจำนวนจุดต่อ 81 จุดต่อและจำนวนเอลิเมนต์ 128 เอลิเมนต์
- 4) รูปแบบจำลองอย่างเป็นทางการเป็นระเบียบ 4 (Mesh S4) ประกอบด้วยจำนวนจุดต่อ 289 จุดต่อและจำนวนเอลิเมนต์ 512 เอลิเมนต์
- 5) รูปแบบจำลองอย่างเป็นทางการเป็นระเบียบ 5 (Mesh S5) ประกอบด้วยจำนวนจุดต่อ 1089 จุดต่อและจำนวนเอลิเมนต์ 2048 เอลิเมนต์
- 6) รูปแบบจำลองอย่างไร้ระเบียบ 1 (Mesh U1) ประกอบด้วยจำนวนจุดต่อ 9 จุดต่อและจำนวนเอลิเมนต์ 8 เอลิเมนต์
- 7) รูปแบบจำลองอย่างไร้ระเบียบ 2 (Mesh U2) ประกอบด้วยจำนวนจุดต่อ 287 จุดต่อและจำนวนเอลิเมนต์ 526 เอลิเมนต์
- 8) รูปแบบจำลองอย่างไร้ระเบียบ 3 (Mesh U3) ประกอบด้วยจำนวนจุดต่อ 219 จุดต่อและจำนวนเอลิเมนต์ 384 เอลิเมนต์

โดยรายละเอียดรูปแบบจำลองไฟไนต์เอลิเมนต์ข้างต้นแสดงในรูปที่ 7.21ก - 7.21ข ขั้นตอนต่อมาคือทำการคำนวณหาค่าการเคลื่อนตัวและค่าความเค้นต่าง ๆ ตามลำดับ จากนั้นนำผลลัพธ์ที่ได้มาคำนวณค่าแอลทูนอร์มสำหรับความคลาดเคลื่อนและค่าเปอร์เซ็นต์ความคลาดเคลื่อน

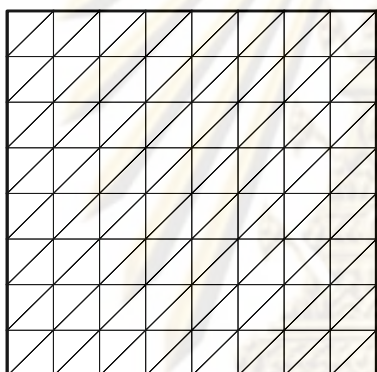
การเปรียบเทียบจำนวนจุดต่อและค่าเปอร์เซ็นต์ความคลาดเคลื่อนของรูปแบบจำลองไฟไนต์เอลิเมนต์ทั้ง 8 รูปแบบ แสดงอยู่ในตารางที่ 7.4 โดยรูปแบบจำลองจากการปรับขนาดเอลิเมนต์ทั้งสองนั้นใช้จำนวนจุดต่อในระดับใกล้เคียงกัน ในขณะที่แบบจำลองสุดท้ายของรูปแบบจำลองอย่างเป็นทางการเป็นระเบียบนั้นใช้จำนวนจุดต่อมากกว่ารูปแบบจำลองจากการปรับขนาดเอลิเมนต์ประมาณสี่เท่าเมื่อเปรียบเทียบในระดับค่าเปอร์เซ็นต์ความคลาดเคลื่อนเดียวกัน



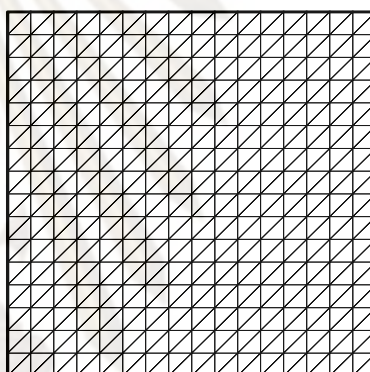
ก) Mesh S1 (U1)



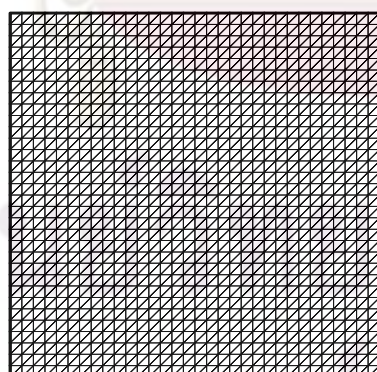
ข) Mesh S2



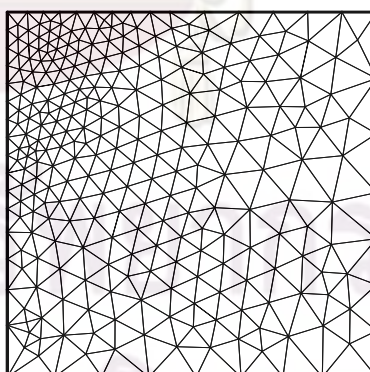
ค) Mesh S3



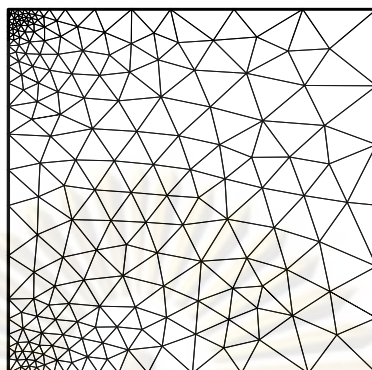
ง) Mesh S4



จ) Mesh S5



ฉ) Mesh U2



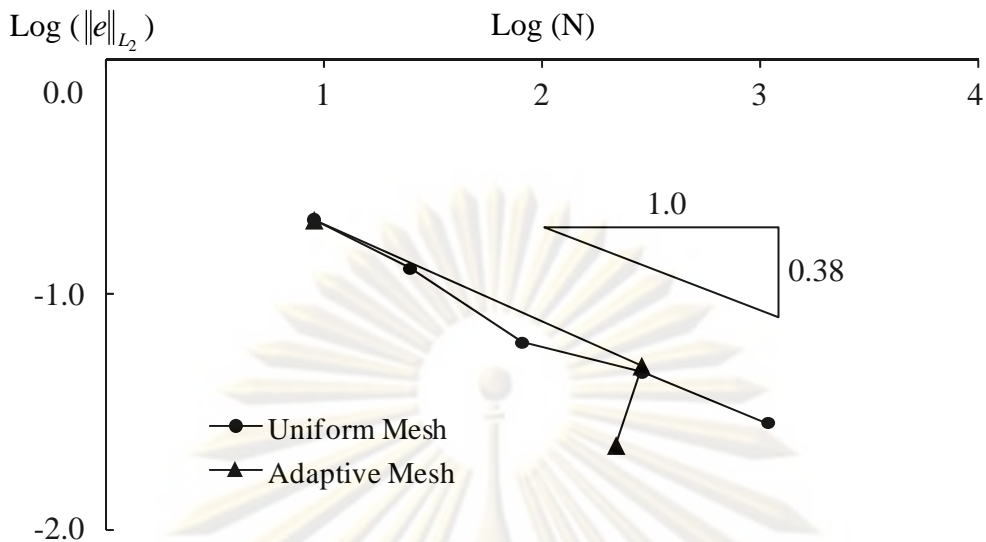
ข) Mesh U3

รูปที่ 7.21 รูปแบบจำลองไฟไนต์เอลิเมนต์อย่างเป็นระเบียบ ก) Mesh S1 (U1), ข) Mesh S2, ค) Mesh S3, ง) Mesh S4, จ) Mesh S5 และรูปแบบจำลองไฟไนต์เอลิเมนต์อย่างไว้ระเบียบ ฉ) Mesh U2, ซ) Mesh U3

ตารางที่ 7.4 ตารางแสดงการเปรียบเทียบจำนวนจุดต่อและค่าความคลาดเคลื่อนระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งแปดรูปแบบของปัญหาคานปลายยื่นชนิดสั้นภายใต้แรงกด

Element	จำนวนจุดต่อ	η_{FE}
Mesh S1 (U1)	9	15.67%
Mesh S2	25	10.19%
Mesh S3	81	4.71%
Mesh S4	289	3.91%
Mesh S5	1,089	2.42%
Mesh U2	287	4.17%
Mesh U3	219	1.97%

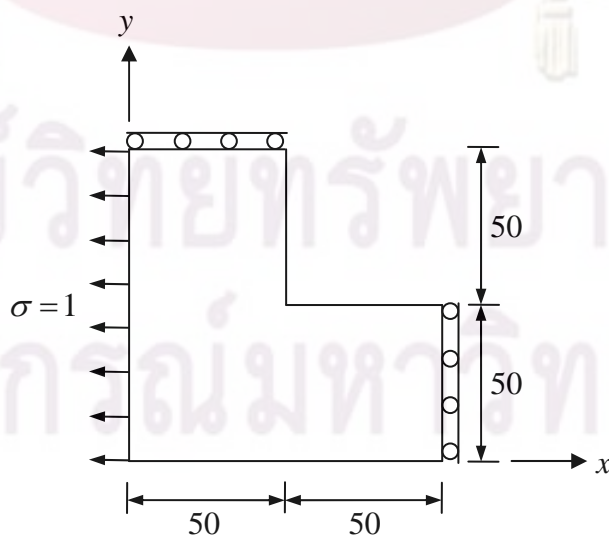
การเปรียบเทียบการลู่เข้าของผลเฉลยโดยประมาณระหว่างรูปแบบจำลองอย่างเป็นระเบียบและรูปแบบจำลองจากการปรับขนาดเอลิเมนต์ดังแสดงในรูปที่ 7.22 แสดงให้เห็นถึงประสิทธิภาพของเทคนิคการปรับขนาดเอลิเมนต์ที่ประยุกต์เข้ากับระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อด้วยการใช้หลักการของฟลักซ์



รูปที่ 7.22 ผลการเปรียบเทียบการลู่เข้าของผลเฉลยโดยประมาณระหว่างรูปแบบจำลองอย่างเป็นระเบียบและรูปแบบจำลองจากการปรับขนาดเอลิเมนต์สำหรับปัญหาคานปลายยื่นชนิดสั้นภายใต้แรงกด

7.5.5 ปัญหาแผ่นโลหะรูปตัวแอล (L-shaped plate) ภายใต้แรงดึง

ตัวอย่างนี้เป็นแผ่นโลหะรูปตัวแอลซึ่งมีรายละเอียดของขนาดและเงื่อนไขขอบเขตแสดงอยู่ในรูปที่ 7.23 แผ่นโลหะนี้ถูกดึงด้วยแรงดึงขนาด $\sigma = 1$ [30] และอยู่ภายใต้เงื่อนไขความเค้นในระนาบโดยมีคุณสมบัติทางกลดังนี้ ค่าโมดูลัสของความยืดหยุ่น $E = 100,000$ psi และค่าอัตราส่วนปัวส์ซง $\nu = 0.3$



รูปที่ 7.23 ปัญหาแผ่นโลหะรูปตัวแอลภายใต้แรงดึง

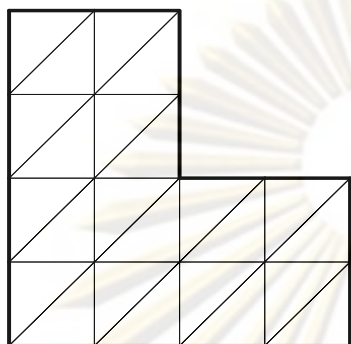
รูปแบบจำลองไฟไนต์เอลิเมนต์สำหรับปัญหาแผ่นโลหะรูปตัวแอลภายใต้แรงดึง ประกอบด้วย รูปแบบจำลองอย่างเป็นระเบียบจำนวน 4 รูปแบบ (S1 ถึง S4) และรูปแบบจำลองอย่างไร้ระเบียบจำนวน 4 รูปแบบ (U1 ถึง U4) ซึ่งเป็นรูปแบบจำลองเริ่มต้น 1 รูปแบบ (U1) และรูปแบบจำลองจากการปรับขนาดเอลิเมนต์ 3 รูปแบบ (U2 และ U4) รายละเอียดของรูปแบบจำลองมีดังนี้

- 1) รูปแบบจำลองอย่างเป็นระเบียบ 1 (Mesh S1) ประกอบด้วยจำนวนจุดต่อ 21 จุดต่อและจำนวนเอลิเมนต์ 24 เอลิเมนต์
- 2) รูปแบบจำลองอย่างเป็นระเบียบ 2 (Mesh S2) ประกอบด้วยจำนวนจุดต่อ 96 จุดต่อและจำนวนเอลิเมนต์ 150 เอลิเมนต์
- 3) รูปแบบจำลองอย่างเป็นระเบียบ 3 (Mesh S3) ประกอบด้วยจำนวนจุดต่อ 341 จุดต่อและจำนวนเอลิเมนต์ 600 เอลิเมนต์
- 4) รูปแบบจำลองอย่างเป็นระเบียบ 4 (Mesh S4) ประกอบด้วยจำนวนจุดต่อ 1281 จุดต่อและจำนวนเอลิเมนต์ 2400 เอลิเมนต์
- 5) รูปแบบจำลองอย่างไร้ระเบียบ 1 (Mesh U1) ประกอบด้วยจำนวนจุดต่อ 166 จุดต่อและจำนวนเอลิเมนต์ 290 เอลิเมนต์
- 6) รูปแบบจำลองอย่างไร้ระเบียบ 2 (Mesh U2) ประกอบด้วยจำนวนจุดต่อ 228 จุดต่อและจำนวนเอลิเมนต์ 390 เอลิเมนต์
- 7) รูปแบบจำลองอย่างไร้ระเบียบ 3 (Mesh U3) ประกอบด้วยจำนวนจุดต่อ 409 จุดต่อและจำนวนเอลิเมนต์ 729 เอลิเมนต์
- 8) รูปแบบจำลองอย่างไร้ระเบียบ 4 (Mesh U4) ประกอบด้วยจำนวนจุดต่อ 257 จุดต่อและจำนวนเอลิเมนต์ 445 เอลิเมนต์

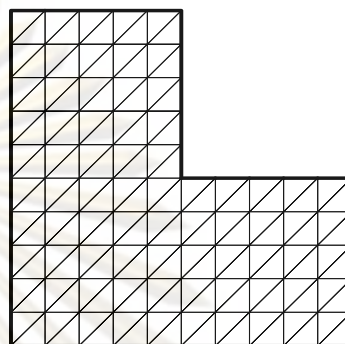
โดยรายละเอียดรูปแบบจำลองไฟไนต์เอลิเมนต์ข้างต้นแสดงในรูปที่ 7.24ก - 7.24ข ขั้นตอนต่อมาคือทำการคำนวณหาค่าการเคลื่อนตัวและค่าความเค้นต่าง ๆ ตามลำดับ จากนั้นนำผลลัพธ์ที่ได้มาคำนวณค่าแอลทูนอร์มสำหรับความคลาดเคลื่อนและค่าเปอร์เซ็นต์ความคลาดเคลื่อน

ตารางที่ 7.5 แสดงการเปรียบเทียบจำนวนจุดต่อและค่าเปอร์เซ็นต์ความคลาดเคลื่อนของรูปแบบจำลองไฟไนต์เอลิเมนต์ทั้ง 8 รูปแบบ โดยรูปแบบจำลองจากการปรับขนาดเอลิเมนต์ทั้งสามนั้นใช้จำนวนจุดต่อในระดับใกล้เคียงกันหรืออาจกล่าวได้ว่าเกือบจะคงที่ ในขณะที่รูปแบบจำลองอย่างเป็นระเบียบ 4 นั้นใช้จำนวนจุดต่อมากกว่ารูปแบบจำลองจากการ

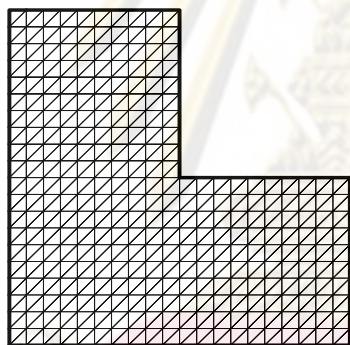
ปรับขนาดเอลิเมนต์ประมาณห้าเท่าเมื่อเปรียบเทียบในระดับค่าเปอร์เซ็นต์ความคลาดเคลื่อนเดียวกัน



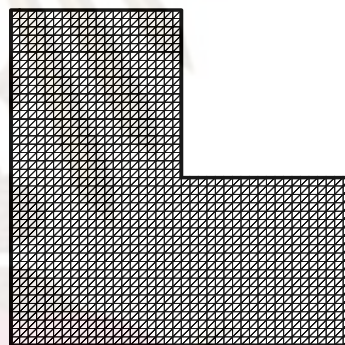
ก) Mesh S1



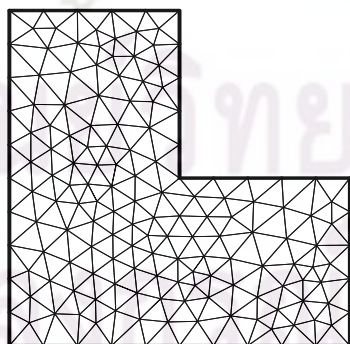
ข) Mesh S2



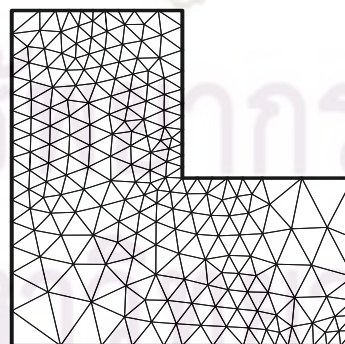
ค) Mesh S3



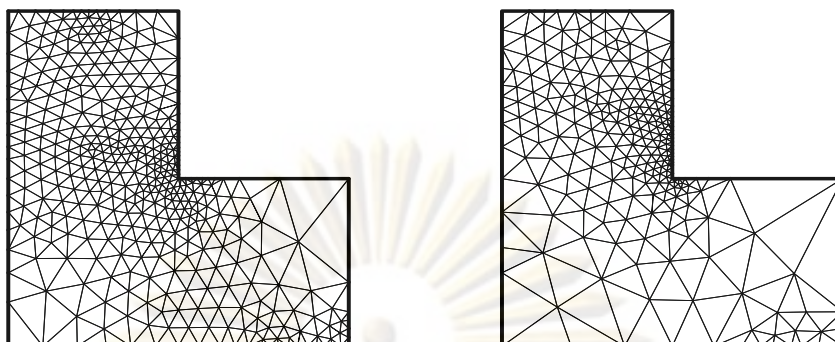
ง) Mesh S4



จ) Mesh U1



ฉ) Mesh U2



ข) Mesh U3

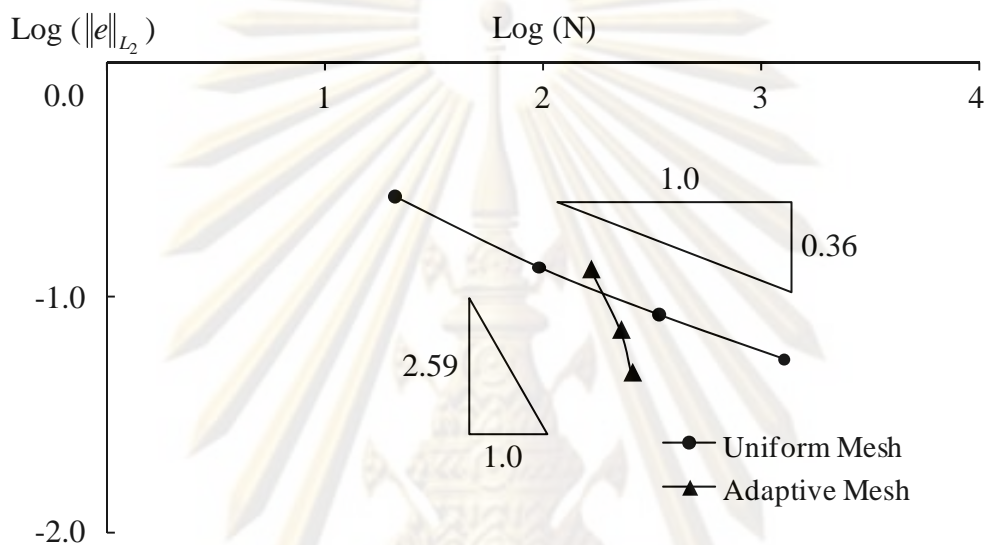
ช) Mesh U4

รูปที่ 7.24 รูปแบบจำลองไฟไนต์เอลิเมนต์อย่างเป็นระเบียบ ก) Mesh S1, ข) Mesh S2, ค) Mesh S3, ง) Mesh S4 และรูปแบบจำลองไฟไนต์เอลิเมนต์อย่างไร้ระเบียบ จ) Mesh U1, ฉ) Mesh U2, ช) Mesh U3, ซ) Mesh U4

ตารางที่ 7.5 ตารางแสดงการเปรียบเทียบจำนวนจุดต่อและค่าความคลาดเคลื่อนระหว่างแบบจำลองไฟไนต์เอลิเมนต์ทั้งแปดรูปแบบของปัญหาแผ่นโลหะรูปตัวแอลภายใต้แรงดึง

Element	จำนวนจุดต่อ	η_{FE}
Mesh S1	21	12.08%
Mesh S2	96	6.35%
Mesh S3	341	4.07%
Mesh S4	1,281	2.67%
Mesh U1	166	6.21%
Mesh U2	228	3.53%
Mesh U3	409	2.24%
Mesh U4	257	2.35%

การเปรียบเทียบการลู่เข้าของผลเฉลยโดยประมาณระหว่างรูปแบบจำลองอย่างเป็นระเบียบและรูปแบบจำลองจากการปรับขนาดเอลิเมนต์ดังแสดงในรูปที่ 7.25 แสดงให้เห็นถึงประสิทธิภาพของเทคนิคการปรับขนาดเอลิเมนต์ที่ประยุกต์เข้ากับระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไว้จุดต่อด้วยการใช้หลักการของฟลักซ์



รูปที่ 7.25 ผลการเปรียบเทียบการลู่เข้าของผลเฉลยโดยประมาณระหว่างรูปแบบจำลองอย่างเป็นระเบียบและรูปแบบจำลองจากการปรับขนาดเอลิเมนต์สำหรับปัญหาแผ่นโลหะรูปตัวแอลภายใต้แรงดึง

บทที่ 8

บทสรุป ปัญหาที่พบและข้อเสนอแนะ

8.1 บทสรุป

วิทยานิพนธ์นี้เน้นการการประดิษฐ์ระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์สำหรับเอลิเมนต์สามเหลี่ยมแบบตัวแปรไร้จุดต่อ และใช้วิธีดังกล่าวในการวิเคราะห์ปัญหาอุณหภูมิและโครงสร้าง รวมไปถึงการนำเสนอการคะเนความคลาดเคลื่อนแบบโพสเทอริออริและนำไปประยุกต์ใช้ร่วมกับการปรับขนาดเอลิเมนต์โดยอัตโนมัติ เพื่อเพิ่มความถูกต้องเที่ยงตรงของผลลัพธ์ที่คำนวณได้จากระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อด้วยการใช้หลักการของฟลักซ์ให้สูงขึ้น

ในการวิเคราะห์ปัญหาอุณหภูมิและโครงสร้างด้วยระเบียบวิธีไฟไนต์เอลิเมนต์ จำเป็นที่จะต้องมีความรู้พื้นฐานในทฤษฎีการถ่ายเทความร้อนและทฤษฎีทางด้านกลศาสตร์ของแข็ง สมการเชิงอนุพันธ์ที่เกี่ยวข้องได้แก่ สมการเชิงอนุพันธ์ของปัญหาการถ่ายเทความร้อนและสมการเชิงอนุพันธ์ของปัญหาการหาค่าความเค้นอันเนื่องมาจากอุณหภูมิภายในของแข็งดังแสดงในบทที่ 2 จากนั้นบทที่ 3 ได้อธิบายถึงขั้นตอนทั่วไปของระเบียบวิธีไฟไนต์เอลิเมนต์โดยวิธีการถ่วงน้ำหนักเศษตค่าง แล้วจึงอธิบายรายละเอียดของระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ และขั้นตอนการประดิษฐ์ระเบียบวิธีไฟไนต์เอลิเมนต์ด้วยการใช้หลักการของฟลักซ์เพื่อใช้แก้ปัญหาการถ่ายเทความร้อนและปัญหาของแข็งตามลำดับ อีกทั้งแสดงขั้นตอนในการประดิษฐ์สมการไฟไนต์เอลิเมนต์สำหรับปัญหาการถ่ายเทความร้อนและสมการไฟไนต์เอลิเมนต์สำหรับปัญหาของแข็งที่สอดคล้องกับสมการเชิงอนุพันธ์ข้างต้น และยังแสดงรายละเอียดของไฟไนต์เอลิเมนต์เมทริกซ์ต่าง ๆ ที่ประดิษฐ์ในรูปแบบที่สามารถนำไปประดิษฐ์เป็นโปรแกรมคอมพิวเตอร์ได้โดยตรง

ในบทที่ 4 เป็นการอธิบายโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นโดยนำเสนอถึงขั้นตอนการทำงาน รายละเอียดของโปรแกรมคอมพิวเตอร์ และรูปแบบของไฟล์ข้อมูลนำเข้าที่จำเป็นต้องเตรียมเพื่อทำการคำนวณสำหรับการวิเคราะห์ปัญหาการถ่ายเทความร้อนและปัญหาความเค้นอันเนื่องมาจากอุณหภูมิตามลำดับ จากนั้นจึงได้อธิบายหลักการของเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติและนำเสนอโปรแกรมคอมพิวเตอร์สำหรับการปรับขนาดเอลิเมนต์โดยอัตโนมัติ (FEMESH) และภาพโดยรวมของโครงสร้างการทำงานของโปรแกรม FEMESH ดังแสดงในบทที่ 5 คำนวณ จากนั้นโปรแกรมคอมพิวเตอร์ประดิษฐ์ขึ้นในบทที่ 4 จะได้รับการตรวจสอบ

ความถูกต้องในบทที่ 6 โดยนำผลลัพธ์ที่ได้จากการวิเคราะห์ด้วยไฟไนต์เอลิเมนต์โปรแกรมคอมพิวเตอร์ไปเปรียบเทียบกับผลเฉลยแม่นยำตรง การตรวจสอบโปรแกรมคอมพิวเตอร์นี้แบ่งการตรวจสอบออกเป็น 4 ส่วน ได้แก่ 1) การวิเคราะห์ปัญหาการถ่ายเทความร้อน 2) การวิเคราะห์ปัญหาของแข็ง 3) การวิเคราะห์ปัญหาความเค้นอันเนื่องมาจากอุณหภูมิ และ 4) การวิเคราะห์ปัญหาด้วยเทคนิคการปรับเอลิเมนต์แบบอัตโนมัติ จำนวนปัญหาที่ใช้ในการตรวจสอบทั้ง 4 ส่วนรวมทั้งสิ้น 7 ปัญหา โดยมีรายละเอียดดังนี้ 1) การวิเคราะห์ปัญหาการถ่ายเทความร้อนนั้นใช้ปัญหาในการตรวจสอบ 1 ปัญหาคือ ปัญหาแผ่นสี่เหลี่ยมที่มีการผลิตความร้อนได้เอง 2) การวิเคราะห์ปัญหาของแข็งใช้ปัญหาในการตรวจสอบ 1 ปัญหาคือปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง 3) การวิเคราะห์ปัญหาความเค้นอันเนื่องมาจากอุณหภูมิใช้ปัญหาในการตรวจสอบ 2 ปัญหา ได้แก่ ปัญหาแผ่นสี่เหลี่ยมที่มีอุณหภูมิเปลี่ยนแปลงเชิงเส้นตามความยาวและปัญหาวงแหวนที่มีอุณหภูมิเปลี่ยนแปลงตามแนวรัศมี และ 4) การวิเคราะห์ปัญหาด้วยเทคนิคการปรับเอลิเมนต์แบบอัตโนมัติใช้ปัญหาในการตรวจสอบ 3 ปัญหา ได้แก่ ปัญหาแผ่นสี่เหลี่ยมจัตุรัสภายใต้กรวยของเกรเดียนท์ความร้อน ปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมตรงกลาง และปัญหาวงแหวนที่มีอุณหภูมิเปลี่ยนแปลงตามแนวรัศมี

สุดท้ายในบทที่ 7 เป็นการอธิบายความหมาย นิยาม และที่มาการคะเนของความคลาดเคลื่อน จากนั้นจึงอธิบายหลักการการคะเนความคลาดเคลื่อนแบบโพสเทอริออริซึ่งเป็นการคะเนความคลาดเคลื่อนภายหลังจากที่ได้ผลเฉลยจากระเบียบวิธีไฟไนต์เอลิเมนต์ โดยมีวัตถุประสงค์เพื่อเป็นดัชนีชี้วัดการยุติการปรับขนาดเอลิเมนต์และแสดงความถูกต้องแม่นยำของเทคนิคการปรับขนาดเอลิเมนต์ แล้วจึงนำเสนอรายละเอียดของการคะเนความคลาดเคลื่อนแบบโพสเทอริออริด้วยการใช้หลักการของฟลักซ์ ก่อนที่จะนำไปประดิษฐ์เป็นโปรแกรมคอมพิวเตอร์สำหรับการคะเนความคลาดเคลื่อนที่คำนวณจากผลต่างของฟลักซ์ โดยโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นจะถูกนำไปทดสอบกับปัญหาที่มีผลเฉลยแม่นยำตรง ผลเฉลยเชิงวิเคราะห์ และปัญหาที่มีความซับซ้อนรวมทั้งสิ้น 5 ปัญหา ประกอบด้วย ปัญหาแผ่นสี่เหลี่ยมผืนผ้าซึ่งถูกกำหนดอุณหภูมิที่ขอบ ปัญหาแผ่นสี่เหลี่ยมจัตุรัสพร้อมแหล่งกำเนิด ปัญหาแผ่นโลหะสี่เหลี่ยมมีรูกลมภายใต้แรงดึงทิศทางเดียว ปัญหาคานปลายยื่นชนิดสั้นภายใต้แรงกด และปัญหาแผ่นโลหะรูปตัวแอลภายใต้แรงดึง การทดสอบแสดงให้เห็นถึงประสิทธิภาพการคะเนความคลาดเคลื่อนแบบโพสเทอริออริด้วยการใช้หลักการของฟลักซ์ที่ได้นำเสนอ และประสิทธิภาพของเทคนิคการปรับขนาดเอลิเมนต์ที่ประยุกต์เข้ากับระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อด้วยการใช้หลักการของฟลักซ์

8.2 ปัญหาที่พบในขณะทำวิทยานิพนธ์

ปัญหาสำคัญที่พบในขณะทำวิทยานิพนธ์ก็คือ การประดิษฐ์ระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อ เนื่องจากเอลิเมนต์สามเหลี่ยมแบบตัวแปรไร้จุดต่อมีความหมายทางกายภาพที่แตกต่างออกไปจากเอลิเมนต์ทั่วไป (เอลิเมนต์สามเหลี่ยมสามจุดต่อเอลิเมนต์หรือเอลิเมนต์สามเหลี่ยมหกจุดต่อ) ทำให้จำเป็นต้องพิจารณาเงื่อนไขเริ่มต้นและเงื่อนไขขอบเขตต่าง ๆ ของเอลิเมนต์สามเหลี่ยมแบบตัวแปรไร้จุดต่อให้สอดคล้องกับกายภาพของปัญหาที่ทำการวิเคราะห์ และอีกปัญหาที่พบก็คือ สมการไฟไนต์เอลิเมนต์ที่ได้จากการวิเคราะห์การกระจายของพลาสมาที่มีลักษณะต่อเนื่องดังแสดงในสมการ (7.22) นั้น ไม่สามารถที่จะใช้ร่วมกับเมทริกซ์มวลแบบรวมที่จุดต่อ (lumped mass matrix) ได้ เนื่องด้วยเมทริกซ์มวลแบบรวมที่จุดต่อมีหลักการประดิษฐ์ที่ขัดกับกายภาพของเอลิเมนต์สามเหลี่ยมแบบตัวแปรไร้จุดต่อนั่นเอง

8.3 ข้อเสนอแนะสำหรับงานวิจัยในอนาคต

การพัฒนางานวิจัยต่อจากวิทยานิพนธ์นี้มีอยู่ด้วยกันหลายแนวทาง ยกตัวอย่างเช่น การปรับปรุงไฟไนต์เอลิเมนต์โปรแกรมคอมพิวเตอร์เพื่อวิเคราะห์ปัญหาในสภาวะชั่วขณะ (transient problems) การประยุกต์ระเบียบวิธีไฟไนต์เอลิเมนต์แบบตัวแปรไร้จุดต่อด้วยการใช้หลักการของพลาสมาเพื่อวิเคราะห์ปัญหาการไหลหรือวิเคราะห์ปัญหาการปฏิสัมพันธ์ร่วมกันระหว่างของไหล ของแข็งและการถ่ายเทความร้อน การพัฒนาและประยุกต์การคะเนความคลาดเคลื่อนแบบโพสเทอริออร์สำหรับวิเคราะห์ปัญหาการไหล และการปรับปรุงวิธีในการแก้ระบบสมการให้ดียิ่งขึ้น

รายการอ้างอิง

- [1] ปราโมทย์ เดชะอำไพ, ไฟไนต์เอลิเมนต์ในงานวิศวกรรม, พิมพ์ครั้งที่ 4. กรุงเทพมหานคร: สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2550.
- [2] Dechaumphai, P. Adaptive finite element technique for thermal stress analysis of built-up structures. JSM International Journal, Series A: Mechanics and Material Engineering 39. 2 (1996): 223-230.
- [3] Huebner, K. H., Thornton, E. A. and Byrom, T. G. The finite element method for engineers. 3rd ed. Newyork: John Wiley & Sons, 1995.
- [4] Thornton, E. A., Dechaumphai, P. and Pandey, A. K. Finite element thermal-structural analyses of a cable-stiffened orbiting antenna. AIAA Journal of Spacecraft and Rockets 23. 6 (1986): 620-624.
- [5] Zienkiewicz, O. C. and Taylor, R. L. The finite element method volume I: The basis. 5th ed. Oxford: Butterworth Heinemann, 2000.
- [6] Dechaumphai, P. and Thornton, E. A. Nodeless variable finite elements for improved thermal-structural analysis. International Conference on Finite Element Methods, Shanghai, China, August 2-6, 1982.
- [7] Phongthanapanich, S., Traivivatana, S., Boonmaruth, P. and Dechaumphai, P. Nodeless variable finite element method for heat transfer analysis by means of flux-based formulation and mesh adaptation. Acta Mechanica Sinica 22 (2006): 138-147.
- [8] Zienkiewicz, O. C. and Zhu, J. Z. A simple error estimator and adaptive procedure for practical engineering analysis. International Journal for Numerical Methods in Engineering 24 (1987): 337-357.
- [9] Zienkiewicz, O. C., Liu, Y. C. and Huang, G. C. Error estimation and adaptivity in flow formulation for forming problems. International Journal for Numerical Methods in Engineering 25 (1988): 23-42.
- [10] Pandey, A. K., Dechaumphai, P. and Wieting, A. R. Thermal-structural finite element analysis using linear flux formulation. Structural Dynamics and Materials Conference, 30th, Mobile, Alabama, April 3-5, 1989.

- [11] Paweenawat, A. and Dechaumphai, P. Nodeless variables finite element method and adaptive meshing technique for viscous flow analysis. Journal of Mechanical Science and Technology 20. 10 (2006): 1730-1740.
- [12] Ainsworth, M., Zhu, J. Z., Craig, A. W. and Zienkiewicz, O. C. Analysis of the Zienkiewicz-Zhu A-Posteriori error estimator in the finite element method. International Journal for Numerical Methods in Engineering 28 (1989): 2161-2174.
- [13] Polesky, S. P. Hierarchical flux-based thermal-structural finite element analysis method. NASA Technical Memorandum, TM-107574, 1992.
- [14] Kelly, D. W., De S.R. Gago, J. P., Zienkiewicz, O. C. and Babuska, I. A Posteriori error analysis and adaptive processes in the finite element method : Part I – Error analysis. International Journal for Numerical Methods in Engineering 19 (1983): 1593-1619.
- [15] Kelly, D. W., De S.R. Gago, J. P., Zienkiewicz, O. C. and Babuska, I. A Posteriori error analysis and adaptive processes in the finite element method : Part II – Adaptive mesh refinement, International Journal for Numerical Methods in Engineering 19 (1983): 1621-1656.
- [16] วิทยา วัฒนนุกุลชัย. การแบ่งย่อยกริดเพื่อวิเคราะห์ความเค้นเนื่องจากอุณหภูมิโดยระเบียบวิธีไฟไนต์เอลิเมนต์. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต, ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2547.
- [17] Incropera, F. P. and De Witt, D. P. Fundamentals of Heat and Mass Transfer. 4th ed. Singapore: John Wiley & Sons, 1996.
- [18] Timoshenko, S. P. and Goodier, J. N. Theory of Elasticity. 3rd ed. Singapore: McGrawHill, 1970.
- [19] Silvester, P. Higher-order polynomial triangular elements for potential problems. International Journal of Engineering Science 7. 8 (1969): 849-861.
- [20] อาชวี ปวีณวัฒน์. ระเบียบวิธีไฟไนต์เอลิเมนต์เพื่อการวิเคราะห์การไหลแบบหนึ่งมิติโดยใช้ตัวแปรไร้จุดต่อ. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต, ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2545.

- [21] สุทธิศักดิ์ พงศ์ธนาพานิช. การพัฒนาโปรแกรมคอมพิวเตอร์สำหรับการไหลไม่คงตัว ความเร็วสูงแบบอัดตัวได้และไร้ความหนืดในสองมิติด้วยระเบียบวิธีไฟไนต์เอลิเมนต์. วิทยานิพนธ์ปริญญาามหาบัณฑิต, ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2544.
- [22] Dechaumphai, P. Adaptive finite element technique for heat transfer problems. Journal of Energy, Heat and Mass Transfer 17 (1995): 87-94.
- [23] ปราโมทย์ เดชะอำไพ และ สุทธิศักดิ์ พงศ์ธนาพานิช. ไฟไนต์เอลิเมนต์อย่างง่ายพร้อมซอฟต์แวร์. พิมพ์ครั้งที่ 1 กรุงเทพมหานคร สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2548.
- [24] วิโรจน์ ลิ้มตระการ. ระเบียบวิธีไฟไนต์เอลิเมนต์สำหรับปฏิสัมพันธ์ระหว่างการไหลความเร็วสูงและโครงสร้าง. วิทยานิพนธ์ปริญญาดุษฎีบัณฑิต, ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2545.
- [25] Traivivatana, S., Phongthanapanich, S. and Dechaumphai, P. Improved numerical solution accuracy of Poisson's equation by adaptive nodeless variable finite elements with flux-based formulation Transactions of the Canadian Society for Mechanical Engineering 32. 1 (2008): 23-42.
- [26] Dechaumphai, P. and Phongthanapanich, S. Easy Finite Element Method with Software. 1st ed. Oxford. Alpha Science, 2009.
- [27] Lesaint, P. and Zlamal, M. Superconvergence of the gradient of finite element solutions, RAIRO Analyse Numerique 13 (1979): 139-166.
- [28] Zienkiewicz, O. C. and Zhu, J. Z. The superconvergent patch recovery and a *posteriori* error estimates : Part I – The recovery technique. International Journal for Numerical Methods in Engineering 33 (1992): 1331-1364.
- [29] Phongthanapanich, S. and Dechaumphai, P. Nodeless variable finite element method for stress analysis using flux-based formulation. Journal of Mechanical Science and Technology 22. 4 (2004): 639-646.
- [30] Zienkiewicz, O. C. and Zhu, J. Z. The superconvergent patch recovery and a *posteriori* error estimates : Part II – Error estimates and adaptivity. International Journal for Numerical Methods in Engineering 33 (1992): 1365-1382.



ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก ก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

รายละเอียดของโปรแกรมวิเคราะห์ปัญหาการถ่ายเทความร้อน

```

PROGRAM HEAT_CONDUCT

use HC

implicit none

call MAIN()

END PROGRAM HEAT_CONDUCT

!-----

MODULE HC

implicit none

character(len=20)          :: namel

integer(4), allocatable, dimension(:,:) :: intmat      ! Node connection on element
integer(4), allocatable, dimension(:)  :: ibcp        ! boundary condition for variable
integer(4), dimension(20)             :: text         ! buffer for reading

real(8), allocatable, dimension(:,:)  :: coord, sysk, sysky ! Coordinate of each node
real(8), allocatable, dimension(:)    :: p, flux
real(8), allocatable, dimension(:)    :: sysf, one, sysfy
real(8), allocatable, dimension(:)    :: Q, qxfb, qyfb, qxex, qyex

integer(4)                    :: neq, npoi, nelem, nflux, ip, ie, ierror
integer(4)                    :: i, j, k, ii, jj, kk
integer(4)                    :: ll, mm, nn
integer(4)                    :: nlines, iline, nside, nboun, exact

real(8)                       :: xg1, xg2, xg3, yg1, yg2, yg3, theta
real(8)                       :: b1, b2, b3, c1, c2, c3
real(8)                       :: area, x, y, zero
real(8)                       :: tc, lo, cee, tk
real(8)                       :: aread, efind
real(8)                       :: rprfb, rprst, rprfe, ergfb, ergst, ergfe
real(8)                       :: maxlfbx, maxlfby, maxlstx, maxlsty, maxlfex, maxlfey

real(8), dimension(6,6)       :: kele
real(8), dimension(6,3)       :: deex, deey
real(8), dimension(3,6)       :: eeel, feel
real(8), dimension(6)         :: r, rboun, rsurf

integer(4)                    :: tnode, lflux, fcount
integer(4), allocatable, dimension(:,:) :: number
integer(4), allocatable, dimension(:,,:) :: acheck

CONTAINS

!-----

Subroutine MAIN()

use solve

implicit none

call READ_INPUT()
call GEN_NL()
call CST()
call SYMPCG(sysk, sysf, p, neq)
call ERR_EST()
call WRITE_OUTPUT()

End Subroutine MAIN

!-----

Subroutine READ_INPUT()

implicit none

write(6,10)
10 format(/, ' PLEASE ENTER INPUT FILE NAME:',/)

```

```

read(5, '(A') namel
open(unit=7, file=namel, status='OLD', ACTION='read', iostat=ierror)

!
! read input data
!

read(7,*) nlines

do iline = 1,nlines
  read(7,1) text
  1 format(20A4)
enddo

read(7,1) text
read(7,*) tnode, nelelem, nboun, nflux

read(7,1) text
read(7,*) tc, lo, cee, tk, exact

nside = (3*nelem+nboun)/2
npoi = tnode+nside
neq = npoi

allocate(      intmat(nelem,6),      coord(npoi,2))
allocate(      sysk(neq,neq),      sysf(neq))
allocate(      ibcp(npoi),      p(neq))
allocate(      Q(nelem),      flux(tnode),      one(tnode))
allocate(      number(tnode,tnode), acheck(tnode,tnode,2))
allocate(      qxfb(tnode),      qyfb(tnode))

if (exact == 1) allocate(qxex(tnode), qyex(tnode))

read(7,1) text
do ip = 1,tnode
  read(7,*) i, ibcp(i), (coord(i,k), k=1,2), p(i), flux(i)
  if(i /= ip) write(6,40) ip
  40 format(/, ' NODE NO.', I5,' IN DATA FILE IS MISSING')
  if(i /= ip) stop
enddo

read(7,1) text
do ie = 1,nelem
  read(7,*) i, (intmat(i,j), j=1,3), Q(i)
  if(i /= ie) write(6,50) ie
  50 format(/, ' ELEMENT NO.', I5, ' IN DATA FILE IS MISSING')
  if(i /= ie) stop
enddo

close(unit=7, status='keep')

write(6,60) tnode, nelelem
60 format(/, ' THE FINITE ELEMENT MODEL CONSISTS OF :',/, &
' NUMBER OF TIP NODES      =', I6, /, &
' NUMBER OF ELEMENTS      =', I6, /)

End Subroutine READ_INPUT

!-----

Subroutine CST()

implicit none

sysf = 0.
sysk = 0.

!
! Loop over the number of elements:
!

const: DO ie = 1,nelem
! Find element local coordinates:

  ii = intmat(ie,1)
  jj = intmat(ie,2)
  kk = intmat(ie,3)
  ll = intmat(ie,4)
  mm = intmat(ie,5)
  nn = intmat(ie,6)

  xg1 = coord(ii,1)
  xg2 = coord(jj,1)

```

```

    xg3 = coord(kk,1)

    yg1 = coord(ii,2)
    yg2 = coord(jj,2)
    yg3 = coord(kk,2)

    area = 0.5*(xg2*(yg3-yg1)+xg1*(yg2-yg3)+xg3*(yg1-yg2))

    b1 = (yg2 - yg3)
    b2 = (yg3 - yg1)
    b3 = (yg1 - yg2)

    c1 = (xg3 - xg2)
    c2 = (xg1 - xg3)
    c3 = (xg2 - xg1)

! Setup & clear matrix:

    kele = 0.
    r = 0.
    rboun = 0.

! Set the value for
! the coefficient of diffusion term

    call CnK_ELE()

! Compute all loads

    call ALL_LOAD()

! Assemble element

    call ASMP()

Enddo const

call APPLBC()

End Subroutine CST

!-----
Subroutine CnK_ELE()

implicit none

deex(1,1) = 4.*b1
deex(1,2) = deex(1,1)
deex(1,3) = deex(1,1)

deey(1,1) = 4.*c1
deey(1,2) = deey(1,1)
deey(1,3) = deey(1,1)

deex(2,1) = 4.*b2
deex(2,2) = deex(2,1)
deex(2,3) = deex(2,1)

deey(2,1) = 4.*c2
deey(2,2) = deey(2,1)
deey(2,3) = deey(2,1)

deex(3,1) = 4.*b3
deex(3,2) = deex(3,1)
deex(3,3) = deex(3,1)

deey(3,1) = 4.*c3
deey(3,2) = deey(3,1)
deey(3,3) = deey(3,1)

deex(4,1) = b2 + b3
deex(4,2) = b2 + 2.*b3
deex(4,3) = 2.*b2 + b3

deey(4,1) = c2 + c3
deey(4,2) = c2 + 2.*c3
deey(4,3) = 2.*c2 + c3

deex(5,1) = b1 + 2.*b3
deex(5,2) = b1 + b3
deex(5,3) = 2.*b1 + b3

deey(5,1) = c1 + 2.*c3
deey(5,2) = c1 + c3

```

```

deey(5,3) = 2.*c1 + c3

deex(6,1) = b1 + 2.*b2
deex(6,2) = 2.*b1 + b2
deex(6,3) = b1 + b2

deey(6,1) = c1 + 2.*c2
deey(6,2) = 2.*c1 + c2
deey(6,3) = c1 + c2

deex = deex*tk/24.
deey = deey*tk/24.

eele(1,1) = b1
eele(2,1) = eele(1,1)
eele(3,1) = eele(1,1)

eele(1,2) = b2
eele(2,2) = eele(1,2)
eele(3,2) = eele(1,2)

eele(1,3) = b3
eele(2,3) = eele(1,3)
eele(3,3) = eele(1,3)

eele(1,4) = 0.
eele(1,5) = b3
eele(1,6) = b2

eele(2,4) = b3
eele(2,5) = 0.
eele(2,6) = b1

eele(3,4) = b2
eele(3,5) = b1
eele(3,6) = 0.

fele(1,1) = c1
fele(2,1) = fele(1,1)
fele(3,1) = fele(1,1)

fele(1,2) = c2
fele(2,2) = fele(1,2)
fele(3,2) = fele(1,2)

fele(1,3) = c3
fele(2,3) = fele(1,3)
fele(3,3) = fele(1,3)

fele(1,4) = 0.
fele(1,5) = c3
fele(1,6) = c2

fele(2,4) = c3
fele(2,5) = 0.
fele(2,6) = c1

fele(3,4) = c2
fele(3,5) = c1
fele(3,6) = 0.

eele = eele/2./area
fele = fele/2./area

kele = kele + matmul(deex,eele) + matmul(deey,fele)

End Subroutine CnK_ELE

!-----

Subroutine ALL_LOAD()
implicit none
real(8) :: dx, dy, dl, QS
real(8) :: xgc, ygc, sqcond, pi

! Compute Nodal Flux
if (nflux /= 0) then
  if ((flux(ii) /= 0.) .and. (flux(jj) /= 0.)) then
    dx = xg2 - xg1
    dy = yg2 - yg1
    dl = sqrt(dx*dx + dy*dy)

```



```

      QS = (flux(ii) + flux(jj))/2.
      rboun(1) = QS*d1/2.
      rboun(2) = QS*d1/2.
      rboun(6) = 0.
    endif

    if ((flux(jj) /= 0.) .and. (flux(kk) /= 0.)) then
      dx = xg3 - xg2
      dy = yg3 - yg2
      dl = sqrt(dx*dx + dy*dy)
      QS = (flux(jj) + flux(kk))/2.
      rboun(2) = QS*d1/2.
      rboun(3) = QS*d1/2.
      rboun(4) = 0.
    endif

    if ((flux(kk) /= 0.) .and. (flux(ii) /= 0.)) then
      dx = xg3 - xg1
      dy = yg3 - yg1
      dl = sqrt(dx*dx + dy*dy)
      QS = (flux(kk) + flux(ii))/2.
      rboun(1) = QS*d1/2.
      rboun(3) = QS*d1/2.
      rboun(5) = 0.
    endif

  endif

  ! Compute Constant Surface Heating
  ! with specified function

  xgc = abs((xg1+xg2+xg3)/3.)
  ygc = abs((yg1+yg2+yg3)/3.)
  sqcond = sqrt((xgc-0.25)*(xgc-0.25)+(ygc-0.25)*(ygc-0.25))
  pi = 4.*atan(1.)

  if (sqcond .gt. 0.075) then
    Q(ie) = 0.
  else
    Q(ie) = 10.*pi/0.075*(pi/0.075*cos(sqcond*pi/0.075)+sin(sqcond*pi/0.075)/sqcond)
  endif

  rsurf(1) = Q(ie)*area*tk/3.
  rsurf(2) = rsurf(1)
  rsurf(3) = rsurf(1)
  rsurf(4) = 0.
  rsurf(5) = rsurf(4)
  rsurf(6) = rsurf(4)

  ! Compute Total Load

  r = rboun + rsurf

End Subroutine ALL_LOAD

!-----

Subroutine ASMP()

implicit none

! Assembling system load vector

do i = 1,6
  ii = intmat(ie,i)
  do j = 1,6
    jj = intmat(ie,j)
    sysk(ii,jj) = sysk(ii,jj) + kele(i,j)
  enddo
  sysf(ii) = sysf(ii) + r(i)
enddo

End Subroutine ASMP

!-----

Subroutine aplbc()

implicit none

integer(4)          ::  ieq, ir, ic

outer1: do ieq = 1,neq
          if(ibcp(ieq) == 0) cycle outer1

```

```

        inner1: do ir = 1,neq
                if(ir == ieq) cycle inner1
                sysf(ir) = sysf(ir) - sysk(ir,ieq)*p(ieq)
                sysk(ir,ieq) = 0.
        enddo inner1

        do ic = 1,neq
                sysk(ieq,ic) = 0.
        enddo

        sysk(ieq,ieq) = 1.
        sysf(ieq)      = p(ieq)

    enddo outer1
End Subroutine applbc

!-----
Subroutine ERR_EST()
use solve
implicit none

real(8)                :: pi

real(8), dimension(1,1) :: eqxfb, eqyfb, eqxst, eqyst, eqxfe, eqyfe
real(8), dimension(1,1) :: tlqfe, tlqfb
real(8), dimension(3,1) :: bele, cele, qxfbe, qyfbe
real(8), dimension(1,3) :: belet, celet
real(8), dimension(6,1) :: tbar
real(8), dimension(1,6) :: tbart
real(8), dimension(3,3) :: mele
real(8), dimension(3,6) :: geex, geey
real(8), dimension(6,6) :: kcx, kcy

deallocate(sysk, sysf, stat=ierror)
allocate (sysk(tnode,tnode), sysky(tnode,tnode), sysf(tnode), sysfy(tnode))

    sysk = 0.
    sysf = 0.
    Sysky = 0.
    sysfy = 0.
    aread = 0.
    one = 0.
    qxfb = 0.
    qyfb = 0.
    tlqfe = 0.
    tlqfb = 0.
    ergfb = 0.
    ergst = 0.
    ergfe = 0.
    maxlfbx = 0.
    maxlfby = 0.
    maxlstx = 0.
    maxlsty = 0.
    maxlfex = 0.
    maxlfey = 0.

    pi = 4.*atan(1.)

    !
    ! Setup element matrices to compute nodal fluxes :
    !
    Do ie = 1,nelem

        ii = intmat(ie,1)
        jj = intmat(ie,2)
        kk = intmat(ie,3)
        ll = intmat(ie,4)
        mm = intmat(ie,5)
        nn = intmat(ie,6)

        xg1 = coord(ii,1)
        xg2 = coord(jj,1)
        xg3 = coord(kk,1)

        yg1 = coord(ii,2)
        yg2 = coord(jj,2)
        yg3 = coord(kk,2)

        area = 0.5*(xg2*(yg3-yg1)+xg1*(yg2-yg3)+xg3*(yg1-yg2))

```

```

b1 = (yg2 - yg3)
b2 = (yg3 - yg1)
b3 = (yg1 - yg2)

c1 = (xg3 - xg2)
c2 = (xg1 - xg3)
c3 = (xg2 - xg1)

tbar(1,1) = p(ii)
tbar(2,1) = p(jj)
tbar(3,1) = p(kk)
tbar(4,1) = p(ll)
tbar(5,1) = p(mm)
tbar(6,1) = p(nn)

! Setup mass matrix [M] :

mele(1,1) = 2.
mele(1,2) = 1.
mele(1,3) = 1.
mele(2,1) = 1.
mele(2,2) = 2.
mele(2,3) = 1.
mele(3,1) = 1.
mele(3,2) = 1.
mele(3,3) = 2.

mele = mele*area/12.

! Setup matrix [Gx] & [Gy] :

geex(1,1) = 4.*b1
geex(2,1) = geex(1,1)
geex(3,1) = geex(1,1)

geey(1,1) = 4.*c1
geey(2,1) = geey(1,1)
geey(3,1) = geey(1,1)

geex(1,2) = 4.*b2
geex(2,2) = geex(1,2)
geex(3,2) = geex(1,2)

geey(1,2) = 4.*c2
geey(2,2) = geey(1,2)
geey(3,2) = geey(1,2)

geex(1,3) = 4.*b3
geex(2,3) = geex(1,3)
geex(3,3) = geex(1,3)

geey(1,3) = 4.*c3
geey(2,3) = geey(1,3)
geey(3,3) = geey(1,3)

geex(1,4) = b2 + b3
geex(2,4) = b2 + 2.*b3
geex(3,4) = 2.*b2 + b3

geey(1,4) = c2 + c3
geey(2,4) = c2 + 2.*c3
geey(3,4) = 2.*c2 + c3

geex(1,5) = b1 + 2.*b3
geex(2,5) = b1 + b3
geex(3,5) = 2.*b1 + b3

geey(1,5) = c1 + 2.*c3
geey(2,5) = c1 + c3
geey(3,5) = 2.*c1 + c3

geex(1,6) = b1 + 2.*b2
geex(2,6) = 2.*b1 + b2
geex(3,6) = b1 + b2

geey(1,6) = c1 + 2.*c2
geey(2,6) = 2.*c1 + c2
geey(3,6) = c1 + c2

geex = geex/24.
geey = geey/24.

! Compute element load vectors (qxfbe, qyfbe) :

```

```

qxfbe = -tc*matmul(geex,tbar)
qyfbe = -tc*matmul(geey,tbar)

! Assemble element

do i=1,3
  ii = intmat(ie,i)
  do j=1,3
    jj = intmat(ie,j)
    sysk(ii,jj) = sysk(ii,jj) + mele(i,j)
    sysky(ii,jj) = sysky(ii,jj) + mele(i,j)
  enddo
  sysf(ii) = sysf(ii) + qxfbe(i,1)
  sysfy(ii) = sysfy(ii) + qyfbe(i,1)
enddo

Enddo

! Compute NLFB nodal flux (qxfb, qyfb) :
call sympcg( sysk, sysf, qxfb, tnode)
call sympcg( sysky, sysfy, qyfb, tnode)

! Compute exact nodal flux (qxex, qyex) :
! with specified function

If (exact == 1) then
  do ip = 1, tnode
    qxex(ip) = -tc*0.5*pi*dcos(0.5*pi*coord(ip,1))*dsinh(0.5*pi*coord(ip,2))/dsinh(0.5*pi)
    qyex(ip) = -tc*0.5*pi*dsin(0.5*pi*coord(ip,1))*dcosh(0.5*pi*coord(ip,2))/dsinh(0.5*pi)
  enddo
Endif

!
! Compute square of L2-norm error both x&y directions
! which in the form |B|[M]{B}, |B|[M]{C}, |C|[M]{C}
!

Do ie = 1,nelem

  ii = intmat(ie,1)
  jj = intmat(ie,2)
  kk = intmat(ie,3)
  ll = intmat(ie,4)
  mm = intmat(ie,5)
  nn = intmat(ie,6)

  xg1 = coord(ii,1)
  xg2 = coord(jj,1)
  xg3 = coord(kk,1)

  yg1 = coord(ii,2)
  yg2 = coord(jj,2)
  yg3 = coord(kk,2)

  area = 0.5*(xg2*(yg3-yg1)+xg1*(yg2-yg3)+xg3*(yg1-yg2))

  b1 = (yg2 - yg3)
  b2 = (yg3 - yg1)
  b3 = (yg1 - yg2)

  c1 = (xg3 - xg2)
  c2 = (xg1 - xg3)
  c3 = (xg2 - xg1)

  aread = aread + area

  tbar(1,1) = p(ii)
  tbar(2,1) = p(jj)
  tbar(3,1) = p(kk)
  tbar(4,1) = p(ll)
  tbar(5,1) = p(mm)
  tbar(6,1) = p(nn)

  tbart = transpose(tbar)

! Setup mass matrix [M] :

mele(1,1) = 2.
mele(1,2) = 1.
mele(1,3) = 1.
mele(2,1) = 1.
mele(2,2) = 2.
mele(2,3) = 1.
mele(3,1) = 1.

```



```

mele(3,2) = 1.
mele(3,3) = 2.

mele = mele*area/12.

! Setup matrix [Gx] & [Gy] :

geex(1,1) = 4.*b1
geex(2,1) = geex(1,1)
geex(3,1) = geex(1,1)

geey(1,1) = 4.*c1
geey(2,1) = geey(1,1)
geey(3,1) = geey(1,1)

geex(1,2) = 4.*b2
geex(2,2) = geex(1,2)
geex(3,2) = geex(1,2)

geey(1,2) = 4.*c2
geey(2,2) = geey(1,2)
geey(3,2) = geey(1,2)

geex(1,3) = 4.*b3
geex(2,3) = geex(1,3)
geex(3,3) = geex(1,3)

geey(1,3) = 4.*c3
geey(2,3) = geey(1,3)
geey(3,3) = geey(1,3)

geex(1,4) = b2 + b3
geex(2,4) = b2 + 2.*b3
geex(3,4) = 2.*b2 + b3

geey(1,4) = c2 + c3
geey(2,4) = c2 + 2.*c3
geey(3,4) = 2.*c2 + c3

geex(1,5) = b1 + 2.*b3
geex(2,5) = b1 + b3
geex(3,5) = 2.*b1 + b3

geey(1,5) = c1 + 2.*c3
geey(2,5) = c1 + c3
geey(3,5) = 2.*c1 + c3

geex(1,6) = b1 + 2.*b2
geex(2,6) = 2.*b1 + b2
geex(3,6) = b1 + b2

geey(1,6) = c1 + 2.*c2
geey(2,6) = 2.*c1 + c2
geey(3,6) = c1 + c2

geex = geex/24.
geey = geey/24.

! Setup matrix [Kcx] & [Kcy] :

kcx(1,1) = 12.*b1*b1
kcx(1,2) = 12.*b1*b2
kcx(1,3) = 12.*b1*b3
kcx(2,1) = kcx(1,2)
kcx(2,2) = 12.*b2*b2
kcx(2,3) = 12.*b2*b3
kcx(3,1) = kcx(1,3)
kcx(3,2) = kcx(2,3)
kcx(3,3) = 12.*b3*b3

kcy(1,1) = 12.*c1*c1
kcy(1,2) = 12.*c1*c2
kcy(1,3) = 12.*c1*c3
kcy(2,1) = kcy(1,2)
kcy(2,2) = 12.*c2*c2
kcy(2,3) = 12.*c2*c3
kcy(3,1) = kcy(1,3)
kcy(3,2) = kcy(2,3)
kcy(3,3) = 12.*c3*c3

kcx(1,4) = 4.*b1*(b2+b3)
kcx(1,5) = 4.*b1*(b1+b3)
kcx(1,6) = 4.*b1*(b1+b2)
kcx(2,4) = 4.*b2*(b2+b3)
kcx(2,5) = 4.*b2*(b3+b1)

```

```

kcx(2,6) = 4.*b2*(b1+b2)
kcx(3,4) = 4.*b3*(b2+b3)
kcx(3,5) = 4.*b3*(b3+b1)
kcx(3,6) = 4.*b3*(b1+b2)

kcy(1,4) = 4.*c1*(c2+c3)
kcy(1,5) = 4.*c1*(c1+c3)
kcy(1,6) = 4.*c1*(c1+c2)
kcy(2,4) = 4.*c2*(c2+c3)
kcy(2,5) = 4.*c2*(c3+c1)
kcy(2,6) = 4.*c2*(c1+c2)
kcy(3,4) = 4.*c3*(c2+c3)
kcy(3,5) = 4.*c3*(c3+c1)
kcy(3,6) = 4.*c3*(c1+c2)

kcx(4,1) = kcx(1,4)
kcx(4,2) = kcx(2,4)
kcx(4,3) = kcx(3,4)
kcx(5,1) = kcx(1,5)
kcx(5,2) = kcx(2,5)
kcx(5,3) = kcx(3,5)
kcx(6,1) = kcx(1,6)
kcx(6,2) = kcx(2,6)
kcx(6,3) = kcx(3,6)

kcy(4,1) = kcy(1,4)
kcy(4,2) = kcy(2,4)
kcy(4,3) = kcy(3,4)
kcy(5,1) = kcy(1,5)
kcy(5,2) = kcy(2,5)
kcy(5,3) = kcy(3,5)
kcy(6,1) = kcy(1,6)
kcy(6,2) = kcy(2,6)
kcy(6,3) = kcy(3,6)

kcx(4,4) = 2.*(b2*b2 + b2*b3 + b3*b3)
kcx(4,5) = 1.*(b3*b3 + b1*b3 + b2*b3 + 2.*b1*b2)
kcx(4,6) = 1.*(b2*b2 + b1*b2 + b2*b3 + 2.*b1*b3)
kcx(5,4) = kcx(4,5)
kcx(5,5) = 2.*(b1*b1 + b1*b3 + b3*b3)
kcx(5,6) = 1.*(b1*b1 + b1*b2 + b1*b3 + 2.*b2*b3)
kcx(6,4) = kcx(4,6)
kcx(6,5) = kcx(5,6)
kcx(6,6) = 2.*(b1*b1 + b1*b2 + b2*b2)

kcy(4,4) = 2.*(c2*c2 + c2*c3 + c3*c3)
kcy(4,5) = 1.*(c3*c3 + c1*c3 + c2*c3 + 2.*c1*c2)
kcy(4,6) = 1.*(c2*c2 + c1*c2 + c2*c3 + 2.*c1*c3)
kcy(5,4) = kcy(4,5)
kcy(5,5) = 2.*(c1*c1 + c1*c3 + c3*c3)
kcy(5,6) = 1.*(c1*c1 + c1*c2 + c1*c3 + 2.*c2*c3)
kcy(6,4) = kcy(4,6)
kcy(6,5) = kcy(5,6)
kcy(6,6) = 2.*(c1*c1 + c1*c2 + c2*c2)

kcx = kcx/48./area
kcy = kcy/48./area

! Compute exact & approximate total fluxes :

if (exact == 1) then

    bele(1,1) = qxex(ii)
    bele(2,1) = qxex(jj)
    bele(3,1) = qxex(kk)

    belet = transpose(bele)

    cele(1,1) = qyex(ii)
    cele(2,1) = qyex(jj)
    cele(3,1) = qyex(kk)

    celet = transpose(cele)

    tlqfb = tlqfb + matmul(belet,matmul(mele,bele)) + matmul(celet,matmul(mele,cele))

endif

bele(1,1) = qxfb(ii)
bele(2,1) = qxfb(jj)
bele(3,1) = qxfb(kk)

belet = transpose(bele)

cele(1,1) = qyfb(ii)

```

```

cele(2,1) = qyfb(jj)
cele(3,1) = qyfb(kk)

celet = transpose(cele)

tlqfe = tlqfe + matmul(belet,matmul(mele,bele)) + matmul(celet,matmul(mele,cele))

! Compute error of flux-based and standard Nodeless :

If (exact == 1) then

  bele(1,1) = qxex(ii)
  bele(2,1) = qxex(jj)
  bele(3,1) = qxex(kk)

  belet = transpose(bele)

  cele(1,1) = qxfb(ii)
  cele(2,1) = qxfb(jj)
  cele(3,1) = qxfb(kk)

  celet = transpose(cele)

  eqxfb = matmul(belet,matmul(mele,bele)) - 2.*matmul(belet,matmul(mele,cele)) + &
          matmul(celet,matmul(mele,cele))
  eqxst = matmul(belet,matmul(mele,bele)) + 2.*tc*matmul(belet,matmul(geex,tbar)) + &
          tc*tc*matmul(tbart,matmul(kcx,tbar))

  if (abs(eqxfb(1,1)) < 0.0000000001) eqxfb = 0.
  if (abs(eqxst(1,1)) < 0.0000000001) eqxst = 0.

  bele(1,1) = qyex(ii)
  bele(2,1) = qyex(jj)
  bele(3,1) = qyex(kk)

  belet = transpose(bele)

  cele(1,1) = qyfb(ii)
  cele(2,1) = qyfb(jj)
  cele(3,1) = qyfb(kk)

  celet = transpose(cele)

  eqyfb = matmul(belet,matmul(mele,bele)) - 2.*matmul(belet,matmul(mele,cele)) + &
          matmul(celet,matmul(mele,cele))
  eqyst = matmul(belet,matmul(mele,bele)) + 2.*tc*matmul(belet,matmul(geey,tbar)) + &
          tc*tc*matmul(tbart,matmul(kcy,tbar))

  if (abs(eqyfb(1,1)) < 0.0000000001) eqyfb = 0.
  if (abs(eqyst(1,1)) < 0.0000000001) eqyst = 0.

  ergfb = ergfb + eqxfb(1,1) + eqyfb(1,1)
  ergst = ergst + eqxst(1,1) + eqyst(1,1)

  if (eqxfb(1,1) > maxlfbx) maxlfbx = eqxfb(1,1)
  if (eqyfb(1,1) > maxlfby) maxlfby = eqyfb(1,1)

  if (eqxst(1,1) > maxlstx) maxlstx = eqxst(1,1)
  if (eqyst(1,1) > maxlsty) maxlsty = eqyst(1,1)

Endif

! Compute error of approximate nodal flux :

bele(1,1) = qxfb(ii)
bele(2,1) = qxfb(jj)
bele(3,1) = qxfb(kk)

belet = transpose(bele)

cele(1,1) = qyfb(ii)
cele(2,1) = qyfb(jj)
cele(3,1) = qyfb(kk)

celet = transpose(cele)

eqxfe = matmul(belet,matmul(mele,bele)) + 2.*tc*matmul(belet,matmul(geex,tbar)) + &
        tc*tc*matmul(tbart,matmul(kcx,tbar))
eqyfe = matmul(celet,matmul(mele,cele)) + 2.*tc*matmul(celet,matmul(geey,tbar)) + &
        tc*tc*matmul(tbart,matmul(kcy,tbar))

if (abs(eqxfe(1,1)) < 0.0000000001) eqxfe = 0.
if (abs(eqyfe(1,1)) < 0.0000000001) eqyfe = 0.

ergfe = ergfe + eqxfe(1,1) + eqyfe(1,1)

```

```

    if (eqxfe(1,1) > maxlfex) maxlfex = eqxfe(1,1)
    if (eqyfe(1,1) > maxlfey) maxlfey = eqyfe(1,1)

Enddo

ergfe = sqrt(ergfe/aread)
tlqfe = sqrt(tlqfe/aread)
rprfe = ergfe/(ergfe + tlqfe(1,1))*100

If (exact == 1) then

    ergfb = sqrt(ergfb/aread)
    tlqfb = sqrt(tlqfb/aread)
    rprfb = ergfb/(ergfb + tlqfb(1,1))*100

    ergst = sqrt(ergst/aread)
    rprst = ergst/(ergst + tlqfb(1,1))*100
    efind = ergfe/ergst

    write(6,100) ergst
    100 format(' Exact-ST global error is :', e12.4)
    write(6,125) maxlstx
    125 format(' Max x-local EST error is :', e12.4)
    write(6,150) maxlsty
    150 format(' Max y-local EST error is :', e12.4)
    write(6,175) rprst
    175 format(' Relative EST % error is :', e12.4, ' %',/)

    write(6,200) ergfb
    200 format(' Exact-FB global error is :', e12.4)
    write(6,225) maxlfbx
    225 format(' Max x-local EFB error is :', e12.4)
    write(6,250) maxlfby
    250 format(' Max y-local EFB error is :', e12.4)
    write(6,275) rprfb
    275 format(' Relative EFB % error is :', e12.4, ' %',/)

    write(6,300) efind
    300 format(' Effectivity Index is :', e12.4,/)

Endif

write(6,400) ergfe
400 format(' Approximate global error is :', e12.4)
write(6,425) maxlfex
425 format(' Max x-local approx error is :', e12.4)
write(6,450) maxlfey
450 format(' Max y-local approx error is :', e12.4)
write(6,500) rprfe
500 format(' Relative percentage error is :', e12.4, ' %')

End Subroutine ERR_EST

!-----
Subroutine WRITE_OUTPUT()
implicit none

integer(4)                :: leng

leng = len_trim(name1) - 4
open(unit=8,file=name1(1:leng)//'.oup',status='unknown',iostat=ierror)

    if (exact == 1) then
        write(8,100) ergst
        100 format(' Exact-ST global error is :', e16.6)
        write(8,125) maxlstx
        125 format(' Max x-local EST error is :', e16.6)
        write(8,150) maxlsty
        150 format(' Max y-local EST error is :', e16.6)
        write(8,175) rprst
        175 format(' Relative EST % error is :', e16.6, ' %',/)
        write(8,200) ergfb
        200 format(' Exact-FB global error is :', e16.6)
        write(8,225) maxlfbx
        225 format(' Max x-local EFB error is :', e16.6)
        write(8,250) maxlfby
        250 format(' Max y-local EFB error is :', e16.6)
        write(8,275) rprfb
        275 format(' Relative EFB % error is :', e16.6, ' %',/)
        write(8,300) efind
        300 format(' Effectivity Index is :', e16.6,/)
    end if
end subroutine WRITE_OUTPUT

```



```

endif

write(8,400) ergfe
400 format(' Approximate global error is :', e16.6)
write(8,425) maxlfex
425 format(' Max x-local approx error is :', e16.6)
write(8,450) maxlfey
450 format(' Max y-local approx error is :', e16.6)
write(8,500) rprfe
500 format(' Relative percentage error is :', e16.6, ' %',/)

write(8,600)
600 format(' Node      X-coor      Y-coor      Temp      Xflux      Yflux')

do i = 1,tnode
  write(8,700) i, coord(i,1), coord(i,2), p(i), qxfb(i), qyfb(i)
  700 format(i6,2f12.6,3e16.6)
enddo

do i = tnode+1,npoi
  write(8,710) i, coord(i,1), coord(i,2), p(i)
  710 format(i6,2f12.6,e16.6)
enddo

do ie = 1, nelelem
  write(8,750) ie, (intmat(ie,j),j=1,6)
  750 format(7i6)
enddo

close(unit=8, status='keep')

open(unit=9,file=name1(1:leng)//'.plt',status='unknown',iostat=ierror)

write(9,800) tnode, nelelem
800 format('Variables = "X-coor","Y-coor","Temp","Xflux","Yflux",/,',ZONE &
          N=',I6,',E=',I6,',F=FEPOINT,ET=TRIANGLE')

do ip = 1, tnode
  write(9,850) coord(ip,1), coord(ip,2), p(ip), qxfb(ip), qyfb(ip)
  850 format(2f12.6,3e16.6)
enddo

do ie = 1, nelelem
  write(9,900) (intmat(ie,j),j=1,3)
  900 format(3i6)
enddo

close(unit=9, status='keep')

End Subroutine WRITE_OUTPUT

!-----
Subroutine GEN_NL()
implicit none
real(8) :: pi
pi = 4.*atan(1.)

!
! Generate Nodeless
!
acheck = 0
npoi = tnode

Do ie = 1,nelelem
  ii = intmat(ie,1)
  jj = intmat(ie,2)
  kk = intmat(ie,3)
! Node 4 is between Node 2 & 3

  acheck(jj,kk,2) = acheck(jj,kk,2) + 1
  acheck(kk,jj,2) = acheck(kk,jj,2) + 1
  if (acheck(jj,kk,1)==1) goto 10
  acheck(jj,kk,1) = 1
  acheck(kk,jj,1) = 1
  npoi = npoi + 1
  intmat(ie,4) = npoi
  number(jj,kk) = npoi
  number(kk,jj) = npoi
  coord(npoi,1) = (coord(jj,1)+coord(kk,1))/2.

```

```

coord(npoi,2) = (coord(jj,2)+coord(kk,2))/2.

! Check boundary for node 4 <p(4)=0.>

if (ibcp(jj)==1 .and. ibcp(kk)==1) then
  ibcp(npoi) = 1

! specified boundary condition
p(npoi) = 0.

endif
goto 15
10 continue
  intmat(ie,4) = number(jj,kk)
15 continue

! Node 5 is between Node 3 & 1

acheck(ii,kk,2) = acheck(ii,kk,2) + 1
acheck(kk,ii,2) = acheck(kk,ii,2) + 1
if (acheck(ii,kk,1)==1) goto 20
acheck(ii,kk,1) = 1
acheck(kk,ii,1) = 1
npoi = npoi + 1
intmat(ie,5) = npoi
number(ii,kk) = npoi
number(kk,ii) = npoi
coord(npoi,1) = (coord(ii,1)+coord(kk,1))/2.
coord(npoi,2) = (coord(ii,2)+coord(kk,2))/2.

! Check boundary for node 5 <p(5)=0.>

if (ibcp(ii)==1 .and. ibcp(kk)==1) then
  ibcp(npoi) = 1

! specified boundary condition
p(npoi) = 0.

endif
goto 25
20 continue
  intmat(ie,5) = number(ii,kk)
25 continue

! Node 6 is between Node 1 & 2

acheck(ii,jj,2) = acheck(ii,jj,2) + 1
acheck(jj,ii,2) = acheck(jj,ii,2) + 1
if (acheck(ii,jj,1)==1) goto 30
acheck(ii,jj,1) = 1
acheck(jj,ii,1) = 1
npoi = npoi + 1
intmat(ie,6) = npoi
number(ii,jj) = npoi
number(jj,ii) = npoi
coord(npoi,1) = (coord(ii,1)+coord(jj,1))/2.
coord(npoi,2) = (coord(ii,2)+coord(jj,2))/2.

! Check boundary for node 6 <p(6)=0.>

if (ibcp(ii)==1 .and. ibcp(jj)==1) then
  ibcp(npoi) = 1

! specified boundary condition
p(npoi) = 0.

endif
goto 35
30 continue
  intmat(ie,6) = number(ii,jj)
35 continue

Enddo
End Subroutine GEN_NL

!-----
END MODULE HC
!-----

```

```

Module solve

implicit none

contains

!
! Subroutine preconditioned conjugate gradient method
!

subroutine unsympcg(a, b, x, n)

implicit none

integer(4)          :: maxit, iter, count
integer(4)          :: i, j, n
real(8)             :: del, dnew, tol, bottom
real(8)             :: alpha, dold, beta, check, thresh
real(8), dimension(n,n) :: a
real(8), dimension(n)  :: q, r, d, pre, s, x, b, ax, atb

integer(4), allocatable, dimension(:) :: ija, ijat, ijata
real(8), allocatable, dimension(:) :: sa, sat, sata

    iter = 1
    tol = 1.d-6
    maxit = 50000
    thresh = 1.e-10

    count = 1000000

    write(6,*) count

    allocate( sa(count), ija(count))
    allocate( sat(count), ijat(count))
    allocate(sata(count), ijata(count))

    !-----
    ! Convert square matrix into row-indexed sparse
    !-----
    call sprsin( a,n,n,thresh,count, sa, ija)
    call sprsint(a,n,n,thresh,count,sat,ijat)

    !-----
    ! Compute At*A
    !-----
    call sprstm(sat,ijat,sat,ijat,sata,ijata,thresh,count)

    !-----
    ! Compute At*b
    !-----
    call sprsax(sat,ijat,b,atb,n,count)

    !-----
    ! Construct precondition matrix P (but this will find P inverse)
    !-----
    do i = 1,n
        pre(i) = 1./abs(sata(i))
    end do

    !-----
    ! Find Ro
    !-----
    call sprsax(sata,ijata,x,ax,n,count)
    do i = 1,n
        r(i) = atb(i) - ax(i)
    end do

    !-----
    ! Find Do
    !-----
    do i = 1,n
        d(i) = pre(i)*r(i)
    end do

    !-----
    ! Find D-New
    !-----
    dnew = 0.
    do i = 1,n
        dnew = dnew + r(i)*d(i)
    end do

    del = dnew

```

```

Do
  if(iter >= maxit.OR.dnew <= (tol*tol*del)) exit
!-----
! Find Q(i+1)
!-----
  call sprsax(sata,ijata,d,q,n,count)

!-----
! Find Alpha
!-----
  bottom = 0.
  do i = 1,n
    bottom = bottom + d(i)*q(i)
  end do

  alpha = dnew/bottom
  do i = 1,n
    x(i) = x(i) + alpha*d(i)
  end do

!-----
! Find R(i+1)
!-----
  check = mod(iter,50)
  if(check == 0.) then
    call sprsax(sata,ijata,x,ax,n,count)
    do i = 1,n
      r(i) = atb(i) - ax(i)
    end do
  else
    do i = 1,n
      r(i) = r(i) - alpha*q(i)
    end do
  endif

!-----
! Find S(i+1)
!-----
  do i = 1,n
    s(i) = pre(i)*r(i)
  end do

!-----
! Find beta
!-----
  dold = dnew
  dnew = 0.
  do i = 1,n
    dnew = dnew + r(i)*s(i)
  end do

  beta = dnew/dold
  do i = 1,n
    d(i) = s(i) + beta*d(i)
  end do

  iter = iter + 1

End do

end subroutine unsympcg
!-----
subroutine sympcg(a, b, x, n)
implicit none

integer(4)          :: maxit, iter, count
integer(4)          :: i, j, n
real(8)             :: del, dnew, tol, bottom
real(8)             :: alpha, dold, beta, check, thresh
real(8), dimension(n,n) :: a
real(8), dimension(n)  :: q, r, d, pre, s, x, b, ax

integer(4), allocatable, dimension(:) :: ija
real(8), allocatable, dimension(:) :: sa

  iter = 1
  tol = 1.d-6
  maxit = 50000
  thresh = 1.e-10

  count = 10000000

```



```

allocate(sa(count), ija(count))

!-----
! Construct precondition matrix P (but this will find P inverse)
!-----
do i = 1,n
  pre(i) = 1./abs(a(i,i))
end do

!-----
! Convert square matrix into row-indexed sparse
!-----
call sprsin(a,n,n,thresh,count,sa,ija)

!-----
! Find Ro
!-----
call sprsax(sa,ija,x,ax,n,count)
do i = 1,n
  r(i) = b(i) - ax(i)
end do

!-----
! Find Do
!-----
do i = 1,n
  d(i) = pre(i)*r(i)
end do

!-----
! Find D-New
!-----
dnew = 0.
do i = 1,n
  dnew = dnew + r(i)*d(i)
end do

del = dnew

Do
  if(iter >= maxit.OR.dnew <= (tol*tol*del)) exit

!-----
! Find Q(i+1)
!-----
call sprsax(sa,ija,d,q,n,count)

!-----
! Find Alpha
!-----
bottom = 0.
do i = 1,n
  bottom = bottom + d(i)*q(i)
end do

alpha = dnew/bottom
do i = 1,n
  x(i) = x(i) + alpha*d(i)
end do

!-----
! Find R(i+1)
!-----
check = mod(iter,50)
if(check == 0.) then
  call sprsax(sa,ija,x,ax,n,count)
  do i = 1,n
    r(i) = b(i) - ax(i)
  end do
else
  do i = 1,n
    r(i) = r(i) - alpha*q(i)
  end do
endif

!-----
! Find S(i+1)
!-----
do i = 1,n
  s(i) = pre(i)*r(i)
end do

```

```

!-----
! Find beta
!-----
dold = dnew
dnew = 0.
do i = 1,n
  dnew = dnew + r(i)*s(i)
end do

beta = dnew/dold
do i = 1,n
  d(i) = s(i) + beta*d(i)
end do

iter = iter + 1

End do

end subroutine sympcg

!-----
subroutine sprsin(a,n,np,thresh,nmax,sa,ija)

implicit none

integer(4)          :: n, nmax, np
integer(4)          :: i, j, k
integer(4), dimension(nmax) :: ija

real(8)            :: thresh
real(8), dimension(np,np) :: a
real(8), dimension(nmax) :: sa

do j = 1,n
  sa(j) = a(j,j)
enddo

ija(1) = n+2
k = n+1
do i = 1,n
  do j = 1,n
    if (abs(a(i,j)) >= thresh) then
      if (i /= j) then
        k = k+1
        if (k > nmax) then
          write(6,*) k, nmax !, 'nmax too small in sprsin'
          pause 'nmax too small in sprsin'
        endif
        sa(k) = a(i,j)
        ija(k) = j
      endif
    endif
  enddo
  ija(i+1) = k+1
enddo

END subroutine sprsin

!-----

subroutine sprsax(sa,ija,x,b,n,nmax)

! Multiply a matrix in sparse storage by a vector x(1:n)
! and the result is a vector b(1:n)

implicit none

integer(4)          :: n, nmax, i, k
integer(4), dimension(nmax) :: ija

real(8), dimension(n) :: b, x
real(8), dimension(nmax) :: sa

if (ija(1) /= n+2) pause 'mismatched vector and matrix in sprsax'
do i = 1,n
  b(i) = sa(i) * x(i)
  do k = ija(i),ija(i+1)-1
    b(i) = b(i) + sa(k) * x(ija(k))
  enddo
enddo

END subroutine sprsax

!-----

```

```
subroutine sprstm(sa,ija,sb,ijb,sc,ijc,thresh,nmax)
```

```
!
! Multiply At*A
!
! Input for sa must be At
! Input for sb must be At
! output of sc will be At*A
!
implicit none

integer(4)          :: n, nmax
integer(4)          :: i, ijma, ijmb, j, k, ma, mb, mbb

integer(4), dimension(nmax) :: ija, ijb, ijc
real(8) , dimension(nmax) :: sa, sb, sc

real(8)             :: sum, thresh

if(ija(1) /= ijb(1)) pause 'sprstm sizes do not match'
k = ija(1)
ijc(1) = k
do i = 1, ija(1)-2
  do j = 1, ijb(1)-2
    if(i==j) then
      sum = sa(i)*sb(j)
    else
      sum = 0.d0
    endif
    mb = ijb(j)
    do ma = ija(i), ija(i+1)-1
      ijma = ija(ma)
      if(ijma == j) then
        sum = sum + sa(ma)*sb(j)
      else
2       if(mb < ijb(j+1)) then
          ijmb = ijb(mb)
          if(ijmb == i) then
            sum = sum + sa(i)*sb(mb)
            mb = mb + 1
            goto 2
          else if(ijmb < ijma) then
            mb = mb + 1
            goto 2
          else if(ijmb == ijma) then
            sum = sum + sa(ma)*sb(mb)
            mb = mb + 1
            goto 2
          endif
        endif
      endif
    enddo

    do mbb = mb, ijb(j+1)-1
      if(ijb(mbb) == i) then
        sum = sum + sa(i)*sb(mbb)
      endif
    enddo

    if(i == j) then
      sc(i) = sum
    else if(abs(sum) > thresh) then
      if(k > nmax) pause 'sprstm: nmax too small'
      sc(k) = sum
      ijc(k) = j
      k = k + 1
    endif
  enddo
  ijc(i+1) = k
enddo
end subroutine sprstm
```

```
!-----
```

```
subroutine sprsint(a,n,np,thresh,nmax,sa,ija)
```

```
implicit none

integer(4)          :: n, nmax, np
integer(4)          :: i, j, k
integer(4), dimension(nmax) :: ija
```

```
real(8)                :: thresh
real(8), dimension(np,np) :: a
real(8), dimension(nmax)  :: sa

do j = 1,n
  sa(j) = a(j,j)
enddo
ija(1) = n+2
k = n+1
do j = 1,n
  do i = 1,n
    if (abs(a(i,j)) >= thresh) then
      if (i /= j) then
        k = k+1
        if (k > nmax) pause 'nmax too small in sprsin'
        sa(k) = a(i,j)
        ija(k) = i
      endif
    endif
  enddo
  ija(j+1) = k+1
enddo

END subroutine sprsint
!-----
end module solve
```



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก ข

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข
รายละเอียดของโปรแกรมวิเคราะห์ปัญหาความเค้น
อันเนื่องมาจากอุณหภูมิ

```

Program SOLID_STRESS

use SOLID

implicit none

call MAIN()

end Program SOLID_STRESS

!-----
MODULE SOLID

implicit none

character(len=20)          :: namel

integer(4), allocatable, dimension(:,:) :: intmat          ! Node connection on element
integer(4), allocatable, dimension(:)   :: ibcx, ibcy       ! boundary condition for variable
integer(4), dimension(20)              :: text              ! buffer for reading

real(8), allocatable, dimension(:,:)    :: coord, sysk      ! Coordinate of each node
real(8), allocatable, dimension(:)      :: p, displx, displx
real(8), allocatable, dimension(:)      :: sysf, temp,   sxx, syy, sxy, sigma
real(8), allocatable, dimension(:)      :: sxxex, syyex, sxyex

integer(4)                        :: neq, npoi, nelem, ierror, ip, ie
integer(4)                        :: i, j, k, ii, jj, kk
integer(4)                        :: ll, mm, nn
integer(4)                        :: nlines, iline
integer(4)                        :: nside, nboun

real(8)                            :: xg1, xg2, xg3, yg1, yg2, yg3, theta
real(8)                            :: b1, b2, b3, c1, c2, c3
real(8)                            :: uj, vj
real(8)                            :: area, x, y, sum, xx, yy
real(8)                            :: elas, pr, tref, alpha, thick
real(8)                            :: c11, c12, c21, c22, c33, lfa
real(8)                            :: aread, efind
real(8)                            :: rprfb, rprst, rprfe, ergfb, ergst, ergfe
real(8)                            :: maxlfb, maxlst, maxlfe

real(8), dimension(12,12)           :: kele
real(8), dimension(12)              :: r, rsur, rtemp

integer(4)                          :: tnode, stype, exact
integer(4), allocatable, dimension(:,:) :: number
integer(4), allocatable, dimension(:,,:) :: acheck

CONTAINS

!-----
Subroutine MAIN()

use solve

implicit none

call READ_INPUT()
call GEN_NL()
call ARRANGE()
call CST()
call SYMPCG(sysk, sysf, p, neq)
call STRESS()
call ERR_EST()
call WRITE_OUTPUT()

End Subroutine MAIN

!-----

```

```

Subroutine read_input()

implicit none

write(6,10)
10 format(/, ' PLEASE ENTER INPUT FILE NAME:',/)
read(5, '(A)') namel

open(unit=7, file=namel, status='OLD', action='read', iostat=ierror)

!
! read input data
!

read(7,*) nlines

do iline = 1,nlines
  read(7,1) text
  1 format(20A4)
enddo

read(7,1) text
read(7,*) tnode, nelem, nboun, stype, exact

nside = (3*nelem+nboun)/2
npoi = tnode+nside
neq = 2*npoi

allocate(      intmat(nelem,6),      coord(npoi,2),      p(neq))
allocate(      ibcx(npoi),      ibcy(npoi), displx(npoi), disply(npoi), temp(tnode))
allocate(      sysk(neq,neq),      sysf(neq),      sxx(npoi),      syy(npoi),      sxy(npoi))
allocate( number(tnode,tnode),      acheck(tnode,tnode,2))

if (exact == 1) allocate(sxxex(tnode), syyex(tnode), sxyex(tnode))

read(7,1) text
read(7,*) elas, pr, tref, alpha, thick
read(7,1) text
do ip = 1,tnode
  read(7,*) i, ibcx(i), ibcy(i), (coord(i,k), k=1,2), displx(i), disply(i), &
    sxx(i), syy(i), temp(i))
  if(i /= ip) write(6,40) ip
  40 format(/, ' NODE NO.', I5, ' IN DATA FILE IS MISSING')
  if(i /= ip) stop
enddo

read(7,1) text
do ie = 1,nelem
  read(7,*) i, (intmat(i,j), j=1,3)
  if(i /= ie) write(6,50) ie
  50 format(/, ' ELEMENT NO.', I5, ' IN DATA FILE IS MISSING')
  if(i /= ie) stop
enddo

close(unit=7, status='keep')

write(6,60) tnode, nelem
60 format(/, ' THE FINITE ELEMENT MODEL CONSISTS OF :',/, &
  ' NUMBER OF TIP NODES =', I6, '/', &
  ' NUMBER OF ELEMENTS =', I6, '/')

End Subroutine read_input

!-----
Subroutine CST()

implicit none

sysf = 0.
sysk = 0.

if (stype == 0) then
  c11 =      elas / (1 - pr*pr)
  c12 =      pr * elas / (1 - pr*pr)
  c33 = (1 - pr) * elas / (1 - pr*pr)/2.
  lfa = alpha
endif

if (stype == 1) then
  c11 =      (1 - pr) * elas / (1 + pr) / (1 - 2.*pr)
  c12 =      pr * elas / (1 + pr) / (1 - 2.*pr)
  c33 = (1 - 2.*pr) * elas / (1 + pr) / (1 - 2.*pr)/2.
  lfa = alpha*(1+pr)
endif

```

```

c21 = c12
c22 = c11

!
! Loop over the number of elements:
!

const: do ie = 1,nelem

! Find element local coordinates:

ii = intmat(ie,1)
jj = intmat(ie,2)
kk = intmat(ie,3)
ll = intmat(ie,4)
mm = intmat(ie,5)
nn = intmat(ie,6)

xg1 = coord(ii,1)
xg2 = coord(jj,1)
xg3 = coord(kk,1)

yg1 = coord(ii,2)
yg2 = coord(jj,2)
yg3 = coord(kk,2)

area = 0.5*(xg2*(yg3-yg1)+xg1*(yg2-yg3)+xg3*(yg1-yg2))

b1 = (yg2 - yg3)
b2 = (yg3 - yg1)
b3 = (yg1 - yg2)

c1 = (xg3 - xg2)
c2 = (xg1 - xg3)
c3 = (xg2 - xg1)

! Setup & clear matrices:

kele = 0.
rsur = 0.
rtemp = 0.
r = 0.

! Set the value for
! the coefficient of diffusion term

call CnK_ELE()

! Compute all loads

call ALL_LOAD()

! Assemble element

call asmp()

enddo const

call applbc()

end Subroutine CST

!-----
Subroutine CnK_ELE()
implicit none

!
! upper left (quadrant 2)
!
kele(1,1) = 12.*(c11*b1*b1 + c33*c1*c1)
kele(1,2) = 12.*(c11*b1*b2 + c33*c1*c2)
kele(1,3) = 12.*(c11*b1*b3 + c33*c1*c3)
kele(2,1) = kele(1,2)
kele(2,2) = 12.*(c11*b2*b2 + c33*c2*c2)
kele(2,3) = 12.*(c11*b2*b3 + c33*c2*c3)
kele(3,1) = kele(1,3)
kele(3,2) = kele(2,3)
kele(3,3) = 12.*(c11*b3*b3 + c33*c3*c3)

kele(1,4) = 4.*(c11*b1*(b2+b3) + c33*c1*(c2+c3))
kele(2,4) = 4.*(c11*b2*(b2+b3) + c33*c2*(c2+c3))
kele(3,4) = 4.*(c11*b3*(b2+b3) + c33*c3*(c2+c3))
kele(1,5) = 4.*(c11*b1*(b3+b1) + c33*c1*(c3+c1))

```



```

kele(2,5) = 4.*(c11*b2*(b3+b1) + c33*c2*(c3+c1))
kele(3,5) = 4.*(c11*b3*(b3+b1) + c33*c3*(c3+c1))
kele(1,6) = 4.*(c11*b1*(b1+b2) + c33*c1*(c1+c2))
kele(2,6) = 4.*(c11*b2*(b1+b2) + c33*c2*(c1+c2))
kele(3,6) = 4.*(c11*b3*(b1+b2) + c33*c3*(c1+c2))

kele(4,1) = kele(1,4)
kele(4,2) = kele(2,4)
kele(4,3) = kele(3,4)
kele(5,1) = kele(1,5)
kele(5,2) = kele(2,5)
kele(5,3) = kele(3,5)
kele(6,1) = kele(1,6)
kele(6,2) = kele(2,6)
kele(6,3) = kele(3,6)

kele(4,4) = c11*(2.*b2*b2 + 2.*b2*b3 + 2.*b3*b3) + c33*(2.*c2*c2 + 2.*c2*c3 + 2.*c3*c3)
kele(5,5) = c11*(2.*b1*b1 + 2.*b1*b3 + 2.*b3*b3) + c33*(2.*c1*c1 + 2.*c1*c3 + 2.*c3*c3)
kele(6,6) = c11*(2.*b1*b1 + 2.*b1*b2 + 2.*b2*b2) + c33*(2.*c1*c1 + 2.*c1*c2 + 2.*c2*c2)
kele(4,5) = c11*(b3*b3 + b1*b3 + b2*b3 + 2.*b1*b2) + c33*(c3*c3 + c1*c3 + c2*c3 + 2.*c1*c2)
kele(4,6) = c11*(b2*b2 + b1*b2 + b2*b3 + 2.*b1*b3) + c33*(c2*c2 + c1*c2 + c2*c3 + 2.*c1*c3)
kele(5,6) = c11*(b1*b1 + b1*b2 + b1*b3 + 2.*b2*b3) + c33*(c1*c1 + c1*c2 + c1*c3 + 2.*c2*c3)
kele(5,4) = kele(4,5)
kele(6,4) = kele(4,6)
kele(6,5) = kele(5,6)

!
! lower right (quardrant 4)
!
kele(7,7) = 12.*(c33*b1*b1 + c22*c1*c1)
kele(7,8) = 12.*(c33*b1*b2 + c22*c1*c2)
kele(7,9) = 12.*(c33*b1*b3 + c22*c1*c3)
kele(8,7) = kele(7,8)
kele(8,8) = 12.*(c33*b2*b2 + c22*c2*c2)
kele(8,9) = 12.*(c33*b2*b3 + c22*c2*c3)
kele(9,7) = kele(7,9)
kele(9,8) = kele(8,9)
kele(9,9) = 12.*(c33*b3*b3 + c22*c3*c3)

kele(7,10) = 4.*(c33*b1*(b2+b3) + c22*c1*(c2+c3))
kele(8,10) = 4.*(c33*b2*(b2+b3) + c22*c2*(c2+c3))
kele(9,10) = 4.*(c33*b3*(b2+b3) + c22*c3*(c2+c3))
kele(7,11) = 4.*(c33*b1*(b3+b1) + c22*c1*(c3+c1))
kele(8,11) = 4.*(c33*b2*(b3+b1) + c22*c2*(c3+c1))
kele(9,11) = 4.*(c33*b3*(b3+b1) + c22*c3*(c3+c1))
kele(7,12) = 4.*(c33*b1*(b1+b2) + c22*c1*(c1+c2))
kele(8,12) = 4.*(c33*b2*(b1+b2) + c22*c2*(c1+c2))
kele(9,12) = 4.*(c33*b3*(b1+b2) + c22*c3*(c1+c2))

kele(10,7) = kele(7,10)
kele(10,8) = kele(8,10)
kele(10,9) = kele(9,10)
kele(11,7) = kele(7,11)
kele(11,8) = kele(8,11)
kele(11,9) = kele(9,11)
kele(12,7) = kele(7,12)
kele(12,8) = kele(8,12)
kele(12,9) = kele(9,12)

kele(10,10) = c33*(2.*b2*b2 + 2.*b2*b3 + 2.*b3*b3) + c22*(2.*c2*c2 + 2.*c2*c3 + 2.*c3*c3)
kele(11,11) = c33*(2.*b1*b1 + 2.*b1*b3 + 2.*b3*b3) + c22*(2.*c1*c1 + 2.*c1*c3 + 2.*c3*c3)
kele(12,12) = c33*(2.*b1*b1 + 2.*b1*b2 + 2.*b2*b2) + c22*(2.*c1*c1 + 2.*c1*c2 + 2.*c2*c2)
kele(10,11) = c33*(b3*b3 + b1*b3 + b2*b3 + 2.*b1*b2) + c22*(c3*c3 + c1*c3 + c2*c3 + 2.*c1*c2)
kele(10,12) = c33*(b2*b2 + b1*b2 + b2*b3 + 2.*b1*b3) + c22*(c2*c2 + c1*c2 + c2*c3 + 2.*c1*c3)
kele(11,12) = c33*(b1*b1 + b1*b2 + b1*b3 + 2.*b2*b3) + c22*(c1*c1 + c1*c2 + c1*c3 + 2.*c2*c3)
kele(11,10) = kele(10,11)
kele(12,10) = kele(10,12)
kele(12,11) = kele(11,12)

!
! upper right (quardrant 1)
!
kele(1,7) = 12.*(c12*b1*c1 + c33*b1*c1)
kele(2,7) = 12.*(c12*b2*c1 + c33*b1*c2)
kele(3,7) = 12.*(c12*b3*c1 + c33*b1*c3)
kele(1,8) = 12.*(c12*b1*c2 + c33*b2*c1)
kele(2,8) = 12.*(c12*b2*c2 + c33*b2*c2)
kele(3,8) = 12.*(c12*b3*c2 + c33*b2*c3)
kele(1,9) = 12.*(c12*b1*c3 + c33*b3*c1)
kele(2,9) = 12.*(c12*b2*c3 + c33*b3*c2)
kele(3,9) = 12.*(c12*b3*c3 + c33*b3*c3)

kele(1,10) = 4.*(c12*b1*(c3+c2) + c33*(b3+b2)*c1)
kele(2,10) = 4.*(c12*b2*(c3+c2) + c33*(b3+b2)*c2)
kele(3,10) = 4.*(c12*b3*(c3+c2) + c33*(b3+b2)*c3)

```

```

kele(1,11) = 4.*(c12*b1*(c1+c3) + c33*(b1+b3)*c1)
kele(2,11) = 4.*(c12*b2*(c1+c3) + c33*(b1+b3)*c2)
kele(3,11) = 4.*(c12*b3*(c1+c3) + c33*(b1+b3)*c3)
kele(1,12) = 4.*(c12*b1*(c2+c1) + c33*(b2+b1)*c1)
kele(2,12) = 4.*(c12*b2*(c2+c1) + c33*(b2+b1)*c2)
kele(3,12) = 4.*(c12*b3*(c2+c1) + c33*(b2+b1)*c3)

kele(4,7) = 4.*(c12*(b3+b2)*c1 + c33*b1*(c3+c2))
kele(5,7) = 4.*(c12*(b1+b3)*c1 + c33*b1*(c1+c3))
kele(6,7) = 4.*(c12*(b2+b1)*c1 + c33*b1*(c2+c1))
kele(4,8) = 4.*(c12*(b3+b2)*c2 + c33*b2*(c3+c2))
kele(5,8) = 4.*(c12*(b1+b3)*c2 + c33*b2*(c1+c3))
kele(6,8) = 4.*(c12*(b2+b1)*c2 + c33*b2*(c2+c1))
kele(4,9) = 4.*(c12*(b3+b2)*c3 + c33*b3*(c3+c2))
kele(5,9) = 4.*(c12*(b1+b3)*c3 + c33*b3*(c1+c3))
kele(6,9) = 4.*(c12*(b2+b1)*c3 + c33*b3*(c2+c1))

kele(4,10) = c12*(2.*b3*c3 + b3*c2 + b2*c3 + 2.*b2*c2) + &
c33*(2.*b3*c3 + b3*c2 + b2*c3 + 2.*b2*c2)
kele(5,10) = c12*( b1*c3 + 2.*b1*c2 + b3*c3 + b3*c2) + &
c33*( b3*c1 + b3*c3 + 2.*b2*c1 + b2*c3)
kele(6,10) = c12*( b2*c3 + b2*c2 + 2.*b1*c3 + b1*c2) + &
c33*( b3*c2 + 2.*b3*c1 + b2*c2 + b2*c1)
kele(4,11) = c12*( b3*c1 + b3*c3 + 2.*b2*c1 + b2*c3) + &
c33*( b1*c3 + 2.*b1*c2 + b3*c3 + b3*c2)
kele(5,11) = c12*(2.*b1*c1 + b1*c3 + b3*c1 + 2.*b3*c3) + &
c33*(2.*b1*c1 + b1*c3 + b3*c1 + 2.*b3*c3)
kele(6,11) = c12*( b2*c1 + 2.*b2*c3 + b1*c1 + b1*c3) + &
c33*( b1*c2 + b1*c1 + 2.*b3*c2 + b3*c1)
kele(4,12) = c12*( b3*c2 + 2.*b3*c1 + b2*c2 + b2*c1) + &
c33*( b2*c3 + b2*c2 + 2.*b1*c3 + b1*c2)
kele(5,12) = c12*( b1*c2 + b1*c1 + 2.*b3*c2 + b3*c1) + &
c33*( b2*c1 + 2.*b2*c3 + b1*c1 + b1*c3)
kele(6,12) = c12*(2.*b2*c2 + b2*c1 + b1*c2 + 2.*b1*c1) + &
c33*(2.*b2*c2 + b2*c1 + b1*c2 + 2.*b1*c1)
!
! lower left (quadrant 3)
!
kele(7,1) = kele(1,7)
kele(7,2) = kele(2,7)
kele(7,3) = kele(3,7)
kele(7,4) = kele(4,7)
kele(7,5) = kele(5,7)
kele(7,6) = kele(6,7)

kele(8,1) = kele(1,8)
kele(8,2) = kele(2,8)
kele(8,3) = kele(3,8)
kele(8,4) = kele(4,8)
kele(8,5) = kele(5,8)
kele(8,6) = kele(6,8)

kele(9,1) = kele(1,9)
kele(9,2) = kele(2,9)
kele(9,3) = kele(3,9)
kele(9,4) = kele(4,9)
kele(9,5) = kele(5,9)
kele(9,6) = kele(6,9)

kele(10,1) = kele(1,10)
kele(10,2) = kele(2,10)
kele(10,3) = kele(3,10)
kele(10,4) = kele(4,10)
kele(10,5) = kele(5,10)
kele(10,6) = kele(6,10)

kele(11,1) = kele(1,11)
kele(11,2) = kele(2,11)
kele(11,3) = kele(3,11)
kele(11,4) = kele(4,11)
kele(11,5) = kele(5,11)
kele(11,6) = kele(6,11)

kele(12,1) = kele(1,12)
kele(12,2) = kele(2,12)
kele(12,3) = kele(3,12)
kele(12,4) = kele(4,12)
kele(12,5) = kele(5,12)
kele(12,6) = kele(6,12)

kele = kele*thick/area/48.

End Subroutine CnK_ELE
!-----

```

```

Subroutine ALL_LOAD()

implicit none

real(8)                :: dx, dy, dl, SL, tavg, Q, xx, yy

real(8), dimension(12,3) :: bmat
real(8), dimension( 3,3) :: cmat
real(8), dimension(3)    :: lmat

! surface load

if ((sxx(ii) /= 0.) .and. (sxx(jj) /= 0.)) then

  dx = xg2 - xg1
  dy = yg2 - yg1
  dl = sqrt(dx*dx + dy*dy)

  SL = (sxx(ii) + sxx(jj))/2.

  rsur(1) = SL*thick*dl/2.
  rsur(2) = SL*thick*dl/2.
  rsur(6) = 0.

endif

if ((sxx(jj) /= 0.) .and. (sxx(kk) /= 0.)) then

  dx = xg3 - xg2
  dy = yg3 - yg2
  dl = sqrt(dx*dx + dy*dy)

  SL = (sxx(jj) + sxx(kk))/2.

  rsur(2) = SL*thick*dl/2.
  rsur(3) = SL*thick*dl/2.
  rsur(4) = 0.

endif

if ((sxx(kk) /= 0.) .and. (sxx(ii) /= 0.)) then

  dx = xg3 - xg1
  dy = yg3 - yg1
  dl = sqrt(dx*dx + dy*dy)

  SL = (sxx(kk) + sxx(ii))/2.

  rsur(1) = SL*thick*dl/2.
  rsur(3) = SL*thick*dl/2.
  rsur(5) = 0.

endif

if ((syy(ii) /= 0.) .and. (syy(jj) /= 0.)) then

  dx = xg2 - xg1
  dy = yg2 - yg1
  dl = sqrt(dx*dx + dy*dy)

  SL = (syy(ii) + syy(jj))/2.

  rsur(7) = SL*thick*dl/2.
  rsur(8) = SL*thick*dl/2.
  rsur(12) = 0.

endif

if ((syy(jj) /= 0.) .and. (syy(kk) /= 0.)) then

  dx = xg3 - xg2
  dy = yg3 - yg2
  dl = sqrt(dx*dx + dy*dy)

  SL = (syy(jj) + syy(kk))/2.

  rsur(8) = SL*thick*dl/2.
  rsur(9) = SL*thick*dl/2.
  rsur(10) = 0.

endif

if ((syy(kk) /= 0.) .and. (syy(ii) /= 0.)) then

  dx = xg3 - xg1

```

```

dy = yg3 - yg1
dl = sqrt(dx*dx + dy*dy)

SL = (syy(kk) + syy(ii))/2.

rsur(7) = SL*thick*d1/2.
rsur(9) = SL*thick*d1/2.
rsur(11) = 0.

endif

! Compute Constant Surface Heating
! with specified function

xx = (xg1+xg2+xg3)/3.
yy = (yg1+yg2+yg3)/3.
Q = ((xx*xx+yy*yy)*(xx*3.+yy*yy)-6.*xx-2.)*dsin(xx*yy) - 2.*xx*yy*(3.*xx+2.)*dcos(xx*yy)
rsur(1) = Q*area*thick/3.
rsur(2) = rsur(1)
rsur(3) = rsur(1)
rsur(4) = 0.
rsur(5) = rsur(4)
rsur(6) = rsur(4)

! average element temperature

tavg = (temp(ii) + temp(jj) + temp(kk))/3.

Q = tavg - tref

bmat = 0.
cmat = 0.
lmat = 0.

bmat(1,1) = b1/2.
bmat(2,1) = b2/2.
bmat(3,1) = b3/2.
bmat(4,1) = (b2+b3)/6.
bmat(5,1) = (b1+b3)/6.
bmat(6,1) = (b2+b1)/6.

bmat(7,2) = c1/2.
bmat(8,2) = c2/2.
bmat(9,2) = c3/2.
bmat(10,2) = (c2+c3)/6.
bmat(11,2) = (c1+c3)/6.
bmat(12,2) = (c2+c1)/6.

bmat(1,3) = c1/2.
bmat(2,3) = c2/2.
bmat(3,3) = c3/2.
bmat(4,3) = (c2+c3)/6.
bmat(5,3) = (c1+c3)/6.
bmat(6,3) = (c2+c1)/6.

bmat(7,3) = b1/2.
bmat(8,3) = b2/2.
bmat(9,3) = b3/2.
bmat(10,3) = (b2+b3)/6.
bmat(11,3) = (b1+b3)/6.
bmat(12,3) = (b2+b1)/6.

cmat(1,1) = c11
cmat(1,2) = c12
cmat(2,1) = c21
cmat(2,2) = c22
cmat(3,3) = c33

lmat(1) = lfa
lmat(2) = lfa

! temp load
rtemp = matmul(bmat,matmul(cmat,lmat))*Q*thick

! total load = surface load + temp load

r = rsur + rtemp

End Subroutine ALL_LOAD

!-----

```



```

Subroutine ASMP()
implicit none
! Assembling system load vector
! Contribution of coefficients associated with psi
do i = 1,6
  do j = 1,6
    ii = intmat(ie,i)
    jj = intmat(ie,j)
    sysk(ii ,jj ) = sysk(ii ,jj ) + kele(i ,j )
    sysk(ii+npoi,jj ) = sysk(ii+npoi,jj ) + kele(i+6,j )
    sysk(ii ,jj+npoi) = sysk(ii ,jj+npoi) + kele(i ,j+6)
    sysk(ii+npoi,jj+npoi) = sysk(ii+npoi,jj+npoi) + kele(i+6,j+6)
  enddo
  sysf(ii) = sysf(ii) + r(i)
  sysf(ii+npoi) = sysf(ii+npoi) + r(i+6)
enddo
End Subroutine ASMP
!-----
Subroutine applbc()
implicit none
integer(4)      ::   ieq, ir, ic
outer1: do ieq = 1,npoi
  if(ibcx(ieq) == 0) cycle outer1
  inner1: do ir = 1,neq
    if(ir == ieq) cycle inner1
    sysf(ir) = sysf(ir) - sysk(ir,ieq)*p(ieq)
    sysk(ir,ieq) = 0.
  enddo inner1
  do ic = 1,neq
    sysk(ieq,ic) = 0.
  enddo
  sysk(ieq,ieq) = 1.
  sysf(ieq) = p(ieq)
enddo outer1
outer: do ieq = 1,npoi
  if(ibcy(ieq) == 0) cycle outer
  inner: do ir = 1,neq
    if(ir == ieq) cycle inner
    sysf(ir) = sysf(ir) - sysk(ir,ieq+npoi)*p(ieq+npoi)
    sysk(ir,ieq+npoi) = 0.
  enddo inner
  do ic = 1,neq
    sysk(ieq+npoi,ic) = 0.
  enddo
  sysk(ieq+npoi,ieq+npoi) = 1.
  sysf(ieq+npoi) = p(ieq+npoi)
enddo outer
End Subroutine applbc
!-----
Subroutine STRESS()
use solve
implicit none
integer(4)      :: 1, m, nnode, ndf, nsr, ner, nsc, nec

```

```

real(8)                :: tavg

real(8), dimension(3)  :: lmat
real(8), dimension(9)  :: lvec
real(8), dimension(12) :: tbar
real(8), dimension(3,3) :: cmat
real(8), dimension(9,3) :: pmat
real(8), dimension(9,9) :: mele
real(8), dimension(9,12) :: gmat

deallocate(sysk, sysf, stat=ierror)
allocate(sysk(3*tnode,3*tnode), sysf(3*tnode), sigma(3*tnode))

sxx = 0.
syy = 0.
sxy = 0.
sysk = 0.
sysf = 0.
cmat = 0.
lmat = 0.
sigma = 0.

cmat(1,1) = c11
cmat(1,2) = c12
cmat(2,1) = c21
cmat(2,2) = c22
cmat(3,3) = c33

lmat(1) = lfa
lmat(2) = lfa

!
! Setup element matrices to compute nodal stress :
!

Do ie=1,nelem

  ii = intmat(ie,1)
  jj = intmat(ie,2)
  kk = intmat(ie,3)
  ll = intmat(ie,4)
  mm = intmat(ie,5)
  nn = intmat(ie,6)

  xg1 = coord(ii,1)
  xg2 = coord(jj,1)
  xg3 = coord(kk,1)

  yg1 = coord(ii,2)
  yg2 = coord(jj,2)
  yg3 = coord(kk,2)

  area = 0.5*(xg2*(yg3-yg1)+xg1*(yg2-yg3)+xg3*(yg1-yg2))

  b1 = (yg2 - yg3)
  b2 = (yg3 - yg1)
  b3 = (yg1 - yg2)

  c1 = (xg3 - xg2)
  c2 = (xg1 - xg3)
  c3 = (xg2 - xg1)

  tbar(1) = p(ii)
  tbar(2) = p(jj)
  tbar(3) = p(kk)
  tbar(4) = p(ll)
  tbar(5) = p(mm)
  tbar(6) = p(nn)
  tbar(7) = p(ii + npoi)
  tbar(8) = p(jj + npoi)
  tbar(9) = p(kk + npoi)
  tbar(10) = p(ll + npoi)
  tbar(11) = p(mm + npoi)
  tbar(12) = p(nn + npoi)

! Setup mass matrix [M] :

mele = 0.

mele(1,1) = 2.
mele(1,4) = 1.
mele(1,7) = 1.
mele(2,2) = 2.
mele(2,5) = 1.

```

```

mele(2,8) = 1.
mele(3,3) = 2.
mele(3,6) = 1.
mele(3,9) = 1.
mele(4,1) = 1.
mele(4,4) = 2.
mele(4,7) = 1.
mele(5,2) = 1.
mele(5,5) = 2.
mele(5,8) = 1.
mele(6,3) = 1.
mele(6,6) = 2.
mele(6,9) = 1.
mele(7,1) = 1.
mele(7,4) = 1.
mele(7,7) = 2.
mele(8,2) = 1.
mele(8,5) = 1.
mele(8,8) = 2.
mele(9,3) = 1.
mele(9,6) = 1.
mele(9,9) = 2.

```

```
mele = mele*area/12.
```

```
! Setup mass matrix [G] :
```

```

gmat(1,1) = 4.*c11*b1
gmat(1,2) = 4.*c11*b2
gmat(1,3) = 4.*c11*b3
gmat(2,1) = 4.*c21*b1
gmat(2,2) = 4.*c21*b2
gmat(2,3) = 4.*c21*b3
gmat(3,1) = 4.*c33*c1
gmat(3,2) = 4.*c33*c2
gmat(3,3) = 4.*c33*c3

```

```

gmat(4,1) = gmat(1,1)
gmat(4,2) = gmat(1,2)
gmat(4,3) = gmat(1,3)
gmat(5,1) = gmat(2,1)
gmat(5,2) = gmat(2,2)
gmat(5,3) = gmat(2,3)
gmat(6,1) = gmat(3,1)
gmat(6,2) = gmat(3,2)
gmat(6,3) = gmat(3,3)

```

```

gmat(7,1) = gmat(1,1)
gmat(7,2) = gmat(1,2)
gmat(7,3) = gmat(1,3)
gmat(8,1) = gmat(2,1)
gmat(8,2) = gmat(2,2)
gmat(8,3) = gmat(2,3)
gmat(9,1) = gmat(3,1)
gmat(9,2) = gmat(3,2)
gmat(9,3) = gmat(3,3)

```

```

gmat(1,4) = c11*( b2 + b3)
gmat(1,5) = c11*( b1 + 2.*b3)
gmat(1,6) = c11*( b1 + 2.*b2)
gmat(2,4) = c21*( b2 + b3)
gmat(2,5) = c21*( b1 + 2.*b3)
gmat(2,6) = c21*( b1 + 2.*b2)
gmat(3,4) = c33*( c2 + c3)
gmat(3,5) = c33*( c1 + 2.*c3)
gmat(3,6) = c33*( c1 + 2.*c2)

```

```

gmat(4,4) = c11*( b2 + 2.*b3)
gmat(4,5) = c11*( b1 + b3)
gmat(4,6) = c11*( 2.*b1 + b2)
gmat(5,4) = c21*( b2 + 2.*b3)
gmat(5,5) = c21*( b1 + b3)
gmat(5,6) = c21*( 2.*b1 + b2)
gmat(6,4) = c33*( c2 + 2.*c3)
gmat(6,5) = c33*( c1 + c3)
gmat(6,6) = c33*( 2.*c1 + c2)

```

```

gmat(7,4) = c11*( 2.*b2 + b3)
gmat(7,5) = c11*( 2.*b1 + b3)
gmat(7,6) = c11*( b1 + b2)
gmat(8,4) = c21*( 2.*b2 + b3)
gmat(8,5) = c21*( 2.*b1 + b3)
gmat(8,6) = c21*( b1 + b2)
gmat(9,4) = c33*( 2.*c2 + c3)
gmat(9,5) = c33*( 2.*c1 + c3)

```

```

gmat(9,6) = c33*( c1 + c2)

gmat(1,7) = 4.*c21*c1
gmat(1,8) = 4.*c21*c2
gmat(1,9) = 4.*c21*c3
gmat(2,7) = 4.*c11*c1
gmat(2,8) = 4.*c11*c2
gmat(2,9) = 4.*c11*c3
gmat(3,7) = 4.*c33*b1
gmat(3,8) = 4.*c33*b2
gmat(3,9) = 4.*c33*b3

gmat(4,7) = gmat(1,7)
gmat(4,8) = gmat(1,8)
gmat(4,9) = gmat(1,9)
gmat(5,7) = gmat(2,7)
gmat(5,8) = gmat(2,8)
gmat(5,9) = gmat(2,9)
gmat(6,7) = gmat(3,7)
gmat(6,8) = gmat(3,8)
gmat(6,9) = gmat(3,9)

gmat(7,7) = gmat(1,7)
gmat(7,8) = gmat(1,8)
gmat(7,9) = gmat(1,9)
gmat(8,7) = gmat(2,7)
gmat(8,8) = gmat(2,8)
gmat(8,9) = gmat(2,9)
gmat(9,7) = gmat(3,7)
gmat(9,8) = gmat(3,8)
gmat(9,9) = gmat(3,9)

gmat(1,10) = c21*( c2 + c3)
gmat(1,11) = c21*( c1 + 2.*c3)
gmat(1,12) = c21*( c1 + 2.*c2)
gmat(2,10) = c11*( c2 + c3)
gmat(2,11) = c11*( c1 + 2.*c3)
gmat(2,12) = c11*( c1 + 2.*c2)
gmat(3,10) = c33*( b2 + b3)
gmat(3,11) = c33*( b1 + 2.*b3)
gmat(3,12) = c33*( b1 + 2.*b2)

gmat(4,10) = c21*( c2 + 2.*c3)
gmat(4,11) = c21*( c1 + c3)
gmat(4,12) = c21*(2.*c1 + c2)
gmat(5,10) = c11*( c2 + 2.*c3)
gmat(5,11) = c11*( c1 + c3)
gmat(5,12) = c11*(2.*c1 + c2)
gmat(6,10) = c33*( b2 + 2.*b3)
gmat(6,11) = c33*( b1 + b3)
gmat(6,12) = c33*(2.*b1 + b2)

gmat(7,10) = c21*(2.*c2 + c3)
gmat(7,11) = c21*(2.*c1 + c3)
gmat(7,12) = c21*( c1 + c2)
gmat(8,10) = c11*(2.*c2 + c3)
gmat(8,11) = c11*(2.*c1 + c3)
gmat(8,12) = c11*( c1 + c2)
gmat(9,10) = c33*(2.*b2 + b3)
gmat(9,11) = c33*(2.*b1 + b3)
gmat(9,12) = c33*( b1 + b2)

gmat = gmat/24.

! Setup mass matrix [P] :
pmat = 0.

pmat(1,1) = 1.
pmat(2,2) = 1.
pmat(3,3) = 1.
pmat(4,1) = 1.
pmat(5,2) = 1.
pmat(6,3) = 1.
pmat(7,1) = 1.
pmat(8,2) = 1.
pmat(9,3) = 1.

pmat = pmat*area/3.

! Compute element load vectors :

tavg = (temp(ii) + temp(jj) + temp(kk))/3.
lvec = matmul(gmat,tbar) - matmul(pmat,matmul(cmat,lmat))*(tavg - tref)

```



```

! Assemble element eqn into system eqn

nnode = 3
ndf = 3

do i=1,nnode
  ii = intmat(ie,i)

  do l=1, ndf

!   denote: nsr = row position in the system eqs.
!           ner = row position in the element eqs.
    nsr = (ii-1)*ndf + 1
    ner = ( i-1)*ndf + 1
    sysf(nsr) = sysf(nsr) + lvec(ner)

    do j=1,nnode
      jj = intmat(ie,j)

      do m=1, ndf

!   denote: nsc = column position in the system eqs.
!           nec = column position in the element eqs.
        nsc = (jj-1)*ndf + m
        nec = ( j-1)*ndf + m
        sysk(nsr,nsc) = sysk(nsr,nsc) + mele(ner,nec)

      enddo
    enddo

  enddo
enddo

Enddo

!
! Compute NLFB nodal stress (Sxx, Syy, Sxy) :
!

call sympcg( sysk, sysf, sigma, 3*tnode)

do ip = 1,tnode
  sxx(ip) = sigma(3*ip-2)
  syy(ip) = sigma(3*ip-1)
  sxy(ip) = sigma(3*ip )
enddo

End Subroutine STRESS

!-----
Subroutine ERR_EST()
implicit none

real(8)          :: d1, d2, d3, xx, yy
real(8)          :: pi, rr, zeta

real(8), dimension(1,1)  :: tlgfe, tlgfb
real(8), dimension(1,1)  :: eqfbe, eqste, eqfee

real(8), dimension(1,9)  :: belet, celet
real(8), dimension(1,12) :: tbart
real(8), dimension(9,1)  :: bele, cele
real(8), dimension(9,9)  :: mele
real(8), dimension(9,12) :: gmat
real(8), dimension(12,1) :: tbar
real(8), dimension(12,12) :: hmat

aread = 0.
tlgfb = 0.
tlgfe = 0.
ergfb = 0.
ergst = 0.
ergfe = 0.
maxlfb = 0.
maxlst = 0.
maxlfe = 0.

pi = 4.*atan(1.)

! Compute exact nodal stress :
! with specified function

if (exact == 1) then

```

```

do ip = 1, tnode
  xx = coord(ip,1)
  yy = coord(ip,2)
  rr = dsqrt(xx*xx + yy*yy)
  if (xx == 0.) then
    zeta = 0.5*pi
  else
    zeta = datan(yy/xx)
  endif
  sxxex(ip) = 1. - (1.5*cos(2.*zeta) + cos(4.*zeta))/rr/rr + 1.5*cos(4.*zeta)/rr**4.
  syyex(ip) = - (0.5*cos(2.*zeta) - cos(4.*zeta))/rr/rr - 1.5*cos(4.*zeta)/rr**4.
  sxyex(ip) = - (0.5*sin(2.*zeta) + sin(4.*zeta))/rr/rr + 1.5*sin(4.*zeta)/rr**4.
enddo
endif

! Compute square of L2-norm error both x&y directions
! which in the form |B|[M]{B}, |B|[M]{C}, |C|[M]{C}

d1 = c11*c11 + c12*c12
d2 = 2.*c11*c12
d3 = c33*c33

Do ie = 1,nelem

  ii = intmat(ie,1)
  jj = intmat(ie,2)
  kk = intmat(ie,3)
  ll = intmat(ie,4)
  mm = intmat(ie,5)
  nn = intmat(ie,6)

  xg1 = coord(ii,1)
  xg2 = coord(jj,1)
  xg3 = coord(kk,1)

  yg1 = coord(ii,2)
  yg2 = coord(jj,2)
  yg3 = coord(kk,2)

  area = 0.5*(xg2*(yg3-yg1)+xg1*(yg2-yg3)+xg3*(yg1-yg2))

  b1 = (yg2 - yg3)
  b2 = (yg3 - yg1)
  b3 = (yg1 - yg2)

  c1 = (xg3 - xg2)
  c2 = (xg1 - xg3)
  c3 = (xg2 - xg1)

  aread = aread + area

  tbar(1,1) = p(ii)
  tbar(2,1) = p(jj)
  tbar(3,1) = p(kk)
  tbar(4,1) = p(ll)
  tbar(5,1) = p(mm)
  tbar(6,1) = p(nn)
  tbar(7,1) = p(ii + npoi)
  tbar(8,1) = p(jj + npoi)
  tbar(9,1) = p(kk + npoi)
  tbar(10,1) = p(ll + npoi)
  tbar(11,1) = p(mm + npoi)
  tbar(12,1) = p(nn + npoi)

  tbart = transpose(tbar)

! Setup mass matrix [M] :

mele = 0.

mele(1,1) = 2.
mele(1,4) = 1.
mele(1,7) = 1.
mele(2,2) = 2.
mele(2,5) = 1.
mele(2,8) = 1.
mele(3,3) = 2.
mele(3,6) = 1.
mele(3,9) = 1.
mele(4,1) = 1.
mele(4,4) = 2.
mele(4,7) = 1.
mele(5,2) = 1.
mele(5,5) = 2.
mele(5,8) = 1.

```

```

mele(6,3) = 1.
mele(6,6) = 2.
mele(6,9) = 1.
mele(7,1) = 1.
mele(7,4) = 1.
mele(7,7) = 2.
mele(8,2) = 1.
mele(8,5) = 1.
mele(8,8) = 2.
mele(9,3) = 1.
mele(9,6) = 1.
mele(9,9) = 2.

```

```
mele = mele*area/12.
```

```
! Setup mass matrix [G] :
```

```

gmat(1,1) = 4.*c11*b1
gmat(1,2) = 4.*c11*b2
gmat(1,3) = 4.*c11*b3
gmat(2,1) = 4.*c21*b1
gmat(2,2) = 4.*c21*b2
gmat(2,3) = 4.*c21*b3
gmat(3,1) = 4.*c33*c1
gmat(3,2) = 4.*c33*c2
gmat(3,3) = 4.*c33*c3

```

```

gmat(4,1) = gmat(1,1)
gmat(4,2) = gmat(1,2)
gmat(4,3) = gmat(1,3)
gmat(5,1) = gmat(2,1)
gmat(5,2) = gmat(2,2)
gmat(5,3) = gmat(2,3)
gmat(6,1) = gmat(3,1)
gmat(6,2) = gmat(3,2)
gmat(6,3) = gmat(3,3)

```

```

gmat(7,1) = gmat(1,1)
gmat(7,2) = gmat(1,2)
gmat(7,3) = gmat(1,3)
gmat(8,1) = gmat(2,1)
gmat(8,2) = gmat(2,2)
gmat(8,3) = gmat(2,3)
gmat(9,1) = gmat(3,1)
gmat(9,2) = gmat(3,2)
gmat(9,3) = gmat(3,3)

```

```

gmat(1,4) = c11*( b2 + b3)
gmat(1,5) = c11*( b1 + 2.*b3)
gmat(1,6) = c11*( b1 + 2.*b2)
gmat(2,4) = c21*( b2 + b3)
gmat(2,5) = c21*( b1 + 2.*b3)
gmat(2,6) = c21*( b1 + 2.*b2)
gmat(3,4) = c33*( c2 + c3)
gmat(3,5) = c33*( c1 + 2.*c3)
gmat(3,6) = c33*( c1 + 2.*c2)

```

```

gmat(4,4) = c11*( b2 + 2.*b3)
gmat(4,5) = c11*( b1 + b3)
gmat(4,6) = c11*( 2.*b1 + b2)
gmat(5,4) = c21*( b2 + 2.*b3)
gmat(5,5) = c21*( b1 + b3)
gmat(5,6) = c21*( 2.*b1 + b2)
gmat(6,4) = c33*( c2 + 2.*c3)
gmat(6,5) = c33*( c1 + c3)
gmat(6,6) = c33*( 2.*c1 + c2)

```

```

gmat(7,4) = c11*( 2.*b2 + b3)
gmat(7,5) = c11*( 2.*b1 + b3)
gmat(7,6) = c11*( b1 + b2)
gmat(8,4) = c21*( 2.*b2 + b3)
gmat(8,5) = c21*( 2.*b1 + b3)
gmat(8,6) = c21*( b1 + b2)
gmat(9,4) = c33*( 2.*c2 + c3)
gmat(9,5) = c33*( 2.*c1 + c3)
gmat(9,6) = c33*( c1 + c2)

```

```

gmat(1,7) = 4.*c21*c1
gmat(1,8) = 4.*c21*c2
gmat(1,9) = 4.*c21*c3
gmat(2,7) = 4.*c11*c1
gmat(2,8) = 4.*c11*c2
gmat(2,9) = 4.*c11*c3
gmat(3,7) = 4.*c33*b1
gmat(3,8) = 4.*c33*b2

```

```

gmat(3,9) = 4.*c33*b3

gmat(4,7) = gmat(1,7)
gmat(4,8) = gmat(1,8)
gmat(4,9) = gmat(1,9)
gmat(5,7) = gmat(2,7)
gmat(5,8) = gmat(2,8)
gmat(5,9) = gmat(2,9)
gmat(6,7) = gmat(3,7)
gmat(6,8) = gmat(3,8)
gmat(6,9) = gmat(3,9)

gmat(7,7) = gmat(1,7)
gmat(7,8) = gmat(1,8)
gmat(7,9) = gmat(1,9)
gmat(8,7) = gmat(2,7)
gmat(8,8) = gmat(2,8)
gmat(8,9) = gmat(2,9)
gmat(9,7) = gmat(3,7)
gmat(9,8) = gmat(3,8)
gmat(9,9) = gmat(3,9)

gmat(1,10) = c21*( c2 + c3)
gmat(1,11) = c21*( c1 + 2.*c3)
gmat(1,12) = c21*( c1 + 2.*c2)
gmat(2,10) = c11*( c2 + c3)
gmat(2,11) = c11*( c1 + 2.*c3)
gmat(2,12) = c11*( c1 + 2.*c2)
gmat(3,10) = c33*( b2 + b3)
gmat(3,11) = c33*( b1 + 2.*b3)
gmat(3,12) = c33*( b1 + 2.*b2)

gmat(4,10) = c21*( c2 + 2.*c3)
gmat(4,11) = c21*( c1 + c3)
gmat(4,12) = c21*(2.*c1 + c2)
gmat(5,10) = c11*( c2 + 2.*c3)
gmat(5,11) = c11*( c1 + c3)
gmat(5,12) = c11*(2.*c1 + c2)
gmat(6,10) = c33*( b2 + 2.*b3)
gmat(6,11) = c33*( b1 + b3)
gmat(6,12) = c33*(2.*b1 + b2)

gmat(7,10) = c21*(2.*c2 + c3)
gmat(7,11) = c21*(2.*c1 + c3)
gmat(7,12) = c21*( c1 + c2)
gmat(8,10) = c11*(2.*c2 + c3)
gmat(8,11) = c11*(2.*c1 + c3)
gmat(8,12) = c11*( c1 + c2)
gmat(9,10) = c33*(2.*b2 + b3)
gmat(9,11) = c33*(2.*b1 + b3)
gmat(9,12) = c33*( b1 + b2)

gmat = gmat/24.

! Setup mass matrix [H] :
! upper left (quadrant 2)

hmat(1,1) = 12.*(d1*b1*b1 + d3*c1*c1)
hmat(1,2) = 12.*(d1*b1*b2 + d3*c1*c2)
hmat(1,3) = 12.*(d1*b1*b3 + d3*c1*c3)
hmat(2,1) = hmat(1,2)
hmat(2,2) = 12.*(d1*b2*b2 + d3*c2*c2)
hmat(2,3) = 12.*(d1*b2*b3 + d3*c2*c3)
hmat(3,1) = hmat(1,3)
hmat(3,2) = hmat(2,3)
hmat(3,3) = 12.*(d1*b3*b3 + d3*c3*c3)

hmat(1,4) = 4.*(d1*b1*(b2+b3) + d3*c1*(c2+c3))
hmat(2,4) = 4.*(d1*b2*(b2+b3) + d3*c2*(c2+c3))
hmat(3,4) = 4.*(d1*b3*(b2+b3) + d3*c3*(c2+c3))
hmat(1,5) = 4.*(d1*b1*(b3+b1) + d3*c1*(c3+c1))
hmat(2,5) = 4.*(d1*b2*(b3+b1) + d3*c2*(c3+c1))
hmat(3,5) = 4.*(d1*b3*(b3+b1) + d3*c3*(c3+c1))
hmat(1,6) = 4.*(d1*b1*(b1+b2) + d3*c1*(c1+c2))
hmat(2,6) = 4.*(d1*b2*(b1+b2) + d3*c2*(c1+c2))
hmat(3,6) = 4.*(d1*b3*(b1+b2) + d3*c3*(c1+c2))

hmat(4,1) = hmat(1,4)
hmat(4,2) = hmat(2,4)
hmat(4,3) = hmat(3,4)
hmat(5,1) = hmat(1,5)
hmat(5,2) = hmat(2,5)
hmat(5,3) = hmat(3,5)
hmat(6,1) = hmat(1,6)
hmat(6,2) = hmat(2,6)

```



```

hmat(6,3) = hmat(3,6)

hmat(4,4) = d1*(2.*b2*b2 + 2.*b2*b3 + 2.*b3*b3) + d3*(2.*c2*c2 + 2.*c2*c3 + 2.*c3*c3)
hmat(5,5) = d1*(2.*b1*b1 + 2.*b1*b3 + 2.*b3*b3) + d3*(2.*c1*c1 + 2.*c1*c3 + 2.*c3*c3)
hmat(6,6) = d1*(2.*b1*b1 + 2.*b1*b2 + 2.*b2*b2) + d3*(2.*c1*c1 + 2.*c1*c2 + 2.*c2*c2)
hmat(4,5) = d1*(b3*b3 + b1*b3 + b2*b3 + 2.*b1*b2) + d3*(c3*c3 + c1*c3 + c2*c3 + 2.*c1*c2)
hmat(4,6) = d1*(b2*b2 + b1*b2 + b2*b3 + 2.*b1*b3) + d3*(c2*c2 + c1*c2 + c2*c3 + 2.*c1*c3)
hmat(5,6) = d1*(b1*b1 + b1*b2 + b1*b3 + 2.*b2*b3) + d3*(c1*c1 + c1*c2 + c1*c3 + 2.*c2*c3)
hmat(5,4) = hmat(4,5)
hmat(6,4) = hmat(4,6)
hmat(6,5) = hmat(5,6)

! lower right (quadrant 4)

hmat(7,7) = 12.*(d3*b1*b1 + d1*c1*c1)
hmat(7,8) = 12.*(d3*b1*b2 + d1*c1*c2)
hmat(7,9) = 12.*(d3*b1*b3 + d1*c1*c3)
hmat(8,7) = hmat(7,8)
hmat(8,8) = 12.*(d3*b2*b2 + d1*c2*c2)
hmat(8,9) = 12.*(d3*b2*b3 + d1*c2*c3)
hmat(9,7) = hmat(7,9)
hmat(9,8) = hmat(8,9)
hmat(9,9) = 12.*(d3*b3*b3 + d1*c3*c3)

hmat(7,10) = 4.*(d3*b1*(b2+b3) + d1*c1*(c2+c3))
hmat(8,10) = 4.*(d3*b2*(b2+b3) + d1*c2*(c2+c3))
hmat(9,10) = 4.*(d3*b3*(b2+b3) + d1*c3*(c2+c3))
hmat(7,11) = 4.*(d3*b1*(b3+b1) + d1*c1*(c3+c1))
hmat(8,11) = 4.*(d3*b2*(b3+b1) + d1*c2*(c3+c1))
hmat(9,11) = 4.*(d3*b3*(b3+b1) + d1*c3*(c3+c1))
hmat(7,12) = 4.*(d3*b1*(b1+b2) + d1*c1*(c1+c2))
hmat(8,12) = 4.*(d3*b2*(b1+b2) + d1*c2*(c1+c2))
hmat(9,12) = 4.*(d3*b3*(b1+b2) + d1*c3*(c1+c2))

hmat(10,7) = hmat(7,10)
hmat(10,8) = hmat(8,10)
hmat(10,9) = hmat(9,10)
hmat(11,7) = hmat(7,11)
hmat(11,8) = hmat(8,11)
hmat(11,9) = hmat(9,11)
hmat(12,7) = hmat(7,12)
hmat(12,8) = hmat(8,12)
hmat(12,9) = hmat(9,12)

hmat(10,10) = d3*(2.*b2*b2 + 2.*b2*b3 + 2.*b3*b3) + d1*(2.*c2*c2 + 2.*c2*c3 + 2.*c3*c3)
hmat(11,11) = d3*(2.*b1*b1 + 2.*b1*b3 + 2.*b3*b3) + d1*(2.*c1*c1 + 2.*c1*c3 + 2.*c3*c3)
hmat(12,12) = d3*(2.*b1*b1 + 2.*b1*b2 + 2.*b2*b2) + d1*(2.*c1*c1 + 2.*c1*c2 + 2.*c2*c2)
hmat(10,11) = d3*(b3*b3 + b1*b3 + b2*b3 + 2.*b1*b2) + d1*(c3*c3 + c1*c3 + c2*c3 + 2.*c1*c2)
hmat(10,12) = d3*(b2*b2 + b1*b2 + b2*b3 + 2.*b1*b3) + d1*(c2*c2 + c1*c2 + c2*c3 + 2.*c1*c3)
hmat(11,12) = d3*(b1*b1 + b1*b2 + b1*b3 + 2.*b2*b3) + d1*(c1*c1 + c1*c2 + c1*c3 + 2.*c2*c3)
hmat(11,10) = hmat(10,11)
hmat(12,10) = hmat(10,12)
hmat(12,11) = hmat(11,12)

! upper right (quadrant 1)

hmat(1,7) = 12.*(d2*b1*c1 + d3*b1*c1)
hmat(2,7) = 12.*(d2*b2*c1 + d3*b1*c2)
hmat(3,7) = 12.*(d2*b3*c1 + d3*b1*c3)
hmat(1,8) = 12.*(d2*b1*c2 + d3*b2*c1)
hmat(2,8) = 12.*(d2*b2*c2 + d3*b2*c2)
hmat(3,8) = 12.*(d2*b3*c2 + d3*b2*c3)
hmat(1,9) = 12.*(d2*b1*c3 + d3*b3*c1)
hmat(2,9) = 12.*(d2*b2*c3 + d3*b3*c2)
hmat(3,9) = 12.*(d2*b3*c3 + d3*b3*c3)

hmat(1,10) = 4.*(d2*b1*(c3+c2) + d3*(b3+b2)*c1)
hmat(2,10) = 4.*(d2*b2*(c3+c2) + d3*(b3+b2)*c2)
hmat(3,10) = 4.*(d2*b3*(c3+c2) + d3*(b3+b2)*c3)
hmat(1,11) = 4.*(d2*b1*(c1+c3) + d3*(b1+b3)*c1)
hmat(2,11) = 4.*(d2*b2*(c1+c3) + d3*(b1+b3)*c2)
hmat(3,11) = 4.*(d2*b3*(c1+c3) + d3*(b1+b3)*c3)
hmat(1,12) = 4.*(d2*b1*(c2+c1) + d3*(b2+b1)*c1)
hmat(2,12) = 4.*(d2*b2*(c2+c1) + d3*(b2+b1)*c2)
hmat(3,12) = 4.*(d2*b3*(c2+c1) + d3*(b2+b1)*c3)

hmat(4,7) = 4.*(d2*(b3+b2)*c1 + d3*b1*(c3+c2))
hmat(5,7) = 4.*(d2*(b1+b3)*c1 + d3*b1*(c1+c3))
hmat(6,7) = 4.*(d2*(b2+b1)*c1 + d3*b1*(c2+c1))
hmat(4,8) = 4.*(d2*(b3+b2)*c2 + d3*b2*(c3+c2))
hmat(5,8) = 4.*(d2*(b1+b3)*c2 + d3*b2*(c1+c3))
hmat(6,8) = 4.*(d2*(b2+b1)*c2 + d3*b2*(c2+c1))
hmat(4,9) = 4.*(d2*(b3+b2)*c3 + d3*b3*(c3+c2))
hmat(5,9) = 4.*(d2*(b1+b3)*c3 + d3*b3*(c1+c3))
hmat(6,9) = 4.*(d2*(b2+b1)*c3 + d3*b3*(c2+c1))

```

```

hmat(4,10) = d2*(2.*b3*c3 + b3*c2 + b2*c3 + 2.*b2*c2) + &
d3*(2.*b3*c3 + b3*c2 + b2*c3 + 2.*b2*c2)
hmat(5,10) = d2*( b1*c3 + 2.*b1*c2 + b3*c3 + b3*c2) + &
d3*( b3*c1 + b3*c3 + 2.*b2*c1 + b2*c3)
hmat(6,10) = d2*( b2*c3 + b2*c2 + 2.*b1*c3 + b1*c2) + &
d3*( b3*c2 + 2.*b3*c1 + b2*c2 + b2*c1)
hmat(4,11) = d2*( b3*c1 + b3*c3 + 2.*b2*c1 + b2*c3) + &
d3*( b1*c3 + 2.*b1*c2 + b3*c3 + b3*c2)
hmat(5,11) = d2*(2.*b1*c1 + b1*c3 + b3*c1 + 2.*b3*c3) + &
d3*(2.*b1*c1 + b1*c3 + b3*c1 + 2.*b3*c3)
hmat(6,11) = d2*( b2*c1 + 2.*b2*c3 + b1*c1 + b1*c3) + &
d3*( b1*c2 + b1*c1 + 2.*b3*c2 + b3*c1)
hmat(4,12) = d2*( b3*c2 + 2.*b3*c1 + b2*c2 + b2*c1) + &
d3*( b2*c3 + b2*c2 + 2.*b1*c3 + b1*c2)
hmat(5,12) = d2*( b1*c2 + b1*c1 + 2.*b3*c2 + b3*c1) + &
d3*( b2*c1 + 2.*b2*c3 + b1*c1 + b1*c3)
hmat(6,12) = d2*(2.*b2*c2 + b2*c1 + b1*c2 + 2.*b1*c1) + &
d3*(2.*b2*c2 + b2*c1 + b1*c2 + 2.*b1*c1)

! lower left (quadrant 3)

hmat(7,1) = hmat(1,7)
hmat(7,2) = hmat(2,7)
hmat(7,3) = hmat(3,7)
hmat(7,4) = hmat(4,7)
hmat(7,5) = hmat(5,7)
hmat(7,6) = hmat(6,7)

hmat(8,1) = hmat(1,8)
hmat(8,2) = hmat(2,8)
hmat(8,3) = hmat(3,8)
hmat(8,4) = hmat(4,8)
hmat(8,5) = hmat(5,8)
hmat(8,6) = hmat(6,8)

hmat(9,1) = hmat(1,9)
hmat(9,2) = hmat(2,9)
hmat(9,3) = hmat(3,9)
hmat(9,4) = hmat(4,9)
hmat(9,5) = hmat(5,9)
hmat(9,6) = hmat(6,9)

hmat(10,1) = hmat(1,10)
hmat(10,2) = hmat(2,10)
hmat(10,3) = hmat(3,10)
hmat(10,4) = hmat(4,10)
hmat(10,5) = hmat(5,10)
hmat(10,6) = hmat(6,10)

hmat(11,1) = hmat(1,11)
hmat(11,2) = hmat(2,11)
hmat(11,3) = hmat(3,11)
hmat(11,4) = hmat(4,11)
hmat(11,5) = hmat(5,11)
hmat(11,6) = hmat(6,11)

hmat(12,1) = hmat(1,12)
hmat(12,2) = hmat(2,12)
hmat(12,3) = hmat(3,12)
hmat(12,4) = hmat(4,12)
hmat(12,5) = hmat(5,12)
hmat(12,6) = hmat(6,12)

hmat = hmat/48./area
! Compute exact & approximate total stress :
if (exact == 1) then
    bele(1,1) = sxxex(ii)
    bele(2,1) = syyex(ii)
    bele(3,1) = sxyex(ii)
    bele(4,1) = sxxex(jj)
    bele(5,1) = syyex(jj)
    bele(6,1) = sxyex(jj)
    bele(7,1) = sxxex(kk)
    bele(8,1) = syyex(kk)
    bele(9,1) = sxyex(kk)

    belet = transpose(bele)

    tlgfb = tlgfb + matmul(belet,matmul(mele,bele))
endif
endif

```

```

bele(1,1) = sxx(ii)
bele(2,1) = syy(ii)
bele(3,1) = sxy(ii)
bele(4,1) = sxx(jj)
bele(5,1) = syy(jj)
bele(6,1) = sxy(jj)
bele(7,1) = sxx(kk)
bele(8,1) = syy(kk)
bele(9,1) = sxy(kk)

belet = transpose(bele)

tlgfe = tlgfe + matmul(belet,matmul(mele,bele))

! Compute error of flux-based and standard Nodeless :

if (exact == 1) then

  bele(1,1) = sxxex(ii)
  bele(2,1) = syyex(ii)
  bele(3,1) = sxyex(ii)
  bele(4,1) = sxxex(jj)
  bele(5,1) = syyex(jj)
  bele(6,1) = sxyex(jj)
  bele(7,1) = sxxex(kk)
  bele(8,1) = syyex(kk)
  bele(9,1) = sxyex(kk)

  belet = transpose(bele)

  cele(1,1) = sxx(ii)
  cele(2,1) = syy(ii)
  cele(3,1) = sxy(ii)
  cele(4,1) = sxx(jj)
  cele(5,1) = syy(jj)
  cele(6,1) = sxy(jj)
  cele(7,1) = sxx(kk)
  cele(8,1) = syy(kk)
  cele(9,1) = sxy(kk)

  celet = transpose(cele)

  eqfbe = matmul(belet,matmul(mele,bele)) - 2.*matmul(belet,matmul(mele,cele)) + &
    matmul(celet,matmul(mele,cele))
  eqste = matmul(belet,matmul(mele,bele)) - 2.*matmul(belet,matmul(gmat,tbar)) + &
    matmul(tbart,matmul(hmat,tbar))

  if (abs(eqfbe(1,1)) < 0.0000000001) eqfbe = 0.
  if (abs(eqste(1,1)) < 0.0000000001) eqste = 0.

  ergfb = ergfb + eqfbe(1,1)
  ergst = ergst + eqste(1,1)

  if (eqfbe(1,1) > maxlfb) maxlfb = eqfbe(1,1)
  if (eqste(1,1) > maxlst) maxlst = eqste(1,1)

endif

! Compute error of approximate nodal stress :

cele(1,1) = sxx(ii)
cele(2,1) = syy(ii)
cele(3,1) = sxy(ii)
cele(4,1) = sxx(jj)
cele(5,1) = syy(jj)
cele(6,1) = sxy(jj)
cele(7,1) = sxx(kk)
cele(8,1) = syy(kk)
cele(9,1) = sxy(kk)

celet = transpose(cele)

eqfee = matmul(celet,matmul(mele,cele)) - 2.*matmul(celet,matmul(gmat,tbar)) + &
  matmul(tbart,matmul(hmat,tbar))

if (abs(eqfee(1,1)) < 0.0000000001) eqfee = 0.

ergfe = ergfe + eqfee(1,1)

if (eqfee(1,1) > maxlfe) maxlfe = eqfee(1,1)

Enddo

ergfe = sqrt(ergfe/aread)

```

```

tlgfe = sqrt(tlgfe/aread)
rprfe = ergfe/(ergfe + tlgfe(1,1))*100

if (exact == 1) then

  ergfb = sqrt(ergfb/aread)
  tlgfb = sqrt(tlgfb/aread)
  rprfb = ergfb/(ergfb + tlgfb(1,1))*100

  ergst = sqrt(ergst/aread)
  rprst = ergst/(ergst + tlgfb(1,1))*100
  efind = ergfe/ergst

  write(6,100) ergst
  100 format(' Exact-ST global error is :', e12.4)
  write(6,125) maxlst
  125 format(' Max local EST error is :', e12.4)
  write(6,150) rprst
  150 format(' Relative EST % error is :', e12.4, ' %',/)

  write(6,200) ergfb
  200 format(' Exact-FB global error is :', e12.4)
  write(6,225) maxlfb
  225 format(' Max local EFB error is :', e12.4)
  write(6,250) rprfb
  250 format(' Relative EFB % error is :', e12.4, ' %',/)

  write(6,300) efind
  300 format(' Effectivity Index is :', e12.4,/)

endif

write(6,400) ergfe
400 format(' Approximate global error is :', e12.4)
write(6,425) maxlfe
425 format(' Max local approx error is :', e12.4)
write(6,450) rprfe
450 format(' Relative percentage error is :', e12.4, ' %')

End Subroutine ERR_EST

!-----
Subroutine write_output()
implicit none
integer(4)          :: leng
leng = len_trim(name1) - 4
open(unit=8,file=name1(1:leng)//'.oup',status='unknown',iostat=ierror)

if (exact == 1) then
  write(8,100) ergst
  100 format(' Exact-ST global error is :', e16.6)
  write(8,125) maxlst
  125 format(' Max local EST error is :', e16.6)
  write(8,150) rprst
  150 format(' Relative EST % error is :', e16.6, ' %',/)

  write(8,200) ergfb
  200 format(' Exact-FB global error is :', e16.6)
  write(8,225) maxlfb
  225 format(' Max local EFB error is :', e16.6)
  write(8,250) rprfb
  250 format(' Relative EFB % error is :', e16.6, ' %',/)

  write(8,300) efind
  300 format(' Effectivity Index is :', e16.6,/)
endif

write(8,400) ergfe
400 format(' Approximate global error is :', e16.6)
write(8,425) maxlfe
425 format(' Max local approx error is :', e16.6)
write(8,450) rprfe
450 format(' Relative percentage error is :', e16.6, ' %',/)

write(8,500)
500 format(' Node      X-coor      Y-coor      DisX      DisY      &
          SigXX          SigYY          SigXY')

do i = 1,tnode
  write(8,600) i, coord(i,1), coord(i,2), p(i), p(i+npoi), sxx(i), syy(i), sxy(i)

```



```

        600 format(i6,2f12.6,5e16.6)
    enddo

    do i = tnode+1,npoi
        write(8,700) i, coord(i,1), coord(i,2), p(i), p(i+npoi)
        700 format(i6,2f12.6,2e16.6)
    enddo

close(unit=8, status='keep')

open(unit=9,file=name1(1:leng)//'.plt',status='unknown',iostat=ierror)

write(9,800) tnode, nelem
800 format('Variables = "X-coor","Y-coor","DispX","DispY","SigXX","SigYY","SigXY" &
,/, 'ZONE N=',I6,',E=',I6,',F=FEPOINT,ET=TRIANGLE')

do i = 1,tnode
    write(9,850) coord(i,1), coord(i,2), p(i), p(i+npoi), sxx(i), syy(i), sxy(i)
850 format(2f12.6,5e16.6)
enddo

do ie = 1,nelem
    write(9,900) (intmat(ie,j),j=1,3)
900 format(3i6)
enddo

close(unit=9, status='keep')

End Subroutine write_output

!-----
Subroutine GEN_NL()
implicit none

!
! Generate Nodeless
!

acheck = 0
npoi = tnode

do ip = 1, nside
    displx(ip+tnode) = 0.
    displx(ip+tnode) = 0.
enddo

do ie = 1,nelem
    ii = intmat(ie,1)
    jj = intmat(ie,2)
    kk = intmat(ie,3)

! Node 4 is between Node 2 & 3

    acheck(jj,kk,2) = acheck(jj,kk,2) + 1
    acheck(kk,jj,2) = acheck(kk,jj,2) + 1
    if (acheck(jj,kk,1)==1) goto 10
    acheck(jj,kk,1) = 1
    acheck(kk,jj,1) = 1
    npoi = npoi + 1
    intmat(ie,4) = npoi
    number(jj,kk) = npoi
    number(kk,jj) = npoi
    coord(npoi,1) = (coord(jj,1) + coord(kk,1))/2.
    coord(npoi,2) = (coord(jj,2) + coord(kk,2))/2.

! Check boundary for node 4 <p(4)=0.>

    if (ibcx(jj)==1 .and. ibcx(kk)==1) then
        ibcx(npoi) = 1
        displx(npoi) = 0.
    endif
    if (ibcy(jj)==1 .and. ibcy(kk)==1) then
        ibcy(npoi) = 1
        displx(npoi) = 0.
    endif

    goto 15
10 continue
    intmat(ie,4) = number(jj,kk)
15 continue

```

```

! Node 5 is between Node 3 & 1

  acheck(ii,kk,2) = acheck(ii,kk,2) + 1
  acheck(kk,ii,2) = acheck(kk,ii,2) + 1
  if (acheck(ii,kk,1)==1) goto 20
  acheck(ii,kk,1) = 1
  acheck(kk,ii,1) = 1
  npoi = npoi + 1
  intmat(ie,5) = npoi
  number(ii,kk) = npoi
  number(kk,ii) = npoi
  coord(npoi,1) = (coord(ii,1) + coord(kk,1))/2.
  coord(npoi,2) = (coord(ii,2) + coord(kk,2))/2.

! Check boundary for node 5 <p(5)=0.>

  if (ibcx(ii)==1 .and. ibcx(kk)==1) then
    ibcx(npoi) = 1
    displx(npoi) = 0.
  endif
  if (ibcy(ii)==1 .and. ibcy(kk)==1) then
    ibcy(npoi) = 1
    disply(npoi) = 0.
  endif

  goto 25
20 continue
  intmat(ie,5) = number(ii,kk)
25 continue

```

```

! Node 6 is between Node 1 & 2

  acheck(ii,jj,2) = acheck(ii,jj,2) + 1
  acheck(jj,ii,2) = acheck(jj,ii,2) + 1
  if (acheck(ii,jj,1)==1) goto 30
  acheck(ii,jj,1) = 1
  acheck(jj,ii,1) = 1
  npoi = npoi + 1
  intmat(ie,6) = npoi
  number(ii,jj) = npoi
  number(jj,ii) = npoi
  coord(npoi,1) = (coord(ii,1) + coord(jj,1))/2.
  coord(npoi,2) = (coord(ii,2) + coord(jj,2))/2.

! Check boundary for node 6 <p(6)=0.>

  if (ibcx(ii)==1 .and. ibcx(jj)==1) then
    ibcx(npoi) = 1
    displx(npoi) = 0.
  endif
  if (ibcy(ii)==1 .and. ibcy(jj)==1) then
    ibcy(npoi) = 1
    disply(npoi) = 0.
  endif

  goto 35
30 continue
  intmat(ie,6) = number(ii,jj)
35 continue

```

enddo

End Subroutine GEN_NL

!-----

Subroutine ARRANGE()

implicit none

```

do ip=1,npoi
  p(ip) = displx(ip)
  p(ip+npoi) = disply(ip)
enddo
deallocate(displx, disply, stat=ierror)

```

End Subroutine ARRANGE

!-----

END MODULE SOLID

!-----

```

Module solve

implicit none

contains

!
! Subroutine preconditioned conjugate gradient method
!

subroutine unsympcg(a, b, x, n)

implicit none

integer(4)          :: maxit, iter, count
integer(4)          :: i, j, n
real(8)             :: del, dnew, tol, bottom
real(8)             :: alpha, dold, beta, check, thresh
real(8), dimension(n,n) :: a
real(8), dimension(n)  :: q, r, d, pre, s, x, b, ax, atb

integer(4), allocatable, dimension(:) :: ija, ijat, ijata
real(8), allocatable, dimension(:) :: sa, sat, sata

    iter = 1
    tol = 1.d-6
    maxit = 50000
    thresh = 1.e-10

    count = 1000000

    write(6,*) count

    allocate( sa(count), ija(count))
    allocate( sat(count), ijat(count))
    allocate(sata(count), ijata(count))

    !-----
    ! Convert square matrix into row-indexed sparse
    !-----
    call sprsin( a,n,n,thresh,count, sa, ija)
    call sprsint(a,n,n,thresh,count,sat,ijat)

    !-----
    ! Compute At*A
    !-----
    call sprstm(sat,ijat,sat,ijat,sata,ijata,thresh,count)

    !-----
    ! Compute At*b
    !-----
    call sprsax(sat,ijat,b,atb,n,count)

    !-----
    ! Construct precondition matrix P (but this will find P inverse)
    !-----
    do i = 1,n
        pre(i) = 1./abs(sata(i))
    end do

    !-----
    ! Find Ro
    !-----
    call sprsax(sata,ijata,x,ax,n,count)
    do i = 1,n
        r(i) = atb(i) - ax(i)
    end do

    !-----
    ! Find Do
    !-----
    do i = 1,n
        d(i) = pre(i)*r(i)
    end do

    !-----
    ! Find D-New
    !-----
    dnew = 0.
    do i = 1,n
        dnew = dnew + r(i)*d(i)
    end do

    del = dnew

```

```

Do
  if(iter >= maxit.OR.dnew <= (tol*tol*del)) exit
!-----
! Find Q(i+1)
!-----
  call sprsax(sata,ijata,d,q,n,count)

!-----
! Find Alpha
!-----
  bottom = 0.
  do i = 1,n
    bottom = bottom + d(i)*q(i)
  end do

  alpha = dnew/bottom
  do i = 1,n
    x(i) = x(i) + alpha*d(i)
  end do

!-----
! Find R(i+1)
!-----
  check = mod(iter,50)
  if(check == 0.) then
    call sprsax(sata,ijata,x,ax,n,count)
    do i = 1,n
      r(i) = atb(i) - ax(i)
    end do
  else
    do i = 1,n
      r(i) = r(i) - alpha*q(i)
    end do
  endif

!-----
! Find S(i+1)
!-----
  do i = 1,n
    s(i) = pre(i)*r(i)
  end do

!-----
! Find beta
!-----
  dold = dnew
  dnew = 0.
  do i = 1,n
    dnew = dnew + r(i)*s(i)
  end do

  beta = dnew/dold
  do i = 1,n
    d(i) = s(i) + beta*d(i)
  end do

  iter = iter + 1

End do

end subroutine unsympcg
!-----
subroutine sympcg(a, b, x, n)
implicit none

integer(4)          :: maxit, iter, count
integer(4)          :: i, j, n
real(8)             :: del, dnew, tol, bottom
real(8)             :: alpha, dold, beta, check, thresh
real(8), dimension(n,n) :: a
real(8), dimension(n)  :: q, r, d, pre, s, x, b, ax

integer(4), allocatable, dimension(:) :: ija
real(8),   allocatable, dimension(:) :: sa

  iter = 1
  tol = 1.d-6
  maxit = 50000
  thresh = 1.e-10

  count = 10000000

```



```

allocate(sa(count), ija(count))

!-----
! Construct precondition matrix P (but this will find P inverse)
!-----
do i = 1,n
  pre(i) = 1./abs(a(i,i))
end do

!-----
! Convert square matrix into row-indexed sparse
!-----
call sprsin(a,n,n,thresh,count,sa,ija)

!-----
! Find Ro
!-----
call sprsax(sa,ija,x,ax,n,count)
do i = 1,n
  r(i) = b(i) - ax(i)
end do

!-----
! Find Do
!-----
do i = 1,n
  d(i) = pre(i)*r(i)
end do

!-----
! Find D-New
!-----
dnew = 0.
do i = 1,n
  dnew = dnew + r(i)*d(i)
end do

del = dnew

Do
  if(iter >= maxit.OR.dnew <= (tol*tol*del)) exit

!-----
! Find Q(i+1)
!-----
call sprsax(sa,ija,d,q,n,count)

!-----
! Find Alpha
!-----
bottom = 0.
do i = 1,n
  bottom = bottom + d(i)*q(i)
end do

alpha = dnew/bottom
do i = 1,n
  x(i) = x(i) + alpha*d(i)
end do

!-----
! Find R(i+1)
!-----
check = mod(iter,50)
if(check == 0.) then
  call sprsax(sa,ija,x,ax,n,count)
  do i = 1,n
    r(i) = b(i) - ax(i)
  end do
else
  do i = 1,n
    r(i) = r(i) - alpha*q(i)
  end do
endif

!-----
! Find S(i+1)
!-----
do i = 1,n
  s(i) = pre(i)*r(i)
end do

```

```

!-----
! Find beta
!-----
dold = dnew
dnew = 0.
do i = 1,n
  dnew = dnew + r(i)*s(i)
end do

beta = dnew/dold
do i = 1,n
  d(i) = s(i) + beta*d(i)
end do

iter = iter + 1

End do

end subroutine sympcg

!-----
subroutine sprsin(a,n,np,thresh,nmax,sa,ija)

implicit none

integer(4)          :: n, nmax, np
integer(4)          :: i, j, k
integer(4), dimension(nmax) :: ija

real(8)            :: thresh
real(8), dimension(np,np) :: a
real(8), dimension(nmax) :: sa

do j = 1,n
  sa(j) = a(j,j)
enddo

ija(1) = n+2
k = n+1
do i = 1,n
  do j = 1,n
    if (abs(a(i,j)) >= thresh) then
      if (i /= j) then
        k = k+1
        if (k > nmax) then
          write(6,*) k, nmax !, 'nmax too small in sprsin'
          pause 'nmax too small in sprsin'
        endif
        sa(k) = a(i,j)
        ija(k) = j
      endif
    endif
  enddo
  ija(i+1) = k+1
enddo

END subroutine sprsin

!-----

subroutine sprsax(sa,ija,x,b,n,nmax)

! Multiply a matrix in sparse storage by a vector x(1:n)
! and the result is a vector b(1:n)

implicit none

integer(4)          :: n, nmax, i, k
integer(4), dimension(nmax) :: ija

real(8), dimension(n) :: b, x
real(8), dimension(nmax) :: sa

if (ija(1) /= n+2) pause 'mismatched vector and matrix in sprsax'
do i = 1,n
  b(i) = sa(i) * x(i)
  do k = ija(i),ija(i+1)-1
    b(i) = b(i) + sa(k) * x(ija(k))
  enddo
enddo

END subroutine sprsax

!-----

```

```
subroutine sprstm(sa,ija,sb,ijb,sc,ijc,thresh,nmax)
```

```

!
! Multiply At*A
!
! Input for sa must be At
! Input for sb must be At
! output of sc will be At*A
!
implicit none

integer(4)          :: n, nmax
integer(4)          :: i, ijma, ijmb, j, k, ma, mb, mbb

integer(4), dimension(nmax) :: ija, ijb, ijc
real(8) , dimension(nmax)  :: sa, sb, sc

real(8)             :: sum, thresh

if(ija(1) /= ijb(1)) pause 'sprstm sizes do not match'
k = ija(1)
ijc(1) = k
do i = 1, ija(1)-2
  do j = 1, ijb(1)-2
    if(i==j) then
      sum = sa(i)*sb(j)
    else
      sum = 0.d0
    endif
    mb = ijb(j)
    do ma = ija(i), ija(i+1)-1
      ijma = ija(ma)
      if(ijma == j) then
        sum = sum + sa(ma)*sb(j)
      else
        if(mb < ijb(j+1)) then
          ijmb = ijb(mb)
          if(ijmb == i) then
            sum = sum + sa(i)*sb(mb)
            mb = mb + 1
            goto 2
          else if(ijmb < ijma) then
            mb = mb + 1
            goto 2
          else if(ijmb == ijma) then
            sum = sum + sa(ma)*sb(mb)
            mb = mb + 1
            goto 2
          endif
        endif
      endif
    enddo

    do mbb = mb, ijb(j+1)-1
      if(ijb(mbb) == i) then
        sum = sum + sa(i)*sb(mbb)
      endif
    enddo

    if(i == j) then
      sc(i) = sum
    else if(abs(sum) > thresh) then
      if(k > nmax) pause 'sprstm: nmax to small'
      sc(k) = sum
      ijc(k) = j
      k = k + 1
    endif
  enddo
  ijc(i+1) = k
enddo
end subroutine sprstm

```

```
subroutine sprsint(a,n,np,thresh,nmax,sa,ija)
```

```

implicit none

integer(4)          :: n, nmax, np
integer(4)          :: i, j, k
integer(4), dimension(nmax) :: ija

```

```
real(8)                :: thresh
real(8), dimension(np,np) :: a
real(8), dimension(nmax)  :: sa

do j = 1,n
  sa(j) = a(j,j)
enddo
ija(1) = n+2
k = n+1
do j = 1,n
  do i = 1,n
    if (abs(a(i,j)) >= thresh) then
      if (i /= j) then
        k = k+1
        if (k > nmax) pause 'nmax too small in sprsin'
        sa(k) = a(i,j)
        ija(k) = i
      endif
    endif
  enddo
  ija(j+1) = k+1
enddo

END subroutine sprsint
!-----
end module solve
```



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายสุธี ไตรวิวัฒนา เกิดเมื่อวันที่ 20 เดือนมกราคม พุทธศักราช 2521 จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิตจากภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2542 สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรมหาบัณฑิตจากภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2546 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรดุษฎีบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2547



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย