ระเบียบวิธีไฟไนต์เอลิเมนต์ชนิดปรับขนาดได้สำหรับกลศาสตร์การแตกหักแบบอิลาสติก-พลาสติก

นายกอบศักดิ์ พจนานภาศิริ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2548

ISBN 974-53-2366-7

ADAPTIVE FINITE ELEMENT METHOD FOR ELASTIC-PLASTIC FRACTURE MECHANICS

Mr. Kobsak Potjananapasiri

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering Program in Mechanical Engineering

Department of Mechanical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2005

ISBN 974-53-2366-7

หัวข้อวิทยานิพนธ์	ระเบียบวิธีไฟไนต์เอลิเมนต์ชนิดปรับขนาดได้สำหรับกลศาสตร์การแตกหักแบบอิลาสติก-พลาสติก
โดย	นายกอบศักดิ์ พจนานภาศิริ
สาขาวิชา	วิศวกรรมเครื่องกล
อาจารย์ที่ปรึกษา	ศาสตราจารย์ ดร.ปราโมทย์ เดชะอำไพ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.... คณบดี คณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

.... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ตุลย์ มณีวัฒนา)

.... อาจารย์ที่ปรึกษา

(ศาสตราจารย์ ดร.ปราโมทย์ เดชะอำไพ)

.... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ไพโรจน์ สิงหถนัดกิจ)

.... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.กุณฑินี มณีรัตน์)

กอบศักดิ์ พจนานภาศิริ : ระเบียบวิธีไฟไนต์เอลิเมนต์ชนิดปรับขนาดได้สำหรับกลศาสตร์การแตกหักแบบอิลาสติก-พลาสติก. (ADAPTIVE FINITE ELEMENT METHOD FOR ELASTIC-PLASTIC FRACTURE MECHANICS) อ. ที่ปรึกษา : ศาสตราจารย์ ดร. ปราโมทย์ เดชะอำไพ, 241 หน้า. ISBN 974-53-2366-7

วิทยานิพนธ์ฉบับนี้ทำการศึกษาการวิเคราะห์ปัญหารอยร้าวภายใต้สภาวะความเค้นในระนาบ ความเครียดในระนาบและปัญหาสมมาตรรอบแกนด้วยระเบียบวิธีไฟไนต์เอลิเมนต์ โดยประดิษฐ์โปรแกรมคอมพิวเตอร์เพื่อใช้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัล

ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลเพื่อใช้ในการวิเคราะห์รอยร้าวได้ใช้ระเบียบวิธีโดเมนอินทิกรัลซึ่งสมการค่าพารามิเตอร์เจอินทิกรัลได้ถูกเปลี่ยนจากรูปการอินทิเกรตบนเส้นมาอยู่ในรูปการอินทิเกรตบนพื้นที่โดเมนใด ๆ รอบปลายรอยร้าว เอลิเมนต์ที่ใช้ในแบบจำลองประกอบด้วยเอลิเมนต์ที่ปลายรอยร้าวซึ่งเป็นเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อที่ด้านหนึ่งของเอลิเมนต์ถูกยุบมารวมกันที่ตำแหน่งปลายรอยร้าวและเอลิเมนต์ในบริเวณอื่นซึ่งเป็นเอลิเมนต์แบบสามเหลี่ยมหกจุดต่อทั้งหมด นอกจากนี้ยังได้นำเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติและเทคนิคการถ่ายทอดผลเฉลยมาใช้เพื่อให้ผลการคำนวณที่ได้มีความถูกต้องมากยิ่งขึ้น โดยเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติทำการสร้างเอลิเมนต์ขนาดเล็กในบริเวณที่ค่าอนุพันธ์อันดับสองของค่าความเค้นวอนมิสเซสมีค่าสูง ในขณะเดียวกันก็สร้างเอลิเมนต์ขนาดใหญ่ในบริเวณที่ค่าอนุพันธ์อันดับสองของค่าความเค้นวอนมิสเซสมีค่าต่ำ สำหรับเทคนิคการถ่ายทอดผลเฉลยนั้นทำการถ่ายทอดผลเฉลยค่าการเคลื่อนตัวที่จุดต่อจากโครงตาข่ายก่อนการปรับขนาดเอลิเมนต์ไปสู่จุดต่อต่าง ๆ ในโครงตาข่ายที่ทำการปรับขนาดเอลิเมนต์แล้วเพื่อให้การคำนวณสามารถดำเนินต่อไปได้โดยไม่จำเป็นต้องเริ่มใหม่ที่ระดับภาระเริ่มต้นทุกครั้งหลังการปรับขนาดเอลิเมนต์

การตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นนั้นทำโดยการเปรียบเทียบผลการคำนวณค่าพารามิเตอร์เจอินทิกรัลที่ได้จากโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นกับผลลัพธ์จากงานวิจัยอื่น ๆ ที่มีสำหรับปัญหาเดียวกัน โดยผลลัพธ์ที่ได้แสดงให้เห็นถึงประสิทธิภาพของระเบียบวิธีไฟไนต์เอลิเมนต์ที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติพร้อมกับการถ่ายทอดผลเฉลยในการคำนวณค่าพารามิเตอร์เจอินทิกรัลได้เป็นอย่างดี

ภาควิชา วิศวกรรมเครื่องกล ลายมือชื่อนิสิต
สาขาวิชา วิศวกรรมเครื่องกล ลายมือชื่ออาจารย์ที่ปรึกษา
ปีการศึกษา 2548

4570212821 : MAJOR MECHANICAL ENGINEERING
KEY WORD: FINITE ELEMENT / ADAPTIVE REMESHING TECHNIQUE / *J*-INTEGRAL / DOMAIN INTEGRAL METHOD / SOLUTION MAPPING
KOBSAK POTJANANAPASIRI : ADAPTIVE FINITE ELEMENT METHOD FOR ELASTIC-PLASTIC FRACTURE MECHANICS. THESIS ADVISOR: PROF. PRAMOTE DECHAUMPHAI, Ph.D. 241 pp. ISBN 974-53-2366-7.

A finite element method for two-dimensional crack problems under plane stress, plane strain and axisymmetric conditions is presented. A corresponding finite element computer program has been developed to estimate the *J*-integral parameter.

The domain integral method, for which the *J*-integral expression has been changed from a line-integral expression into a domain form, is utilized as the *J*-integral solution scheme. The 6-node triangular element mesh is enhanced by 9-node degenerated elements as crack tip elements. The adaptive remeshing technique is implemented for automatically generating small elements in the regions where large changes in the von Mises stress gradients occur. At the same time, larger elements are generated in the other regions where the stress is nearly uniform. After the new refined mesh has been generated in a load level, a solution mapping scheme is employed to transfer the old-mesh displacement fields onto those of the new mesh to provide good initial fields for the new load level.

The finite element computer program was verified by calculating the *J*-integral of many benchmark examples of which the solutions are presented in the literature. The results have demonstrated that the combined domain integral and finite element method with adaptive remeshing technique and solution mapping scheme is efficient in determining the *J*-integral.

Department Mechanical Engineering Student's signature Kobsak potjananapasiri
Field of study Mechanical Engineering Advisor's signature Pramote Dechaumphai
Academic Year 2005

กิตติกรรมประกาศ

ผู้วิจัยขอกราบขอบพระคุณ ศาสตราจารย์ ดร.ปราโมทย์ เดชะอำไพ อาจารย์ที่ปรึกษาวิทยานิพนธ์เป็นอย่างสูง ที่ท่านได้ให้ความรู้ คำปรึกษา ตลอดจนข้อคิดที่มีคุณค่ายิ่งในการทำวิจัยและการทำงาน

ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ตุลย์ มณีวัฒนา ประธานกรรมการ ผู้ช่วยศาสตราจารย์ ดร.ไพโรจน์ สิงหถนัดกิจ และผู้ช่วยศาสตราจารย์ ดร.กุณฑินี มณีรัตน์ กรรมการ ที่ได้ให้การอบรมสั่งสอน คำแนะนำ การช่วยเหลือและถ่ายทอดความรู้ตลอดระยะเวลาในการศึกษาและทำงานวิจัยนี้ซึ่งทำให้ผู้วิจัยได้รับความรู้และวิทยานิพนธ์ฉบับนี้มีความสมบูรณ์มากยิ่งขึ้น

ขอกราบขอบพระคุณผู้ช่วยศาสตราจารย์ ดร. จิรพงศ์ กสิวิทย์อำนวย ที่กรุณาให้ความช่วยเหลือผู้วิจัยด้วยความเต็มใจเสมอมาทั้งการค้นหาเอกสารทางวิชาการและคำแนะนำอันเป็นประโยชน์ต่อผู้วิจัยอย่างสูง

ขอขอบพระคุณ พี่สุทธิศักดิ์ พงศ์ธนาพาณิช พี่วิโรจน์ ลิ่มตระการ พี่นิพนธ์ วรรณโสภาคย์ พี่ธนวัช ศรีเจริญชัย พี่สุธี โอฬารฤทธินันท์ พี่สุธี ไตรวิวัฒนา พี่พัชรี ธีระเอก คุณอธิพงษ์ มาลาทิพย์ คุณปริญญา บุญมาเลิศ คุณคมกฤษณ์ ชัยโย และคุณกิตติศักดิ์ คู่วรัญญู สำหรับความช่วยเหลือและกำลังใจตลอดการทำวิจัยนี้

ขอขอบคุณ พี่กวี ศรีทองอินทร์ พี่กิตติพงศ์ บุญโล่ง พี่อนวัช ณ สงขลา คุณมนตรี แจ่มแจ้ง คุณวิทยา วัฒนนุกูลชัย คุณสรสิทธิ์ อรัญพิทักษ์ คุณชวนันท์ สุภาศักดิ์ คุณอมรศักดิ์ ฉ่ำแก้ว คุณพงศภรณ์ อาดัม สำหรับความช่วยเหลือและมิตรภาพซึ่งทำให้การทำงานตลอดจนชีวิตโดยรวมของผู้วิจัยมีสีสันมากขึ้น

ท้ายสุดนี้ผู้วิจัยขอกราบขอบพระคุณบิดามารดา น้องชายและญาติของผู้วิจัยที่คอยเป็นกำลังใจและสนับสนุนการศึกษาของผู้วิจัยมาโดยตลอด อนึ่งประโยชน์และคุณค่าอันใดที่ได้รับจากวิทยานิพนธ์นี้ขอมอบเป็นกตัญญุตาบูชาแด่บิดามารดา ครูอาจารย์ ตลอดจนผู้มีพระคุณทุกท่าน

สารบัญ

		หน้า
บทคัดย่อภาษาไทย		**ง**
บทคัดย่อภาษาอังกฤษ		**จ**
กิตติกรรมประกาศ		**ฉ**
สารบัญ		**ช**
สารบัญตาราง		**ญ**
สารบัญภาพ		**ฎ**
คำอธิบายสัญลักษณ์		**ฌ**
บทที่ 1	**บทนำ**	**1**
	1.1 ความสำคัญและที่มาของวิทยานิพนธ์	1
	1.2 วัตถุประสงค์ของวิทยานิพนธ์	2
	1.3 ขอบเขตของวิทยานิพนธ์	3
	1.4 ขั้นตอนการดำเนินงาน	3
	1.5 ประโยชน์ที่ได้รับจากวิทยานิพนธ์	4
	1.6 ปริทัศน์วรรณกรรม	4
บทที่ 2	**กลศาสตร์การแตกหักแบบอิลาสติก-พลาสติก**	**9**
	2.1 พารามิเตอร์เจอินทิกรัล	9
	2.2 พารามิเตอร์เจอินทิกรัลในรูปอินทิเกรตบนโดเมน	17
	2.2.1 พารามิเตอร์เจอินทิกรัลสำหรับปัญหาสองมิติ	20
	2.2.2 พารามิเตอร์เจอินทิกรัลสำหรับปัญหาสมมาตรรอบแกน	21
	2.3 ข้อจำกัดของพารามิเตอร์เจอินทิกรัล	24
	2.4 บทสรุป	26
บทที่ 3	**การวิเคราะห์ปัญหารอยร้าวด้วยระเบียบวิธีไฟไนต์เอลิเมนต์**	**27**
	3.1 ความสัมพันธ์ระหว่างค่าความความเครียดและค่าเค้น	27
	3.1.1 ปัญหาความเค้นในระนาบ	30
	3.1.2 ปัญหาความเครียดในระนาบ	31
	3.1.3 ปัญหาสมมาตรรอบแกน	32

หน้า

3.2 ฟังก์ชั่นการประมาณภายในเอลิเมนต์ 33
3.2.1 เอลิเมนต์สี่เหลี่ยมเก้าจุดต่อ 34
3.2.2 เอลิเมนต์สามเหลี่ยมหกจุดต่อ 35
3.3 สมการไฟไนต์เอลิเมนต์ 36
3.4 วิธีการทำซ้ำ 41
3.4.1 ปัญหาความเค้นในระนาบ 44
3.4.2 ปัญหาความเครียดในระนาบ 50
3.4.3 ปัญหาสมมาตรรอบแกน 51
3.5 การคำนวณค่าความเค้นที่จุดต่อ 54
3.5.1 เอลิเมนต์สี่เหลี่ยมเก้าจุดต่อ 54
3.5.2 เอลิเมนต์สามเหลี่ยมหกจุดต่อ 58
3.6 สมการไฟไนต์เอลิเมนต์สำหรับคำนวณค่าพารามิเตอร์เจอินทิกรัลในรูปโดเมน 62
3.7 บทสรุป 66

บทที่ 4 เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติและการถ่ายทอดผลเฉลย 67
4.1 เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติ 67
4.2 การถ่ายทอดผลเฉลย 76
4.3 บทสรุป 81

บทที่ 5 โปรแกรมคอมพิวเตอร์สำหรับคำนวณค่าพารามิเตอร์เจอินทิกรัล 82
5.1 ลักษณะของโปรแกรม JFACTOR 82
5.2 รายละเอียดของโปรแกรม JFACTOR 87
5.3 ลักษณะของไฟล์ข้อมูลนำเข้า 87
5.4 ลักษณะของไฟล์ผลลัพธ์ 96
5.4.1 ไฟล์ที่นำไปใช้สำหรับปรับขนาดเอลิเมนต์ 96
5.4.2 ไฟล์ที่นำไปใช้สำหรับแสดงผลกราฟฟิกบนจอคอมพิวเตอร์ 97
5.4.3 ไฟล์ที่นำไปใช้เพื่อเริ่มต้นการคำนวณที่สถานะการคำนวณที่ได้ทำการวิเคราะห์ไปแล้ว 98
5.5 บทสรุป 98

หน้า

บทที่ 6 การตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์ **99**
6.1 การชิ้นทดสอบมาตรฐานแบบ CT 99
6.2 การชิ้นทดสอบมาตรฐานแบบ DENT 106
6.3 ท่อทรงกระบอกที่มีรอยร้าววางตัวตามแนวแกนที่ผิวด้านในภายใต้ความดัน 112
6.4 ท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความหนาของท่อ 118
6.5 แผ่นสี่เหลี่ยมแบบที่มีรอยร้าวที่ขอบข้างเดียวภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความกว้างของแผ่น 124
6.6 บทสรุป 131

บทที่ 7 บทสรุป ปัญหาที่พบและข้อเสนอแนะ **132**
7.1 บทสรุปรวม 132
7.2 ปัญหาที่พบในขณะทำวิทยานิพนธ์ 133
7.3 ข้อเสนอแนะสำหรับงานวิจัยในอนาคต 135

รายการอ้างอิง **137**

ภาคผนวก **140**
ภาคผนวก ก. คุณสมบัติของวัสดุยืดหยุ่นและพลังงานความเครียดหนาแน่น 141
ภาคผนวก ข. คุณสมบัติที่พารามิเตอร์เจอินทิกรัลไม่ขึ้นกับเส้นทางเดิน 144
ภาคผนวก ค. พารามิเตอร์เจอินทิกรัลในฐานะพารามิเตอร์ตัวประกอบความเข้มของความเค้น 146
ภาคผนวก ง. เอลิเมนต์ที่ปลายรอยร้าว 151
ภาคผนวก จ. รายละเอียดของโปรแกรมคอมพิวเตอร์ JFACTOR 159
ภาคผนวก ฉ. งานวิจัยที่ได้ตีพิมพ์ในการประชุมเครือข่ายวิศวกรรมเครื่องกลแห่งประเทศไทย ครั้งที่ 18 236

ประวัติผู้เขียนวิทยานิพนธ์ **241**

สารบัญตาราง

ตาราง		หน้า
ตารางที่ 6.1	ค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่ค่าแรงดึงต่าง ๆ สำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ CT	102
ตารางที่ 6.2	ค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่ค่าความเค้นดึงระดับต่าง ๆ สำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ DENT	108
ตารางที่ 6.3	ค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลคำนวณที่ค่าความดันระดับต่าง ๆ สำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าววางตัวตามแนวแกนที่ผิวด้านในภายใต้ความดัน	115
ตารางที่ 6.4	ค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่ค่าความเค้นดึงระดับต่าง ๆ สำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ	121
ตารางที่ 6.5	ค่าเปอร์เซ็นต์การขึ้นกับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่ค่าความเค้นดึงรัดับต่าง ๆ สำหรับปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความกว้างของแผ่น	127

สารบัญภาพ

หน้า

รูปที่ 2.1 กราฟเปรียบเทียบความสัมพันธ์ระหว่างค่าความเค้นกับค่าความเครียดที่เกิดขึ้นในชิ้นงานของวัตถุแบบอิลาสติก-พลาสติกและแบบยืดหยุ่นไม่เชิงเส้น 10

รูปที่ 2.2 ปริมาตรควบคุมที่เปลี่ยนแปลงตามเวลาเนื่องจากผิวด้านนอกเคลื่อนที่ด้วยความเร็ว v_j 13

รูปที่ 2.3 วัตถุที่มีรอยร้าววางตัวอยู่ในระนาบ 14

รูปที่ 2.4 ปริมาตรทรงกระบอกเล็ก ๆ รอบขอบรอยร้าว 14

รูปที่ 2.5 ปริมาตรวงแหวนรอบขอบรอยร้าว 18

รูปที่ 2.6 การวิเคราะห์รอยร้าวแบบระนาบสำหรับปัญหาสองมิติ 20

รูปที่ 2.7 การวิเคราะห์รอยร้าวแบบระนาบสำหรับปัญหาสมมาตรรอบแกน 22

รูปที่ 2.8 เอลิเมนต์เล็ก ๆ ของวัตถุสมมาตรรอบแกน 23

รูปที่ 2.9 บริเวณที่สนามเอกพันธ์ของ HRR มีอิทธิพลสูงและบริเวณที่ค่าความเครียดจำกัดมีอิทธิพลสูงที่บริเวณปลายรอยร้าว 25

รูปที่ 2.10 นิยามของค่าพารามิเตอร์ระยะการเคลื่อนตัวเปิดที่ปลายรอยร้าว δ_t 25

รูปที่ 3.1 ไอโซพาราเมตริกซ์เอลิเมนต์แบบสี่เหลี่ยมเก้าจุดต่อซึ่งใช้เป็นเอลิเมนต์ที่ปลายรอยร้าว 34

รูปที่ 3.2 ไอโซพาราเมตริกซ์เอลิเมนต์แบบสามเหลี่ยมหกจุดต่อ 36

รูปที่ 3.3 ลักษณะการกระจายของค่าฟังก์ชัน q ที่ใช้ในวิทยานิพนธ์นี้

(ก) ฟังก์ชั่นรูปปิรามิดฐานสี่เหลี่ยมจตุรัส 65

(ข) ฟังก์ชั่นรูปกรวยฐานวงกลม 65

(ค) ฟังก์ชั่นรูปปิรามิดฐานสี่เหลี่ยมผืนผ้า 66

รูปที่ 4.1 พิกัด x-y และพิกัด X-Y 68

รูปที่ 4.2 เปรียบเทียบค่าความเค้นกับอนุพันธ์อันดับสองของผลเฉลยเทียบกับแกนพิกัดต่าง ๆ 70

รูปที่ 4.3 สมการ (4.6) ซึ่งถูกเขียนอยู่ในรูปความสัมพันธ์ของสามเหลี่ยมมุมฉาก 71

รูปที่ 4.4 การเรียงตัวของเอลิเมนต์ล้อมรอบจุดต่อ i 73

รูปที่ 4.5 จุดต่อของโครงตาข่ายใหม่ p บนเอลิเมนต์ในโครงตาข่ายเก่า

(ก) เอลิเมนต์สามเหลี่ยมหกจุดต่อ 76

(ข) เอลิเมนต์ที่ปลายรอยร้าว 76

รูปที่ 4.6 การถ่ายทอดผลเฉลยที่ขอบเขตปัญหาเป็นเส้นโค้ง 77

หน้า

รูปที่ 4.7 การหาว่าจุดต่อที่ปลายรอยร้าวของเอลิเมนต์ในโครงต่ข่ายใหม่อยู่บนเอลิเมนต์ที่ปลายรอยร้าวใดในโครงตาข่ายเก่า
(ก) เอลิเมนต์ที่ปลายรอยร้าวของโครงตาข่ายใหม่และโครงตาข่ายเก่า 78
(ข) นิยามของมุมต่าง ๆ ที่ใช้ในการตรวจสอบ 78
รูปที่ 4.8 เอลิเมนต์ที่ปลายรอยร้าวในโครงตาข่ายเก่าที่บรรจุจุดต่อ p 80
รูปที่ 5.1 ลำดับขั้นตอนการทำงานของโปรแกรม JFACTOR 86
รูปที่ 5.2 การกำหนดมุมของระนาบรอยร้าว รหัสตัวเลขผิวรอยร้าว และขนาดของโดเมน 90
รูปที่ 5.3 การกำหนดแรงที่กระทำกับจุดต่อที่ปลายรอยร้าวและจุดต่อที่อยู่ไกลที่สุดบนผิวรอยร้าวในโดเมนที่ทำการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัล 90
รูปที่ 5.4 การกำหนดขนาดของโดเมนรูปสี่เหลี่ยมผืนผ้า 92
รูปที่ 5.5 กำหนดขนาดโดเมนรูปสี่เหลี่ยมผืนผ้าสำหรับปัญหาสมมาตรบนระนาบรอยร้าว 93
รูปที่ 5.6 รูปแบบของไฟล์ผลลัพธ์ที่ได้จากโปรแกรม JFACTOR เพื่อนำไปใช้สำหรับการปรับขนาดเอลิเมนต์ 97
รูปที่ 5.7 รูปแบบของไฟล์ผลลัพธ์ที่ได้จากโปรแกรม JFACTOR เพื่อนำไปใช้สำหรับการแสดงผลกราฟฟิกบนจอคอมพิวเตอร์ด้วยโปรแกรม Tecplot 9.0 97
รูปที่ 5.8 รูปแบบของไฟล์ที่นำไปใช้เพื่อเริ่มต้นการคำนวณที่สถานะการคำนวณก่อนหน้านี้ที่ได้ทำการวิเคราะห์เสร็จเรียบร้อยไปแล้ว 98
รูปที่ 6.1 รูปร่างของปัญหาชิ้นทดสอบมาตรฐานแบบ CT 100
รูปที่ 6.2 รูปร่างและรายละเอียดของปัญหาชิ้นทดสอบมาตรฐานแบบ CT ที่นำมาพิจารณา 100
รูปที่ 6.3 ลักษณะโครงตาข่ายเริ่มต้นและโดเมนที่ใช้ในการคำนวณสำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ CT 103
รูปที่ 6.4 ลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซส, MPa บนโครงตาข่ายเริ่มต้นสำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ CT ที่ค่าแรงดึง $p = 1100$ N 103
รูปที่ 6.5 ลักษณะโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติและโดเมนที่ใช้ในการคำนวณสำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ CT 103

หน้า

รูปที่ 6.6 ลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซส, MPa สำหรับโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติสำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ CT ที่ค่าแรงดึง $p = 1100$ N 104

รูปที่ 6.7 ค่าพารามิเตอร์เจอินทิกรัลของชิ้นงานมาตรฐานแบบ CT สำหรับปัญหาความเครียดในระนาบ ....

(ก) ที่ค่าระดับภาระสูง 105

(ข) ที่ค่าระดับภาระต่ำถึงปานกลาง 105

รูปที่ 6.8 รูปร่างของปัญหาชิ้นทดสอบมาตรฐานแบบ DENT 106

รูปที่ 6.9 รูปร่างและรายละเอียดของปัญหาปัญหาชิ้นทดสอบมาตรฐานแบบ DENT ที่นำมาพิจารณา 107

รูปที่ 6.10 ลักษณะโครงตาข่ายเริ่มต้นของปัญหาชิ้นทดสอบมาตรฐานแบบ DENT พร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลและการกระจายตัวของสนามความเค้นวอนมิสเซส, MPa ที่ค่าความเค้นดึง $\sigma_\infty = 480$ MPa 109

รูปที่ 6.11 ลักษณะโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลและการกระจายตัวของสนามความเค้นวอนมิสเซส, MPa สำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ DENT ที่ค่าความเค้นดึง $\sigma_\infty = 480$ MPa 110

รูปที่ 6.12 ค่าพารามิเตอร์เจอินทิกรัลของชิ้นงานมาตรฐานแบบ DENT สำหรับปัญหาความเค้นในระนาบ ....

(ก) ที่ค่าระดับภาระสูง 111

(ข) ที่ค่าระดับภาระต่ำถึงปานกลาง 111

รูปที่ 6.13 รูปร่างของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดัน 112

รูปที่ 6.14 รูปร่างและรายละเอียดของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดันที่นำมาพิจารณา 113

รูปที่ 6.15 ค่าพารามิเตอร์ตัวประกอบความเข้มของความเค้นที่ระดับความดันต่าง ๆ สำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านใน 114

รูปที่ 6.16 ลักษณะของโครงตาข่ายเริ่มต้นพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลสำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดันภายใน 116

หน้า

รูปที่ 6.17 ลักษณะการกระจายตัวของค่าความเค้นวอนมิสเซส, MPa สำหรับโครงตาข่ายเริ่มต้นที่ค่าความดัน $p = 10$ MPa ของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดันภายใน 116

รูปที่ 6.18 ลักษณะของโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์พร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลโดยอัตโนมัติสำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดันภายใน 117

รูปที่ 6.19 ลักษณะการกระจายตัวของค่าความเค้นวอนมิสเซส, MPa สำหรับโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติที่ค่าความดัน $p = 10$ MPa สำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดันภายใน 117

รูปที่ 6.20 ลักษณะการกระจายตัวของค่าความเค้นตามแนวเส้นรอบวง $\sigma_{\theta\theta}$, MPa สำหรับโครงตาข่ายเริ่มต้นและโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติที่ค่าความดัน $p = 10$ MPa 118

รูปที่ 6.21 รูปร่างของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ 119

รูปที่ 6.22 รูปร่างและรายละเอียดของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิที่นำมาพิจารณา 120

รูปที่ 6.23 ลักษณะโครงตาข่ายเริ่มต้นพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลสำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ 122

รูปที่ 6.24 ลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซส, ksi สำหรับโครงตาข่ายเริ่มต้นที่ค่าภาระความเค้นดึง $\sigma^{\infty} = 90$ ksi ของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวง 122

รูปที่ 6.25 ลักษณะโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลสำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ 123

หน้า

รูปที่ 6.26 ลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซส, ksi สำหรับโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติที่ค่าภาระความเค้นดึง $\sigma^{\infty} = 90$ ksi ของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวง 123

รูปที่ 6.27 ค่าพารามิเตอร์เจอินทิกรัลของท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ 124

รูปที่ 6.28 รูปร่างของปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความกว้างของแผ่น 125

รูปที่ 6.29 รูปร่างและรายละเอียดของปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความกว้างของแผ่นที่นำมาพิจารณา 126

รูปที่ 6.30 ลักษณะโครงตาข่ายเริ่มต้นพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลสำหรับปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวและการกระจายตัวของสนามความเค้นวอนมิสเซส, ksi ที่ค่าความเค้นดึง $\sigma_{\infty} = 79$ ksi 128

รูปที่ 6.31 ลักษณะโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลและการกระจายตัวของสนามความเค้นวอนมิสเซส, ksi ที่ค่าความเค้นดึง $\sigma_{\infty} = 79$ ksi ของปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียว 129

รูปที่ 6.32 ค่าพารามิเตอร์เจอินทิกรัลของปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความกว้างของแผ่นภายใต้เงื่อนไขความเค้นในระนาบ ....
(ก) ที่ค่าแรงดึงระดับสูง 130
(ข) ที่ค่าแรงดึงระดับต่ำถึงปานกลาง 130

รูปที่ ข.1 รอยร้าวที่วางตัวตามแนวแกน x_1 บนชิ้นงานสองมิติ 144

รูปที่ ง.1 กราฟค่าความเค้นและค่าความเครียดของวัสดุที่ค่า $0 \le n \le \infty$ 151

รูปที่ ง.2 แสดงเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อที่ด้านหนึ่งถูกยุบมารวมกันเป็นเอลิเมนต์สามเหลี่ยมและจุดต่อกึ่งกลางด้านและกึ่งกลางเอลิเมนต์ถูกเลื่อนมาอยู่ที่ตำแหน่งหนึ่งในสี่ 153

คำอธิบายสัญลักษณ์

A_e	พื้นที่ของเอลิเมนต์
A_T	พื้นที่ที่ทำการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัล
$[B]$	เมตริกซ์ความสัมพันธ์ระหว่างค่าความเครียดและระยะเคลื่อนตัว
CNG	จำนวนจุดเกาส์ที่ใช้ในการอินทิเกรตพจน์ที่เกี่ยวข้องกับค่าความเค้นดึงที่ผิวรอยร้าว
$d\Omega$	การอินทิเกรตบนปริมาตร
dS	การอินทิเกรตบนพื้นที่ผิว
$\left[\dfrac{\partial F^{vol}}{\partial u}\right]$	เมตริกซ์ความแข็งเกร็งสัมผัสเนื่องจากค่าความเค้นตั้งฉากเฉลี่ย
$\left[\dfrac{\partial F^{dev}}{\partial u}\right]$	เมตริกซ์ความแข็งเกร็งสัมผัสเนื่องจากค่าความเค้นดิเวียทอริก
DNG	จำนวนจุดเกาส์ที่ใช้ในการอินทิเกรตพจน์ที่เกี่ยวข้องกับค่าความเค้นดิเวียทอริก
e_e	ความเครียดประสิทธิผล
e_{ij}	ความเครียดดิเวียทอริก
e_{ij}^p	ความเครียดพลาสติก
E	ค่าคงที่ของการยืดหยุ่น
f_i	แรงวัตถุ
$\{\bar{f}\}$	เวกเตอร์แรงวัตถุ
$\{F^{vol}\}$	โหลดเวกเตอร์เนื่องจากค่าความเค้นตั้งฉากเฉลี่ย
$\{F^{dev}\}$	โหลดเวกเตอร์เนื่องจากค่าความเค้นดิเวียทอริก
$\{F^{trac}\}$	โหลดเวกเตอร์เนื่องจากค่าความเค้นดึงที่ผิว
$\{F^{body}\}$	โหลดเวกเตอร์เนื่องจากค่าแรงวัตถุ
F	งานหนาแน่นเนื่องจากแรงวัตถุ
$\{g\}$	เวกเตอร์ความไม่สมดุลย์ของแรง
$\lVert g\rVert_o$	นอร์มของเวกเตอร์ความไม่สมดุลย์ของแรงในช่วง Incremental Solution Scheme

$\|g\|_i$ นอร์มของเวกเตอร์ความไม่สมดุลย์ของแรงในช่วง Iterative Solution Scheme

h_{min} ขนาดของเอลิเมนต์ที่เล็กที่สุดที่กำหนดโดยผู้ใช้

H_{jk} เทนเซอร์โมเมนตัมเนื่องจากพจน์พลังงานต่าง ๆ

J พารามิเตอร์เจอินทิกรัล

$\bar{J}$ พลังงานที่ถูกปลดปล่อยออกจากวัตถุเฉลี่ยต่อความยาวรอยร้าว

$[J]$ ยาโคบีเมตริกซ์

k ขนาดของฟังก์ชั่นความเค้น

K_I พารามิเตอร์ตัวประกอบความเข้มของความเค้น

L พลังงานจลน์หนาแน่น

ความหนาของชิ้นงาน

m จำนวนเอลิเมนต์ทั้งหมดที่มีจุดต่อ i เป็นจุดต่อร่วม

n เลขยกกำลังของการทำให้แข็งด้วยความเครียด

จำนวนจุดต่อทั้งหมดบนเอลิเมนต์

NG จำนวนจุดเกาส์ที่ใช้ในการอินทิเกรต

N_i ฟังก์ชันการประมาณภายใน

P_{jk} เทนเซอร์โมเมนตัมพลังงานยืดหยุ่นของเอสเชลบาย

q_i เวกเตอร์ฟังก์ชั่นต่อเนื่องใด ๆ

$|\hat{q}|$ ขนาดของเวกเตอร์ q_i

r ระยะทางในแนวรัศมีวัดจากแกนสมมาตร

r_a ระยะทางในแนวรัศมีจากแกนสมมาตรถึงปลายรอยร้าว

R_i ระยะรัศมีภายในของท่อทรงกระบอก

S_T ผิวปริมาตรวงแหวน

S_{ij} ความเค้นดิเวียทอริก

$\{S\}$ เวกเตอร์ความเค้นดิเวียทอริก

t_i ความเค้นดึงที่ผิวในรูปเท็นเซอร์

$\{\bar{T}\}$ เวกเตอร์ความเค้นดึงที่ผิว

$[TR]$ เมตริกซ์ความสัมพันธ์ระหว่างค่าความเค้นที่จุดเกาส์และจุดต่อ

u การเคลื่อนตัวในทิศทางตามแนวรัศมี

u_i การเคลื่อนตัว

$\ddot{u}_i$ ความเร่ง

$\{u\}$ เวกเตอร์การเคลื่อนที่จุดต่อ

$\bar{U}_i$ ค่าเฉลี่ยของค่าความเค้นวอนมิสเซสที่จุดต่อ

U_e^j ค่าความเค้นวอนมิสเซสที่จุดต่อของเอลิเมนต์ที่ j ซึ่งมีจุดต่อนี้เป็นจุดต่อร่วม

VNG จำนวนจุดเกาส์ที่ใช้ในการอินทิเกรตพจน์ที่เกี่ยวข้องกับค่าความเค้นตั้งฉากเฉลี่ย

V_T ปริมาตรวงแหวน

w_p น้ำหนักที่สอดคล้องกับการอินทิเกรตบนพื้นที่ของจุดเกาส์ที่ p

ws_p น้ำหนักที่สอดคล้องกับการอินทิเกรตตามเส้นของจุดเกาส์ที่ p

W พลังงานความเครียดหนาแน่น

W^e พลังงานความเครียดหนาแน่นเนื่องจากค่าความเครียดยืดหยุ่นเชิงเส้น

W^p พลังงานความเครียดหนาแน่นเนื่องจากค่าความเครียดพลาสติก

α ค่าคงที่ของวัสดุ

δ_t พารามิเตอร์ระยะการเคลื่อนตัวเปิดที่ปลายรอยร้าว

δ_{jk} ครอนเน็คเคอร์เด็ลต้า

$\delta l(s)$ ความยาวขอบรอยร้าวที่เพิ่มขึ้นที่ตำแหน่งบนขอบรอยร้าว s

δu สนามการเคลื่อนตัวเสมือนในแนวแกน x

δu_i ค่าการเคลื่อนตัวเสมือนในแนวแกน x ที่จุดต่อ

$\delta\{\bar{u}\}$ เวกเตอร์การเคลื่อนตัวเสมือน

δv สนามการเคลื่อนตัวเสมือนในแนวแกน y

δv_i ค่าการเคลื่อนตัวเสมือนในแนวแกน y ที่จุดต่อ

δV งานเสมือน

$\delta\{\varepsilon\}$	เวกเตอร์ความเครียดเสมือน
ε	ความเผื่อของการลู่เข้าที่ต้องการ
ε_o	ความเครียดที่จุดคราก
ε_p	ความเครียดพลาสติกประสิทธิผล
ε_{ij}	ความเครียดสุทธิ
ε_{ij}^{e}	ความเครียดยืดหยุ่นเชิงเส้น
ε_{ij}^{m}	ค่าความเครียดทางกล
ε_{ij}^{ther}	ความเครียดเริ่มต้นเนื่องจากอุณหภูมิ
$\bar{\varepsilon}$	ความเครียดตั้งฉากเฉลี่ย
$\{\varepsilon\}$	เวกเตอร์ความเครียดสุทธิ
$\{\varepsilon_o\}$	เวกเตอร์ความเครียดเนื่องจากอุณหภูมิ
γ_{ij}	ความเครียดเฉือนทางวิศวกรรม
κ	ค่าสัมประสิทธิ์การขยายตัวเนื่องจากสนามอุณหภูมิ
$\{\kappa\}$	เวกเตอร์สัมประสิทธิ์การขยายตัวเนื่องจากความร้อน
λ_{max}	ขนาดของค่าเจาะจงที่มีค่ามากที่สุดในโดเมนปัญหา
ν	อัตราส่วนปัวส์ซง
$\nu_k(s)$	เวกเตอร์หนึ่งหน่วยตั้งฉากกับขอบรอยร้าววางตัวในระนาบรอยร้าว
Φ_i	ตัวแปรไม่ทราบค่าใด ๆ ที่จุดต่อ
$\tilde{\Phi}$	ฟังก์ชันไร้หน่วยที่ขึ้นกับค่ามุม θ
ρ	ความหนาแน่น
σ_o	ความเค้นที่จุดคราก
$\bar{\sigma}$	ความเค้นตั้งฉากเฉลี่ย
σ_e	ความเค้นประสิทธิผล
σ_{ji}	ความเค้นสุทธิ
$\{\sigma\}$	เวกเตอร์ความเค้นสุทธิ

$\{\overline{\sigma}\}$	เวกเตอร์ความเค้นตั้งฉากเฉลี่ย
θ	มุมในระบบพิกัดเชิงขั้ว
θ_C	มุมที่ระนาบรอยร้าวกระทำกับแกน x_1
Θ	สนามอุณหภูมิ
$\{\Theta\}$	เวกเตอร์อุณหภูมิที่จุดต่อ
$\{\tau_e\}_{nodes}$	เวกเตอร์ค่าความเค้นที่จุดต่อ
$\{\tau_e\}_{opt}$	เวกเตอร์ค่าความเค้นที่จุดเกาส์
Δ	ผลต่างของตัวแปรอิสระระหว่างค่าใหม่และเก่า

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของวิทยานิพนธ์

ชิ้นส่วนเครื่องจักรกลและโครงสร้างโดยทั่วไปนั้นมักถูกกระทำด้วยภาระที่มีขนาดเปลี่ยนแปลงตามเวลาซึ่งอาจก่อให้เกิดความเสียหายทางกลโดยปรากฏออกมาในรูปของรอยร้าว (Crack) ถ้าชิ้นส่วนดังกล่าวยังคงรับภาระต่อไปจะทำให้เกิดการเติบของรอยร้าวจนกระทั่งชิ้นส่วนเกิดความเสียหายในที่สุด

การทำนายค่าภาระที่ทำให้รอยร้าวที่ความยาวต่าง ๆ เกิดการเติบโตจึงเป็นสิ่งจำเป็นเพื่อช่วยให้วิศวกรสามารถตัดสินใจได้ว่าจะใช้ชิ้นงานนั้นต่อไปได้อย่างปลอดภัยหรือไม่ซึ่งก่อให้เกิดประโยชน์เป็นอย่างมากเนื่องจากช่วยลดความสูญเสียทั้งเวลาและค่าใช้จ่าย นอกจากนี้แล้วยังสามารถทำให้เกิดความมั่นใจได้ว่าชิ้นงานที่ใช้งานอยู่นั้นจะมีความปลอดภัยไม่ก่อให้เกิดความเสียหายอย่างรุนแรงจนทำให้เกิดอันตรายต่อชีวิตและทรัพย์สิน

แนวทางที่ใช้ในการเข้าถึงปัญหาชิ้นส่วนที่มีรอยร้าวโดยทั่วไปนั้นมีอยู่หลายแนวทาง โดยแนวทางกลศาสตร์การแตกหักแบบอิลาสติก-พลาสติก (Elastic-Plastic Fracture Mechanics, EPFM) เป็นแนวทางทางวิศวกรรมหนึ่งที่เหมาะสมกับการวิเคราะห์ชิ้นงานที่มีรอยร้าวที่การเสียรูปแบบไม่เชิงเส้นที่บริเวณปลายรอยร้าวมีขนาดใหญ่เมื่อเปรียบเทียบกับขนาดรูปร่างของชิ้นงานเกินกว่าที่จะประยุกต์ใช้แนวทางกลศาสตร์การแตกหักแบบยืดหยุ่นเชิงเส้น (Linear-Elastic Fracture Mechanics, LEFM) ได้ โดยในแนวทางกลศาสตร์การแตกหักแบบอิลาสติก-พลาสติกนี้จะอาศัยค่าพารามิเตอร์ต่าง ๆ ที่ได้ถูกพัฒนาขึ้นเพื่อใช้บ่งบอกสถานะความรุนแรงของสนามความเค้นและความเครียดที่บริเวณปลายรอยร้าวซึ่งจะถูกนำไปใช้ในการทำนายขนาดของภาระที่มากระทำกับชิ้นงานที่ทำให้ค่าพารามิเตอร์เหล่านี้มีถึงค่าวิกฤตที่ทำให้รอยร้าวเกิดการเติบโต

เนื่องจากความซับซ้อนของปัญหาตลอดจนการพัฒนาไปอย่างมากของเครื่องคอมพิวเตอร์ในปัจจุบัน การวิเคราะห์ปัญหาต่าง ๆ ทางวิศวกรรมศาสตร์โดยทั่วไปนั้นจึงได้นำเอาระเบียบวิธีเชิงตัวเลขเข้ามาใช้เพื่อลดระยะเวลาและต้นทุนในการทำงานลง โดยระเบียบวิธีไฟไนต์เอลิเมนต์เป็นระเบียบวิธีเชิงตัวเลขหนึ่งที่ได้รับความนิยมเป็นอย่างมากสำหรับการวิเคราะห์ปัญหาทางวิศวกรรม

หลักการของระเบียบวิธีไฟไนต์เอลิเมนต์โดยทั่วไปนั้นจะเริ่มจากการแบ่งขอบเขตรูปร่างลักษณะของปัญหาออกเป็นเอลิเมนต์ย่อย ๆ ซึ่งในแต่ละเอลิเมนต์จะประกอบไปด้วยจุดต่อต่าง ๆ ซึ่งลักษณะการกระจายของตัวแปรใด ๆ บนเอลิเมนต์จะเกิดขึ้นจากการประมาณโดยใช้ค่าของตัวแปรที่จุดต่อและฟังก์ชั่นการประมาณภายในเอลิเมนต์ (Element Interpolation Function) ที่สอดคล้องกับจุดต่อนั้น ๆ จากนั้นจะทำการสร้างสมการไฟไนต์เอลิเมนต์จากเอลิเมนต์ทั้งหมดแล้วนำสมการที่ได้มาประกอบกันเป็นสมการระบบรวม ทำการประยุกต์เงื่อนไขขอบเขต (Boundary Condition) ที่เหมาะสม จากนั้นจึงทำการแก้ระบบสมการเพื่อหาค่าที่จุดต่อของตัวแปรไม่ทราบค่าต่าง ๆ

เนื่องจากความถูกต้องของผลลัพธ์ที่ได้จากการวิเคราะห์ด้วยระเบียบวิธีไฟไนต์เอลิเมนต์นั้นจะขึ้นอยู่กับขนาดของเอลิเมนต์ที่ใช้ในแบบจำลองไฟไนต์เอลิเมนต์นั้น ๆ อย่างไรก็ตามการใช้เอลิเมนต์ขนาดเล็กเป็นจำนวนมากแม้จะทำให้ผลลัพธ์ที่ได้มีความถูกต้องสูงแต่ก็ทำให้สิ้นเปลืองเวลาและหน่วยความจำ (Ram) ที่ให้ในการคำนวณด้วยเครื่องคอมพิวเตอร์สูงด้วยเช่นกัน ด้วยเหตุนี้จึงได้มีการนำเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติ (Adaptive Rem eshing Technique) มาใช้ร่วมกับระเบียบวิธีไฟไนต์เอลิเมนต์เพื่อให้การแก้ปัญหามีประสิทธิภาพมากยิ่งขึ้น โดยเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติจะทำการสร้างเอลิเมนต์ขนาดเล็กในบริเวณที่การเปลี่ยนแปลงความชันของผลลัพธ์มีค่าสูงและสร้างเอลิเมนต์ขนาดใหญ่ในบริเวณที่การเปลี่ยนแปลงความชันของผลลัพธ์มีค่าต่ำซึ่งทำให้แบบจำลองไฟไนต์เอลิเมนต์ที่ได้ประกอบด้วยเอลิเมนต์ที่มีขนาดเล็กเฉพาะในบริเวณที่จำเป็นจริง ๆ

งานวิทยานิพนธ์นี้ขอนำเสนอการนำระเบียบวิธีไฟไนต์เอลิเมนต์ที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติในการวิเคราะห์ชิ้นงานที่มีรอยร้าว โดยทำการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลซึ่งเป็นพารามิเตอร์ที่สำคัญพารามิเตอร์หนึ่งในแนวทางกลศาสตร์การแตกหักแบบอิลาสติก-พลาสติกที่ใช้บ่งบอกระดับความรุนแรงของสนามความเครียดและสนามความเค้นที่บริเวณปลายรอยร้าวได้ โดยผลลัพธ์ที่ได้จะถูกตรวจสอบโดยนำไปเปรียบเทียบกับผลลัพธ์ที่มีสำหรับปัญหาเดียวกันในงานวิจัยอื่น ๆ

1.2 วัตถุประสงค์ของวิทยานิพนธ์

1.2.1 เพื่อประดิษฐ์สมการไฟไนต์เอลิเมนต์ที่สอดคล้องกับปัญหาของแข็งแบบอิลาสติก-พลาสติกที่มีรอยร้าวได้ โดยสมการที่ได้นี้จะนำไปใช้ในการคำนวณค่าพารามิเตอร์เจอินทิกรัลอีกทีหนึ่ง

1.2.2 เพื่อประดิษฐ์โปรแกรมคอมพิวเตอร์ที่สอดคล้องกับสมการไฟไนต์เอลิเมนต์ที่ได้ประดิษฐ์ขึ้นและโปรแกรมคอมพิวเตอร์นี้สามารถทำการคำนวณบนเครื่องคอมพิวเตอร์ได้

1.2.3 เพื่อนำโปรแกรมคอมพิวเตอร์ที่ได้ประดิษฐ์ขึ้นไปประยุกต์ใช้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลที่เกิดขึ้นในชิ้นงานที่มีรอยร้าวได้

1.2.4 เพื่อศึกษาเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติและนำไปประยุกต์ใช้ร่วมกับโปรแกรมคอมพิวเตอร์ที่ได้ประดิษฐ์ขึ้นอย่างมีประสิทธิภาพได้

1.3 ขอบเขตของวิทยานิพนธ์

1.3.1 ประดิษฐ์สมการไฟไนต์เอลิเมนต์เพื่อนำไปประดิษฐ์โปรแกรมคอมพิวเตอร์สำหรับวิเคราะห์ปัญหาของแข็งแบบอิลาสติก-พลาสติกที่มีรอยร้าวได้

1.3.2 สามารถนำโปรแกรมคอมพิวเตอร์ที่ได้ประดิษฐ์ขึ้นไปประยุกต์ใช้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลที่เกิดขึ้นในชิ้นงานที่มีรอยร้าวได้

1.3.3 สามารถนำเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติมาประยุกต์ใช้ร่วมกับโปรแกรมคอมพิวเตอร์ที่ได้ประดิษฐ์ขึ้นอย่างมีประสิทธิภาพได้

1.3.4 ตรวจสอบผลลัพธ์จากโปรแกรมคอมพิวเตอร์ที่ได้ประดิษฐ์ขึ้นกับผลลัพธ์จากงานวิจัยอื่นที่มีสำหรับปัญหาเดียวกัน

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาและทำความเข้าใจในทฤษฎีกลศาสตร์การการแตกหักแบบอิลาสติก-พลาสติก

1.4.2 ประดิษฐ์สมการไฟไนต์เอลิเมนต์เพื่อใช้ในวิเคราะห์ปัญหาของแข็งแบบอิลาสติก-พลาสติกที่มีรอยร้าว

1.4.3 ประดิษฐ์โปรแกรมคอมพิวเตอร์ที่สอดคล้องกับสมการไฟไนต์เอลิเมนต์ที่ได้ประดิษฐ์ขึ้นด้วยภาษาฟอร์แทรน (Fortran) โดยที่โปรแกรมคอมพิวเตอร์นี้สามารถทำการคำนวณบนเครื่องคอมพิวเตอร์ส่วนบุคคลได้

1.4.4 ตรวจสอบความถูกต้องของโปรแกรมที่ได้ประดิษฐ์ขึ้นโดยนำผลลัพธ์ที่ได้จากการคำนวณไปเปรียบเทียบกับผลลัพธ์ที่มีในงานวิจัยอื่นสำหรับปัญหาเดียวกัน

1.4.5 ศึกษาและทำความเข้าใจเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติและนำมาประยุกต์ใช้ร่วมกับโปรแกรมคอมพิวเตอร์ที่ได้ประดิษฐ์ขึ้นเพื่อให้การวิเคราะห์รอยร้าวด้วยระเบียบวิธีไฟไนต์เอลิเมนต์เป็นไปอย่างมีประสิทธิภาพ

1.4.6 สรุปผลที่เกิดขึ้นรวมทั้งข้อเสนอแนะ

1.4.7 จัดพิมพ์วิทยานิพนธ์

1.5 ประโยชน์ที่ได้รับจากวิทยานิพนธ์

1.5.1 ก่อให้เกิดความเข้าใจพื้นฐานในการประดิษฐ์สมการไฟไนต์เอลิเมนต์เพื่อประดิษฐ์โปรแกรมคอมพิวเตอร์ที่สอดคล้องกันต่อไป

1.5.2 สามารถนำโปรแกรมคอมพิวเตอร์ที่ได้ประดิษฐ์ขึ้นไปใช้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลที่เกิดขึ้นในชิ้นงานที่มีรอยร้าวเพื่อระบุความรุนแรงของสนามความเค้นและสนามความเครียดที่บริเวณปลายรอยร้าวได้

1.5.3 สามารถใช้เป็นแนวทางเริ่มต้นในการศึกษาและพัฒนาระเบียบวิธีไฟไนต์เอลิเมนต์บนแนวทางกลศาสตร์การแตกหักแบบอิลาสติก-พลาสติกที่ใช้พารามิเตอร์อื่นในการวิเคราะห์รอยร้าวที่มีลักษณะซับซ้อนได้

1.5.4 สามารถนำโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นมาไปประยุกต์ใช้ร่วมกับเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติเพื่อลดหน่วยความจำและเวลาที่ต้องใช้ในการคำนวณบนเครื่องคอมพิวเตอร์ส่วนบุคคลได้

1.6 ปริทัศน์วรรณกรรม

Parks [1, 2] และ Hellen [3] ได้ทำการศึกษาการหาค่าอัตราการปลดปล่อยพลังงาน (Energy Release Rate) สำหรับวัสดุแบบยืดหยุ่นเชิงเส้นด้วยระเบียบวิธีไฟไนต์เอลิเมนต์โดยเปลี่ยนวิธีการหาค่าอัตราการปลดปล่อยพลังงานซึ่งแต่เดิมต้องทำการวิเคราะห์จากการแก้ปัญหาสองครั้งที่ความยาวรอยร้าวต่างกันเพื่อหาค่าผลต่างของพลังงานศักย์มาเป็นการวิเคราะห์ปัญหาเพียงครั้งเดียวที่ค่าความยาวรอยร้าวที่ต้องการ โดยความยาวรอยร้าวที่เพิ่มขึ้นเกิดจากการเลื่อนจุดต่อที่ตำแหน่งปลายรอยร้าวออกไปและทำการคำนวณหาค่าอัตราการปลดปล่อยพลังงานเฉพาะเอลิเมนต์ที่ได้รับผลกระทบจากการเลื่อนจุดต่อที่ตำแหน่งปลายรอยร้าวนี้เท่านั้น นอกจากนั้น Parks ยังแสดงให้เห็นว่าค่าอัตราการปลดปล่อยพลังงานที่ได้จากการวิเคราะห์ปัญหาด้วยวิธีนี้จะมีค่าเท่ากับค่าพารามิเตอร์เจอินทิกรัล (*J*-integral) ซึ่ง Parks ได้เรียกวิธีการหาค่าอัตราการปลดปล่อยพลังงาน

นี้ว่า Virtual Crack Extension Method นอกจากนี้ Parks ยังได้ประยุกต์วิธีการแก้ปัญหานี้เข้ากับวัสดุที่มีพฤติกรรมการเสียรูปแบบไม่เชิงเส้นที่บริเวณปลายรอยร้าวอีกด้วย

Henshell and Shaw [4] ได้ทำการศึกษาการใช้ระเบียบวิธีไฟไนต์เอลิเมนต์ในการคำนวณหาค่าพารามิเตอร์ตัวประกอบความเข้มของความเค้น (Stress Intensity Factor) ซึ่งเป็นพารามิเตอร์ที่สำคัญพารามิเตอร์หนึ่งสำหรับแนวทางกลศาสตร์การแตกหักแบบยืดหยุ่นเชิงเส้น โดยใช้ไอโซพาราเมตริกซ์เอลิเมนต์แบบแปดจุดต่อที่ทำการย้ายจุดต่อกึ่งกลางด้านที่อยู่ติดกับปลายรอยร้าวไปที่ตำแหน่งหนึ่งในสี่ของความยาวด้านใกล้กับปลายรอยร้าวซึ่งเอลิเมนต์ชนิดนี้จะก่อให้เกิดลักษณะการกระจายของค่าสนามความเค้นและสนามความเครียดที่บริเวณปลายรอยร้าวเช่นเดียวกับผลเฉลยที่ได้จากการวิเคราะห์ทางทฤษฎีทำให้สามารถใช้ค่าผลเฉลยความเค้นที่ได้จากระเบียบวิธีไฟไนต์เอลิเมนต์มาคำนวณหาค่าพารามิเตอร์ตัวประกอบความเข้มของความเค้นได้

Barsoum [5, 6] ได้ศึกษาการใช้ระเบียบวิธีไฟไนต์เอลิเมนต์ในการคำนวณหาค่าพารามิเตอร์ตัวประกอบความเข้มของความเค้นโดยใช้ไอโซพาราเมตริกซ์เอลิเมนต์อันดับสองทั้งแบบสองและสามมิติซึ่ง Barsoum ได้แสดงให้เห็นว่าความเป็นซิงกิวลาริตี้ (Singularity) ซึ่งแปรผันกับระยะจากปลายรอยร้าวแบบ $1/r$ ของเอลิเมนต์เหล่านี้จะเกิดขึ้นเมื่อด้านหนึ่งของเอลิเมนต์ถูกยุบมารวมกันที่ตำแหน่งปลายรอยร้าวแล้วปล่อยให้จุดต่อของด้านที่ถูกยุบมารวมกันนี้สามารถเคลื่อนที่ได้เป็นอิสระต่อกัน ขณะที่ความเป็นซิงกิวลาริตี้แบบ $1/\sqrt{r}$ จะเกิดขึ้นเมื่อจุดต่อที่ตำแหน่งกึ่งกลางของด้านที่อยู่ติดกับปลายรอยร้าวถูกเลื่อนไปที่ตำแหน่งหนึ่งในสี่ของความยาวด้านใกล้กับปลายรอยร้าวซึ่งความเป็นซิงกิวลาริตี้แบบ $1/\sqrt{r}$ นี้เหมาะสำหรับการวิเคราะห์รอยร้าวในวัสดุแบบยืดหยุ่นเชิงเส้น ขณะที่ความเป็นซิงกิวลาริตี้แบบ $1/r$ เหมาะสำหรับการวิเคราะห์รอยร้าวในวัสดุแบบพลาสติกสมบูรณ์ (Perfectly Plastic) นอกจากนั้น Barsoum ยังแสดงให้เห็นว่าซิงกิวลาริตี้เอลิเมนต์ต่าง ๆ เหล่านี้ยังสามารถนำไปประยุกต์ใช้ได้กับปัญหาความเครียดเริ่มต้นเนื่องจากอุณหภูมิอีกด้วย

deLorenzi [7, 8] ได้เปลี่ยนวิธี Virtual Crack Extension Method จากสมการไฟไนต์เอลิเมนต์มาอยู่ในรูปสมการเชิงวิเคราะห์ของอัตราการปลดปล่อยพลังงานโดยอาศัยหลักการ Mapping รูปร่างปัญหาที่ความยาวรอยร้าวหนึ่งไปเป็นรูปร่างปัญหาที่มีความยาวรอยร้าวเพิ่มขึ้นเพื่อหาค่าผลต่างของพลังงานศักย์ซึ่งสมการเชิงวิเคราะห์ที่ได้นี้สามารถนำไปประยุกต์ใช้ได้กับระเบียบวิธีเชิงตัวเลขแบบใดก็ได้ deLorenzi ได้นำวิธีที่ได้ประดิษฐ์ขึ้นนี้ไปประยุกต์ใช้กับระเบียบวิธีไฟไนต์เอลิเมนต์โดยสมการอัตราการปลดปล่อยพลังงานที่ได้จะถูกเขียนอยู่ในรูปการอินทิเกรตบนพื้นที่สำหรับการวิเคราะห์ปัญหาแบบสองมิติและอยู่ในรูปการอินทิเกรตบนปริมาตรสำหรับการวิเคราะห์ปัญหาแบบสามมิติ

Li et al. [9] และ Shih et al. [10] ได้นำเสนอระเบียบวิธี Domain Integral Method ซึ่งได้จากการเปลี่ยนสมการค่าพารามิเตอร์เจอินทิกรัลซึ่งแต่เดิมอยู่ในรูปของการอินทิเกรตตามเส้นรอบปลายรอยร้าวให้อยู่ในรูปของการอินทิเกรตบนพื้นที่รอบปลายรอยร้าวใด ๆ สำหรับปัญหาสองมิติและอยู่ในรูปของการอินทิเกรตบนปริมาตรใด ๆ รอบปลายรอยร้าวสำหรับปัญหาสามมิติโดยกำหนดฟังก์ชั่นต่อเนื่องใด ๆ ขึ้นมาแล้วทำการประยุกต์ทฤษฎีบทของเกาส์ (Gauss's Theorem) โดยสมการที่ได้นี้จะมีลักษณะคล้ายกับสมการของ deLorenzi ซึ่งได้จากวิธี Virtual Crack Extension Method นอกจากนี้ Li et al. ยังได้ทำการเปรียบเทียบผลคำนวณค่าพารามิเตอร์เจอินทิกรัลด้วยระเบียบวิธีไฟไนต์เอลิเมนต์จากวิธีดั้งเดิมซึ่งอยู่ในรูปการอินทิเกรตตามเส้นกับวิธีใหม่ซึ่งอยู่ในรูปของการอินทิเกรตบนพื้นที่โดเมนรอบปลายรอยร้าว โดยผลการเปรียบเทียบนั้นปรากฏว่าวิธีใหม่ให้ผลการคำนวณที่มีความถูกต้องดีกว่ามาก Shih et al. ยังได้ทำการทดสอบหาค่าพารามิเตอร์เจอินทิกรัลซึ่งเกิดจากผลของค่าความเค้นดึงที่ผิวรอยร้าวและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิด้วยวิธีใหม่นี้โดยผลลัพธ์ที่ได้แสดงให้เห็นถึงประสิทธิภาพของระเบียบวิธีใหม่นี้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลด้วยระเบียบวิธีไฟไนต์เอลิเมนต์

Moran and Shih [11, 12] ได้แสดงให้เห็นว่าสมการค่าพารามิเตอร์เจอินทิกรัลหรือค่าพารามิเตอร์ต่าง ๆ ที่เขียนอยู่ในรูปการอินทิเกรตบนเส้นหรือบนพื้นที่รอบปลายรอยร้าวนั้นคือรูป ๆ หนึ่งของสมการในรูปทั่วไปซึ่งสร้างขึ้นจากสมการความสมดุลย์ของโมเมนตัม (Momentum Balance) โดยสมการในรูปทั่วไปนี้สามารถนำไปประยุกต์ใช้ได้กับวัสดุและรอยร้าวที่มีพฤติกรรมต่าง ๆ ได้ Moran and Shih ยังได้แสดงเงื่อนไขที่สมการเหล่านี้จะไม่ขึ้นอยู่กับการอินทิเกรตบนเส้นรอบปลายรอยร้าว ทั้งบนเส้นที่มีขนาดเล็กเข้าสู่ศูนยที่ปลายรอยร้าว (Local Path Independence) และบนเส้นที่มีขนาดใหญ่ซึ่งอยู่ไกลออกไปจากปลายรอยร้าว (Global Path Independence) นอกจากนี้ยังแสดงวิธีการเปลี่ยนสมการเหล่านี้ให้อยู่ในรูปของการอินทิเกรตบนโดเมนเพื่อนำไปประยุกต์ใช้กับระเบียบวิธีไฟไนต์เอลิเมนต์โดยเฉพาะ

Omori et al. [13] ได้นำเสนอพารามิเตอร์ T_{ε}^{*}-integral เพื่อใช้ในการวิเคราะห์ร้อยร้าวที่เติบโตภายใต้สถานะอยู่ตัว (Stable Crack Growth) ซึ่งนิยามด้วยสมการเดียวกันกับสมการค่าพารามิเตอร์เจอินทิกรัลแต่ใช้ทฤษฎี Incremental Theory of Plasticity ในการคำนวณหาค่าความเค้นและค่าความเครียด โดยเส้นรอบปลายรอยร้าวที่ใช้ในการอินทิเกรตค่าพารามิเตอร์ T_{ε}^{*}-integral จะครอบคลุมรอยร้าวตั้งแต่ขนาดเริ่มต้นก่อนการเติบโตไปจนถึงความยาวรอยร้าวขนาดต่าง ๆ เมื่อรอยร้าวได้เติบไปแล้วและระยะห่างจากรอยร้าวถึงเส้นที่ทำการอินทิเกรตนี้จะมีค่าเท่ากับความหนาของชิ้นงาน โดย Omori et al. ได้ทำการเปรียบเทียบค่าพารามิเตอร์ T_{ε}^{*}-integral

ที่ได้จากการทดลองและจากระเบียบวิธีไฟไนต์เอลิเมนต์ นอกจากนี้ยังเปรียบเทียบค่าที่ได้กับค่าพารามิเตอร์เจอินทิกรัลที่ได้จากการอินทิเกรตบนเส้นใกล้ปลายรอยร้าวที่กำลังเติบโต (Near Field *J*-integral) กับค่าพารามิเตอร์เจอินทิกรัลที่ได้จากการอินทิเกรตบนเส้นที่ไกลจากปลายรอยร้าว (Far Field *J*-integral) จากการทดลองพบว่าค่าพารามิเตอร์ T_ε^*-integral ที่ได้จากการทดลองและระเบียบวิธีไฟไนต์เอลิเมนต์มีความสอดคล้องกัน ส่วนค่าพารามิเตอร์เจอินทิกรัลที่ได้จากการอินทิเกรตบนเส้นที่ใกล้ปลายรอยร้าวกับเส้นที่ไกลจากปลายรอยร้าวนั้นมีค่าแตกต่างกันมาก โดยค่าพารามิเตอร์เจอินทิกรัลที่ได้จากเส้นใกล้ปลายรอยร้าวนั้นจะมีค่าเข้าใกล้ศูนย์เมื่อรอยร้าวมีการเติบโต นอกจากนี้ Omori et al. ยังได้ทำการเปรียบเทียบค่าที่ได้จากการทดลองกับค่าที่ได้จากระเบียบวิธีไฟไนต์เอลิเมนต์สำหรับชิ้นทดสอบมาตรฐานรูปร่างต่าง ๆ ซึ่งผลที่ได้แสดงให้เห็นว่าค่าพารามิเตอร์ T_ε^*-integral ที่ได้มีความสอดคล้องกันและค่าที่ได้จะลู่เข้าสู่ค่าเดียวกันสำหรับชิ้นทดสอบมาตรฐานต่าง ๆ ซึ่งแสดงให้เห็นถึงความเป็นไปได้ในการนำค่าพารามิเตอร์ T_ε^*-integral มาใช้ในการวิเคราะห์ร้อยร้าวที่กำลังเติบโตภายใต้สถานะอยู่ตัวแทนค่าพารามิเตอร์เจอินทิกรัล

Newman et al. [14] ได้สรุปความก้าวหน้าของการนำค่าพารามิเตอร์ CTOA (Crack-Tip-Opening Angle) และค่าพารามิเตอร์ CTOD (Crack-Tip-Opening Displacem ent) ซึ่งเป็นค่าพารามิเตอร์มุมเปิดที่ปลายรอยร้าวและค่าพารามิเตอร์ระยะการเคลื่อนตัวของผิวรอยร้าวที่บริเวณปลายรอยร้าวตามลำดับ ไปใช้ในการวิเคราะห์ร้อยร้าวที่กำลังเติบโตภายใต้สถานะอยู่ตัว ในช่วงแรกของการนำค่าพารามิเตอร์ทั้งสองนี้ไปใช้นั้นยังไม่เป็นที่แพร่หลายมากนักเนื่องจากการทดลองหาค่ามุมเปิดวิกฤตที่ปลายรอยร้าวที่ค่าความยาวรอยร้าวที่เพิ่มขึ้นต่าง ๆ ด้วยระเบียบวิธีไฟไนต์เอลิเมนต์แบบสองมิติโดยใช้เงื่อนไขการคำนวณแบบความเค้นในระนาบหรือความเครียดในระนาบอย่างใดอย่างหนึ่งนั้น ให้ผลการคำนวณค่ามุมเปิดวิกฤตที่ปลายรอยร้าวมีค่าไม่คงที่ในช่วงแรกของค่าความยาวรอยร้าวที่เพิ่มขึ้น ภายหลัง Newman et al. ได้ทำการวิเคราะห์แล้วพบว่าพฤติกรรมของปลายรอยร้าวที่กำลังเติบโตนั้นเป็นแบบสามมิติซึ่งไม่สามารถจำลองได้ด้วยเงื่อนไขความเค้นหรือความเครียดในระนาบเพียงอย่างใดอย่างหนึ่งได้ เนื่องจากที่บริเวณปลายรอยร้าวด้านในของชิ้นงานนั้นสถานะความเค้น (State of Stress) จะมีลักษณะคล้ายกับกรณีเงื่อนไขความเครียดในระนาบ ขณะที่บริเวณปลายรอยร้าวที่ผิวของชิ้นงานนั้นสถานะความเค้นจะเป็นแบบความเค้นในระนาบ ด้วยเหตุนี้จึงมีการนำเสนอการวิเคราะห์แบบสองมิติที่เรียกว่า Plane Strain Core ซึ่งเป็นการใช้เอลิเมนต์ภายใต้สถานะความเครียดในระนาบร่วมกับเอลิเมนต์ภายใต้สถานะความเค้นในระนาบ โดยเอลิเมนต์แบบสถานะความเครียดในระนาบจะเรียงตัวกันอยู่ตามรอยร้าว ส่วนเอลิเมนต์แบบสถานะความเค้นในระนาบจะกระจายตัวอยู่ในบริเวณอื่นซึ่งผลการคำนวณที่ได้จากวิธีนี้นั้นมีความสอดคล้องกับผลการวิเคราะห์แบบสามมิติซึ่งให้ค่ามุมเปิดวิกฤตที่ปลายรอยร้าวมีค่าคงที่ที่ค่าความยาวรอยร้าวที่เพิ่มขึ้นขนาดต่าง ๆ นอกจากนี้ Newman et al. ยังได้ทำการ

คำนวณค่ามุมเปิดวิกฤตที่ปลายรอยร้าวจากชิ้นทดสอบมาตรฐานลักษณะต่าง ๆ เทียบกับค่าความยาวรอยร้าวที่เพิ่มขึ้นแล้วพบว่าหากค่าอัตราส่วนความยาวรอยร้าวต่อความหนา (Crack-Length-to-Thickness Ratio) และค่าอัตราส่วนความกว้างที่เหลืออยู่ต่อความหนา (Uncracked-Ligament-to-Thickness Ratio) มีค่ามากกว่าหรือเท่ากับสี่แล้วค่ามุมเปิดวิกฤตที่ปลายรอยร้าวที่ได้จะมีค่าเท่ากันในทุกชิ้นทดสอบมาตรฐาน

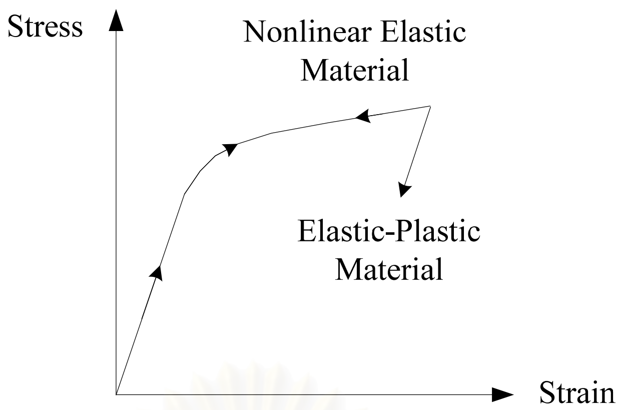
บทที่ 2

กลศาสตร์การแตกหักแบบอิลาสติก-พลาสติก

แนวทางกลศาสตร์การแตกหักแบบยืดหยุ่นเชิงเส้น (Linear Elastic Fracture Mechanics, LEFM) สามารถประยุกต์ใช้ได้กับชิ้นงานที่มีรอยร้าวตราบเท่าที่การเสียรูปแบบไม่เชิงเส้นที่บริเวณปลายรอยร้าวถูกจำกัดอยู่ในบริเวณเล็ก ๆ รอบปลายรอยร้าว ในวัสดุหลาย ๆ ชนิดโดยทั่วไปแล้วการเสียรูปแบบไม่เชิงเส้นที่บริเวณปลายรอยร้าวจะมีขนาดใหญ่เมื่อเปรียบเทียบกับขนาดรูปร่างของชิ้นงานเกินกว่าที่จะประยุกต์ใช้แนวทางกลศาสตร์การแตกหักแบบยืดหยุ่นเชิงเส้นได้ ด้วยเหตุนี้จึงต้องมีการนำเอาทฤษฎีกลศาสตร์การแตกหักอีกแขนงหนึ่งมาประยุกต์ใช้ซึ่งก็คือ กลศาสตร์การแตกหักแบบอิลาสติก-พลาสติก (Elastic-Plastic Fracture Mechanics, EPFM) โดยงานในวิทยานิพนธ์นี้จะเน้นไปที่การคำนวณหาค่าพารามิเตอร์เจอินทิกรัล (*J*-integral) ซึ่งเป็นค่าพารามิเตอร์ที่สำคัญค่าหนึ่งที่ใช้ในการบ่งบอกความรุนแรงของค่าสนามความเค้นที่บริเวณปลายรอยร้าวได้ โดยในทฤษฎีกลศาสตร์การแตกหักแบบอิลาสติก-พลาสติกนั้นยังมีค่าพารามิเตอร์อื่น ๆ ที่มีความสำคัญอีกเช่น ค่าพารามิเตอร์ CTOD [15], ค่าพารามิเตอร์ CTOA [14] และค่าพารามิเตอร์ T_{ε}^{*}-integral [13] เนื่องจากทฤษฎีที่ใช้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลนั้นจริง ๆ แล้วเป็นทฤษฎีที่ใช้กับวัตถุที่มีการเสียรูปแบบยืดหยุ่นไม่เชิงเส้น (Nonlinear Elastic Material) ซึ่งเป็นทฤษฎีที่ไม่ยากมากนัก ดังนั้นในวิทยานิพนธ์นี้จึงเลือกคำนวณค่าพารามิเตอร์เจอินทิกรัลนี้ต่อจากค่าตัวประกอบความเข้มของความเค้น (Stress Intensity Factor) ที่ใช้ในกลศาสตร์การแตกหักแบบยืดหยุ่นเชิงเส้นซึ่งได้ทำการศึกษาไปแล้วในงานวิทยานิพนธ์ก่อนหน้านี้ [16]

2.1 พารามิเตอร์เจอินทิกรัล

พารามิเตอร์เจอินทิกรัลเป็นพารามิเตอร์ที่ใช้บ่งบอกความรุนแรงของสนามความเค้นที่บริเวณปลายรอยร้าวสำหรับวัตถุที่มีการเสียรูปแบบยืดหยุ่นไม่เชิงเส้น โดยความแตกต่างระหว่างการเสียรูปแบบยืดหยุ่นไม่เชิงเส้นกับการเสียรูปแบบอิลาสติก-พลาสติก (Elastic-Plastic deformation) นั้นสามารถอธิบายได้ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 กราฟเปรียบเทียบความสัมพันธ์ระหว่างค่าความเค้นกับค่าความเครียดที่เกิดขึ้นในชิ้นงานของวัตถุแบบอิลาสติก-พลาสติกและแบบยืดหยุ่นไม่เชิงเส้น

ขณะที่วัตถุทั้งสองชนิดรับภาระ กราฟความสัมพันธ์ระหว่างค่าความเค้นและค่าความเครียดที่ได้ของวัตถุทั้งสองชนิดจะเหมือนกัน โดยความแตกต่างจะเกิดขึ้นเมื่อมีการปล่อยภาระที่กระทำกับชิ้นงาน (Unloading) ที่ค่าความเค้นเกินค่าความเค้นคราก (Yield Stress) โดยวัตถุแบบยืดหยุ่นไม่เชิงเส้นนั้นกราฟจะกลับมาที่สถานะเดิมก่อนการรับภาระ ขณะที่วัตถุแบบอิลาสติก-พลาสติกนั้นกราฟจะลดลงด้วยความชันเท่ากับค่าคงที่ของการยืดหยุ่น (Modulus of Elasticity) และเกิดค่าความเครียดถาวรขึ้นในวัตถุทำให้สมการความสัมพันธ์ระหว่างค่าความเค้นและค่าความเครียดจะต้องเขียนอยู่ในรูปความสัมพันธ์ระหว่างค่าความเค้นและค่าความเครียดที่เพิ่มขึ้น (Incremental Form) แทนค่าความเค้นและค่าความเครียดสุทธิ (Total Form) ที่ใช้กับวัตถุแบบยืดหยุ่นไม่เชิงเส้น โดยจากความสัมพันธ์ที่แตกต่างกันนี้เราจึงเรียกทฤษฎีที่ใช้กับวัตถุแบบอิลาสติก-พลาสติกว่าทฤษฎี Incremental Theory of Plasticity และเรียกทฤษฎีที่ใช้กับวัตถุแบบยืดหยุ่นไม่เชิงเส้นว่าทฤษฎี Deformation Theory of Plasticity โดยทฤษฎีทั้งสองจะให้ผลลัพธ์ที่เท่ากันหากภาระความเค้นที่เกิดขึ้นในชิ้นงานมีค่าเพิ่มขึ้นในอัตราส่วนเดียวกันและไม่เกิดการปล่อยภาระขึ้นในช่วงที่มีภาระมากระทำกับชิ้นงาน [15, 17]

Shih et al. [12] สร้างสมการค่าพารามิเตอร์เจอินทิกรัลขึ้นจากกฎทรงพลังงาน (Energy Balance Law) โดยเริ่มจากสมการการเคลื่อนที่ (Equation of Motion) ซึ่งเขียนอยู่ในรูป

$$\sigma_{ji,j} = -f_i + \rho \ddot{u}_i \quad (2.1)$$

โดยที่ σ_{ji} แทนค่าความเค้นสุทธิ (Total Stress)

f_i แทนค่าแรงวัตถุ (Body Force)

ρ แทนค่าความหนาแน่น (Density)

$\ddot{u}_i$ แทนค่าความเร่ง (Acceleration)

คูณสมการ (2.1) ด้วยความเร็ว $\dot{u}_i$ ทั้งสองข้างแล้วจัดรูปใหม่จะได้

$$\left(\sigma_{ji}\dot{u}_i\right)_{,j} = \sigma_{ji}\dot{u}_{i,j} - f_i\dot{u}_i + \rho\ddot{u}_i\dot{u}_i \quad (2.2)$$

ในกรณีที่ค่าความเครียดมีค่าน้อย (Small Strain) สมการความสัมพันธ์ระหว่างค่าความเครียดสุทธิและค่าการเคลื่อนตัวสามารถเขียนแทนได้ดังสมการ

$$\varepsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right) \quad (2.3)$$

โดยที่ ε_{ij} แทนค่าความเครียดสุทธิ (Total Strain)

u_i แทนค่าการเคลื่อนตัว (Displacement)

พจน์แรกทางด้านขวาของสมการ (2.2) สามารถเขียนใหม่ได้เป็น

$$\sigma_{ji}\dot{u}_{i,j} = \frac{1}{2}\left(\sigma_{ji}\dot{u}_{i,j} + \sigma_{ji}\dot{u}_{i,j}\right) \quad (2.4)$$

สมการ (2.4) เมื่อกระจายออกให้อยู่ในรูปผลบวกที่ค่า i และ $j = 1,2,3$ เราสามารถรวมสมการที่กระจายออกแล้วเขียนใหม่โดยใช้คุณสมบัติความสมมาตรของเมตริกซ์ความเค้น $\sigma_{ji} = \sigma_{ij}$ ได้เป็น

$$\sigma_{ji}\dot{u}_{i,j} = \sigma_{ij}\dot{\varepsilon}_{ij} \quad (2.5)$$

กำหนดให้ค่าความเครียดสุทธิประกอบด้วยค่าความเครียดต่าง ๆ ดังนี้

$$\varepsilon_{ij} = \varepsilon_{ij}^{m} + \varepsilon_{ij}^{ther} \quad (2.6)$$

โดยที่ ε_{ij}^{m} แทนค่าความเครียดทางกล (Mechanical Strain)

ε_{ij}^{ther} แทนค่าความเครียดเริ่มต้นเนื่องจากอุณหภูมิ (Thermal Strain)

เนื่องจากค่าความเครียดเริ่มต้นเนื่องจากอุณหภูมิมีค่าคงที่ไม่ขึ้นกับเวลา ดังนั้นสมการ (2.5) จะลดรูปลงเหลือ

$$\sigma_{ji}\dot{u}_{i,j} = \sigma_{ij}\dot{\varepsilon}_{ij}^{m} \quad (2.7)$$

จากภาคผนวก ก ในกรณีที่วัตถุเป็นของแข็งแบบยืดหยุ่น (Elastic Solid) ค่าความเค้นสามารถเขียนอยู่ในรูปของค่าพลังงานความเครียดหนาแน่นและค่าความเครียดทางกลได้เป็น

$$\sigma_{ij} = \frac{\partial W}{\partial \varepsilon_{ij}^{m}} \quad (2.8)$$

โดยที่ $W = \int_{0}^{\varepsilon_{ij}^{m}} \sigma_{ij} d\varepsilon_{ij}^{m}$ แทนค่าพลังงานความเครียดหนาแน่น (Strain Energy Density)

แทนสมการ (2.8) ลงในสมการ (2.7) จะได้ความสัมพันธ์

$$\sigma_{ji}\dot{u}_{i,j} = \frac{\partial W}{\partial \varepsilon_{ij}^{m}} \frac{\partial \varepsilon_{ij}^{m}}{\partial t} = \dot{W} \quad (2.9)$$

ในทำนองเดียวกันจากพจน์ที่สองและสามทางด้านขวาของสมการ (2.2) จะได้

$$F = \int_{0}^{t} f_i \dot{u}_i dt \quad (2.10)$$

$$L = \int_{0}^{t} \rho \ddot{u}_i \dot{u}_i dt \quad (2.11)$$

โดยที่ F แทนงานหนาแน่นเนื่องจากแรงวัตถุ (Work Density due to Body Force)

L แทนพลังงานจลน์หนาแน่น (Kinetic Energy Density)

ในกรณีที่วัตถุมีค่าความหนาแน่นคงที่เราสามารถอินทิเกรตสมการ (2.11) ได้เท่ากับ

$$L = \frac{1}{2}\rho \dot{u}_i^2 \quad (2.12)$$

จากสมการ (2.9-11) สมการ (2.2) สามารถลดรูปให้อยู่ในรูปทั่วไปได้เป็น

$$\varphi_{j,j} = \dot{\psi} \quad (2.13)$$

โดยที่

$$\varphi_j = \sigma_{ji}\dot{u}_i \quad (2.14\text{ ก})$$

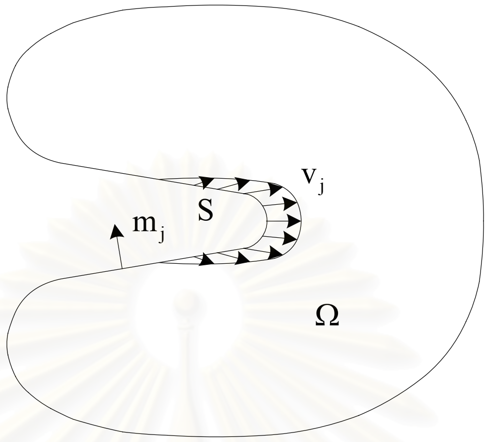
$$\dot{\psi} = \dot{W} - \dot{F} + \dot{L} \quad (2.14\text{ ข})$$

อินทิเกรตสมการ (2.13) บนปริมาตรใด ๆ แล้วประยุกต์ทฤษฎีบทของเกาส์ (Gauss's Theorem) เข้ากับพจน์แรกจะได้

$$\int_{\Omega} \varphi_{j,j} d\Omega = \int_{S} \varphi_j m_j dS \quad (2.15)$$

โดยที่ m_j แทนเวกเตอร์หนึ่งหน่วยตั้งฉากมีทิศพุ่งออกจากผิว S ที่ล้อมรอบปริมาตร Ω

พิจารณาปริมาตร Ω ใด ๆ ที่ผิวของปริมาตร S เคลื่อนที่ด้วยความเร็ว v_j ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 ปริมาตรควบคุมที่เปลี่ยนแปลงตามเวลาเนื่องจากผิวด้านนอกเคลื่อนที่ด้วยความเร็ว v_j

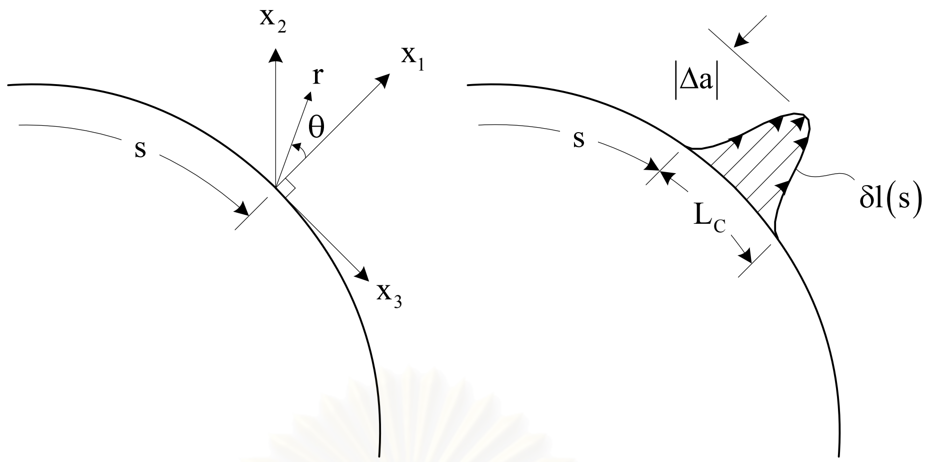
อัตราการเปลี่ยนแปลงของปริมาณ ψ ทั้งหมดบนปริมาตรควบคุมนี้ที่เวลาใด ๆ สามารถหาได้จากทฤษฎีบทการเคลื่อนตัวของเรย์โนล์ด (Reynolds Transport Theorem) เป็น

$$\frac{d}{dt}\int_{\Omega}\psi\, d\Omega = \int_{\Omega}\dot{\psi}\, d\Omega + \int_{S}\psi\, v_j m_j dS \quad (2.16)$$

จากสมการ (2.13) และ (2.15) สมการ (2.16) สามารถเขียนได้ใหม่เป็น

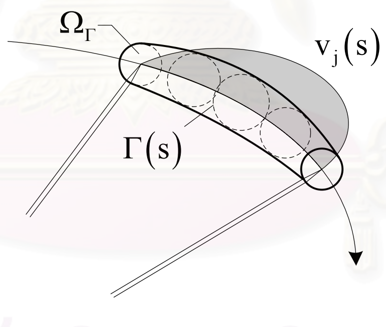
$$\frac{d}{dt}\int_{\Omega}\psi\, d\Omega = \int_{S}\left(\varphi_j + \psi\, v_j\right) m_j dS \quad (2.17)$$

เพื่อหาค่าอัตราพลังงานที่ถูกปลดปล่อยออกจากวัตถุ (Energy Release Rate) ที่ตำแหน่ง s ใด ๆ บนขอบรอยร้าว (Crack Front) เนื่องจากขอบรอยร้าวเติบโตออกไปด้วยขนาดต่อเนื่องใด ๆ $\delta l(s)$ บนความยาวขอบรอยร้าว L_C ที่มีขนาดสูงสุดเท่ากับ $|\Delta a|$ ที่ตำแหน่ง s ซึ่งเป็นค่าที่ใช้ในการวิเคราะห์ความรุนแรงของรอยร้าวในปัญหาสามมิติ พิจารณาวัตถุปริมาตร Ω ซึ่งมีรอยร้าววางตัวอยู่ในระนาบ $x_1 - x_3$ (Planar Crack) และขอบรอยร้าวมีลักษณะต่อเนื่อง โดยรอยร้าวจะเติบโตอยู่ในระนาบในทิศตั้งฉากกับขอบรอยร้าว x_1 ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 วัตถุที่มีรอยร้าววางตัวอยู่ในระนาบ

แบ่งปริมาตรของวัตถุนี้ออกเป็นสองส่วน โดยส่วนแรกเป็นปริมาตรทรงกระบอกเล็ก ๆ Ω_Γ ล้อมรอบขอบรอยร้าวด้วยเส้น $\Gamma(s)$ เป็นระยะ L_C เคลื่อนที่ด้วยความเร็วเดียวกับขอบรอยร้าว $v_j(s)$ ที่ตำแหน่งขอบรอยร้าว s ต่าง ๆ ตลอดทั้งปริมาตรและมีผิว S_Γ ล้อมรอบปริมาตรนี้ โดยที่ปลายทั้งสองของปริมาตรความเร็ว $v_j(s)$ มีค่าเป็นศูนย์ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 ปริมาตรทรงกระบอกเล็ก ๆ รอบขอบรอยร้าว

ส่วนที่สองเป็นปริมาตรที่เหลือทั้งหมดของวัตถุนี้ $\Omega - \Omega_\Gamma$ ซึ่งผิวของปริมาตรนี้ที่ไม่สัมผัสกับผิวของปริมาตรทรงกระบอก $S - S_\Gamma$ ถูกกำหนดให้ไม่มีการเคลื่อนที่ ดังนั้นทำการประยุกต์สมการ (2.17) เข้ากับปริมาตร $\Omega - \Omega_\Gamma$ จะได้

$$\frac{d}{dt}\int_{\Omega-\Omega_\Gamma} \psi \, d\Omega = \int_{S-S_\Gamma} \varphi_j m_j dS + \int_{S_\Gamma} \left(\varphi_j + \psi v_j\right) m_j dS \quad (2.18)$$

โดยพจน์ทางซ้ายของสมการ (2.18) แทนอัตราการเพิ่มขึ้นของพลังงานภายใน (Internal Energy) พจน์แรกทางขวาแทนอัตราการไหลของพลังงานเข้าสู่ปริมาตร $\Omega-\Omega_\Gamma$ ผ่านผิว $S-S_\Gamma$ ขณะที่พจน์สุดท้ายแทนอัตราการไหลของพลังงานผ่านเข้าสู่ผิว S_Γ เพื่อหาค่าอัตราพลังงานที่ถูกปลดปล่อยออกจากวัตถุปริมาตร Ω เนื่องจากขอบรอยร้าวเติบโตออกด้วยความเร็ว v_j เราจึงกำหนดให้ $\Omega_\Gamma \to 0$ หรือ $\Gamma \to 0$ ดังนั้นจากสมการ (2.18) อัตราพลังงานที่ถูกปลดปล่อยออกจากวัตถุเนื่องจากการเติบโตของขอบรอยร้าวตลอดขอบ δs จะมีค่าเท่ากับ

$$J_\Gamma = -\lim_{\Gamma\to 0} \oint_{S_\Gamma} \left(\varphi_j + \psi_j\right) m_j dS \quad (2.19)$$

เนื่องจาก m_j ในสมการ (2.19) แทนเวกเตอร์หนึ่งหน่วยตั้งฉากมีทิศพุ่งออกจากผิวของปริมาตร $\Omega-\Omega_\Gamma$ ซึ่งมีทิศตรงข้ามกับเวกเตอร์หนึ่งหน่วยตั้งฉาก n_j ที่ผิวของปริมาตร Ω_Γ ดังนั้นสมการ (2.19) สามารถเขียนใหม่ในรูปอินทิเกรตบนผิวของปริมาตร Ω_Γ ได้เป็น

$$J_\Gamma = \lim_{\Gamma\to 0} \oint_{S_\Gamma} \left(\varphi_j + \psi_j\right) n_j dS \quad (2.20)$$

พิจารณาระบบพิกัดฉากใด ๆ ที่เคลื่อนที่ไปพร้อมกับขอบรอยร้าวที่ตำแหน่ง s ด้วยความเร็วเดียวกับขอบรอยร้าว $v_k(s)$ จากการกระจายอนุกรมเทย์เลอร์ (Taylor's Series) จะได้ความสัมพันธ์ [18]

$$\frac{du_i}{dt} = \frac{\partial u_i}{\partial t} + \frac{\partial u_i}{\partial x_1} v_1 + \frac{\partial u_i}{\partial x_2} v_2 + \frac{\partial u_i}{\partial x_3} v_3 \quad (2.21)$$

โดยที่ $\frac{\partial u_i}{\partial t}$ แทนอัตราการเปลี่ยนแปลงของการเคลื่อนตัวที่สังเกตุจากระบบพิกัดฉากที่อยู่กับที่ซึ่งก็คือ $\dot{u}_i$ ที่แสดงในสมการก่อนหน้านี้ $\frac{du_i}{dt}$ แทนอัตราการเปลี่ยนแปลงของการเคลื่อนตัวที่สังเกตุจากระบบพิกัดฉากที่เคลื่อนที่ด้วยความเร็ว v_k ไปพร้อมกับปลายรอยร้าว ในกรณีที่ขอบรอยร้าวเติบโตภายใต้สถานะอยู่ตัว (Steady State Crack Growth) พจน์ทางด้านซ้ายของสมการ (2.21) มีค่าเท่ากับศูนย์ซึ่งมีความหมายทางกายภาพว่าการเคลื่อนตัวที่ตำแหน่งเทียบจากขอบรอยร้าวเดียวกันที่ความยาวรอยร้าวต่าง ๆ จะมีค่าเท่ากัน โดยทั่วไปแล้วที่บริเวณขอบรอยร้าวสามพจน์สุดท้ายของสมการ (2.21) จะมีค่ามากกว่าพจน์แรกทางด้านซ้ายมากเนื่องจากที่บริเวณขอบรอยร้าวการเปลี่ยนแปลงของการเคลื่อนตัวเทียบกับตำแหน่งซึ่งแปรผันตรงกับค่าความเครียดจะมีค่าเข้าสู่อนันต์ดังนั้นสมการ (2.21) เมื่อนำมาใช้ที่บริเวณขอบรอยร้าวสามารถลดรูปลงเหลือ

$$\dot{u}_i = -u_{i,k} v_k \quad (2.22)$$

ดังนั้นสมการ (2.14ก) สามารถเขียนใหม่ได้เป็น

$$\varphi_j \;=\; -\sigma_{ji}u_{i,k}v_k \quad (2.23)$$

เนื่องจากที่ผิว S_e^+ และ S_e^- เวกเตอร์ v_j มีค่าเป็นศูนย์ ดังนั้นจากสมการ (2.14 ข) และ (2.23) เราสามารถเขียนสมการ (2.20) ได้ใหม่เป็น

$$J_\Gamma \;=\; \lim_{\Gamma\to 0}\int_{S_t} n_j H_{jk} v_k dS \quad (2.24)$$

เนื่องจากแรงวัตถุและการเคลื่อนตัวมีค่าไม่เข้าสู่อนันต์ที่บริเวณขอบรอยร้าว ดังนั้นเมื่อเส้น $\Gamma\to 0$ อินทิเกรตพจน์ที่เกี่ยวข้องกับงานหนาแน่นเนื่องจากแรงวัตถุจะมีค่าเป็นศูนย์ทำให้ H_{jk} ในสมการ (2.24) ลดรูปลงเหลือ

$$H_{jk} \;=\; \left(W+L\right)\delta_{jk} - \sigma_{ji}u_{i,k} \quad (2.25)$$

โดยที่ H_{jk} แทนเทนเซอร์โมเมนตัมเนื่องจากพจน์พลังงานต่าง ๆ

δ_{jk} แทนครอนเน็คเคอร์เด็ลต้า (Kronecker Delta)

ในวิทยานิพนธ์ฉบับนี้เราพิจารณาเฉพาะวัตถุยืดหยุ่นที่มีคุณสมบัติทางอุณหภูมิไม่ขึ้นกับทิศทางและภาระที่กระทำกับวัตถุเป็นแบบค่อย ๆ เพิ่มขึ้น (Elastic Solid under Quasistatic Isotherm al Conditions) ดังนั้นสมการ (2.25) สามารถเขียนใหม่ในกรณีนี้ได้เป็น

$$P_{jk} \;=\; W\delta_{jk} - \sigma_{ji}u_{i,k} \quad (2.26)$$

โดยที่ P_{jk} แทนเทนเซอร์โมเมนตัมพลังงานยืดหยุ่นของเอสเชลบาย (Eshelby's Elastic Energy-Momentum Tensor)

พิจารณารูปที่ 2.3 หากเรากำหนดให้ $v_k = \dfrac{|\Delta a| l_k}{dt}$ โดยที่ $|\Delta a|$ แทนขนาดที่ขอบรอยร้าวเติบโตสูงสุดบนช่วง L_C และ l_k แทนเวกเตอร์การเติบโตของขอบรอยร้าวต่อระยะการเติบโตสูงสุด ดังนั้นค่าพลังงานที่ถูกปลดปล่อยออกจากวัตถุเนื่องจากขอบรอยร้าวเติบโต $\delta l(s)$ ในช่วงขอบรอยร้าว L_C ดังแสดงในรูปที่ 2.3 มีค่าเท่ากับ

$$\bar{J}|\Delta a| \;=\; |\Delta a|\lim_{\Gamma\to 0}\int_{S_t} n_j P_{jk} l_k dS \quad (2.27)$$

โดยที่ $\bar{J}$ แทนค่าพลังงานที่ถูกปลดปล่อยออกจากวัตถุเฉลี่ยต่อความยาวรอยร้าว

เพื่อหาค่าพารามิเตอร์เจอินทิกรัลซึ่งก็คือค่าพลังงานที่ถูกปลดปล่อยออกจากวัตถุที่ตำแหน่ง s ใด ๆ บนขอบรอยร้าวต่อพื้นที่รอยร้าวที่เพิ่มขึ้น $J(\mathrm{s})$ สมการ (2.27) สามารถเขียนอยู่ในรูปใหม่ได้เป็น

$$\bar{J}|\Delta \mathrm{a}| = \int_{\mathrm{L_C}} J(\mathrm{s})\delta \mathrm{l}(\mathrm{s})\mathrm{ds} \quad (2.28)$$

$\delta \mathrm{l}(\mathrm{s})$ แทนค่าความยาวขอบรอยร้าวที่เพิ่มขึ้นที่ตำแหน่งบนขอบรอยร้าว s โดยมีค่าเท่ากับ

$$\delta \mathrm{l}(\mathrm{s}) = |\Delta \mathrm{a}|\mathrm{l_k}(\mathrm{s})\nu_{\mathrm{k}}(\mathrm{s}) \quad (2.29)$$

โดยที่ $\nu_{\mathrm{k}}(\mathrm{s})$ แทนเวกเตอร์หนึ่งหน่วยตั้งฉากกับขอบรอยร้าววางตัวในระนาบรอยร้าว

จากสมการ (2.27-29) จะได้ความสัมพันธ์เพื่อหาค่า $J(\mathrm{s})$ ซึ่งสามารถนำไปประยุกต์ใช้ในระเบียบวิธีไฟไนต์เอลิเมนต์ต่อไปเป็น

$$\int_{\mathrm{L_C}} J(\mathrm{s})\mathrm{l_k}(\mathrm{s})\nu_{\mathrm{k}}(\mathrm{s})\mathrm{ds} = \lim_{\Gamma \to 0} \int_{\mathrm{S_t}} \mathrm{n_j P_{jk} l_k dS} \quad (2.30)$$

สำหรับปัญหารอยร้าวสองมิติและสมมาตรรอบแกน ค่าพารามิเตอร์เจอินทิกรัลจะมีค่าคงที่ตลอดความยาวขอบรอยร้าวดังนั้นสมการ (2.30) สามารถเขียนอยู่ในรูป

$$J = \frac{\lim_{\Gamma \to 0} \int_{\mathrm{S_t}} \mathrm{n_j P_{jk} l_k dS}}{\int_{\mathrm{L_C}} \mathrm{l_k}(\mathrm{s})\nu_{\mathrm{k}}(\mathrm{s})\mathrm{ds}} \quad (2.31)$$

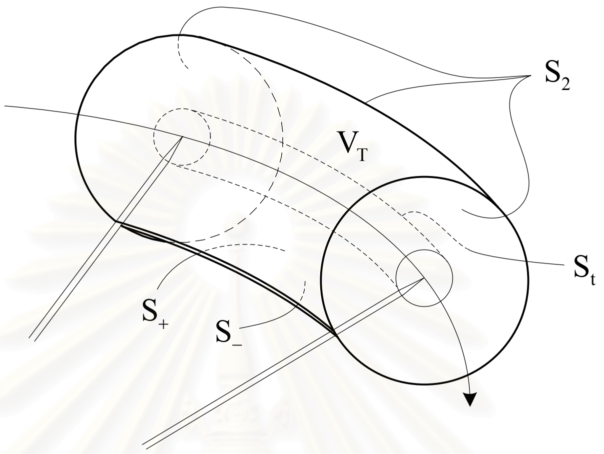
โดยสมการทั้งหมดสามารถนำไปใช้คำนวณได้ในระบบพิกัดฉากใด ๆ

2.2 พารามิเตอร์เจอินทิกรัลในรูปอินทิเกรตบนโดเมน

ในการหาค่าพารามิเตอร์เจอินทิกรัลด้วยระเบียบวิธีไฟไนต์เอลิเมนต์จากสมการ (2.30) หรือ (2.31) ให้ได้ค่าที่ถูกต้องนั้นไม่สามารถทำได้โดยง่ายเนื่องจากสมการอยู่ในรูปการอินทิเกรตบนผิวทรงกระบอกเล็ก ๆ รอบขอบรอยร้าวทำให้เอลิเมนต์ที่ใช้ต้องมีขนาดเล็กมากตามไปด้วย จากภาคผนวก ข เราทราบว่าที่บริเวณขอบรอยร้าวค่าความเค้นและค่าความเครียดมีค่าเข้าสู่อนันต์เนื่องจากสนามเอกพันธ์ของ HRR (Hutchinson-Rice-Rosengren Singularity) ทำให้ผลเฉลยค่าความเค้นและค่าความเครียดที่ได้จากระเบียบวิธีไฟไนต์เอลิเมนต์มีค่าความผิดพลาดสูง ดังนั้นเพื่อให้การคำนวณด้วยระเบียบวิธีไฟไนต์เอลิเมนต์มีความถูกต้องมากขึ้น Shih et al. [12] จึงได้

เปลี่ยนสมการที่ใช้คำนวณค่าพารามิเตอร์เจอินทิกรัลเสียใหม่ดังนี้ พิจารณาสมการ (2.27) ซึ่งเขียนใหม่อยู่ในรูป

$$\bar{J} = \lim_{\Gamma\to 0} \int_{S_t} m_j P_{jk} l_k \, dS \quad (2.32)$$



รูปที่ 2.5 ปริมาตรวงแหวนรอบขอบรอยร้าว

พิจารณารูปที่ 2.5 หากเรากำหนดให้เวกเตอร์ q_k มีค่าเท่ากับเวกเตอร์ l_k บนผิว S_t มีค่าเป็นศูนย์บนผิว S_2 มีทิศทางตั้งฉากกับเวกเตอร์ m_k บนผิว S_+ และ S_- และมีค่าใด ๆ ในปริมาตรวงแหวน สมการ (2.32) สามารถเขียนใหม่ในรูปเวกเตอร์ q_k และเวกเตอร์หนึ่งหน่วยตั้งฉากมีทิศพุ่งออกจากผิวปริมาตรวงแหวน S_T ได้เป็น

$$\bar{J} = -\oint_{S_T} m_j P_{jk} q_k \, dS + \int_{S_+ + S_-} t_i u_{i,k} q_k \, dS \quad (2.33)$$

โดยที่ $t_i = m_j \sigma_{ji}$ แทนค่าความเค้นที่ผิว (Surface Traction)

ประยุกต์ทฤษฎีบทของเกาส์เข้ากับพจน์แรกทางด้านขวาของสมการ (2.33) จะได้

$$\bar{J} = -\int_{V_T} \left(P_{jk} q_k\right)_{,j} dV + \int_{S_+ + S_-} t_i u_{i,k} q_k \, dS \quad (2.34)$$

แทนสมการ (2.26) ลงในพจน์แรกทางด้านขวาของสมการ (2.34) ได้

$$-\int_{V_T} \left(P_{jk} q_k\right)_{,j} dV = \int_{V_T} \left(\sigma_{ji} u_{i,k} - W \delta_{jk}\right) q_{k,j} \, dV + \int_{V_T} \left(\sigma_{ji} u_{i,k} - W \delta_{jk}\right)_{,j} q_k \, dV \quad (2.35)$$

กระจายพจน์ที่สองด้านขวามือของสมการ (2.35) ได้

$$\int_{V_T} \left(\sigma_{ji} u_{i,k} - W\delta_{jk}\right)_{,j} q_k dV \;=\; \int_{V_T} \left(\sigma_{ji,j} u_{i,k} + \sigma_{ji} u_{i,j,k} - W_{,k}\right) q_k dV \quad (2.36)$$

จากสมการ (2.1) พจน์แรกทางด้านขวาของสมการ (2.36) เมื่อไม่พิจารณาผลของความเฉื่อย สามารถเขียนได้ใหม่เป็น

$$\sigma_{ji,j} u_{i,k} \;=\; -f_i u_{i,k} \quad (2.37)$$

ในทำนองเดียวกันกับสมการ (2.5) พจน์ที่สองทางขวาของสมการ (2.36) สามารถเขียนอยู่ในรูป

$$\sigma_{ji} u_{i,j,k} \;=\; \sigma_{ij} \varepsilon_{ij,k} \quad (2.38)$$

จากสมการ (2.8) ค่าอนุพันธ์ของค่าพลังงานความเครียดหนาแน่นเทียบกับพิกัดฉากสามารถเขียนใหม่ได้

$$W_{,k} \;=\; \sigma_{ij} \varepsilon^{m}_{ij,k} \;+\; \left.\frac{\partial W}{\partial x_k}\right|_{\text{explicit}} \quad (2.39)$$

พจน์สุดท้ายของสมการ (2.39) เกิดขึ้นในกรณีที่คุณสมบัติของวัสดุ (Material Property) มีค่าไม่คงที่ขึ้นอยู่กับตำแหน่งบนพิกัดฉาก หากกำหนดให้ค่าความเครียดเริ่มต้นเนื่องจากอุณหภูมิดังแสดงในสมการ (2.6) มีค่าเท่ากับ

$$\varepsilon^{\text{ther}}_{ij} \;=\; \kappa \Theta \delta_{ij} \quad (2.40)$$

โดยที่ κ แทนค่าสัมประสิทธิ์การขยายตัวเนื่องจากสนามอุณหภูมิ (Coefficient of Thermal Expansion)

Θ แทนค่าสนามอุณหภูมิ (Temperature)

ดังนั้นจากสมการ (2.6) และ (2.40) สมการ (2.39) สามารถเขียนใหม่ได้ในกรณีที่คุณสมบัติของวัสดุมีค่าคงที่เป็น

$$W_{,k} \;=\; \sigma_{ij} \varepsilon_{ij,k} - \kappa \sigma_{ii} \Theta_{,k} \quad (2.41)$$

แทนสมการ (2.35-38) และ (2.41) ลงในสมการ (2.34) จะได้

$$\bar{J} \;=\; \int_{V_T} \left[\left(\sigma_{ji} u_{i,k} - W \delta_{jk}\right) q_{k,j} + \left(\kappa \sigma_{ii} \Theta_{,k} - f_i u_{i,k}\right) q_k \right] dV \;-\; \int_{S_+ + S_-} t_i u_{i,k} q_k dS \quad (2.42)$$

ซึ่งอยู่ในรูปที่เหมาะกับการประยุกต์ใช้กับระเบียบวิธีไฟไนต์เอลิเมนต์เพราะเป็นการอินทิเกรตบนปริมาตรและพื้นผิวจำกัดต่างจากสมการ (2.32) ซึ่งอยู่ในรูปการอินทิเกรตบนผิวเล็ก ๆ เข้าสู่ศูนย์รอบขอบรอยร้าว แทนสมการ (2.42) ลงในสมการ (2.30) จะได้สมการความสัมพันธ์ของค่าพารามิเตอร์เจอินทิกรัลตามขอบรอยร้าว $J(\mathrm{s})$ เป็น

$$\int_{\mathrm{L_C}} J(\mathrm{s})\mathrm{l_k}(\mathrm{s})\nu_{\mathrm{k}}(\mathrm{s})\mathrm{ds} \quad = \bar{J} \quad (2.43)$$

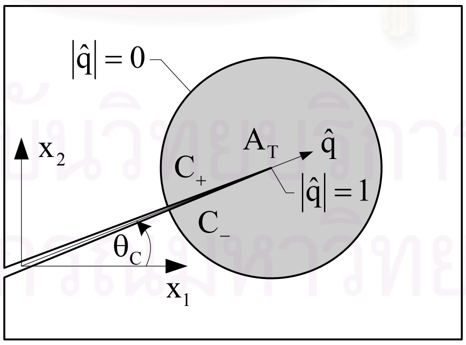
สำหรับปัญหารอยร้าวสองมิติและสมมาตรรอบแกน ค่าพารามิเตอร์เจอินทิกรัลจะมีค่าคงที่ตลอดความยาวขอบรอยร้าว ดังนั้นจากสมการ (2.43) จะได้ความสัมพันธ์

$$J \quad = \frac{\bar{J}}{\int_{\mathrm{L_C}} \mathrm{l_k}(\mathrm{s})\nu_{\mathrm{k}}(\mathrm{s})\mathrm{ds}} \quad (2.44)$$

โดย $\bar{J}$ มีค่าดังแสดงในสมการ (2.42)

2.2.1 พารามิเตอร์เจอินทิกรัลสำหรับปัญหาสองมิติ

พิจารณารอยร้าวแบบระนาบสำหรับปัญหาสองมิติที่มีแกนพิกัดฉากแกนหนึ่งมีทิศทางตามขอบรอยร้าวซึ่งวางตัวอยู่ตามความหนาของชิ้นงานและแกนที่เหลือวางตัวอยู่บนระนาบของชิ้นงานโดยมีรอยร้าววางตัวทำมุมใด ๆ กับแกนทั้งสองบนระนาบนี้ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 การวิเคราะห์รอยร้าวแบบระนาบสำหรับปัญหาสองมิติ

สำหรับปัญหาสองมิติค่าตัวแปรทั้งหมดไม่ขึ้นกับทิศทางตามความหนาดังนั้นสมการ (2.42) สามารถเขียนอยู่ในรูปการอินทิเกรตบนพื้นที่และบนเส้นในระนาบได้เป็น

$$\bar{J} = \int_{A_T}\Big[\big(\sigma_{ji}u_{i,k} - W\delta_{jk}\big)q_{k,j} + \big(\kappa\sigma_{ii}\Theta_{,k} - f_i u_{i,k}\big)q_k\Big]dA$$
$$- \int_{C_+ + C_-} L\,t_i u_{i,k} q_k dC \quad (2.45)$$

โดยที่ L แทนค่าความหนาของชิ้นงาน

เนื่องจาก $q_3 = f_3 = t_3 = q_{k,3} = q_{3,j} = \sigma_{13} = \sigma_{23} = 0$ และ $l_k(s)\nu_k(s) = 1$ ดังนั้นจากสมการ (2.44) และ (2.45) จะได้สมการค่าพารามิเตอร์เจอินทิกรัลเป็น

$$J = \int_{A_T}\Big[\big(\boldsymbol{\sigma}_{ji}\;{}_{i,k}W \quad \delta_{jk}\big)q_{k,j} + \big(3\kappa\bar{\sigma}\Theta_{,k} - f_i u_{i,k}\big)q_k\Big]dA - \int_{C_+ + C_-} t_i u_{i,k} q_k d\Gamma \quad (2.46)$$

โดยที่ $i, j, k = 1, 2$ แทนแกนพิกัดฉากทั้งสองบนระนาบของชิ้นงาน

$\bar{\sigma} = \dfrac{(\sigma_{11} + \sigma_{22} + \sigma_{33})}{3}$ แทนค่าความเค้นตั้งฉากเฉลี่ย (Mean Stress)

C_+, C_- แทนเส้นที่ต้องทำการอินทิเกรตบนผิวรอยร้าวทั้งสอง

ในกรณีปัญหาสองมิติจะได้ขนาดของเวกเตอร์ q_k มีค่าเท่ากับหนึ่งที่ตำแหน่งปลายรอยร้าวและมีค่าเท่ากับศูนย์ที่ขอบของพื้นที่ A_T ขณะที่มีค่าอยู่ระหว่างหนึ่งถึงศูนย์บนพื้นที่ A_T โดยเวกเตอร์ q_k เมื่อเขียนอยู่ในพิกัด x_1-x_2 ดังแสดงในรูปที่ 2.6 แล้วสามารถเขียนแทนได้ดังสมการ

$$q_1 = |\hat{q}|\cos\theta_C \quad (2.47\text{ก})$$

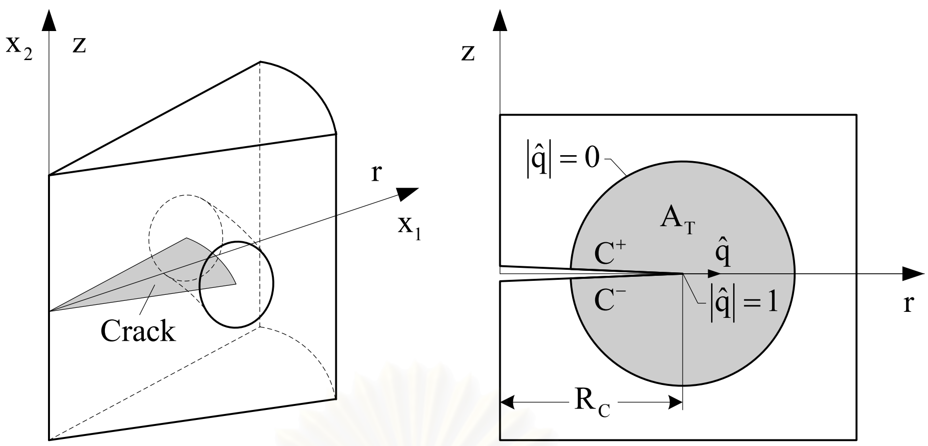
$$q_2 = |\hat{q}|\sin\theta_C \quad (2.47\text{ข})$$

โดยที่ $|\hat{q}|$ แทนขนาดของเวกเตอร์ q_k

θ_C แทนมุมที่ระนาบรอยร้าวกระทำกับแกน x_1

2.2.2 พารามิเตอร์เจอินทิกรัลสำหรับปัญหาสมมาตรรอบแกน

พิจารณารอยร้าวแบบระนาบสำหรับปัญหาสมมาตรรอบแกนที่มีระบบแกนพิกัดฉากและพิกัดทรงกระบอกดังแสดงในรูปที่ 2.7



รูปที่ 2.7 การวิเคราะห์รอยร้าวแบบระนาบสำหรับปัญหาสมมาตรรอบแกน

สำหรับปัญหาสมมาตรรอบแกนค่าตัวแปรต่างจะไม่ขึ้นอยู่กับค่ามุม θ ในพิกัดทรงกระบอกและรอยร้าวจะเติบโตออกในแนวรัศมีเท่านั้น ดังนั้นสมการ (2.42) สามารถเขียนอยู่ในรูปอินทิเกรตบนพื้นที่และบนเส้นในระนาบหน้าตัดได้เป็น

$$\bar{J} \;=\; \pi \int_{A_T} \Big[\big(\sigma_{ji} u_{i,k} - W\delta_{jk} \big) q_{k,j} + \big(\kappa\sigma_{ii}\Theta_{,k} - f_i u_{i,k} \big) q_k \Big] r\,dA$$

$$-\;\pi \int_{C_+ + C_-} t_i u_{i,k} q_k r\,dC \quad (2.48)$$

บนระนาบนี้แกนพิกัด $\hat{i}_1 = \hat{i}_r$, $\hat{i}_2 = \hat{i}_z$ และ $\hat{i}_3 = -\hat{i}_\theta$ ดังนั้นจะได้ $q_1 = q_r \neq 0$ และ $f_3 = t_3 = 0$ ซึ่งทำให้สองพจน์สุดท้ายของสมการ (2.48) ลดรูปลงเหลือ

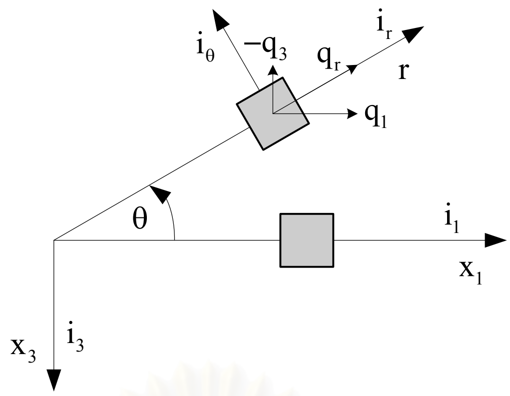
$$\int_{A_T} \big(\kappa\sigma_{ii}\Theta_{,k} + f_i u_{i,k} \big) q_k r\,dA \;=\; \int_{A_T} \big(3\kappa\bar{\sigma}\Theta_{,r} - f_\gamma u_{\gamma,r} \big) q_r r\,dA \quad (2.49\text{ก})$$

$$\int_{C_+ + C_-} t_i u_{i,k} q_k r\,dC \;=\; \int_{C_+ + C_-} t_\gamma u_{\gamma,r} q_r r\,dC \quad (2.49\text{ข})$$

โดยที่ $\gamma = r, z$ แทนแกนพิกัดทรงกระบอกทั้งสองบนระนาบ

$\bar{\sigma} = \dfrac{(\sigma_{rr} + \sigma_{zz} + \sigma_{\theta\theta})}{3}$ แทนค่าความเค้นตั้งฉากเฉลี่ย

จากการสังเกตุจะได้ $q_{1,1} = q_{r,r}$, $q_{1,2} = q_{r,z}$, $q_{2,1} = q_{2,2} = q_{2,3} = q_{3,1} = q_{3,2} = 0$ บนระนาบนี้ พิจารณารูปที่ 2.8 จะได้สมการความสัมพันธ์



รูปที่ 2.8 เอลิเมนต์เล็ก ๆ ของวัตถุสมมาตรรอบแกน

$$q_1 = q_r \cos\theta \qquad (2.50ก)$$

$$q_3 = q_r \sin\theta \qquad (2.50ข)$$

ดังนั้นค่าอนุพันธ์ $q_{1,3}$ และ $q_{3,3}$ จะสามารถคำนวณได้เท่ากับ

$$q_{1,3} = \lim_{\theta\to 0} \frac{q_r - (q_r\cos\theta)}{0-(-r\theta)} = 0 \qquad (2.51ก)$$

$$q_{3,3} = \lim_{\theta\to 0} \frac{0-(-q_r\sin\theta)}{0-(-r\theta)} = \frac{q_r}{r} \qquad (2.51ข)$$

ในทำนองเดียวกันบนระนาบนี้จะได้ความสัมพันธ์ของเมตริกซ์ค่าความเค้นเป็น

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} = \begin{bmatrix} \sigma_{rr} & \sigma_{rz} & \sigma_{r\theta} \\ \sigma_{zr} & \sigma_{zz} & \sigma_{z\theta} \\ \sigma_{\theta r} & \sigma_{\theta z} & \sigma_{\theta\theta} \end{bmatrix} = \begin{bmatrix} \sigma_{rr} & \sigma_{rz} & 0 \\ \sigma_{rz} & \sigma_{zz} & 0 \\ 0 & 0 & \sigma_{\theta\theta} \end{bmatrix} \qquad (2.52)$$

ทำให้พจน์แรกทางด้านขวาของสมการ (2.48) สามารถเขียนอยู่ในรูปใหม่ได้เป็น

$$\int_{A_T} \left(\sigma_{ji}u_{i,k} - W\delta_{jk}\right)q_{k,j}r\,dA = \int_{A_T} \left[\left(\sigma_{\beta\gamma}u_{\gamma,r} - W\delta_{\beta r}\right)q_{r,\beta}r + \left(\sigma_{\theta\theta}u_{3,3} - W\right)q_r\right]dA \qquad (2.53)$$

ในทำนองเดียวกันกับสมการ (2.50ข) และ (2.51ข) จะได้

$$u_{3,3} = \lim_{\theta\to 0} \frac{0-(-u_r\sin\theta)}{0-(-r\theta)} = \frac{u_r}{r} \qquad (2.54)$$

เนื่องจากค่าตัวแปรต่างจะไม่ขึ้นอยู่กับค่ามุม θ ดังนั้น $l_k(s)\nu_k(s) = 1$ ทำให้

$$\int_{L_C} l_k(s)\nu_k(s)ds \;=\; 2\pi R_C \;(2.55)$$

ดังนั้นจากสมการ (2.44) (2.48-49ข) (2.53) (2.54) และ (2.55) จะได้

$$J \;=\; \frac{1}{R_C}\int_{A_T}\Big[\big(\sigma W_{\beta\gamma}\;_{\gamma,r} - \quad \delta_{\beta r}\big)q_{r,\beta} + \big(3\kappa\bar{\sigma}\Theta_{,r} - f_\gamma u_{\gamma,r}\big)q_r\Big]rdA$$
$$+ W\frac{1}{R_C}\int_{A_T}\left(\sigma_{\theta\theta}\frac{u_r}{r} - \quad\right)q_r dA \;-\frac{1}{R_C}\int_{C_+ + C_-} t_\gamma u_{\gamma,r}q_r rdC \;(2.56)$$

โดยที่ $\gamma,\ \beta \;=\; r,\ z$ คิดจากแกนพิกัดทรงกระบอกดังแสดงในรูปที่ 2.7

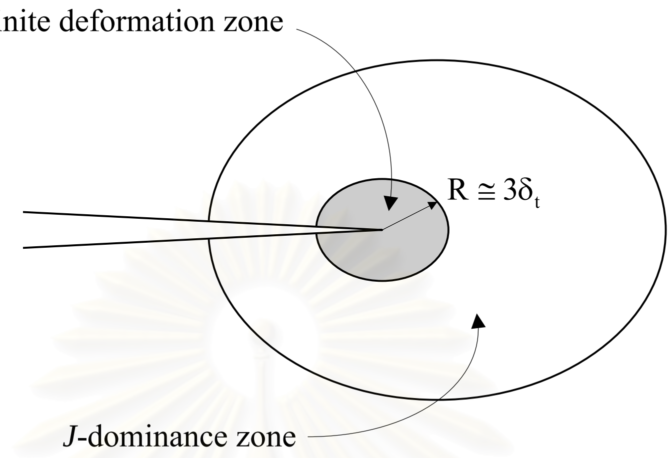
ในกรณีปัญหาสมมาตรรอบแกนจะได้ขนาดของเวกเตอร์ q_k มีค่าเท่ากับหนึ่งที่ตำแหน่งปลายรอยร้าวและมีค่าเท่ากับศูนย์ที่ขอบของพื้นที่ A_T ขณะที่มีค่าอยู่ระหว่างหนึ่งถึงศูนย์บนพื้นที่ A_T โดยเวกเตอร์ q_k เมื่อเขียนอยู่ในพิกัด r-z ดังแสดงในรูปที่ 2.7 แล้วสามารถเขียนแทนได้ดังสมการ

$$q_r \;=\; |\hat{q}| \;(2.57 \qquad ก)$$
$$q_z \;=\; 0 \;(2.57 \qquad ข)$$

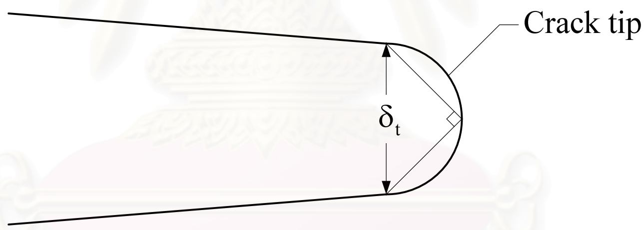
2.3 ข้อจำกัดของพารามิเตอร์เจอินทิกรัล

ข้อจำกัดในการใช้งานของค่าพารามิเตอร์เจอินทิกรัลนั้นจะขึ้นอยู่กับว่าคุณสมบัติของค่าพารามิเตอร์เจอินทิกรัลซึ่งใช้บ่งบอกความรุนแรงของสนามความเค้นที่บริเวณปลายรอยร้าวดังได้อธิบายไว้ในภาคผนวก ค นั้นยังสามารถใช้งานได้เมื่อเปรียบเทียบกับสนามความเค้นที่เกิดขึ้นจริงที่บริเวณปลายรอยร้าว เนื่องจากสนามเอกพันธ์ของ HRR (HRR Singularity) เกิดจากการนำเฉพาะพจน์แรกของผลเฉลยฟังก์ชั่นความเค้นซึ่งเป็นพจน์ที่มีค่าเข้าสู่อนันต์เมื่อระยะตามแนวรัศมีจากปลายรอยร้าว r มีค่าเข้าสู่ศูนย์ ดังนั้นสนามเอกพันธ์ของ HRR จึงไม่สามารถอธิบายสนามความเค้นที่เกิดขึ้นไกลจากปลายรอยร้าวได้ เนื่องจากผลเฉลยสนามเอกพันธ์ของ HRR นั้นได้จากการวิเคราะห์แบบความเครียดน้อย (Small Strain) ขณะที่ความเครียดที่เกิดขึ้นจริงที่ปลายรอยร้าวเป็นแบบจำกัด (Finite Strain) ดังนั้นบริเวณที่สนามเอกพันธ์ของ HRR มีอิทธิพลสูง (*J*-Dominance Zone) จะต้องมีขนาดใหญ่กว่าบริเวณที่ค่าความเครียดจำกัดมีอิทธิพลสูงที่บริเวณปลายรอยร้าว (Finite Def ormation Zone) ดังแสดงในรูปที่ 2.9 ซึ่งมีขนาดประมาณสองถึงสามเท่าของ

ค่าพารามิเตอร์ระยะการเคลื่อนตัวเปิดที่ปลายรอยร้าว δ_t (Crack Tip Opening Displacem ent) ซึ่งนิยามดังแสดงในรูปที่ 2.10



รูปที่ 2.9 บริเวณที่สนามเอกพันธ์ของ HRR มีอิทธิพลสูงและบริเวณที่ค่าความเครียดจำกัดมีอิทธิพลสูงที่บริเวณปลายรอยร้าว



รูปที่ 2.10 นิยามของค่าพารามิเตอร์ระยะการเคลื่อนตัวเปิดที่ปลายรอยร้าว δ_t

ในกรณีที่พลาสติกโซนมีขนาดไม่ใหญ่มากนัก (Contained Plasticity Condition) ผลการคำนวณด้วยระเบียบวิธีไฟไนต์เอลิเมนต์ได้แสดงให้เห็นว่าบริเวณที่สนามเอกพันธ์ของ HRR มีอิทธิพลสูงจะมีขนาดใหญ่กว่าบริเวณที่ค่าความเครียดจำกัดมีอิทธิพลสูงที่บริเวณปลายรอยร้าวค่อนข้างมาก ในกรณีที่พลาสติกโซนมีขนาดใหญ่ (Large Scale Yielding Condition) บริเวณที่สนามเอกพันธ์ของ HRR มีอิทธิพลสูงจะขึ้นกับลักษณะของภาระที่มากระทำกับความกว้างของชิ้นงานที่เหลืออยู่ (Uncracked Ligament) ว่าเป็นภาระความเค้นดึง (Tension) หรือโมเมนต์ดัด (Bending) โดย Shih and German [19] ได้ทำการเปรียบเทียบผลการคำนวณค่าความเค้นที่ระยะต่าง ๆ จากปลายรอยร้าวด้วยระเบียบวิธีไฟไนต์เอลิเมนต์เปรียบเทียบกับผลการคำนวณที่ได้จากสมการ (ค.17) ซึ่งเป็นสมการสนามเอกพันธ์ของ HRR ที่ค่าพารามิเตอร์เจอินทิกรัลเดียวกันสำหรับชิ้นงานภายใต้ภาระความเค้นดึงและโมเมนต์ดัด จากการทดลอง Shih and German พบว่าในกรณี

ที่พลาสติกโซนมีขนาดใหญ่ สำหรับชิ้นงานภายใต้ภาระความเค้นดึง ค่าความกว้างของชิ้นงานที่เหลืออยู่จะต้องมีขนาดมากกว่า $200J/\sigma_o$ โดยที่ σ_o แทนค่าความเค้นที่จุดคราก (Yield Stress) และต้องมีขนาดมากกว่า $25J/\sigma_o$ สำหรับชิ้นงานภายใต้ภาระโมเมนต์ดัด ในกรณีที่บริเวณที่สนามเอกพันธ์ของ HRR มีอิทธิพลสูงมีขนาดเล็กกว่าบริเวณที่ค่าความเครียดจำกัดมีอิทธิพลสูงที่บริเวณปลายรอยร้าวค่าความต้านทานการแตกหัก (Fracture Toughness) ซึ่งได้จากการทดลองหาค่าพารามิเตอร์เจอินทิกรัลที่ทำให้รอยร้าวเติบโต J_{IC} จากชิ้นทดสอบลักษณะต่าง ๆ สำหรับวัสดุชนิดเดียวกันจะมีค่าแตกต่างกันซึ่งแสดงให้เห็นว่าค่า J_{IC} ที่ได้ไม่สามารถนำมาใช้งานในฐานะคุณสมบัติของวัสดุได้อีกต่อไป เพื่อให้ข้อจำกัดในการใช้งานของพารามิเตอร์เจอินทิกรัลลดลงจึงมีนักวิจัยเสนอปรับปรุงการใช้งานใหม่จากแต่เดิมที่ใช้เพียงพารามิเตอร์เจอินทิกรัลเพียงพารามิเตอร์เดียวในการบ่งบอกความรุนแรงของสนามความเค้นที่บริเวณปลายรอยร้าว (One Param eter Characterization of Crack Tip Fields) มาเป็นใช้งานร่วมกับพารามิเตอร์อื่นอีกหนึ่งพารามิเตอร์ (Two Parameter Characterization of Crack Tip Fields) เพื่อให้สามารถบ่งบอกความรุนแรงของสนามความเค้นที่บริเวณปลายรอยร้าวได้ในบริเวณที่มากขึ้น [20, 21, 22, 23] ในกรณีรอยร้าวที่กำลังเติบโต Omori et al. [13] ได้แสดงให้เห็นว่าค่าพารามิเตอร์เจอินทิกรัลที่ได้จากการอินทิเกรตบนเส้นใกล้ปลายรอยร้าวที่กำลังเติบโตจะมีค่าเข้าใกล้ศูนย์ด้วยเหตุนี้ Omori et al. จึงได้นำเสนอพารามิเตอร์ -iT_{ε}^{*} ntegral เพื่อใช้ในการวิเคราะห์ร้อยร้าวที่เติบโตภายใต้สถานะอยู่ตัวโดยผลการคำนวณที่ได้ทั้งจากการทดลองและระเบียบวิธีไฟไนต์เอลิเมนต์สำหรับชิ้นทดสอบมาตรฐานต่าง ๆ แสดงให้เห็นว่าค่าพารามิเตอร์ -iT_{ε}^{*} ntegral ที่ได้มีความสอดคล้องกันและลู่เข้าสู่ค่าเดียวกันสำหรับปัญหาชิ้นทดสอบมาตรฐานต่าง ๆ ซึ่งแสดงให้เห็นว่าพารามิเตอร์ T_{ε}^{*}-integral สามารถนำมาใช้ในการวิเคราะห์ร้อยร้าวที่กำลังเติบโตภายใต้สถานะอยู่ตัวแทนค่าพารามิเตอร์เจอินทิกรัลได้ นอกจากนี้ Newman et al. [14] ยังแสดงให้เห็นว่าพารามิเตอร์มุมเปิดที่ปลายรอยร้าว CTOA สามารถนำไปใช้ในการวิเคราะห์ร้อยร้าวที่กำลังเติบโตภายใต้สถานะอยู่ตัวได้เป็นอย่างดี

2.4 บทสรุป

ในบทนี้สมการค่าพารามิเตอร์เจอินทิกรัลสำหรับปัญหาสามมิติทั้งที่อยู่ในรูปแบบดั้งเดิมซึ่งเกิดจากการอินทิเกรตบนพื้นที่ผิวทรงกระบอกเล็ก ๆ รอบปลายรอยร้าวและในรูปของการอินทิเกรตบนปริมาตรโดเมนได้ถูกประดิษฐ์ขึ้น จากนั้นสมการโดเมนเจอินทิกรัลในแบบสามมิติได้ถูกเปลี่ยนให้อยู่ในรูปสองมิติและสมมาตรรอบแกน นอกจากนั้นข้อจำกัดในการใช้งานพารามิเตอร์เจอินทิกรัลในทางปฏิบัติยังได้ถูกอธิบายในบทนี้ สมการไฟไนต์เอลิเมนต์ที่ใช้ในการวิเคราะห์รอยร้าวในชิ้นงานพร้อมทั้งสมการไฟไนต์เอลิเมนต์สำหรับการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลได้ถูกประดิษฐ์ขึ้นในบทต่อไป

บทที่ 3

การวิเคราะห์ปัญหารอยร้าวด้วยระเบียบวิธีไฟไนต์เอลิเมนต์

ในการวิเคราะห์ปัญหารอยร้าวด้วยระเบียบวิธีไฟไนต์เอลิเมนต์นั้นจะเริ่มจากการวิเคราะห์ความเค้นและการเสียรูปของของแข็งยืดหยุ่นไม่เชิงเส้น (Nonlinear Elastic Solid) ภายใต้ทฤษฎีการเสียรูปแบบพลาสติก (Deformation Theory of Plasticity) ซึ่งรวมผลของค่าความเครียดที่เกิดจากสนามอุณหภูมิเข้าไปด้วย โดยในวิทยานิพนธ์นี้จะทำการวิเคราะห์ทั้งปัญหาความเค้นและความเครียดในระนาบ และปัญหาสมมาตรรอบแกน โดยสมการไฟไนต์เอลิเมนต์จะถูกสร้างจากทฤษฎีงานเสมือน (Principle of Virtual Work) เมื่อทำการวิเคราะห์ปัญหาเสร็จสิ้นแล้วค่าการเคลื่อนตัวที่จุดต่อต่าง ๆ (Nodal Displacements) จะถูกนำมาใช้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลต่อไป

3.1 ความสัมพันธ์ระหว่างค่าความเครียดและค่าความเค้น

ความสัมพันธ์ระหว่างค่าความเครียดและค่าความเค้นในทิศทางเดียวตามแนวแกน (Uniaxial Loading) ของวัสดุที่มีพฤติกรรมแบบ Ramberg-Osgood สามารถแสดงได้ดังสมการ

$$\frac{\varepsilon}{\varepsilon_o} = \frac{\sigma}{\sigma_o} + \alpha\left(\frac{\sigma}{\sigma_o}\right)^n + \frac{\kappa\Theta}{\varepsilon_o} \tag{3.1}$$

$$E = \frac{\sigma_o}{\varepsilon_o} \tag{3.2}$$

โดยที่ σ_o แทนค่าความเค้นที่จุดคราก (Yield Stress)

ε_o แทนค่าความเครียดที่จุดคราก (Yield Strain)

α แทนค่าคงที่ของวัสดุ (A Material Constant or Yield Offset)

n แทนค่ายกกำลังของความเครียด (Strain Hardening Exponent)

κ แทนค่าสัมประสิทธิ์การขยายตัวเนื่องจากสนามอุณหภูมิ (Coefficient of Thermal Expansion)

Θ แทนค่าสนามอุณหภูมิ (Temperature)

E แทนค่าคงที่ของการยืดหยุ่น (Modulus of Elasticity)

ภายใต้ทฤษฎีการเสียรูปแบบพลาสติก ความสัมพันธ์ระหว่างค่าความเค้นและค่าความเครียดที่รวมค่าความเครียดเนื่องจากสนามอุณหภูมิสามารถเขียนอยู่ในรูปสมการเทนเซอร์ (Tensor Notation) ได้เป็น

$$\varepsilon_{ij} = \varepsilon_{ij}^{e} + e_{ij}^{p} + \varepsilon_{ij}^{ther} \tag{3.3}$$

$$\varepsilon_{ij}^{e} = \left(\frac{1-2\nu}{3E}\right)\sigma_{kk}\delta_{ij} + \left(\frac{1+\nu}{E}\right)S_{ij} \tag{3.4}$$

$$e_{ij}^{p} = \frac{3}{2}\frac{\alpha}{E}\left(\frac{\sigma_e}{\sigma_o}\right)^{n-1} S_{ij} \tag{3.5}$$

$$\varepsilon_{ij}^{ther} = \kappa\Theta\delta_{ij} \tag{3.6}$$

$$\sigma_e = \sqrt{\frac{3}{2}S_{ij}S_{ij}} \tag{3.7}$$

โดยที่ ε_{ij} แทนค่าความเครียดสุทธิ (Total Strain)

ε_{ij}^{e} แทนค่าความเครียดยืดหยุ่น (Elastic Strain)

e_{ij}^{p} แทนค่าความเครียดพลาสติก (Plastic Strain)

ε_{ij}^{ther} แทนค่าความเครียดเริ่มต้นเนื่องจากสนามอุณหภูมิ (Thermal Strain)

σ_{ij} แทนค่าความเค้นสุทธิ (Total Stress)

S_{ij} แทนค่าความเค้นดิเวียทอริก (Deviatoric Stress)

δ_{ij} แทนค่าครอเน็คเคอร์เด็ลต้า (Kronecker Delta)

σ_e แทนค่าความเค้นประสิทธิผล (Effective Stress)

ν แทนค่าอัตราส่วนปัวส์ซง (Poisson’s Ratio)

โดยความสัมพันธ์ระหว่างค่าความเค้นสุทธิและความเค้นดิเวียทอริกมีค่าเท่ากับ

$$\sigma_{ij} = \bar{\sigma}\delta_{ij} + S_{ij} \tag{3.8}$$

โดยที่ $\bar{\sigma} = \frac{1}{3}\sigma_{kk}$ แทนค่าความเค้นตั้งฉากเฉลี่ย (Mean Stress)

ค่าความเค้นประสิทธิผลสามารถหาได้จากความสัมพันธ์ในทิศทางเดียวตามแนวแกนของวัสดุแบบ Ramberg-Osgood ดังสมการ [24]

$$\alpha\left(\frac{\sigma_e}{\sigma_o}\right)^n + \frac{2}{3}(1+\nu)\left(\frac{\sigma_e}{\sigma_o}\right) - \left(\frac{e_e}{\varepsilon_o}\right) = 0 \qquad (3.9)$$

$$e_e = \sqrt{\frac{2}{3}e_{ij}e_{ij}} \qquad (3.10)$$

โดยที่ e_e แทนค่าความเครียดประสิทธิผล (Effective Strain)

e_{ij} แทนค่าความเครียดดิเวียทอริก (Deviatoric Strain)

ความสัมพันธ์ระหว่างค่าความเครียดสุทธิและความเครียดดิเวียทอริกมีค่าเท่ากับ

$$\varepsilon_{ij} = \bar{\varepsilon}\delta_{ij} + e_{ij} \qquad (3.11)$$

$$\varepsilon_{ij} = \frac{1}{2}\gamma_{ij}, \quad i \neq j \qquad (3.12)$$

โดยที่ $\bar{\varepsilon} = \frac{1}{3}\varepsilon_{kk}$ แทนค่าความเครียดตั้งฉากเฉลี่ย (Mean Strain)

γ_{ij} แทนค่าความเครียดเฉือนทางวิศวกรรม (Engineering Shear Strain)

จากการแทนสมการ (3.4) (3.5) และ (3.6) ลงในสมการ (3.3) จะได้ความสัมพันธ์ของค่าความเครียดสุทธิเป็น

$$\varepsilon_{ij} = \left(\frac{1-2\nu}{3E}\right)\sigma_{kk}\delta_{ij} + \left(\frac{1+\nu}{E}\right)S_{ij} + \frac{3}{2}\frac{\alpha}{E}\left(\frac{\sigma_e}{\sigma_o}\right)^{n-1}S_{ij} + \kappa\Theta\delta_{ij} \qquad (3.13)$$

ซึ่งสามารถจัดรูปใหม่ให้อยู่ในรูปความเค้นสุทธิได้เป็น

$$\sigma_{ij} = \frac{E}{3(1-2\nu)}(\varepsilon_{kk} - 3\kappa\Theta)\delta_{ij} + \frac{2}{3}\frac{\sigma_e}{e_e}\left(\varepsilon_{ij} - \frac{1}{3}\varepsilon_{kk}\delta_{ij}\right) \qquad (3.14)$$

เทียบสมการ (3.8) กับสมการ (3.14) และแทนสมการ (3.11) สำหรับค่าความเครียดสุทธิจะได้

$$\sigma_{kk} = \frac{E}{(1-2\nu)}(\varepsilon_{kk} - 3\kappa\Theta) \qquad (3.15)$$

$$S_{ij} = \frac{2}{3}\frac{\sigma_e}{e_e}e_{ij} \qquad (3.16)$$

สมการ (3.8) สามารถเขียนใหม่ในรูปผลบวกของเวกเตอร์ความเค้นได้เป็น

$$\{\sigma\} \;=\; \{\bar{\sigma}\} + \{S\} \qquad (3.17)$$

โดยที่ $\{\sigma\}$ แทนเวกเตอร์ความเค้นสุทธิ

$\{\bar{\sigma}\}$ แทนเวกเตอร์ความเค้นตั้งฉากเฉลี่ย

$\{S\}$ แทนเวกเตอร์ความเค้นดิเวียทอริก

โดยลักษณะของเวกเตอร์ต่าง ๆ จะขึ้นกับชนิดของปัญหาว่าเป็นปัญหาความเค้นในระนาบ ความเครียดในระนาบหรือปัญหาสมมาตรรอบแกน

3.1.1 ปัญหาความเค้นในระนาบ (Plane Stress Condition)

สำหรับปัญหาความเค้นในระนาบซึ่ง $\sigma_{33} = \sigma_{31} = \sigma_{23} = \gamma_{31} = \gamma_{23} = 0$ สมการ (3.14) สามารถเขียนอยู่ในรูปเวกเตอร์ความเค้นดังสมการ (3.17) ในรูปความสัมพันธ์ระหว่างเวกเตอร์ความเค้นและความเครียดได้เป็น

$$\{\bar{\sigma}\} \;=\; [E]\{\varepsilon - \varepsilon_o\} \qquad (3.18)$$

$$\{S\} \;=\; [G]\{\varepsilon\} - [H]\{\varepsilon_o\} \qquad (3.19)$$

โดยกำหนดให้

$$\{\sigma\}^T = \lfloor \sigma_{11} \quad \sigma_{22} \quad \sigma_{12} \rfloor \qquad (3.20ก)$$

$$\{\bar{\sigma}\}^T = \lfloor \bar{\sigma} \quad \bar{\sigma} \quad 0 \rfloor \qquad (3.20ข)$$

$$\{S\}^T = \lfloor S_{11} \quad S_{22} \quad S_{12} \rfloor \qquad (3.20ค)$$

$$\{\varepsilon\}^T = \lfloor \varepsilon_{11} \quad \varepsilon_{22} \quad \gamma_{12} \rfloor \qquad (3.20ง)$$

$$\{\varepsilon_o\}^T = \lfloor \kappa\Theta \quad \kappa\Theta \quad 0 \rfloor \qquad (3.20จ)$$

$$[G] \;=\; \frac{\beta E}{(1-2\nu)(1+2\beta)}\begin{bmatrix} 1+\beta & -\beta & 0 \\ -\beta & 1+\beta & 0 \\ 0 & 0 & 0.5+\beta \end{bmatrix} \qquad (3.20ฉ)$$

$$[H] = \frac{\beta E}{(1-2\nu)(1+2\beta)}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.20ช}$$

$$[E] = \frac{\beta E}{(1-2\nu)(1+2\beta)}\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.20ซ}$$

$$\beta = \frac{2}{3}\left(\frac{1-2\nu}{E}\right)\left(\frac{\sigma_e}{e_e}\right) \tag{3.20ฌ}$$

$$\begin{aligned} e_e^2 &= \frac{4}{3}\frac{\left(1+\beta+\beta^2\right)}{(1+2\beta)^2}(\varepsilon_{11}+\varepsilon_{22})^2 - \frac{4}{3}\varepsilon_{11}\varepsilon_{22} \\ &\quad + \frac{1}{3}\gamma_{12}^2 - \frac{4\kappa\Theta}{(1+2\beta)^2}(\varepsilon_{11}+\varepsilon_{22}-\kappa\Theta) \end{aligned} \tag{3.20ญ}$$

โดยที่ $\{\varepsilon\}$ แทนเวกเตอร์ความเครียดสุทธิ

$\{\varepsilon_o\}$ แทนเวกเตอร์ความเครียดเนื่องจากอุณหภูมิ

ในกรณีปัญหาความเค้นในระนาบค่าความเครียดประสิทธิผลเมื่อเขียนอยู่ในรูปของค่าความเครียดบนพิกัดสองมิติ $(\varepsilon_{11}, \varepsilon_{22}, \gamma_{12})$ แล้วจะขึ้นอยู่กับค่าอุณหภูมิด้วย ดังนั้นเมตริกซ์ $[G]$, $[H]$ และ $[E]$ จึงไม่เพียงขึ้นกับค่าความเครียดบนพิกัดสองมิติเท่านั้นแต่ยังขึ้นกับค่าอุณหภูมิด้วย

3.1.2 ปัญหาความเครียดระนาบ (Plane Strain Condition)

สำหรับปัญหาความเครียดในระนาบซึ่ง $\sigma_{31} = \sigma_{23} = \varepsilon_{33} = \gamma_{31} = \gamma_{23} = 0$ สมการ (3.14) สามารถเขียนอยู่ในรูปเวกเตอร์ความเค้นดังสมการ (3.17) ในรูปความสัมพันธ์ระหว่างเวกเตอร์ความเค้นและความเครียดได้เป็น

$$\{\bar{\sigma}\} = [E]\{\varepsilon - \varepsilon_o\} \tag{3.21}$$

$$\{S\} = [G]\{\varepsilon\} \tag{3.22}$$

กำหนดให้

$$\{\sigma\}^T = \lfloor \sigma_{11} \quad \sigma_{22} \quad \sigma_{12} \rfloor \tag{3.23ก}$$

$$\{\bar{\sigma}\}^T = \lfloor \bar{\sigma} \quad \bar{\sigma} \quad 0 \rfloor \tag{3.23ข}$$

$$\{S\}^T = \lfloor S_{11} \quad S_{22} \quad S_{12} \rfloor \tag{3.23ค}$$

$$\{\varepsilon\}^T = \lfloor \varepsilon_{11} \quad \varepsilon_{22} \quad \gamma_{12} \rfloor \tag{3.23ง}$$

$$\{\varepsilon_o\}^T = \lfloor 1.5\kappa\Theta \quad 1.5\kappa\Theta \quad 0 \rfloor \tag{3.23จ}$$

$$[G] = \frac{2}{9}\left(\frac{\sigma_e}{e_e}\right)\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1.5 \end{bmatrix} \tag{3.23ฉ}$$

$$[E] = \frac{E}{3(1-2\nu)}\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.23ช}$$

$$e_e^2 = \frac{4}{9}\left(\varepsilon_{11}^2 + \varepsilon_{22}^2 - \varepsilon_{11}\varepsilon_{22}\right) + \frac{1}{3}\gamma_{12}^2 \tag{3.23ซ}$$

ในกรณีปัญหาความเครียดในระนาบค่าความเครียดประสิทธิผลเมื่อเขียนอยู่ในรูปของค่าความเครียดบนพิกัดสองมิติแล้วจะไม่ขึ้นอยู่กับค่าอุณหภูมิด้วย ดังนั้นเมตริกซ์ $[G]$ และ $[E]$ จึงขึ้นอยู่กับค่าความเครียดบนพิกัดสองมิติเท่านั้นทำให้การสร้างสมการเพื่อใช้ในการวิเคราะห์ทำได้ง่ายกว่าปัญหาความเค้นในระนาบมาก

3.1.3 ปัญหาสมมาตรรอบแกน (Axisymmetric Condition)

สำหรับปัญหาสมมาตรรอบแกนซึ่ง $\sigma_{31} = \sigma_{23} = \gamma_{31} = \gamma_{23} = 0$ สมการ (3.14) สามารถเขียนอยู่ในรูปเวกเตอร์ความเค้นดังสมการ (3.17) ในรูปความสัมพันธ์ระหว่างเวกเตอร์ความเค้นและความเครียดได้เป็น

$$\{\bar{\sigma}\} = [E]\{\varepsilon - \varepsilon_o\} \tag{3.24}$$

$$\{S\} = [G]\{\varepsilon\} \tag{3.25}$$

กำหนดให้

$$\{\sigma\}^T = \lfloor \sigma_{11} \quad \sigma_{22} \quad \sigma_{12} \quad \sigma_{33} \rfloor \tag{3.26ก}$$

$$\{\bar{\sigma}\}^T = \lfloor \bar{\sigma} \quad \bar{\sigma} \quad 0 \quad \bar{\sigma} \rfloor \tag{3.26ข}$$

$$\{S\}^T = \lfloor S_{11} \quad S_{22} \quad S_{12} \quad S_{33} \rfloor \tag{3.26ค}$$

$$\{\varepsilon\}^T = \lfloor \varepsilon_{11} \quad \varepsilon_{22} \quad \gamma_{12} \quad \varepsilon_{33} \rfloor \tag{3.26ง}$$

$$\{\varepsilon_o\}^T = \lfloor \kappa\Theta \quad \kappa\Theta \quad 0 \quad \kappa\Theta \rfloor \tag{3.26จ}$$

$$[G] = \frac{2}{9}\left(\frac{\sigma_e}{e_e}\right)\begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & 0 & -1 \\ 0 & 0 & 1.5 & 0 \\ -1 & -1 & 0 & 2 \end{bmatrix} \tag{3.26ฉ}$$

$$[E] = \frac{E}{3(1-2\nu)}\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \tag{3.26ช}$$

$$e_e^2 = \frac{4}{9}\left(\varepsilon_{11}^2 + \varepsilon_{22}^2 + \varepsilon_{33}^2 - \varepsilon_{11}\varepsilon_{22} - \varepsilon_{22}\varepsilon_{33} - \varepsilon_{33}\varepsilon_{11}\right) + \frac{1}{3}\gamma_{12}^2 \tag{3.26ซ}$$

ในกรณีปัญหาสมมาตรรอบแกนค่าความเครียด ε_{33} ในทิศทางรอบแกนจะมีความสัมพันธ์กับระยะเคลื่อนตัวในแนวรัศมีดังสมการ

$$\varepsilon_{33} = \frac{u}{r} \tag{3.27}$$

โดยที่ u แทนค่าระยะเคลื่อนตัวในทิศทางตามแนวรัศมี

r แทนระยะทางในแนวรัศมีวัดจากแกนสมมาตร

ในกรณีสมมาตรรอบแกนนี้จะคล้ายกับกรณีปัญหาความเครียดในระนาบคือเมตริกซ์ $[G]$ และ $[E]$ จะขึ้นอยู่กับค่าความเครียดบนพิกัดสองมิติเท่านั้นดังนั้นสมการที่ใช้ในการวิเคราะห์จะสร้างได้ง่ายกว่าปัญหาความเค้นในระนาบมาก

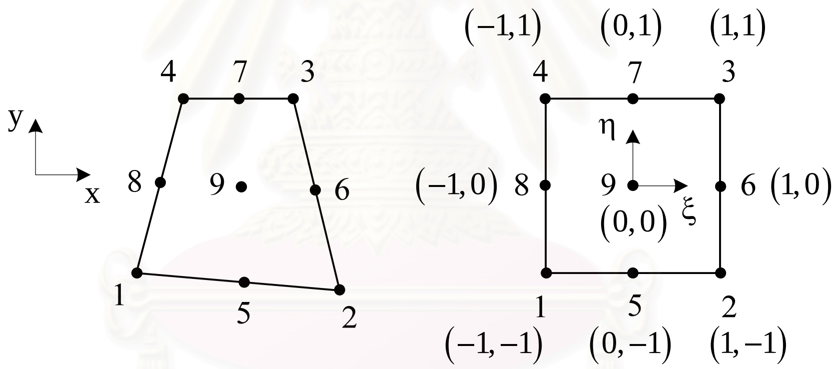
3.2 ฟังก์ชั่นการประมาณภายในเอลิเมนต์

ในวิทยานิพนธ์นี้ใช้เอลิเมนต์สองชนิดในการจำลองปัญหา โดยเอลิเมนต์ที่ปลายรอยร้าว (Crack Tip Element) เป็นไอโซพาราเมตริกซ์เอลิเมนต์ (Isoparametric Element) แบบ

สี่เหลี่ยมเก้าจุดต่อ (Lagrange Family) ที่จุดต่อทั้งสามจุดของด้านหนึ่งถูกยุบมารวมกันที่ปลายรอยร้าวโดยที่จุดต่อทั้งสามยังสามารถเคลื่อนตัวได้เป็นอิสระต่อกันเมื่อเกิดการเสียรูปซึ่งทำให้ผลเฉลยค่าความเครียดที่ได้แปรผันกับระยะตามแนวรัศมีจากปลายรอยร้าว r เป็น $\frac{1}{r}$ ซึ่งสอดคล้องกับผลเฉลยของค่าความเครียดที่บริเวณปลายรอยร้าวสำหรับวัสดุที่มีพฤติกรรมแบบพลาสติกสมบูรณ์ (Perfectly Plastic) [6] ขณะที่เอลิเมนต์ที่บริเวณอื่นจะเป็นไอโซพาราเมตริกซ์เอลิเมนต์แบบสามเหลี่ยมหกจุดต่อ

3.2.1 เอลิเมนต์สี่เหลี่ยมเก้าจุดต่อ (9-node Rectangular Element)

ไอโซพาราเมตริกซ์เอลิเมนต์แบบสี่เหลี่ยมเก้าจุดต่อดังแสดงในรูปที่ 3.1 ประกอบด้วยจุดต่อ (Node) ทั้งหมดเก้าจุด โดยแต่ละจุดจะประกอบไปด้วยตัวแปรไม่ทราบค่าซึ่งในกรณีปัญหาของแข็งที่มีค่าความเครียดเริ่มต้นเนื่องจากอุณหภูมินั้น ตัวแปรไม่ทราบค่าเหล่านี้ได้แก่ ค่าอุณหภูมิและระยะเคลื่อนตัวที่จุดต่อเป็นต้น



รูปที่ 3.1 ไอโซพาราเมตริกซ์เอลิเมนต์แบบสี่เหลี่ยมเก้าจุดต่อซึ่งใช้เป็นเอลิเมนต์ที่ปลายรอยร้าว

ลักษณะการประมาณภายในเอลิเมนต์ชนิดนี้เป็นแบบอันดับสอง (Quadratic Interpolation) ซึ่งความสัมพันธ์ระหว่างพิกัด $x-y$ และพิกัดธรรมชาติ (Natural Coordinates) $\xi-\eta$ สามารถเขียนแทนได้ด้วยสมการ

$$x(\xi,\ \eta) = \lfloor N \rfloor \{x\} \tag{3.28ก}$$

$$y(\xi,\ \eta) = \lfloor N \rfloor \{y\} \tag{3.28ข}$$

$$\lfloor N \rfloor = \lfloor N_1 \quad N_2 \quad N_3 \quad N_4 \quad N_5 \quad N_6 \quad N_7 \quad N_8 \quad N_9 \rfloor \tag{3.29ก}$$

$$\{x\}^T = \lfloor x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \rfloor \tag{3.29ข}$$

$$\{y\}^T = \lfloor y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7 \quad y_8 \rfloor \tag{3.29ค}$$

โดยที่ N_i แทนฟังก์ชันการประมาณภายใน (Interpolation Function) ของจุดต่อ i

x_i แทนค่าพิกัดในแนวแกน x ของจุดต่อ i

y_i แทนค่าพิกัดในแนวแกน y ของจุดต่อ i

โดยฟังก์ชันการประมาณภายในซึ่งเขียนอยู่ในรูปพิกัดธรรมชาติมีค่าเท่ากับ [25]

$$\begin{aligned}
N_1(\xi,\ \eta) &= +\frac{1}{4}\xi(1-\xi)\eta(1-\eta) \\
N_2(\xi,\ \eta) &= -\frac{1}{4}\xi(1+\xi)\eta(1-\eta) \\
N_3(\xi,\ \eta) &= +\frac{1}{4}\xi(1+\xi)\eta(1+\eta) \\
N_4(\xi,\ \eta) &= -\frac{1}{4}\xi(1-\xi)\eta(1+\eta) \\
N_5(\xi,\ \eta) &= -\frac{1}{2}(1-\xi^2)\eta(1-\eta) \\
N_6(\xi,\ \eta) &= +\frac{1}{2}\xi(1+\xi)(1-\eta^2) \\
N_7(\xi,\ \eta) &= +\frac{1}{2}(1-\xi^2)\eta(1+\eta) \\
N_8(\xi,\ \eta) &= -\frac{1}{2}\xi(1-\xi)(1-\eta^2) \\
N_9(\xi,\ \eta) &= (1-\xi^2)(1-\eta^2)
\end{aligned} \tag{3.30}$$

โดยกำหนดให้การกระจายของตัวแปรไม่ทราบค่าต่าง ๆ ภายในเอลิเมนต์มีค่าดังสมการ

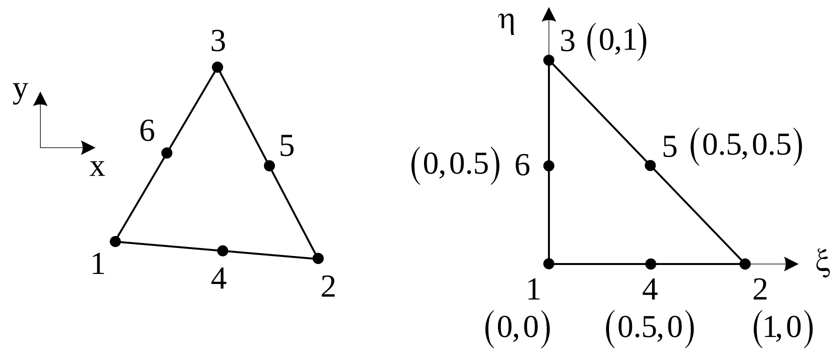
$$\Phi(\xi,\ \eta) = \lfloor N \rfloor \{\Phi\} \tag{3.31}$$

$$\{\Phi\}^T = \lfloor \Phi_1 \quad \Phi_2 \quad \Phi_3 \quad \Phi_4 \quad \Phi_5 \quad \Phi_6 \quad \Phi_7 \quad \Phi_8 \rfloor \tag{3.32}$$

โดยที่ Φ_i แทนค่าของตัวแปรไม่ทราบค่าใด ๆ ที่จุดต่อ i

3.2.2 เอลิเมนต์สามเหลี่ยมหกจุดต่อ (6-node Triangular Element)

สำหรับไอโซพาราเมตริกซ์เอลิเมนต์แบบสามเหลี่ยมหกจุดต่อดังแสดงในรูปที่ 3.2 ลักษณะการประมาณภายในเอลิเมนต์จะเป็นแบบอันดับสอง ความสัมพันธ์ระหว่างพิกัด $x-y$ กับพิกัดธรรมชาติ $\xi-\eta$ และลักษณะการกระจายของตัวแปรไม่ทราบค่าใด ๆ บนเอลิเมนต์สามารถเขียนแทนได้ดังสมการ (3.28ก), (3.28ข) และ (3.31) ตามลำดับโดยเปลี่ยนจำนวนจุดต่อจากเก้าจุดต่อเป็นหกจุดต่อ



รูปที่ 3.2 ไอโซพาราเมตริกซ์เอลิเมนต์แบบสามเหลี่ยมหกจุดต่อ

โดยฟังก์ชันการประมาณภายในซึ่งเขียนอยู่ในรูปพิกัดธรรมชาติมีค่าเท่ากับ [26]

$$
\begin{aligned}
N_1(\xi,\ \eta) &= (1-\xi-\eta)(1-2\xi-2\eta) \\
N_2(\xi,\ \eta) &= \xi(2\xi-1) \\
N_3(\xi,\ \eta) &= \eta(2\eta-1) \\
N_4(\xi,\ \eta) &= 4\xi(1-\xi-\eta) \\
N_5(\xi,\ \eta) &= 4\xi\eta \\
N_6(\xi,\ \eta) &= 4\eta(1-\xi-\eta)
\end{aligned}
\tag{3.33}
$$

3.3 สมการไฟไนต์เอลิเมนต์

จากทฤษฎีงานเสมือน (Principle of Virtual Work) Crisfield [27] ค่างานที่เกิดจากระยะเคลื่อนตัวเสมือนสามารถเขียนอยู่ในรูปผลคูณของเวกเตอร์ต่าง ๆ ได้ดังสมการ

$$\delta V = \int_{\Omega} \{\sigma\}^T \delta\{\varepsilon\} d\Omega - \int_{S} \{\bar{T}\}^T \delta\{\bar{u}\} dS - \int_{\Omega} \{\bar{f}\}^T \delta\{\bar{u}\} d\Omega \tag{3.34}$$

โดยที่ δV แทนค่างานเสมือน (Virtual Work)

$d\Omega$ แทนการอินทิเกรตบนปริมาตร

dS แทนการอินทิเกรตบนพื้นที่ผิว

$\delta\{\bar{u}\}$ แทนเวกเตอร์การเคลื่อนตัวเสมือน (Virtual Displacement Vector)

$\delta\{\varepsilon\}$ แทนเวกเตอร์ความเครียดเสมือน (Virtual Strain Vector)

$\{\bar{T}\}$ แทนเวกเตอร์ความเค้นดึงที่ผิว (Surface Traction Vector)

$\{\bar{f}\}$ แทนเวกเตอร์แรงวัตถุ (Body Force Vector)

โดยเวกเตอร์ต่าง ๆ มีรายละเอียดดังนี้

$$\delta\{\bar{u}\} = \begin{Bmatrix} \delta u \\ \delta v \end{Bmatrix} = \left[\bar{N}\right]\delta\{u\} \tag{3.35ก}$$

$$\{\bar{T}\} = \begin{Bmatrix} T_x \\ T_y \end{Bmatrix} = \left[\bar{N}\right]\{T\} \tag{3.35ข}$$

$$\{\bar{f}\} = \begin{Bmatrix} f_x \\ f_y \end{Bmatrix} = \left[\bar{N}\right]\{f\} \tag{3.35ค}$$

$$\left[\bar{N}\right] = \begin{bmatrix} N_1 & 0 & N_2 & 0 & \dots & N_n & 0 \\ 0 & N_1 & 0 & N_2 & \dots & 0 & N_n \end{bmatrix} \tag{3.36ก}$$

$$\delta\{u\}^T = \lfloor \delta u_1 \quad \delta v_1 \quad \delta u_2 \quad \delta v_2 \quad \cdots \quad \delta u_n \quad \delta v_n \rfloor \tag{3.36ข}$$

$$\{T\}^T = \lfloor T_{x1} \quad T_{y1} \quad T_{x2} \quad T_{y2} \quad \cdots \quad T_{xn} \quad T_{yn} \rfloor \tag{3.36ค}$$

$$\{f\}^T = \lfloor f_{x1} \quad f_{y1} \quad f_{x2} \quad f_{y2} \quad \cdots \quad f_{xn} \quad f_{yn} \rfloor \tag{3.36ง}$$

โดยที่ δu แทนสนามการเคลื่อนตัวเสมือนในแนวแกน x

δv แทนสนามการเคลื่อนตัวเสมือนในแนวแกน y

T_x แทนสนามความเค้นดึงที่ผิวในแนวแกน x

T_y แทนสนามความเค้นดึงที่ผิวในแนวแกน y

f_x แทนสนามแรงวัตถุในแนวแกน x

f_y แทนสนามแรงวัตถุในแนวแกน y

δu_i แทนค่าการเคลื่อนตัวเสมือนในแนวแกน x ที่จุดต่อ i

δv_i แทนค่าการเคลื่อนตัวเสมือนในแนวแกน y ที่จุดต่อ i

T_{xi} แทนค่าความเค้นดึงที่ผิวในแนวแกน x ที่จุดต่อ i

T_{yi} แทนค่าความเค้นดึงที่ผิวในแนวแกน y ที่จุดต่อ i

f_{xi} แทนค่าแรงวัตถุในแนวแกน x ที่จุดต่อ i

f_{yi} แทนค่าแรงวัตถุในแนวแกน x ที่จุดต่อ i

n แทนจำนวนจุดต่อทั้งหมดบนเอลิเมนต์

แทนเวกเตอร์ความเค้นสุทธิด้วยผลบวกของเวกเตอร์ความเค้นตั้งฉากเฉลี่ยและความเค้นดิเวียทอริก จากสมการ (3.17) จะได้สมการงานเสมือนเป็น

$$\delta V = \int_\Omega \{\bar{\sigma}\}^T \delta\{\varepsilon\} d\Omega + \int_\Omega \{S\}^T \delta\{\varepsilon\} d\Omega - \int_S \{\bar{T}\}^T \delta\{\bar{u}\} dS - \int_\Omega \{\bar{f}\}^T \delta\{\bar{u}\} d\Omega \tag{3.37}$$

เวกเตอร์ความเครียดสามารถเขียนอยู่ในรูปความสัมพันธ์ระหว่างเมตริกซ์ความสัมพันธ์ระหว่างค่าความเครียดและระยะเคลื่อนตัว กับเวกเตอร์ค่าการเคลื่อนตัวที่จุดต่อได้ดังสมการ

$$\{\varepsilon\} = [B]\{u\} \tag{3.38}$$

ในกรณีปัญหาความเค้นและความเครียดในระนาบ เมตริกซ์และเวกเตอร์ด้านบนมีรายละเอียดดังนี้

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{Bmatrix} \tag{3.39ก}$$

$$[B] = \begin{bmatrix} \dfrac{\partial N_1}{\partial x} & 0 & \dfrac{\partial N_2}{\partial x} & 0 & \cdots & \dfrac{\partial N_n}{\partial x} & 0 \\ 0 & \dfrac{\partial N_1}{\partial y} & 0 & \dfrac{\partial N_2}{\partial y} & \cdots & 0 & \dfrac{\partial N_n}{\partial y} \\ \dfrac{\partial N_1}{\partial y} & \dfrac{\partial N_1}{\partial x} & \dfrac{\partial N_2}{\partial y} & \dfrac{\partial N_2}{\partial x} & \cdots & \dfrac{\partial N_n}{\partial y} & \dfrac{\partial N_n}{\partial x} \end{bmatrix} \tag{3.39ข}$$

สำหรับปัญหาสมมาตรรอบแกนจะได้ [28]

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \\ \varepsilon_{33} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \\ \dfrac{u}{r} \end{Bmatrix} \tag{3.40ก}$$

$$[B] = \begin{bmatrix} \dfrac{\partial N_1}{\partial x} & 0 & \dfrac{\partial N_2}{\partial x} & 0 & \cdots & \dfrac{\partial N_n}{\partial x} & 0 \\ 0 & \dfrac{\partial N_1}{\partial y} & 0 & \dfrac{\partial N_2}{\partial y} & \cdots & 0 & \dfrac{\partial N_n}{\partial y} \\ \dfrac{\partial N_1}{\partial y} & \dfrac{\partial N_1}{\partial x} & \dfrac{\partial N_2}{\partial y} & \dfrac{\partial N_2}{\partial x} & \cdots & \dfrac{\partial N_n}{\partial y} & \dfrac{\partial N_n}{\partial x} \\ \dfrac{N_1}{r} & 0 & \dfrac{N_2}{r} & 0 & \cdots & \dfrac{N_n}{r} & 0 \end{bmatrix} \tag{3.40ข}$$

$$\{u\}^T = \lfloor u_1 \quad v_1 \quad u_2 \quad v_2 \quad \cdots \quad u_n \quad v_n \rfloor \tag{3.41}$$

โดยที่ $[B]$ แทนเมตริกซ์ความสัมพันธ์ระหว่างค่าความเครียดและระยะเคลื่อนตัว (Strain-Displacement Matrix)

$\{u\}$ แทนเวกเตอร์การเคลื่อนที่จุดต่อ (Nodal Displacement Vector)

u_i แทนค่าการเคลื่อนตัวในแนวแกน x ที่จุดต่อ i

v_i แทนค่าการเคลื่อนตัวในแนวแกน y ที่จุดต่อ i

ในทำนองเดียวกันเวกเตอร์ความเครียดเสมือนสามารถเขียนแทนได้เป็น

$$\delta\{\varepsilon\} = [B]\delta\{u\} \tag{3.42}$$

แทนสมการ (3.35ก) และ (3.42) ลงในสมการ (3.37) จะได้สมการงานเสมือนเป็น

$$\delta V = \int_\Omega \{\bar{\sigma}\}^T [B] d\Omega \delta\{u\} + \int_\Omega \{S\}^T [B] d\Omega \delta\{u\}$$

$$- \int_S \{\bar{T}\}^T [\bar{N}] dS \delta\{u\} - \int_\Omega \{\bar{f}\}^T [\bar{N}] d\Omega \delta\{u\} \tag{3.43}$$

ที่สภาวะสมดุลย์ (Equilibrium State) ค่างานเสมือนจะมีค่าเป็นศูนย์และเนื่องจากค่าการเคลื่อนตัวเสมือนที่จุดต่อเป็นค่าใด ๆ ก็ได้ ดังนั้นสมการไฟไนต์เอลิเมนต์ที่สภาวะสมดุลย์จะมีค่าเท่ากับ

$$\{F^{vol}\} + \{F^{dev}\} - \{F^{trac}\} - \{F^{body}\} = \{0\} \tag{3.44}$$

$$\{F^{vol}\} = \int_\Omega [B]^T \{\bar{\sigma}\} d\Omega \tag{3.45ก}$$

$$\{F^{dev}\} = \int_\Omega [B]^T \{S\} d\Omega \tag{3.45ข}$$

$$\{F^{trac}\} = \int_S [\bar{N}]^T \{\bar{T}\} dS \tag{3.45ค}$$

$$\{F^{body}\} = \int_\Omega [\bar{N}]^T \{\bar{f}\} d\Omega \tag{3.45ง}$$

โดยที่ $\{F^{vol}\}$ แทนโหลดเวกเตอร์เนื่องจากค่าความเค้นตั้งฉากเฉลี่ย

$\{F^{dev}\}$ แทนโหลดเวกเตอร์เนื่องจากค่าความเค้นดิเวียทอริก

$\{F^{trac}\}$ แทนโหลดเวกเตอร์เนื่องจากค่าความเค้นดึงที่ผิว

$\{F^{body}\}$ แทนโหลดเวกเตอร์เนื่องจากค่าแรงวัตถุ

จากสมการ (3.18), (3.21), (3.24) และ (3.38) โหลดเวกเตอร์เนื่องจากค่าความเค้นตั้งฉากเฉลี่ยสามารถเขียนอยู่ในรูปเดียวกันได้ทั้งหมดสำหรับปัญหาความเค้นและความเครียดในระนาบ และปัญหาสมมาตรรอบแกนได้เป็น

$$\left\{F^{vol}\right\} = \int_{\Omega}[B]^{T}[E][B]d\Omega\{u\} - \int_{\Omega}[B]^{T}[E]\{\varepsilon_{o}\}d\Omega \qquad (3.46)$$

สำหรับโหลดเวกเตอร์เนื่องจากค่าความเค้นดิเวียทอริกจะไม่สามารถเขียนอยู่ในรูปที่เหมือนกันได้ทั้งหมด โดยในกรณีปัญหาความเค้นในระนาบจากสมการ (3.19) และ (3.38) จะได้

$$\left\{F^{dev}\right\} = \int_{\Omega}[B]^{T}[G][B]d\Omega\{u\} - \int_{\Omega}[B]^{T}[H]\{\varepsilon_{o}\}d\Omega \qquad (3.47ก)$$

สำหรับปัญหาความเครียดในระนาบและปัญหาสมมาตรรอบแกนจากสมการ (3.22), (3.25) และ (3.38) จะได้

$$\left\{F^{dev}\right\} = \int_{\Omega}[B]^{T}[G][B]d\Omega\{u\} \qquad (3.47ข)$$

แทนสมการ (3.35ข) ลงในสมการ (3.45ค) จะได้โหลดเวกเตอร์เนื่องจากค่าความเค้นดึงที่ผิวเป็น

$$\left\{F^{trac}\right\} = \int_{S}\left[\bar{N}\right]^{T}\left[\bar{N}\right]dS\{T\} \qquad (3.48)$$

ในทำนองเดียวกันแทนสมการ (3.35ค) ลงในสมการ (3.45ง) จะได้โหลดเวกเตอร์เนื่องจากแรงวัตถุมีค่าเท่ากับ

$$\left\{F^{body}\right\} = \int_{\Omega}\left[\bar{N}\right]^{T}\left[\bar{N}\right]d\Omega\{f\} \qquad (3.49)$$

จากสมการ (3.20จ), (3.23จ) และ (3.26จ) ความสัมพันธ์ระหว่างเวกเตอร์ค่าความเครียดเนื่องจากอุณหภูมิและค่าสนามอุณหภูมิสามารถเขียนใหม่ได้เป็น

$$\{\varepsilon_{o}\} = \{\kappa\}\Theta \qquad (3.50)$$

โดยที่ $\{\kappa\}$ แทนเวกเตอร์สัมประสิทธิ์การขยายตัวเนื่องจากความร้อน

กรณีความเค้นในระนาบ

$$\{\kappa\}^{T} = \lfloor \kappa \quad \kappa \quad 0 \rfloor \qquad (3.51ก)$$

กรณีความเครียดในระนาบ

$$\{\kappa\}^{T} = \lfloor 1.5\kappa \quad 1.5\kappa \quad 0 \rfloor \qquad (3.51ข)$$

กรณีสมมาตรรอบแกน

$$\{\kappa\}^{T} = \lfloor \kappa \quad \kappa \quad 0 \quad \kappa \rfloor \qquad (3.51ค)$$

จากสมการ (3.31) และ (3.32) เราสามารถสร้างสมการสนามอุณหภูมิได้เป็น

$$\Theta = \lfloor N \rfloor \{\Theta\} \qquad (3.52)$$

$$\{\Theta\}^{T} = \lfloor \Theta_1 \quad \Theta_2 \quad \cdots \quad \Theta_n \rfloor \qquad (3.53)$$

โดยที่ $\{\Theta\}$ แทนเวกเตอร์อุณหภูมิที่จุดต่อ

Θ_i แทนค่าอุณหภูมิที่จุดต่อ i

แทนสมการ (3.52) ลงในสมการ (3.50) จะได้สมการความสัมพันธ์ระหว่างเวกเตอร์ค่าความเครียดเนื่องจากอุณหภูมิและค่าอุณหภูมิที่จุดต่อเป็น

$$\{\varepsilon_o\} = \{\kappa\}\lfloor N \rfloor \{\Theta\} \qquad (3.54)$$

3.4 วิธีการทำซ้ำ

เนื่องจากสมการไฟไนต์เอลิเมนต์ที่สภาวะสมดุลย์ (3.44) อยู่ในรูปของระบบสมการไม่เชิงเส้น (Nonlinear System of Equations) ดังนั้นเราจึงต้องประยุกต์ใช้วิธีการทำซ้ำเพื่อหาผลเฉลยที่ทำให้ระบบสมการอยู่ในสภาวะสมดุลย์ วิธีการทำซ้ำที่ใช้ในวิทยานิพนธ์นี้คือวิธีการทำซ้ำแบบนิวตัน-ราฟสัน (Newton-Raphson Iteration Scheme) [27] ซึ่งสร้างขึ้นจากการกระจายอนุกรมเทย์เลอร์ (Taylor’s Series) พิจารณาสมการสภาวะสมดุลย์ (3.44) ซึ่งเขียนใหม่ในรูป

$$\{g\} = \{F^{vol}\} + \{F^{dev}\} - \{F^{trac}\} - \{F^{body}\} = \{0\} \qquad (3.55)$$

โดยที่ $\{g\}$ แทนเวกเตอร์ความไม่สมดุลของแรง (Out of Balance Force Vector)

ตัวแปรอิสระทั้งหมดของสมการด้านบนประกอบไปด้วย ค่าเคลื่อนตัว อุณหภูมิ ความเค้นดึงที่ผิวและแรงวัตถุที่จุดต่อ โดยหากค่าตัวแปรอิสระทั้งหมดเป็นผลเฉลยที่ถูกต้องแล้วเวกเตอร์ความไม่

สมดุลย์ของแรงจะมีค่าเป็นศูนย์ ประยุกต์อนุกรมเทย์เลอร์อันดับที่หนึ่ง (1st Order Taylor's Series) เข้ากับสมการ (3.55) จะได้สมการใหม่ซึ่งเขียนอยู่ในรูปเทนเซอร์เป็น

$$\Delta g_i = \frac{\partial g_i}{\partial u_m}\Delta u_m + \frac{\partial g_i}{\partial \Theta_n}\Delta\Theta_n - \frac{\partial g_i}{\partial T_p}\Delta T_p - \frac{\partial g_i}{\partial f_q}\Delta f_q \qquad (3.56)$$

โดยที่ Δ แทนผลต่างของตัวแปรอิสระระหว่างค่าใหม่และเก่า

เนื่องจากค่าความเค้นประสิทธิผลมีความสัมพันธ์โดยตรงกับค่าความเครียดประสิทธิผลดังแสดงในสมการ (3.9) ดังนั้นจากสมการ (3.20ญ), (3.23ซ) และ (3.26ซ) จะได้อัตราส่วนของค่าความเค้นประสิทธิผลต่อค่าความเครียดประสิทธิผลมีค่าขึ้นอยู่กับค่าการเคลื่อนตัวที่จุดต่อด้วย นอกจากนั้นในกรณีปัญหาความเค้นในระนาบค่าอัตราส่วนนี้ยังขึ้นอยู่กับค่าอุณหภูมิที่จุดต่ออีกด้วย ดังนั้นสมการ (3.56) สามารถกระจายออกได้เป็น

$$\Delta g_i = \frac{\partial F_i^{vol}}{\partial u_m}\Delta u_m + \frac{\partial F_i^{dev}}{\partial u_m}\Delta u_m + \frac{\partial F_i^{vol}}{\partial \Theta_n}\Delta\Theta_n$$

$$+ \frac{\partial F_i^{dev}}{\partial \Theta_n}\Delta\Theta_n - \frac{\partial F_i^{trac}}{\partial T_p}\Delta T_p - \frac{\partial F_i^{body}}{\partial f_q}\Delta f_q \qquad (3.57)$$

จากสมการ (3.35ข), (3.35ค) และ (3.36ก) เนื่องจากความสัมพันธ์ระหว่างโหลดเวกเตอร์เนื่องจากความเค้นดึงที่ผิวและค่าความเค้นดึงที่ผิวที่จุดต่อต่าง ๆ เป็นแบบเชิงเส้น เช่นเดียวกับความสัมพันธ์ระหว่างโหลดเวกเตอร์เนื่องจากแรงวัตถุและค่าแรงวัตถุที่จุดต่อต่าง ๆ ดังนั้นสมการ (3.57) สามารถเขียนใหม่ได้เป็น

$$\Delta g_i = \frac{\partial F_i^{vol}}{\partial u_m}\Delta u_m + \frac{\partial F_i^{dev}}{\partial u_m}\Delta u_m + \frac{\partial F_i^{vol}}{\partial \Theta_n}\Delta\Theta_n$$

$$+ \frac{\partial F_i^{dev}}{\partial \Theta_n}\Delta\Theta_n - \Delta F_i^{trac} - \Delta F_i^{body} \qquad (3.58ก)$$

ซึ่งสามารถเขียนอยู่ในรูปเมตริกซ์ได้เป็น

$$\{\Delta g\} = \left[\left[\frac{\partial F^{vol}}{\partial u}\right] + \left[\frac{\partial F^{dev}}{\partial u}\right]\right]\{\Delta u\} + \left[\left[\frac{\partial F^{vol}}{\partial \Theta}\right] + \left[\frac{\partial F^{dev}}{\partial \Theta}\right]\right]\{\Delta\Theta\}$$

$$- \{\Delta F^{trac}\} - \{\Delta F^{body}\} \qquad (3.58ข)$$

โดยที่ $\left[\dfrac{\partial F^{vol}}{\partial u}\right]$ แทนเมตริกซ์ความแข็งเกร็งสัมผัสเนื่องจากค่าความเค้นตั้งฉากเฉลี่ย

$\left[\dfrac{\partial F^{dev}}{\partial u}\right]$ แทนเมตริกซ์ความแข็งเกร็งสัมผัสเนื่องจากค่าความเค้นดิเวียทอริก

ในการหาผลเฉลยของสมการ (3.58) นั้นจะประกอบด้วยขั้นตอนสองขั้นตอน โดยในขั้นตอนแรกจะทำการหาผลเฉลยโดยประมาณของค่าตัวแปรอิสระจากการแทนค่าเวกเตอร์ $\{\Delta g\}=\{0\}$ ลงในสมการ (3.58ข) ซึ่งได้มาจากความเข้าใจที่ว่าหากผลต่างของค่าตัวแปรอิสระมีค่าน้อยมาก ๆ แล้วผลเฉลยของค่าตัวแปรอิสระที่ได้จากสมการอนุกรมเทย์เลอร์อันดับที่หนึ่งจะเป็นผลเฉลยที่ใกล้เคียงกับผลเฉลยที่แท้จริง โดยเรียกขั้นตอนการหาผลเฉลยในช่วงนี้ว่า Incremental Solution Scheme ดังนั้นสมการ (3.58ข) สามารถเขียนได้ใหม่ในขั้นตอนนี้เป็น

$$\left[\left[\frac{\partial F^{vol}}{\partial u}\right]+\left[\frac{\partial F^{dev}}{\partial u}\right]\right]_o\{\Delta u\} = \left\{\Delta F^{trac}\right\} + \left\{\Delta F^{body}\right\} - \left[\left[\frac{\partial F^{vol}}{\partial \Theta}\right]+\left[\frac{\partial F^{dev}}{\partial \Theta}\right]\right]_o\{\Delta\Theta\} \tag{3.59}$$

โดยสัญลักษณ์ $[\ ]_o$ หมายถึงเมตริกซ์ถูกคำนวณโดยใช้ค่าตัวแปรอิสระที่จุดต่อต่าง ๆ จากสถานะการวิเคราะห์ (Analysis State) ก่อนหน้านี้ที่มีผลเฉลยที่อยู่ในสภาวะสมดุลแล้ว (Converged Solution) ในขั้นตอนต่อไปจะใช้วิธีการทำซ้ำเพื่อปรับปรุงผลเฉลยที่ได้ในขั้นตอนแรกให้เข้าสู่สภาวะสมดุลด้วยการทำให้สมการสภาวะสมดุลย์ (3.55) มีค่าเข้าใกล้ศูนย์ให้มากที่สุด โดยเรียกขั้นตอนการหาผลเฉลยในช่วงนี้ว่า Iterative Solution Scheme จัดรูปสมการ (3.58ข) ให้อยู่ในรูปใหม่เป็น

$$\{g\}_{i+1} = \{g\}_i + \left[\left[\frac{\partial F^{vol}}{\partial u}\right]+\left[\frac{\partial F^{dev}}{\partial u}\right]\right]_i\{\Delta u\} + \left[\left[\frac{\partial F^{vol}}{\partial \Theta}\right]+\left[\frac{\partial F^{dev}}{\partial \Theta}\right]\right]_i\{\Delta\Theta\} - \left\{F^{trac}\right\}_{i+1} + \left\{F^{trac}\right\}_i - \left\{F^{body}\right\}_{i+1} + \left\{F^{body}\right\}_i \tag{3.60}$$

โดยที่ i แทนการคำนวณที่การทำซ้ำครั้งปัจจุบัน

$i+1$ แทนการคำนวณที่การทำซ้ำครั้งถัดไป

เนื่องจากในปัญหาในวิทยานิพนธ์นี้ ค่าอุณหภูมิ ค่าความเค้นดึงที่ผิวและค่าแรงวัตถุที่จุดต่อต่าง ๆ เป็นตัวแปรอิสระที่รู้ค่าที่สถานะการวิเคราะห์ต่าง ๆ ทั้งหมด ดังนั้นในช่วงการทำซ้ำนี้สมการ (3.60) จะลดรูปลงเหลือ

$$\{g\}_{i+1} \;=\; \{g\}_i \;+\; \left[\left[\frac{\partial F^{vol}}{\partial u}\right]+\left[\frac{\partial F^{dev}}{\partial u}\right]\right]_i \{\Delta u\} \qquad (3.60)$$

เพื่อที่จะหาผลเฉลยค่าการเคลื่อนตัวที่จุดต่อที่มีความถูกต้องมากขึ้นซึ่งทำให้สมการสภาวะสมดุล (3.55) มีค่าเข้าใกล้ศูนย์ ดังนั้นเราจึงกำหนดให้เวกเตอร์ความไม่สมดุลย์ของแรงที่การทำซ้ำครั้งถัดไปมีค่าเป็นศูนย์ $\{g\}_{i+1} = \{0\}$ ลงในสมการ (3.60) แล้วแทนสมการสภาวะสมดุลย์ (3.55) ที่คำนวณในการทำซ้ำครั้งปัจจุบันจะได้

$$\left[\left[\frac{\partial F^{vol}}{\partial u}\right]+\left[\frac{\partial F^{dev}}{\partial u}\right]\right]_i \{\Delta u\} \;=\; -\{g\}_i \qquad (3.61)$$

โดยที่

$$-\{g\}_i \;=\; \left\{F^{trac}\right\}_i \;+\; \left\{F^{body}\right\}_i \;-\; \left\{F^{vol}\right\}_i \;-\; \left\{F^{dev}\right\}_i \qquad (3.62)$$

สมการ (3.61) และ (3.62) เป็นสมการที่ใช้ในช่วงการทำซ้ำนี้ โดยรายละเอียดของเวกเตอร์และเมตริกซ์ต่าง ๆ ที่ใช้ในขั้นตอนการคำนวณหาผลเฉลยทั้งหมดจะขึ้นอยู่กับชนิดของปัญหาว่าเป็นปัญหาชนิดใดดังต่อไปนี้

3.4.1 ปัญหาความเค้นในระนาบ

เนื่องจากค่าความเครียดประสิทธิผล e_e เมื่อเขียนอยู่ในรูปของค่าความเครียดในสองมิติแล้วจะขึ้นอยู่กับค่าสนามอุณหภูมิด้วยดังแสดงในสมการ (3.20ญ) ดังนั้นเมตริกซ์ต่าง ๆ ที่ใช้ในขั้นตอนการหาผลเฉลยจึงมีความซับซ้อนมากกว่ากรณีปัญหาความเครียดในระนาบและปัญหาสมมาตรรอบแกนมาก โดยค่าอนุพันธ์ของโหลดเวกเตอร์ต่าง ๆ เทียบกับตัวแปรอิสระจะสามารถหาได้โดยง่ายเมื่อเขียนอยู่ในรูปเทนเซอร์ ดังนั้นจากสมการ (3.46) และ (3.54) ค่าโหลดเวกเตอร์เนื่องจากค่าความเค้นตั้งฉากเฉลี่ยจะมีค่าเท่ากับ

$$F_i^{vol} \;=\; \int_\Omega B_{ji} E_{jk} B_{kl} d\Omega u_l \;-\; \int_\Omega B_{ji} E_{jk} \kappa_k N_m d\Omega \Theta_m \qquad (3.63ก)$$

หรือ

$$F_i^{vol} \;=\; \int_\Omega B_{ji} E_{jk} \left(\varepsilon_k - \varepsilon_{ok}\right) d\Omega \qquad (3.63ข)$$

ดังนั้นค่าอนุพันธ์ของโหลดเวกเตอร์นี้เทียบกับค่าเคลื่อนตัวที่จุดต่อจะมีค่าเท่ากับ

$$
\begin{aligned}
\frac{\partial F_i^{vol}}{\partial u_n} &= \int_\Omega B_{ji} E_{jk} B_{kl} d\Omega \frac{\partial u_l}{\partial u_n} + \int_\Omega B_{ji} \frac{\partial E_{jk}}{\partial u_n} B_{kl} d\Omega u_l - \int_\Omega B_{ji} \frac{\partial E_{jk}}{\partial u_n} \kappa_k N_m d\Omega \Theta_m \\
&= \int_\Omega B_{ji} E_{jk} B_{kl} d\Omega \delta_{ln} + \int_\Omega B_{ji} \frac{\partial E_{jk}}{\partial \varepsilon_p} \frac{\partial \varepsilon_p}{\partial u_n} \left(B_{kl} u_l - \kappa_k N_m \Theta_m \right) d\Omega \\
&= \int_\Omega B_{ji} E_{jk} B_{kn} d\Omega + \int_\Omega B_{ji} \frac{\partial E_{jk}}{\partial \varepsilon_p} B_{pn} \left(B_{kl} u_l - \kappa_k N_m \Theta_m \right) d\Omega \\
&= \int_\Omega B_{ji} E_{jk} B_{kn} d\Omega + \int_\Omega B_{ji} \frac{\partial E_{jp}}{\partial \varepsilon_k} \left(B_{pl} u_l - \kappa_p N_m \Theta_m \right) B_{kn} d\Omega \\
&= \int_\Omega B_{ji} \left[E_{jk} + \frac{\partial E_{jp}}{\partial \varepsilon_k} \left(B_{pl} u_l - \kappa_p N_m \Theta_m \right) \right] B_{kn} d\Omega \\
&= \int_\Omega B_{ji} \left[E_{jk} + \frac{\partial E_{jp}}{\partial \varepsilon_k} \left(\varepsilon_p - \kappa_p \Theta \right) \right] B_{kn} d\Omega \qquad (3.64)
\end{aligned}
$$

ในทำนองเดียวกันค่าอนุพันธ์ของโหลดเวกเตอร์นี้เมื่อเทียบกับค่าอุณหภูมิที่จุดต่อจะมีค่าเท่ากับ

$$
\begin{aligned}
\frac{\partial F_i^{vol}}{\partial \Theta_q} &= \int_\Omega B_{ji} \frac{\partial E_{jk}}{\partial \Theta_q} \left(B_{kl} u_l - \kappa_k N_m \Theta_m \right) d\Omega - \int_\Omega B_{ji} E_{jk} \kappa_k N_m d\Omega \frac{\partial \Theta_m}{\partial \Theta_q} \\
&= \int_\Omega B_{ji} \frac{\partial E_{jk}}{\partial \Theta} \frac{\partial \Theta}{\partial \Theta_q} \left(B_{kl} u_l - \kappa_k N_m \Theta_m \right) d\Omega - \int_\Omega B_{ji} E_{jk} \kappa_k N_m d\Omega \delta_{mq} \\
&= \int_\Omega B_{ji} \frac{\partial E_{jk}}{\partial \Theta} N_q \left(B_{kl} u_l - \kappa_k N_m \Theta_m \right) d\Omega - \int_\Omega B_{ji} E_{jk} \kappa_k N_q d\Omega \\
&= \int_\Omega B_{ji} \left[\frac{\partial E_{jk}}{\partial \Theta} \left(B_{kl} u_l - \kappa_k N_m \Theta_m \right) - E_{jk} \kappa_k \right] N_q d\Omega \\
&= \int_\Omega B_{ji} \left[\frac{\partial E_{jk}}{\partial \Theta} \left(\varepsilon_k - \kappa_k \Theta \right) - E_{jk} \kappa_k \right] N_q d\Omega \qquad (3.65)
\end{aligned}
$$

ค่าโหลดเวกเตอร์เนื่องจากค่าความเค้นดิเวียทอริกสามารถหาได้จากสมการ (3.47ก) และ (3.54) มีค่าเท่ากับ

$$
F_i^{dev} = \int_\Omega B_{ji} G_{jk} B_{kl} d\Omega u_l - \int_\Omega B_{ji} H_{jk} \kappa_k N_m d\Omega \Theta_m \qquad (3.66ก)
$$

หรือ

$$
F_i^{dev} = \int_\Omega B_{ji} \left(G_{jk} \varepsilon_k - H_{jk} \varepsilon_{ok} \right) d\Omega \qquad (3.66ข)
$$

ดังนั้นค่าอนุพันธ์ของโหลดเวกเตอร์นี้เมื่อเทียบกับค่าเคลื่อนตัวที่จุดต่อจะมีค่าเท่ากับ

$$\frac{\partial F_i^{dev}}{\partial u_n} = \int_\Omega B_{ji}\left[G_{jk} + \frac{\partial G_{jp}}{\partial \varepsilon_k}\varepsilon_p - \frac{\partial H_{jp}}{\partial \varepsilon_k}\kappa_p\Theta\right]B_{kn}d\Omega \tag{3.67}$$

ในทำนองเดียวกันเมื่อเทียบกับค่าอุณหภูมิที่จุดต่อจะได้

$$\frac{\partial F_i^{dev}}{\partial \Theta_q} = \int_\Omega B_{ji}\left[\frac{\partial G_{jk}}{\partial \Theta}\varepsilon_k - \frac{\partial H_{jk}}{\partial \Theta}\kappa_k\Theta - H_{jk}\kappa_k\right]N_q d\Omega \tag{3.68}$$

จากสมการ (3.48) ค่าโหลดเวกเตอร์เนื่องจากความเค้นดึงที่ผิวสามารถเขียนให้อยู่ในรูปเทนเซอร์ได้เป็น

$$F_i^{trac} = \int_S \bar{N}_{ji}\bar{N}_{jk}dST_k \tag{3.69}$$

ดังนั้นค่าอนุพันธ์ของโหลดเวกเตอร์นี้เทียบกับค่าความเค้นดึงที่จุดต่อจะมีค่าเท่ากับ

$$\frac{\partial F_i^{trac}}{\partial T_m} = \int_S \bar{N}_{ji}\bar{N}_{jk}dS\frac{\partial T_k}{\partial T_m} = \int_S \bar{N}_{ji}\bar{N}_{jk}dS\delta_{km} = \int_S \bar{N}_{ji}\bar{N}_{im}dS \tag{3.70}$$

ในทำนองเดียวกันจากสมการ (3.49) ค่าโหลดเวกเตอร์เนื่องจากค่าแรงวัตถุจะมีค่าเท่ากับ

$$F_i^{trac} = \int_S \bar{N}_{ji}\bar{N}_{jk}dSf_k \tag{3.71}$$

ดังนั้นค่าอนุพันธ์ของโหลดเวกเตอร์นี้เทียบกับค่าแรงวัตถุที่จุดต่อจะมีค่าเท่ากับ

$$\frac{\partial F_i^{trac}}{\partial f_m} = \int_\Omega \bar{N}_{ji}\bar{N}_{jk}d\Omega\frac{\partial f_k}{\partial f_m} = \int_\Omega \bar{N}_{ji}\bar{N}_{jk}d\Omega\delta_{km} = \int_\Omega \bar{N}_{ji}\bar{N}_{im}d\Omega \tag{3.72}$$

จากสมการ (3.64), (3.65), (3.67) และ (3.68) ค่าเทนเซอร์ที่ต้องทำการกระจายให้อยู่ในรูปที่ง่ายต่อการคำนวณจะประกอบไปด้วยพจน์ $\frac{\partial E_{jp}}{\partial \varepsilon_k}(\varepsilon_p - \kappa_p\Theta)$, $\frac{\partial E_{jk}}{\partial \Theta}(\varepsilon_k - \kappa_k\Theta)$, $\frac{\partial G_{jp}}{\partial \varepsilon_k}\varepsilon_p$, $\frac{\partial H_{jp}}{\partial \varepsilon_k}\kappa_p\Theta$, $\frac{\partial G_{jk}}{\partial \Theta}\varepsilon_k$ และ $\frac{\partial H_{jk}}{\partial \Theta}\kappa_k\Theta$ เนื่องจากเมตริกซ์ $[G]$, $[H]$ และ $[E]$ ขึ้นอยู่กับค่า β ซึ่งมีค่าดังแสดงในสมการ (3.20ฌ) ดังนั้นค่าอนุพันธ์ของเมตริกซ์ต่าง ๆ เหล่านี้สามารถเขียนใหม่ได้เท่ากับ

$$\frac{\partial E_{jp}}{\partial \varepsilon_k}(\varepsilon_p - \kappa_p\Theta) = \frac{\partial E_{jp}}{\partial \beta}(\varepsilon_p - \kappa_p\Theta)\frac{\partial \beta}{\partial \varepsilon_k} \tag{3.73ก}$$

$$\frac{\partial E_{jk}}{\partial \Theta}\left(\varepsilon_k - \kappa_k\Theta\right) = \frac{\partial E_{jk}}{\partial \beta}\left(\varepsilon_k - \kappa_k\Theta\right)\frac{\partial \beta}{\partial \Theta} \tag{3.73ข}$$

$$\frac{\partial G_{jp}}{\partial \varepsilon_k}\varepsilon_p = \frac{\partial G_{jp}}{\partial \beta}\varepsilon_p \frac{\partial \beta}{\partial \varepsilon_k} \tag{3.73ค}$$

$$\frac{\partial H_{jp}}{\partial \varepsilon_k}\kappa_p\Theta = \frac{\partial H_{jp}}{\partial \beta}\kappa_p\Theta \frac{\partial \beta}{\partial \varepsilon_k} \tag{3.73ง}$$

$$\frac{\partial G_{jk}}{\partial \Theta}\varepsilon_k = \frac{\partial G_{jk}}{\partial \beta}\varepsilon_k \frac{\partial \beta}{\partial \Theta} \tag{3.73จ}$$

$$\frac{\partial H_{jk}}{\partial \Theta}\kappa_k\Theta = \frac{\partial H_{jk}}{\partial \beta}\kappa_k\Theta \frac{\partial \beta}{\partial \Theta} \tag{3.73ฉ}$$

จากสมการ (3.20ง-ซ) และ (3.50-3.51ก) จะได้

$$\frac{\partial E_{jp}}{\partial \beta}\left(\varepsilon_p - \kappa_p\Theta\right) = \frac{2E}{(1-2\nu)(1+2\beta)^2}\begin{Bmatrix} 0.5\varepsilon_{11} + 0.5\varepsilon_{22} - \kappa\Theta \\ 0.5\varepsilon_{11} + 0.5\varepsilon_{22} - \kappa\Theta \\ 0 \end{Bmatrix} \tag{3.74ก}$$

$$\frac{\partial G_{jp}}{\partial \beta}\varepsilon_p = \frac{2E}{(1-2\nu)(1+2\beta)^2}\begin{Bmatrix} \left(0.5+\beta+\beta^2\right)\varepsilon_{11} - \beta(1+\beta)\varepsilon_{22} \\ -\beta(1+\beta)\varepsilon_{11} + \left(0.5+\beta+\beta^2\right)\varepsilon_{22} \\ \left(0.25+\beta+\beta^2\right)\gamma_{22} \end{Bmatrix} \tag{3.74ข}$$

$$\frac{\partial H_{jk}}{\partial \beta}\kappa_k\Theta = \frac{2E}{(1-2\nu)(1+2\beta)^2}\begin{Bmatrix} 0.5\kappa\Theta \\ 0.5\kappa\Theta \\ 0 \end{Bmatrix} \tag{3.74ค}$$

ดังนั้นจะเหลือพจน์ที่ต้องกระจายให้อยู่ในรูปที่ง่ายขึ้นต่อไปคือ $\frac{\partial \beta}{\partial \varepsilon_k}$ และ $\frac{\partial \beta}{\partial \Theta}$ จากสมการ (3.9) และ (3.20ฌ) เราพบว่า $e_e = f(\beta)$ ดังนั้นจะได้

$$\frac{\partial\left(e_e^2\right)}{\partial \varepsilon_k} = \frac{d\left(e_e^2\right)}{d\beta}\frac{\partial \beta}{\partial \varepsilon_k} = 2e_e\frac{de_e}{d\beta}\frac{\partial \beta}{\partial \varepsilon_k} \tag{3.75}$$

จากสมการ (3.20ฌ) และ (3.9) จะได้

$$\frac{d\beta}{de_e} = \frac{\partial\beta}{\partial\sigma_e}\frac{d\sigma_e}{de_e} + \frac{\partial\beta}{\partial e_e} \tag{3.76}$$

ค่าอนุพันธ์ $\frac{\partial\beta}{\partial\sigma_e}$ และ $\frac{\partial\beta}{\partial e_e}$ สามารถหาได้โดยตรงจากสมการ (3.20ฌ) โดยเมื่อแทนกลับลงไปในสมการด้านบนจะได้

$$\frac{d\beta}{de_e} = \frac{2}{3}\frac{(1-2\nu)}{E}\frac{\sigma_e}{e_e^2}\left[\frac{e_e}{\sigma_e}\frac{d\sigma_e}{de_e} - 1\right] \tag{3.77}$$

ค่าอนุพันธ์ $\frac{d\sigma_e}{de_e}$ สามารถหาได้โดยตรงจากสมการ (3.9) เป็น

$$\frac{d\sigma_e}{de_e} = \frac{\sigma_o/\varepsilon_o}{n\frac{e_e/\varepsilon_o}{\sigma_e/\sigma_o} - \frac{2}{3}(1+\nu)(n-1)} \tag{3.78}$$

หากเรากำหนดให้

$$\bar{A} = \left[1 - \frac{e_e}{\sigma_e}\frac{d\sigma_e}{de_e}\right] = \frac{\frac{2}{3}(1+\nu)\frac{\sigma_e/\sigma_o}{e_e/\varepsilon_o} - 1}{\frac{2}{3}(1+\nu)\frac{\sigma_e/\sigma_o}{e_e/\varepsilon_o} - \frac{n}{n-1}} \tag{3.79}$$

จากสมการ (3.79) และ (3.20ฌ) ดังนั้นสมการ (3.77) สามารถเขียนได้ใหม่เป็น

$$\frac{d\beta}{de_e} = -\frac{\beta}{e_e}\bar{A} \tag{3.80}$$

แทนสมการ (3.80) ลงในสมการ (3.75) จะได้

$$\frac{\partial(e_e^2)}{\partial\varepsilon_k} = 2e_e\frac{1}{\left(\frac{d\beta}{de_e}\right)}\frac{\partial\beta}{\partial\varepsilon_k} = -2\frac{e_e^2}{\beta\bar{A}}\frac{\partial\beta}{\partial\varepsilon_k} \tag{3.81}$$

ในขณะเดียวกันค่าอนุพันธ์ $\frac{\partial(e_e^2)}{\partial\varepsilon_k}$ เองก็สามารถหาได้จากการพิจารณาสมการ (3.20ญ) เป็น

$$\frac{\partial(e_e^2)}{\partial\varepsilon_k} = \frac{\partial(e_e^2)}{\partial\beta}\frac{\partial\beta}{\partial\varepsilon_k} + \left.\frac{\partial(e_e^2)}{\partial\varepsilon_k}\right|_{\beta} \tag{3.82}$$

จากสมการ (3.20ญ) จะได้ค่าอนุพันธ์

$$\frac{\partial\left(e_e^2\right)}{\partial\beta} = -4\left[\frac{\left(\varepsilon_{11}+\varepsilon_{22}\right)^2-4\kappa\Theta\left(\varepsilon_{11}+\varepsilon_{22}-\kappa\Theta\right)}{\left(1+2\beta\right)^3}\right] \tag{3.83ก}$$

$$\left.\frac{\partial\left(e_e^2\right)}{\partial\varepsilon_k}\right|_{\beta} = \frac{8}{3}\frac{1}{\left(1+2\beta\right)^2}\{C\}^T \tag{3.83ข}$$

โดยที่

$$\{C\} = \begin{Bmatrix} \left(1+\beta+\beta^2\right)\varepsilon_{11}+\left(0.5-\beta-\beta^2\right)\varepsilon_{22}-1.5\kappa\Theta \\ \left(0.5-\beta-\beta^2\right)\varepsilon_{11}+\left(1+\beta+\beta^2\right)\varepsilon_{22}-1.5\kappa\Theta \\ \left(0.25+\beta+\beta^2\right)\gamma_{12} \end{Bmatrix} \tag{3.84}$$

จากสมการ (3.81-84) จะได้

$$\frac{\partial\beta}{\partial\varepsilon_k} = -\frac{4}{3}\frac{\beta}{\left(1+2\beta\right)^2}\frac{1}{e_e^2}\frac{\bar{A}\{C\}^T}{\left[1-\dfrac{2\beta\bar{A}}{\left(1+2\beta\right)^3}\left\{\dfrac{\left(\varepsilon_{11}+\varepsilon_{22}\right)^2-4\kappa\Theta\left(\varepsilon_{11}+\varepsilon_{22}-\kappa\Theta\right)}{e_e^2}\right\}\right]} \tag{3.85}$$

ค่าอนุพันธ์ $\dfrac{\partial\left(e_e^2\right)}{\partial\Theta}$ สามารถหาได้ในทำนองเดียวกันกับสมการ (3.75) และ (3.81) เท่ากับ

$$\frac{\partial\left(e_e^2\right)}{\partial\Theta} = \frac{d\left(e_e^2\right)}{d\beta}\frac{\partial\beta}{\partial\Theta} = 2e_e\frac{de_e}{d\beta}\frac{\partial\beta}{\partial\Theta} = -2\frac{e_e^2}{\beta\bar{A}}\frac{\partial\beta}{\partial\Theta} \tag{3.86}$$

ในทำนองเดียวกันกับสมการ (3.82) จะได้

$$\frac{\partial\left(e_e^2\right)}{\partial\Theta} = \frac{\partial\left(e_e^2\right)}{\partial\beta}\frac{\partial\beta}{\partial\Theta} + \left.\frac{\partial\left(e_e^2\right)}{\partial\Theta}\right|_{\beta} \tag{3.87}$$

จากสมการ (3.20ญ) จะได้พจน์ทางด้านขวามือของสมการ (3.87) เท่ากับ

$$\left.\frac{\partial\left(e_e^2\right)}{\partial\Theta}\right|_{\beta} = -\frac{4\kappa\left(\varepsilon_{11}+\varepsilon_{22}-2\kappa\Theta\right)}{\left(1+2\beta\right)^2} \tag{3.88}$$

แทนสมการ (3.88) และ (3.83ก) ลงในสมการ (3.87) แล้วแทนลงในสมการ (3.86) จะได้

$$\frac{\partial\beta}{\partial\Theta} = 2\frac{\beta}{(1+2\beta)^2}\frac{1}{e_e^2}\frac{(\varepsilon_{11}+\varepsilon_{22}-2\kappa\Theta)\kappa\bar{A}}{\left[1-\frac{2\beta\bar{A}}{(1+2\beta)^3}\left\{\frac{(\varepsilon_{11}+\varepsilon_{22})^2-4\kappa\Theta(\varepsilon_{11}+\varepsilon_{22}-\kappa\Theta)}{e_e^2}\right\}\right]} \quad (3.89)$$

เมื่อได้พจน์ต่าง ๆ อยู่ในรูปที่สามารถทำการคำนวณหาได้โดยง่ายแล้ว เราจึงสามารถคำนวณหาค่าเมตริกซ์และเวกเตอร์ต่าง ๆ ที่ใช้ในขั้นตอนการหาค่าผลเฉลยได้ จากสมการ (3.40ก), (3.20ฉ-ซ), (3.74ก-ค) และ (3.85) สิ่งที่ต้องตระหนักถึงในการคำนวณเมตริกซ์ความแข็งเกร็งสัมผัสเนื่องจากค่าความเค้นตั้งฉากเฉลี่ยและค่าความเค้นดิเวียทอริกสำหรับปัญหาความเค้นในระนาบก็คือเมตริกซ์ความแข็งเกร็งสัมผัสทั้งสองนี้ไม่ได้เป็นเมตริกซ์สมมาตรอีกต่อไปจนกว่าเมตริกซ์ทั้งสองจะถูกนำมารวมกันเป็นเมตริกซ์ความแข็งเกร็งสัมผัส (Tangent Stiffness Matrix)

3.4.2 ปัญหาความเครียดในระนาบ

เนื่องจากค่าความเครียดประสิทธิผล e_e เมื่อเขียนอยู่ในรูปของค่าความเครียดในสองมิติแล้วจะไม่ขึ้นอยู่กับค่าสนามอุณหภูมิดังแสดงในสมการ (3.23ซ) ขณะเดียวกันเมตริกซ์ $[E]$ ยังไม่ขึ้นกับค่าอุณหภูมิและค่าการเคลื่อนตัวที่จุดต่อซึ่งทำให้เวกเตอร์และเมตริกซ์ต่าง ๆ ที่ใช้ในขั้นตอนการหาผลเฉลยนั้นไม่ซับซ้อนเหมือนกับกรณีปัญหาความเค้นในระนาบ โดยค่าอนุพันธ์ของโหลดเวกเตอร์เนื่องจากค่าความเค้นตั้งฉากเฉลี่ยเมื่อเทียบกับค่าการเคลื่อนตัวและอุณหภูมิที่จุดต่อสามารถหาได้ในทำนองเดียวกับกรณีปัญหาความเค้นในระนาบเท่ากับ

$$\frac{\partial F_i^{vol}}{\partial u_n} = \int_\Omega B_{ji} E_{jk} B_{kn} d\Omega \quad (3.90)$$

$$\frac{\partial F_i^{vol}}{\partial \Theta_q} = -\int_\Omega B_{ji} E_{jk} \kappa_k N_q d\Omega \quad (3.91)$$

ตามลำดับ ขณะที่ค่าอนุพันธ์ของโหลดเวกเตอร์เนื่องจากความเค้นดิเวียทอริกเทียบกับค่าการเคลื่อนตัวและอุณหภูมิที่จุดต่อมีค่าเท่ากับ

$$\frac{\partial F_i^{dev}}{\partial u_n} = \int_\Omega B_{ji}\left[G_{jk} + \frac{\partial G_{jp}}{\partial \varepsilon_k}\varepsilon_p\right] B_{kn} d\Omega \quad (3.92)$$

$$\frac{\partial F_i^{dev}}{\partial \Theta_q} = 0 \quad (3.93)$$

ตามลำดับ ขณะที่ค่าอนุพันธ์ของโหลดเวกเตอร์เนื่องจากค่าความเค้นดึงที่ผิวและค่าแรงวัตถุเมื่อเทียบกับค่าความเค้นที่ผิวและแรงวัตถุที่จุดต่อจะมีค่าเท่ากับสมการ (3.70) และ (3.72) ตามลำดับ ค่าเทนเซอร์ $\frac{\partial G_{jp}}{\partial \varepsilon_k}\varepsilon_p$ สามารถหาได้จากความสัมพันธ์

$$\frac{\partial G_{jp}}{\partial \varepsilon_k}\varepsilon_p = \frac{\partial G_{jp}}{\partial\left(\sigma_e/e_e\right)}\varepsilon_p \frac{\partial\left(\sigma_e/e_e\right)}{\partial \varepsilon_k} \tag{3.94}$$

จากสมการ (3.23ง) และ (3.23ฉ) จะได้

$$\frac{\partial G_{jp}}{\partial\left(\sigma_e/e_e\right)}\varepsilon_p = \frac{2}{9}\{C\} \tag{3.95}$$

โดยที่

$$\{C\} = \begin{Bmatrix} 2\varepsilon_{11}-\varepsilon_{22} \\ -\varepsilon_{11}+2\varepsilon_{22} \\ 1.5\gamma_{12} \end{Bmatrix} \tag{3.96}$$

จากสมการ (3.9) จะได้ $\sigma_e = f\left(e_e\right)$ ดังนั้น

$$\frac{\partial\left(\sigma_e/e_e\right)}{\partial \varepsilon_k} = \frac{1}{e_e}\frac{d\sigma_e}{de_e}\frac{\partial e_e}{\partial \varepsilon_k} - \frac{\sigma_e}{e_e^2}\frac{\partial e_e}{\partial \varepsilon_k} = -\frac{\sigma_e}{e_e^2}\frac{\partial e_e}{\partial \varepsilon_k}\bar{A} \tag{3.97}$$

โดยที่ $\bar{A}$ มีค่าเท่ากับสมการ (3.79) จากสมการ (3.23ซ) จะได้ค่าอนุพันธ์

$$\frac{\partial e_e}{\partial \varepsilon_k} = \frac{2}{9}\left(\frac{1}{e_e}\right)\{C\}^T \tag{3.98}$$

ดังนั้นจากสมการ (3.96-98) จะได้

$$\frac{\partial\left(\sigma_e/e_e\right)}{\partial \varepsilon_k} = -\frac{2}{9}\left(\frac{\sigma_e}{e_e^3}\right)\bar{A}\{C\}^T \tag{3.99}$$

3.4.3 ปัญหาสมมาตรรอบแกน

เนื่องจากค่าความเครียดประสิทธิผล e_e เมื่อเขียนอยู่ในรูปของค่าความเครียดดังแสดงในสมการ (3.26ง) แล้วจะไม่ขึ้นอยู่กับค่าสนามอุณหภูมิดังแสดงในสมการ (3.26ซ) ในขณะเดียวกัน

เมตริกซ์ [E] ของกรณีปัญหาสมมาตรรอบแกนยังไม่ขึ้นกับค่าอุณหภูมิและการเคลื่อนตัวที่จุดต่อ ซึ่งเหมือนกับกรณีปัญหาความเครียดในระนาบ ดังนั้นสมการในรูปเทนเซอร์ต่าง ๆ ที่ใช้ในขั้นตอนการหาผลเฉลยจึงเหมือนกับกรณีปัญหาความเครียดในระนาบ โดยในกรณีปัญหาสมมาตรรอบแกนนี้จะได้

$$\{C\} = \begin{Bmatrix} 2\varepsilon_{11} - \varepsilon_{22} - \varepsilon_{33} \\ -\varepsilon_{11} + 2\varepsilon_{22} - \varepsilon_{33} \\ 1.5\gamma_{12} \\ -\varepsilon_{11} - \varepsilon_{22} + 2\varepsilon_{33} \end{Bmatrix} \tag{3.100}$$

ในขั้นตอนการหาผลเฉลยทั้งหมดนั้นเราจำเป็นต้องรู้ค่าอัตราส่วนของค่าความเค้นประสิทธิผลต่อค่าความเครียดประสิทธิผล ซึ่งในกรณีปัญหาความเครียดในระนาบและปัญหาสมมาตรรอบแกนนั้นเมื่อเราได้ค่าการเคลื่อนตัวที่จุดต่อทั้งหมดจากการคำนวณที่สถานะการวิเคราะห์ก่อนหน้านี้ที่มีผลเฉลยอยู่ในสภาวะสมดุลแล้วนั้นเราสามารถนำค่าที่ได้มาคำนวณหาค่าความเครียดสุทธิผ่านสมการ (3.38) จากนั้นนำมาหาค่าความเครียดประสิทธิผลโดยใช้สมการ (3.10) ซึ่งได้จาการแทนความสัมพันธ์ในสมการ (3.11-12) ลงไป จากนั้นจึงแก้สมการ (3.9) ซึ่งเป็นสมการไม่เชิงเส้นเพื่อหาค่าความเค้นประสิทธิผลต่อไปโดยประยุกต์ระเบียบวิธีการทำซ้ำแบบวิธีนิวตัน-ราฟสันเข้ากับสมการ (3.9) จะได้ผลต่างของค่าความเค้นประสิทธิผลต่อค่าความเค้นที่จุดครากสำหรับการทำซ้ำในแต่ละครั้งเป็น

$$\left(\frac{\Delta\sigma_e}{\sigma_o}\right)_i = -\left(\frac{\alpha\left(\dfrac{\sigma_e}{\sigma_o}\right)^n + \dfrac{2}{3}(1+\nu)\left(\dfrac{\sigma_e}{\sigma_o}\right) - \left(\dfrac{e_e}{\varepsilon_o}\right)}{\alpha n\left(\dfrac{\sigma_e}{\sigma_o}\right)^{n-1} + \dfrac{2}{3}(1+\nu)}\right)_i \tag{3.101}$$

จากนั้นทำการเดาค่าความเค้นประสิทธิผลเริ่มต้นขึ้นมาหนึ่งค่าแทนลงในสมการ (3.101) เพื่อใช้หาค่าความเค้นประสิทธิผลค่าใหม่ในการทำซ้ำครั้งต่อไปจากสมการ

$$(\sigma_e)_{i+1} = (\sigma_e)_i + (\Delta\sigma_e)_i \tag{3.102}$$

เมื่อได้ค่าความเค้นประสิทธิผลค่าใหม่แล้วจึงเริ่มกระบวนการทำซ้ำครั้งต่อไปจนได้คำตอบอยู่ในช่วงความเผื่อของการลู่เข้า (Convergence Tolerance) ที่ต้องการ

ในกรณีปัญหาความเค้นในระนาบ ค่าความเครียดประสิทธิผลดังแสดงในสมการ (3.20ญ) ขึ้นอยู่กับค่าความเค้นประสิทธิผลและค่าสนามอุณหภูมิ ดังนั้นทั้งค่าความเครียดและค่าความเค้นประสิทธิผลทางด้านขวาของสมการ (3.101) ต่างก็เป็นตัวแปรไม่ทราบค่าทั้งคู่ทำให้ต้องแก้สมการ

หลายสมการไปพร้อมกันโดยเริ่มจากการเดาค่าความเค้นประสิทธิผลขึ้นมาหนึ่งค่าเป็นค่าเริ่มต้นเพื่อใช้คำนวณหาค่าอัตราส่วนของค่าความเค้นประสิทธิผลต่อค่าความเครียดประสิทธิผลโดยจัดรูปสมการ (3.9) ใหม่เป็น

$$\frac{\sigma_e}{e_e} = \frac{E}{\alpha\left(\frac{\sigma_e}{\sigma_o}\right)^{n-1} + \frac{2}{3}(1+\nu)} \tag{3.103}$$

จากนั้นนำค่าที่ได้แทนลงในสมการ (3.20ฌ) จะได้ค่า β เพื่อนำไปใช้แทนในสมการ (3.20ญ) เพื่อหาค่าความเครียดประสิทธิผลจากนั้นจึงแทนค่าที่ได้ลงในสมการ (3.101) พร้อมกับค่าความเค้นประสิทธิผล จากนั้นหาค่าความเค้นประสิทธิผลในการทำซ้ำครั้งต่อไปจากสมการ (3.102) แล้วจึงเริ่มกระบวนการทำซ้ำครั้งใหม่ต่อไปจนกระทั่งได้ผลลัพธ์อยู่ในช่วงพิกัดความเผื่อที่ต้องการ

ในขั้นตอนการหาผลเฉลยทั้งสองขั้นตอนที่อธิบายไว้ข้างต้นนั้น หากเราใช้เฉพาะค่าการเคลื่อนตัวที่จุดต่อซึ่งคำนวณได้ในช่วง Incremental Solution Scheme มาใช้ในการคำนวณเมตริกซ์ความแข็งเกร็งสัมผัสและเวกเตอร์ต่าง ๆ ตลอดการคำนวณในช่วง Iterative Solution Scheme แล้วเราจะเรียกวิธีการทำซ้ำนี้ว่า Modified Newton-Raphson Method ในทางกลับกันหากเราใช้ค่าการเคลื่อนตัวที่จุดต่อที่คำนวณได้ในแต่ละช่วงมาคำนวณเมตริกซ์ความแข็งเกร็งสัมผัสและเวกเตอร์ต่าง ๆ ในช่วงนั้น ๆ แล้วเราจะเรียกวิธีการทำซ้ำแบบนี้ว่า Full Newton-Raphson Method [27] ข้อดีของวิธี Modified Newton-Raphson Method คือไม่ต้องทำสร้างเมตริกซ์ความแข็งเกร็งสัมผัสและแก้ระบบสมการใหม่หมดทุกครั้งที่มีการทำซ้ำเนื่องจากเมตริกซ์ความแข็งเกร็งสัมผัสที่ได้จากช่วง Incremental Solution Scheme ได้ถูกแยกตัวประกอบให้อยู่ในรูปที่ง่ายแก่การแก้ระบบสมการอยู่ก่อนแล้ว ส่วนข้อเสียของวิธีการทำซ้ำนี้เมื่อเทียบกับวิธี Full Newton-Raphson Method ก็คือวิธี Modified Newton-Raphson Method ให้อัตราการลู่เข้าของผลลัพธ์ที่ต่ำกว่า ดังนั้นจึงต้องใช้จำนวนครั้งในการทำซ้ำที่สูงกว่ามาก

ในการตรวจสอบการลู่เข้าของผลลัพธ์เพื่อใช้ในการหยุดการทำซ้ำนั้นจะใช้นอร์มของเวกเตอร์ความไม่สมดุลย์ของแรง (Norm of the Out of Balance Force Vector) เป็นตัวกำหนดโดยนอร์มนี้สามารถเขียนแทนได้เท่ากับ [25]

$$\|g\| = \sqrt{\{g\}^T\{g\}} \tag{3.104}$$

โดยการทำซ้ำจะหยุดลงเมื่อนอร์มของเวกเตอร์ความไม่สมดุลย์ของแรงที่คำนวณได้ในช่วง Iterative Solution Scheme เมื่อเทียบกับนอร์มของเวกเตอร์ความไม่สมดุลย์ของแรงที่คำนวณได้

ในช่วง Incremental Solution Scheme มีค่าน้อยกว่า ε (ปกติแล้วกำหนดให้มีค่าเท่ากับ 10^{-8} หรือต่ำกว่าโดยประมาณ) ดังสมการ

$$\|g\|_i \;\le\; \varepsilon\|g\|_o \tag{3.105}$$

โดยที่ $\|g\|_o$ แทนนอร์มของเวกเตอร์ความไม่สมดุลของแรงในช่วง Incremental Solution Scheme

$\|g\|_i$ แทนนอร์มของเวกเตอร์ความไม่สมดุลของแรงในช่วง Iterative Solution Scheme ที่การทำซ้ำครั้งที่ i

ε แทนค่าความเผื่อของการลู่เข้าที่ต้องการ

3.5 การคำนวณค่าความเค้นที่จุดต่อ

ในการคำนวณหาค่าความเค้นสุทธิที่จุดต่อต่าง ๆ นั้นจะแยกการคำนวณออกเป็นสองส่วน โดยเริ่มจากคำนวณค่าความเค้นตั้งฉากเฉลี่ยที่จุดต่อจากนั้นจึงคำนวณค่าความเค้นดิเวียทอริกที่จุดต่อแล้วนำค่าที่ได้มารวมกันเพื่อหาค่าความเค้นสุทธิที่จุดต่อโดยใช้สมการ (3.8) ในการคำนวณจะเริ่มจากคำนวณค่าความเค้นที่ตำแหน่งจุดเกาส์ (Gauss's Point) ที่ใช้ทั้งหมดในเอลิเมนต์จากนั้นจะทำการแปลงค่าที่ได้ไปที่ตำแหน่งจุดต่อต่าง ๆ โดยใช้เมตริกซ์ความสัมพันธ์ระหว่างค่าความเค้นที่จุดเกาส์และจุดต่อ ในวิทยานิพนธ์ฉบับนี้ผู้วิจัยได้ใช้จำนวนจุดเกาส์เท่ากับ 4 และ 9 จุดสำหรับเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อ และใช้จุดเกาส์เท่ากับ 3 และ 7 จุดสำหรับเอลิเมนต์สามเหลี่ยมหกจุดต่อ โดยเมตริกซ์ที่ใช้แปลงค่าความเค้นจากจุดเกาส์ไปที่จุดต่อสำหรับเอลิเมนต์ทั้งสองชนิดนี้ที่จำนวนจุดเกาส์ต่าง ๆ สามารถหาได้ดังนี้ [26]

3.5.1 เอลิเมนต์สี่เหลี่ยมเก้าจุดต่อ

สำหรับเอลิเมนต์ชนิดนี้ที่จำนวนจุดเกาส์เท่ากับ 9 จุด ความสัมพันธ์ระหว่างค่าความเค้นที่จุดต่อและจุดเกาส์สามารถเขียนอยู่ในรูปเมตริกซ์ได้ดังนี้

$$\{\tau_e\}_{nodes} \;=\; [TR]\{\tau_e\}_{opt} \tag{3.106}$$

โดยที่ $\{\tau_e\}_{nodes}$ แทนเวกเตอร์ค่าความเค้นที่จุดต่อ

$\{\tau_e\}_{opt}$ แทนเวกเตอร์ค่าความเค้นที่จุดเกาส์

$[TR]$ แทนเมตริกซ์ความสัมพันธ์ระหว่างค่าความเค้นที่จุดเกาส์และจุดต่อ

โดยเมตริกซ์ $\left[TR\right]$ สามารถหาได้ดังนี้ พิจารณาสมการสนามความเค้นบนพิกัดธรรมชาติ $(\xi,\ \eta)$ ของเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อ (Biquadratic Shape Function)

$$\tau_s\left(\xi,\ \eta\right) \ = \ c_1 + c_2\xi + c_3\eta + c_4\xi^2 + c_5\xi\eta + c_6\eta^2 + c_7\xi^2\eta + c_8\xi\eta^2 + c_9\xi^2\eta^2 \qquad (3.107)$$

โดยที่ $\tau_s\left(\xi,\ \eta\right)$ แทนสนามความเค้นบนเอลิเมนต์

$c_1,\ c_2,\ \ldots,\ c_9$ แทนค่าคงที่ของสมการ

เนื่องจากเราใช้จำนวนจุดเกาส์เท่ากับจำนวนค่าคงที่ดังนั้นค่าคงที่ $c_1,\ c_2,\ldots,\ c_9$ จึงต้องหาด้วยการแก้ระบบสมการ (Interpolatory Fit) พิจารณาสมการ (3.107) เราสามารถเขียนความสัมพันธ์ระหว่างเวกเตอร์ค่าความเค้นที่จุดเกาส์กับเวกเตอร์ค่าคงที่ได้เป็น

$$\left\{\tau_e\right\}_{opt} \ = \ \left[R\right]\left\{c\right\} \qquad (3.108)$$

โดยที่

$$\left[R\right] \ = \ \begin{bmatrix} 1 & \xi_I & \eta_I & \xi_I^2 & \xi_I\eta_I & \eta_I^2 & \xi_I^2\eta_I & \xi_I\eta_I^2 & \xi_I^2\eta_I^2 \\ 1 & \xi_{II} & \eta_{II} & \xi_{II}^2 & \xi_{II}\eta_{II} & \eta_{II}^2 & \xi_{II}^2\eta_{II} & \xi_{II}\eta_{II}^2 & \xi_{II}^2\eta_{II}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \xi_{IX} & \eta_{IX} & \xi_{IX}^2 & \xi_{IX}\eta_{IX} & \eta_{IX}^2 & \xi_{IX}^2\eta_{IX} & \xi_{IX}\eta_{IX}^2 & \xi_{IX}^2\eta_{IX}^2 \end{bmatrix} \qquad (3.109\text{ก})$$

$$\left\{c\right\}^T \ = \ \left\lfloor c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7 \quad c_8 \quad c_9 \right\rfloor \qquad (3.109\text{ข})$$

ดังนั้นเวกเตอร์ค่าคงที่จะมีค่าเท่ากับ

$$\left\{c\right\} \ = \ \left[R\right]^{-1}\left\{\tau_e\right\}_{opt} \qquad (3.110)$$

โดยพิกัดธรรมชาติของจุดเกาส์ทั้ง 9 จุดมีค่าดังนี้

$$\begin{aligned} \xi_I &= \xi_{II} = \xi_{III} = \eta_I = \eta_{IV} = \eta_{VII} = -\frac{\sqrt{3}}{\sqrt{5}} \\ \xi_{IV} &= \xi_V = \xi_{VI} = \eta_{II} = \eta_V = \eta_{VIII} = 0 \\ \xi_{VII} &= \xi_{VIII} = \xi_{IX} = \eta_{III} = \eta_{VI} = \eta_{XI} = +\frac{\sqrt{3}}{\sqrt{5}} \end{aligned} \qquad (3.111)$$

ในทำนองเดียวกันจากสมการ (3.107) เราสามารถเขียนสมการความสัมพันธ์ระหว่างเวกเตอร์ค่าความเค้นที่จุดต่อกับเวกเตอร์ค่าคงที่ได้เป็น

$$\{\tau_e\}_{\text{nodes}} = [S]\{c\} \tag{3.112}$$

โดยที่

$$[S] = \begin{bmatrix} 1 & \xi_1 & \eta_1 & \xi_1^2 & \xi_1\eta_1 & \eta_1^2 & \xi_1^2\eta_1 & \xi_1\eta_1^2 & \xi_1^2\eta_1^2 \\ 1 & \xi_2 & \eta_2 & \xi_2^2 & \xi_2\eta_2 & \eta_2^2 & \xi_2^2\eta_2 & \xi_2\eta_2^2 & \xi_2^2\eta_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \xi_9 & \eta_9 & \xi_9^2 & \xi_9\eta_9 & \eta_9^2 & \xi_9^2\eta_9 & \xi_9\eta_9^2 & \xi_9^2\eta_9^2 \end{bmatrix} \tag{3.113}$$

โดยพิกัดธรรมชาติของจุดต่อทั้ง 9 จุดมีค่าดังนี้

$$\begin{array}{ccccccc} \xi_1 & = & \xi_4 & = & \xi_8 & = & +1 \\ \xi_5 & = & \xi_7 & = & \xi_9 & = & 0 \\ \xi_2 & = & \xi_3 & = & \xi_6 & = & -1 \\ \eta_1 & = & \eta_2 & = & \eta_5 & = & +1 \\ \eta_6 & = & \eta_8 & = & \eta_9 & = & 0 \\ \eta_3 & = & \eta_4 & = & \eta_7 & = & -1 \end{array} \tag{3.114}$$

แทนค่าเวกเตอร์ $\{c\}$ จากสมการ (3.110) ลงในสมการ (3.112) จะได้สมการความสัมพันธ์ระหว่างเวกเตอร์ค่าความเค้นที่จุดต่อกับเวกเตอร์ค่าความเค้นที่จุดเกาส์เป็น

$$\{\tau_e\}_{\text{nodes}} = [S][R]^{-1}\{\tau_e\}_{\text{opt}} = [TR]\{\tau_e\}_{\text{opt}} \tag{3.115}$$

แทนค่าพิกัดธรรมชาติของจุดเกาส์และจุดต่อทั้งหมดลงในเมตริกซ์ต่าง ๆ จะได้เมตริกซ์ความสัมพันธ์ระหว่างเวกเตอร์ค่าความเค้นที่จุดเกาส์และจุดต่อสำหรับจำนวนจุดเกาส์เท่ากับ 9 จุดเป็น

$$[TR] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{12} & a_{15} & a_{16} & a_{13} & a_{16} & a_{19} \\ a_{13} & a_{16} & a_{19} & a_{12} & a_{15} & a_{16} & a_{11} & a_{12} & a_{13} \\ a_{19} & a_{16} & a_{13} & a_{16} & a_{15} & a_{12} & a_{13} & a_{12} & a_{11} \\ a_{13} & a_{12} & a_{11} & a_{16} & a_{15} & a_{12} & a_{19} & a_{16} & a_{13} \\ 0 & 0 & 0 & a_{54} & a_{55} & a_{56} & 0 & 0 & 0 \\ 0 & a_{56} & 0 & 0 & a_{55} & 0 & 0 & a_{54} & 0 \\ 0 & 0 & 0 & a_{56} & a_{55} & a_{54} & 0 & 0 & 0 \\ 0 & a_{54} & 0 & 0 & a_{55} & 0 & 0 & a_{56} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.116}$$

โดยที่

$$
\begin{aligned}
a_{11} &= 2.186939818390950 \\
a_{12} &= -0.985887038467491 \\
a_{13} &= 0.277777777777777 \\
a_{15} &= 0.444444444444445 \\
a_{16} &= -0.125224072643620 \\
a_{19} &= 0.035282403831273 \\
a_{54} &= 1.478830557701236 \\
a_{55} &= -0.666666666666666 \\
a_{56} &= 0.187836108965430
\end{aligned}
$$

ในกรณีที่จำนวนจุดเกาส์มีค่าเท่ากับ 4 จุดนั้น ความสัมพันธ์ระหว่างค่าความเค้นที่จุดต่อและจุดเกาส์ยังคงเขียนอยู่ในรูปเมตริกซ์ได้ดังสมการ (3.106) สมการสนามความเค้นบนพิกัดธรรมชาติซึ่งมีจำนวนค่าคงที่เท่ากับจำนวนจุดเกาส์สำหรับเอลิเมนต์สี่เหลี่ยมสี่จุดต่อ (Bilinear Shape Function) จะมีค่าเป็น

$$\tau_s(\xi, \eta) = c_1 + c_2\xi + c_3\eta + c_4\xi\eta \tag{3.117}$$

ในทำนองเดียวกันจะได้เมตริกซ์ต่าง ๆ มีเท่ากับ

$$[R] = \begin{bmatrix} 1 & \xi_I & \eta_I & \xi_I\eta_I \\ 1 & \xi_{II} & \eta_{II} & \xi_{II}\eta_{II} \\ 1 & \xi_{III} & \eta_{III} & \xi_{III}\eta_{III} \\ 1 & \xi_{IV} & \eta_{IV} & \xi_{IV}\eta_{IV} \end{bmatrix} \tag{3.118ก}$$

$$\{c\}^T = \lfloor c_1 \quad c_2 \quad c_3 \quad c_4 \rfloor \tag{3.118ข}$$

$$[S] = \begin{bmatrix} 1 & \xi_1 & \eta_1 & \xi_1\eta_1 \\ 1 & \xi_2 & \eta_2 & \xi_2\eta_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \xi_9 & \eta_9 & \xi_9\eta_9 \end{bmatrix} \tag{3.118ค}$$

โดยพิกัดธรรมชาติของจุดเกาส์ทั้ง 4 จุดจะมีค่าเท่ากับ

$$
\begin{aligned}
\xi_I &= \xi_{II} = \eta_I = \eta_{III} = -\frac{1}{\sqrt{3}} \\
\xi_{III} &= \xi_{IV} = \eta_{II} = \eta_{IV} = +\frac{1}{\sqrt{3}}
\end{aligned} \tag{3.119}
$$

แทนค่าพิกัดธรรมชาติของจุดเกาส์และจุดต่อทั้งหมดลงในเมตริกซ์ต่าง ๆ จะได้เมตริกซ์ความสัมพันธ์ระหว่างเวกเตอร์ค่าความเค้นที่จุดเกาส์และจุดต่อสำหรับจำนวนจุดเกาส์เท่ากับ 4 จุด เป็น

$$[TR] = \begin{bmatrix} b_{11} & b_{12} & b_{12} & b_{14} \\ b_{12} & b_{14} & b_{11} & b_{12} \\ b_{14} & b_{12} & b_{12} & b_{11} \\ b_{12} & b_{11} & b_{14} & b_{12} \\ b_{51} & b_{52} & b_{51} & b_{52} \\ b_{52} & b_{52} & b_{51} & b_{51} \\ b_{52} & b_{51} & b_{52} & b_{51} \\ b_{51} & b_{51} & b_{52} & b_{52} \\ b_{91} & b_{91} & b_{91} & b_{91} \end{bmatrix} \tag{3.120}$$

โดยที่

$$\begin{aligned} b_{11} &= 1.866025403784438 \\ b_{12} &= -0.5 \\ b_{14} &= 0.133974596215561 \\ b_{51} &= 0.683012701892219 \\ b_{52} &= 0.183012701892219 \\ b_{91} &= 0.25 \end{aligned}$$

3.5.2 เอลิเมนต์สามเหลี่ยมหกจุดต่อ

สำหรับเอลิเมนต์ชนิดนี้ที่จำนวนจุดเกาส์เท่ากับ 7 จุด ความสัมพันธ์ระหว่างค่าความเค้นที่จุดต่อและจุดเกาส์ยังคงเขียนอยู่ในรูปเมตริกซ์ได้ดังสมการ (3.106) สนามความเค้นของเอลิเมนต์สามเหลี่ยมหกจุดต่อ (Quadratic Shape Function) ซึ่งมีการกระจายของฟังก์ชั่นยกกำลังอันดับสองสมบูรณ์ (Complete 2^{nd} Degree Polynomial) สามารถเขียนอยู่ในรูปพิกัดธรรมชาติได้ดังสมการ

$$\tau_s(\xi, \eta) = c_1 + c_2\xi + c_3\eta + c_4\xi^2 + c_5\xi\eta + c_6\eta^2 \tag{3.121}$$

เนื่องจากจำนวนจุดเกาส์มีมากกว่าจำนวนค่าคงที่ดังนั้นค่าคงที่ $c_1, c_2, \ldots, c_6$ จึงต้องหาด้วยวิธียกกำลังสองน้อยที่สุด (Least Mean Square Method) ดังนี้ พิจารณาค่าความผิดพลาด e ระหว่างค่าความเค้นที่คำนวณได้โดยตรงจากจุดเกาส์กับค่าความเค้นที่คำนวณได้จากสนามความเค้นดังสมการ (3.121)

$$e = \sum_{i=I}^{VII} \left[\tau_s(\xi_i,\ \eta_i) - \tau_i \right]^2 \tag{3.122}$$

จากนั้นหาค่าคงที่ $c_1,\ c_2, \ldots,\ c_6$ ที่ทำให้ค่าความผิดพลาด e มีค่าน้อยที่สุดโดยการหาค่าอนุพันธ์ของค่าความผิดพลาดเทียบกับค่าคงที่ต่าง ๆ ดังสมการ

$$\frac{\partial e}{\partial c_i} = 0, \quad i = 1,\ 2, \ldots,\ 6 \tag{3.123}$$

แทนสมการ (3.121) ลงในสมการ (3.122) แล้วแทนลงในสมการด้านบนจะได้

$$\sum_{i=I}^{VII} \left[c_1 + c_2\xi_i + c_3\eta_i + c_4\xi_i^2 + c_5\xi_i\eta_i + c_6\eta_i^2 - \tau_i \right] = 0 \tag{3.124ก}$$

$$\sum_{i=I}^{VII} \left[c_1 + c_2\xi_i + c_3\eta_i + c_4\xi_i^2 + c_5\xi_i\eta_i + c_6\eta_i^2 - \tau_i \right]\xi_i = 0 \tag{3.124ข}$$

$$\sum_{i=I}^{VII} \left[c_1 + c_2\xi_i + c_3\eta_i + c_4\xi_i^2 + c_5\xi_i\eta_i + c_6\eta_i^2 - \tau_i \right]\eta_i = 0 \tag{3.124ค}$$

$$\sum_{i=I}^{VII} \left[c_1 + c_2\xi_i + c_3\eta_i + c_4\xi_i^2 + c_5\xi_i\eta_i + c_6\eta_i^2 - \tau_i \right]\xi_i^2 = 0 \tag{3.124ง}$$

$$\sum_{i=I}^{VII} \left[c_1 + c_2\xi_i + c_3\eta_i + c_4\xi_i^2 + c_5\xi_i\eta_i + c_6\eta_i^2 - \tau_i \right]\xi_i\eta_i = 0 \tag{3.124จ}$$

$$\sum_{i=I}^{VII} \left[c_1 + c_2\xi_i + c_3\eta_i + c_4\xi_i^2 + c_5\xi_i\eta_i + c_6\eta_i^2 - \tau_i \right]\eta_i^2 = 0 \tag{3.124ฉ}$$

ซึ่งสามารถเขียนให้อยู่ในรูปเมตริกซ์ได้เป็น

$$[P]\{c\} = [Q]\{\tau_e\}_{opt} \tag{3.125}$$

ดังนั้น

$$\{c\} = [P]^{-1}[Q]\{\tau_e\}_{opt} \tag{3.126}$$

โดยที่เมตริกซ์ $[P]$ และ $[Q]$ มีรายละเอียดดังนี้

$$[P] = \sum_{i=I}^{VII} \begin{bmatrix} 1 & \xi_i & \eta_i & \xi_i^2 & \xi_i\eta_i & \eta_i^2 \\ & \xi_i^2 & \xi_i\eta_i & \xi_i^3 & \xi_i^2\eta_i & \xi_i\eta_i^2 \\ & & \eta_i^2 & \xi_i^2\eta_i & \xi_i\eta_i^2 & \eta_i^3 \\ & & & \xi_i^4 & \xi_i^3\eta_i & \xi_i^2\eta_i^2 \\ & \text{Sym.} & & & \xi_i^2\eta_i^2 & \xi_i\eta_i^3 \\ & & & & & \eta_i^4 \end{bmatrix} \tag{3.127ก}$$

$$[Q] = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \xi_I & \xi_{II} & \cdots & \xi_{VII} \\ \eta_I & \eta_{II} & \cdots & \eta_{VII} \\ \xi_I^2 & \xi_{II}^2 & \cdots & \xi_{VII}^2 \\ \xi_I\eta_I & \xi_{II}\eta_{II} & \cdots & \xi_{VII}\eta_{VII} \\ \eta_I^2 & \eta_{II}^2 & \cdots & \eta_{VII}^2 \end{bmatrix} \tag{3.127ข}$$

โดยพิกัดธรรมชาติของจุดเกาส์ทั้ง 7 จุดเท่ากับ

$$\begin{array}{lllll} \xi_I & = 0.333333333333333 & & \xi_I & = 0.333333333333333 \\ \xi_{II} & = 0.101286507323456 & & \xi_{II} & = 0.101286507323456 \\ \xi_{III} & = 0.797426985353087 & & \xi_{III} & = 0.101286507323456 \\ \xi_{IV} & = 0.101286507323456 & & \xi_{IV} & = 0.797426985353087 \\ \xi_V & = 0.470142064105115 & & \xi_V & = 0.470142064105115 \\ \xi_{VI} & = 0.059715871789770 & & \xi_{VI} & = 0.470142064105115 \\ \xi_{VII} & = 0.470142064105115 & & \xi_{VII} & = 0.059715871789770 \end{array} \tag{3.128}$$

ในทำนองเดียวกันกับกรณีเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อจะได้

$$[S] = \begin{bmatrix} 1 & \xi_1 & \eta_1 & \xi_1^2 & \xi_1\eta_1 & \eta_1^2 \\ 1 & \xi_2 & \eta_2 & \xi_2^2 & \xi_2\eta_2 & \eta_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \xi_6 & \eta_6 & \xi_6^2 & \xi_6\eta_6 & \eta_6^2 \end{bmatrix} \tag{3.129}$$

โดยพิกัดธรรมชาติของจุดต่อทั้ง 6 จุดมีค่าเท่ากับ

$$\begin{array}{llll} \xi_1 & = 0 & \eta_1 & = 0 \\ \xi_2 & = 1 & \eta_2 & = 0 \\ \xi_3 & = 0 & \eta_3 & = 1 \\ \xi_4 & = 0.5 & \eta_4 & = 0 \\ \xi_5 & = 0.5 & \eta_5 & = 0.5 \\ \xi_6 & = 0 & \eta_6 & = 0.5 \end{array} \tag{3.130}$$

โดยเมื่อแทนค่าพิกัดธรรมชาติของจุดเกาส์และจุดต่อทั้งหมดลงในเมตริกซ์ต่าง ๆ แล้วจะได้

$$[TR] = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{13} & c_{15} & c_{16} & c_{16} \\ c_{11} & c_{13} & c_{12} & c_{13} & c_{16} & c_{15} & c_{16} \\ c_{11} & c_{13} & c_{13} & c_{12} & c_{16} & c_{16} & c_{15} \\ c_{41} & c_{42} & c_{42} & c_{44} & c_{45} & c_{45} & c_{47} \\ c_{41} & c_{44} & c_{42} & c_{42} & c_{47} & c_{45} & c_{45} \\ c_{41} & c_{42} & c_{44} & c_{42} & c_{45} & c_{47} & c_{45} \end{bmatrix} \quad (3.131)$$

โดยที่

$$\begin{aligned} c_{11} &= -0.692307692307643 \\ c_{12} &= 1.974392460120280 \\ c_{13} &= 0.143444958189243 \\ c_{15} &= 0.256376770648841 \\ c_{16} &= -0.412675727419964 \\ c_{41} &= 0.177514792899405 \\ c_{42} &= 0.098125809310777 \\ c_{44} &= 0.156266015552072 \\ c_{45} &= -0.282297406943914 \\ c_{47} &= 1.034562386814788 \end{aligned}$$

ในกรณีจำนวนจุดเกาส์เท่ากับ 3 จุด ความสัมพันธ์ระหว่างค่าความเค้นที่จุดต่อและจุดเกาส์ยังคงเขียนอยู่ในรูปเมตริกซ์ได้ดังสมการ (3.106) สมการสนามความเค้นบนพิกัดธรรมชาติสำหรับเอลิเมนต์สามเหลี่ยมสามจุดต่อ (Linear Shape Function) ซึ่งมีการกระจายของฟังก์ชั่นยกกำลังอันดับหนึ่งสมบูรณ์ (Complete 1st Degree Polynomial) สามารถเขียนแทนได้ดังสมการ

$$\tau_s(\xi, \eta) = c_1 + c_2\xi + c_3\eta \quad (3.132)$$

เนื่องจากจำนวนจุดเกาส์เท่ากับจำนวนค่าคงที่ ดังนั้นค่าคงที่ $c_1,\ c_2,\ c_3$ จึงต้องหาด้วยการแก้ระบบสมการ โดยเมตริกซ์ต่าง ๆ มีค่าดังนี้

$$[R] = \begin{bmatrix} 1 & \xi_I & \eta_I \\ 1 & \xi_{II} & \eta_{II} \\ 1 & \xi_{III} & \eta_{III} \end{bmatrix} \quad (3.133ก)$$

$$\{c\}^T = \lfloor c_1 \quad c_2 \quad c_3 \rfloor \quad (3.133ข)$$

$$[S] = \begin{bmatrix} 1 & \xi_1 & \eta_1 \\ 1 & \xi_2 & \eta_2 \\ \vdots & \vdots & \vdots \\ 1 & \xi_6 & \eta_6 \end{bmatrix} \qquad (3.133ค)$$

โดยพิกัดธรรมชาติของจุดเกาส์ทั้ง 3 จุดสำหรับเอลิเมนต์สามเหลี่ยมมีค่าดังนี้

$$\begin{aligned} \xi_I &= \frac{1}{6} \qquad & \eta_I &= \frac{1}{6} \\ \xi_{II} &= \frac{2}{3} \qquad & \eta_{II} &= \frac{1}{6} \\ \xi_{III} &= \frac{1}{6} \qquad & \eta_{III} &= \frac{2}{3} \end{aligned} \qquad (3.134)$$

โดยเมื่อแทนค่าพิกัดธรรมชาติของจุดเกาส์และจุดต่อทั้งหมดลงในเมตริกซ์ต่าง ๆ แล้วจะได้

$$[TR] = \frac{1}{3}\begin{bmatrix} 5 & -1 & -1 \\ -1 & 5 & -1 \\ -1 & -1 & 5 \\ 2 & 2 & -1 \\ -1 & 2 & 2 \\ 2 & -1 & 2 \end{bmatrix} \qquad (3.135)$$

3.6 สมการไฟไนต์เอลิเมนต์สำหรับคำนวณค่าพารามิเตอร์เจอินทิกรัลในรูปโดเมน

การหาค่าพารามิเตอร์เจอินทิกรัลด้วยระเบียบวิธีไฟไนต์เอลิเมนต์นั้นสามารถทำได้โดยเปลี่ยนสมการ (2.46) และ (2.56) ให้อยู่ในรูปสมการไฟไนต์เอลิเมนต์โดยในกรณีปัญหาความเค้นและความเครียดในระนาบจะได้

$$\begin{aligned} J = &\sum_{\substack{\text{all}\\ \text{elements}\\ \text{in } A^*}} \sum_{p=1}^{NG} w_p \left\{ \left[\left(\sigma_{ij}\frac{\partial u_j}{\partial x_k} - W\delta_{ik} \right) \frac{\partial q_k}{\partial x_i} + \left(3\kappa\bar{\sigma}\frac{\partial \Theta}{\partial x_j} - f_i \frac{\partial u_i}{\partial x_j} \right) q_j \right] |J| \right\}_p \\ &- \sum_{\substack{\text{all}\\ \text{element}\\ \text{edges on}\\ C^+ + C^-}} \sum_{v=1}^{NG} ws_p \left\{ \left[t_i \frac{\partial u_i}{\partial x_j} q_j \right] |J| \right\}_v \end{aligned} \qquad (3.136)$$

สำหรับกรณีปัญหาสมมาตรรอบแกนจะได้

$$J = \frac{1}{r_a}\sum_{\substack{\text{all}\\ \text{elements}\\ \text{in } A^*}}\sum_{p=1}^{NG} w_p\left\{\left[\left(\sigma_{ij}\frac{\partial u_j}{\partial x_k} - W\delta_{ik}\right)\frac{\partial q_k}{\partial x_i} + \left(3\kappa\bar{\sigma}\frac{\partial \Theta}{\partial x_j} - f_i\frac{\partial u_i}{\partial x_j}\right)q_j\right]r|J|\right\}_p$$
$$+\frac{1}{r_a}\sum_{\substack{\text{all}\\ \text{elements}\\ \text{in } A^*}}\sum_{p=1}^{NG} w_p\left\{\left[\left(\sigma_{33}\frac{u_1}{r} - W\right)q_1\right]|J|\right\}_p$$
$$-\frac{1}{r_a}\sum_{\substack{\text{all}\\ \text{element}\\ \text{edges on}\\ C^+ + C^-}}\sum_{p=1}^{NG} ws_p\left\{\left[t_i\frac{\partial u_i}{\partial x_j}q_j\right]r|J_S|\right\}_p \tag{3.137}$$

โดยที่ NG แทนจำนวนจุดเกาส์ที่ใช้ในการอินทิเกรต

$[J]$ แทนยาโคบีเมตริกซ์ (Jacobian Matrix)

w_p แทนน้ำหนักที่สอดคล้องกับการอินทิเกรตบนพื้นที่ของจุดเกาส์ที่ p

ws_p แทนน้ำหนักที่สอดคล้องกับการอินทิเกรตตามเส้นของจุดเกาส์ที่ p

r แทนระยะทางในแนวรัศมีจากแกนสมมาตร

r_a แทนระยะทางในแนวรัศมีจากแกนสมมาตรถึงปลายรอยร้าว

ค่าในวงเล็บปีกกา $\{\ \}_p$ จะต้องคำนวณที่พิกัดของจุดเกาส์ที่สอดคล้องกับการอินทิเกรตบนพื้นที่โดเมนและตามเส้นผิวรอยร้าว

เนื่องจากในการสร้างสมการความสมดุลย์เราแบ่งค่าความเค้นสุทธิออกเป็นผลบวกของค่าความเค้นตั้งฉากเฉลี่ยและค่าความเค้นดิเวียทอริกซึ่งในการคำนวณพจน์ต่าง ๆ ที่เกี่ยวข้องนี้เราใช้จำนวนจุดเกาส์ในการอินทิเกรตสมการความสมดุลย์ไม่เท่ากันดังนั้นในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลเราจึงสามารถแบ่งการคำนวณออกเป็นพจน์ที่เกี่ยวข้องกับค่าความเค้นตั้งฉากเฉลี่ย พจน์ที่เกี่ยวข้องกับค่าความเค้นดิเวียทอริกและพจน์ที่เกี่ยวข้องกับค่าความเค้นดึงที่ผิวรอยร้าวเป็น [9]

$$J = \sum_{\substack{\text{all}\\ \text{elements}\\ \text{in } A^*}}\left\{\sum_{p=1}^{VNG}\begin{bmatrix}\text{volumetric}\\ \text{terms}\end{bmatrix}_p w_p + \sum_{p=1}^{DNG}\begin{bmatrix}\text{deviatoric}\\ \text{terms}\end{bmatrix}_p w_p\right\}$$
$$-\sum_{\substack{\text{all}\\ \text{element}\\ \text{edges on}\\ C^+ + C^-}}\left\{\sum_{p=1}^{CNG}\begin{bmatrix}\text{crack face}\\ \text{terms}\end{bmatrix}_p w_p\right\} \tag{3.138}$$

โดยในกรณีปัญหาความเค้นและความเครียดในระนาบจะได้

$$\begin{bmatrix}\text{volumetric}\\ \text{terms}\end{bmatrix} = \left[\left(\bar{\sigma}\frac{\partial u_i}{\partial x_j} - \bar{W}\delta_{ij}\right)\frac{\partial q_j}{\partial x_i} + 3\kappa\bar{\sigma}\frac{\partial \Theta}{\partial x_i}q_i\right]|J| \qquad (3.139ก)$$

$$\begin{bmatrix}\text{deviatoric}\\ \text{terms}\end{bmatrix} = \left[\left(S_{ij}\frac{\partial u_j}{\partial x_k} - \Psi\delta_{ik}\right)\frac{\partial q_k}{\partial x_i} - f_i\frac{\partial u_i}{\partial x_j}q_j\right]|J| \qquad (3.139ข)$$

$$\begin{bmatrix}\text{crack face}\\ \text{terms}\end{bmatrix} = \left[t_i\frac{\partial u_i}{\partial x_j}q_j\right]|J| \qquad (3.139ค)$$

สำหรับกรณีปัญหาสมมาตรรอบแกนได้

$$\begin{bmatrix}\text{volumetric}\\ \text{terms}\end{bmatrix} = \left[\left(\bar{\sigma}\frac{\partial u_i}{\partial x_j} - \bar{W}\delta_{ij}\right)\frac{\partial q_j}{\partial x_i} + 3\kappa\bar{\sigma}\frac{\partial \Theta}{\partial x_i}q_i\right]\left(\frac{r}{r_a}\right)|J| + \left[\left(\bar{\sigma}\frac{u_1}{r} - \bar{W}\right)q_1\right]\left(\frac{1}{r_a}\right)|J| \qquad (3.140ก)$$

$$\begin{bmatrix}\text{deviatoric}\\ \text{terms}\end{bmatrix} = \left[\left(S_{ij}\frac{\partial u_j}{\partial x_k} - \Psi\delta_{ik}\right)\frac{\partial q_k}{\partial x_i} - f_i\frac{\partial u_i}{\partial x_j}q_j\right]\left(\frac{r}{r_a}\right)|J| + \left[\left(S_{33}\frac{u_1}{r} - \Psi\right)q_1\right]\left(\frac{1}{r_a}\right)|J| \qquad (3.140ข)$$

$$\begin{bmatrix}\text{crack face}\\ \text{terms}\end{bmatrix} = \left[t_i\frac{\partial u_i}{\partial x_j}q_j\right]\left(\frac{r}{r_a}\right)|J| \qquad (3.140ค)$$

โดยที่

$$W = \bar{W} + \Psi \qquad (3.141ก)$$

$$\bar{W} = \frac{3}{2}\left(\frac{1-2\nu}{E}\right)\bar{\sigma}^2 \qquad (3.141ข)$$

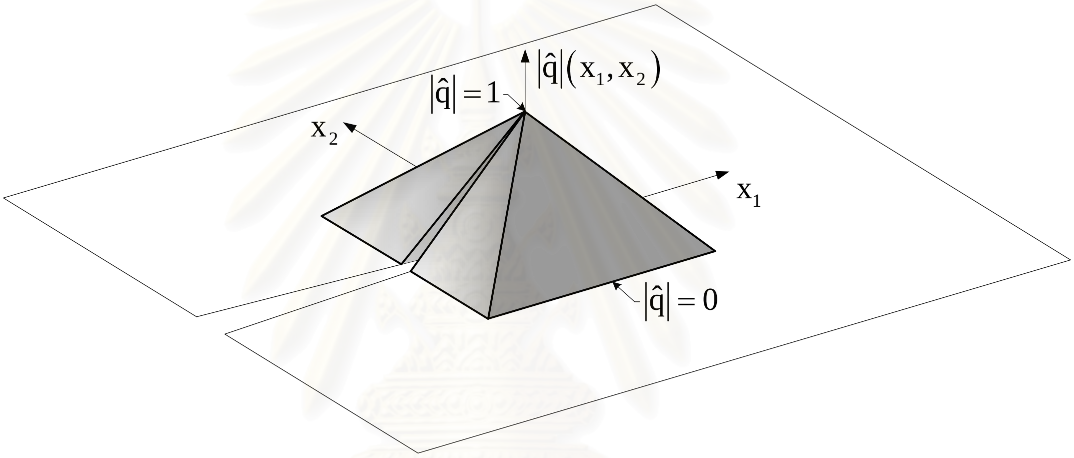
$$\Psi = \frac{1}{3}\left(\frac{1+\nu}{E}\right)\sigma_e^2 + \left(\frac{n}{n+1}\right)\alpha\varepsilon_o\sigma_o\left(\frac{\sigma_e}{\sigma_o}\right)^{n+1} \qquad (3.141ค)$$

โดยที่ VNG แทนจำนวนจุดเกาส์ทั้งหมดที่ใช้ในการอินทิเกรตพจน์ที่เกี่ยวข้องกับค่าความเค้นตั้งฉากเฉลี่ย

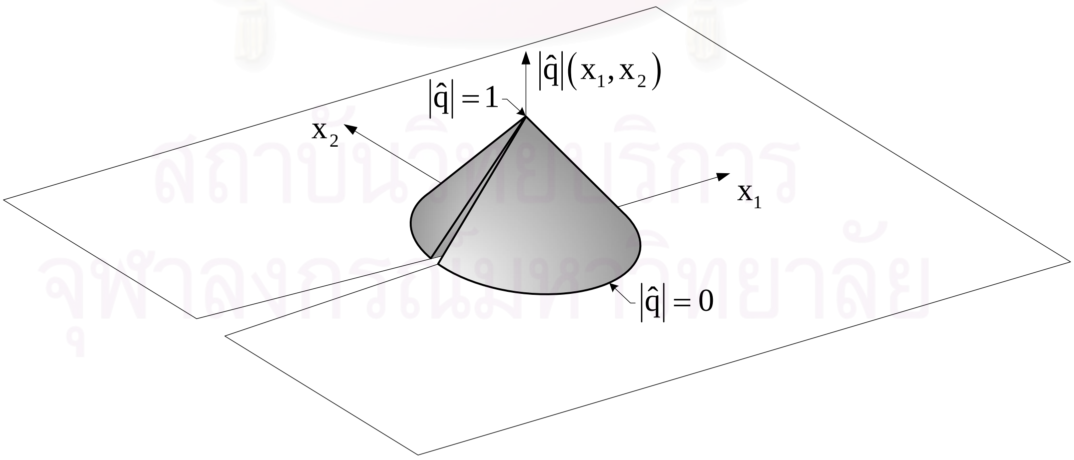
DNG แทนจำนวนจุดเกาส์ทั้งหมดที่ใช้ในการอินทิเกรตพจน์ที่เกี่ยวข้องกับค่าความเค้นดิเวียทอริก

CNG แทนจำนวนจุดเกาส์ทั้งหมดที่ใช้ในการอินทิเกรตพจน์ที่เกี่ยวข้องกับค่าความเค้นดึงที่ผิวรอยร้าว

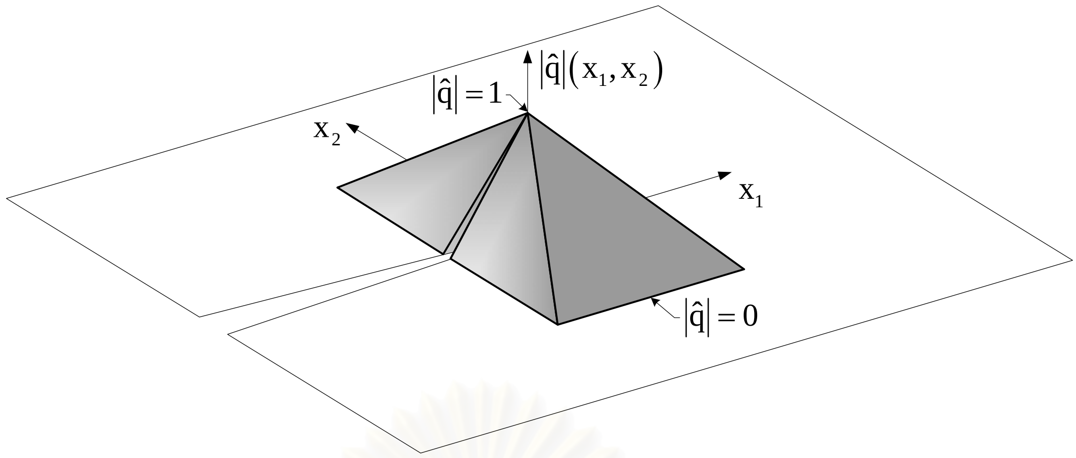
ในการคำนวณค่าพารามิเตอร์เจอินทิกรัลในรูปโดเมนด้วยระเบียบวิธีไฟไนต์เอลิเมนต์นั้น เราจะต้องกำหนดลักษณะการกระจายของค่าเวกเตอร์ q_k ลงบนพื้นที่โดเมนที่ใช้ในการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัล Shih et al. [10] ได้ทดลองใช้การกระจายของค่าเวกเตอร์ q_k ในลักษณะต่าง ๆ แล้วพบว่าค่าพารามิเตอร์เจอินทิกรัลที่ได้นั้นไม่แตกต่างกันมากนัก โดยในวิทยานิพนธ์นี้จะใช้ลักษณะการกระจายของค่าเวกเตอร์ q_k ด้วยกัน 3 แบบคือแบบปิรามิดฐานสี่เหลี่ยมจัตุรัส แบบกรวยฐานวงกลมและแบบปิรามิดฐานสี่เหลี่ยมผืนผ้าดังแสดงในรูปที่ 3.3



(ก) ฟังก์ชั่นรูปปิรามิดฐานสี่เหลี่ยมจัตุรัส



(ข) ฟังก์ชั่นรูปกรวยฐานวงกลม



(ค) ฟังก์ชั่นรูปปิรามิดฐานสี่เหลี่ยมผืนผ้า

รูปที่ 3.3 ลักษณะการกระจายของเวกเตอร์ q_k ที่ใช้ในวิทยานิพนธ์นี้

เวกเตอร์ q_k ที่กระจายบนเอลิเมนต์สามารถเขียนให้อยู่ในรูปสมการไฟไนต์เอลิเมนต์ได้เป็น

$$|\hat{q}|(\xi, \eta) = \sum_{i=1}^{n} N_i q_i \tag{3.142}$$

โดยที่ q_i แทนค่าขนาดของเวกเตอร์ q_k ที่จุดต่อ i

3.7 บทสรุป

ในบทนี้สมการไฟไนต์เอลิเมนต์ที่ใช้ในการวิเคราะห์รอยร้าวในชิ้นงานสำหรับปัญหาความเค้นระนาบ ความเครียดระนาบและปัญหาสมมาตรรอบแกนซึ่งผลของค่าความเครียดเริ่มต้นเนื่องจากอุณหภูมิได้ถูกประดิษฐ์ขึ้นพร้อมทั้งเมตริกซ์ความสัมพันธ์ระหว่างค่าความเค้นที่จุดเกาส์และจุดต่อสำหรับเอลิเมนต์สามเหลี่ยมหกจุดต่อและเอลิเมนต์ที่ปลายรอยร้าวแบบสี่เหลี่ยมเก้าจุดต่อ จากนั้นสมการค่าพารามิเตอร์เจอินทิกรัลได้ถูกเปลี่ยนให้อยู่ในรูปของสมการไฟไนต์เอลิเมนต์โดยแบ่งการคำนวณออกเป็นพจน์ต่าง ๆ ที่เกี่ยวข้องกับความเค้นตั้งฉากเฉลี่ย ความเค้นดิเวียทอริกและความเค้นดึงที่ผิวรอยร้าว เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติพร้อมทั้งเทคนิคการถ่ายทอดผลเฉลยระยะการเคลื่อนตัวจากโครงตาข่ายก่อนการปรับขนาดเอลิเมนต์ไปสู่โครงตาข่ายที่ทำการปรับขนาดเอลิเมนต์แล้วได้ถูกอธิบายในบทต่อไป

บทที่ 4

เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติและการถ่ายทอดผลเฉลย

ในการวิเคราะห์ปัญหาด้วยระเบียบวิธีไฟไนต์เอลิเมนต์นั้นขนาดของเอลิเมนต์จะมีผลต่อความแม่นยำของผลเฉลย โดยการใช้เอลิเมนต์ที่มีขนาดเล็กจำนวนมากจะส่งผลให้คำตอบที่ได้มีความถูกต้องมากขึ้นในขณะเดียวกันระยะเวลาในการคำนวณตลอดจนหน่วยความจำของเครื่องคอมพิวเตอร์ (Ram) ก็ต้องใช้มากขึ้นตามไปด้วย เพื่อแก้ไขปัญหาดังกล่าวจึงมีการนำเอาเทคนิคการปรับขนาดของเอลิเมนต์โดยอัตโนมัติมาใช้ร่วมกับระเบียบวิธีไฟไนต์เอลิเมนต์ โดยในเทคนิคนี้ขนาดของเอลิเมนต์จะถูกปรับให้เหมาะสมตามลักษณะการกระจายตัวของผลเฉลยทำให้ผลเฉลยที่ได้มีความแม่นยำโดยไม่จำเป็นต้องใช้เอลิเมนต์จำนวนมาก เมื่อมีการปรับขนาดเอลิเมนต์ผลเฉลยการเคลื่อนตัวที่จุดต่อก่อนการปรับขนาดเอลิเมนต์จะถูกถ่ายทอด (Mapping) ลงมาในโครงตาข่าย (Mesh) ใหม่หลังการปรับขนาดเอลิเมนต์ทำให้การคำนวณที่ค่าระดับภาระต่าง ๆ สามารถดำเนินต่อเนื่องไปได้โดยไม่ต้องเริ่มใหม่ที่ระดับภาระเริ่มต้นทุกครั้งหลังการปรับขนาดเอลิเมนต์ โดยเนื้อหาในบทนี้จะอธิบายถึงหลักการที่ใช้ในการปรับขนาดเอลิเมนต์และการถ่ายทอดผลเฉลยจากโครงตาข่ายเก่ามาสู่โครงตาข่ายใหม่

4.1 เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติ

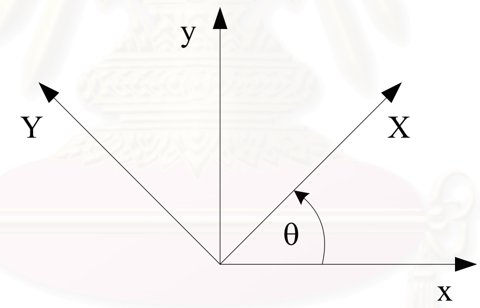
จากอนุกรมของเทย์เลอร์ [25] ค่าความผิดพลาด (Error) ของผลเฉลยค่าความเครียดหรือค่าความเค้นในเอลิเมนต์ใด ๆ จะแปรผันกับขนาดของเอลิเมนต์ h เป็น $O\left(h^{p+1-m}\right)$ โดยที่ p แทนอันดับของฟังก์ชั่นการประมาณภายในที่ใช้ในเอลิเมนต์นั้น ๆ m แทนอันดับของอนุพันธ์ของค่าระยะการเคลื่อนตัวที่ใช้ในการนิยามค่าความเครียด เนื่องจากเอลิเมนต์ทั้งหมดที่ใช้ในวิทยานิพนธ์ฉบับนี้เป็นแบบอันดับสอง (Quadratic Element) และค่าความเครียดถูกนิยามด้วยอนุพันธ์อันดับหนึ่งของระยะการเคลื่อนตัว ดังนั้นค่าความผิดพลาดของผลเฉลยของค่าความเครียดและค่าความเค้นในเอลิเมนต์จะแปรผันกับขนาดของเอลิเมนต์เป็น $O\left(h^2\right)$ ซึ่งจากอนุกรมของเทย์เลอร์ค่าความผิดพลาดนี้ก็จะแปรผันกับค่าอนุพันธ์อันดับสองของค่าความเครียดและค่าความเค้นด้วย ดังนั้นหลักการของเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติจะทำการปรับขนาดเอลิเมนต์ให้มีขนาดเล็กในบริเวณที่ขนาดของค่าอนุพันธ์อันดับสองของผลเฉลยมีค่าสูงขณะที่ใช้เอลิเมนต์ขนาดใหญ่ในบริเวณที่ขนาดของค่าอนุพันธ์อันดับสองของผลเฉลยมีค่าต่ำซึ่งหลักการที่ใช้ในการคำนวณขนาดของค่าอนุพันธ์อันดับสองสามารถอธิบายได้ดังนี้ พิจารณาพิกัด x-y และพิกัด X-Y ใด ๆ ดังแสดง

ในรูปที่ 4.1 ค่าอนุพันธ์อันดับสองของผลเฉลย Φ ซึ่งเขียนอยู่ในรูปของฟังก์ชั่นพิกัด x-y เทียบกับพิกัด X-Y สามารถหาได้จากสมการ

$$\frac{\partial^2\Phi}{\partial X^2} = \frac{\partial\Phi}{\partial x}\frac{\partial^2 x}{\partial X^2} + \frac{\partial x}{\partial X}\left[\frac{\partial^2\Phi}{\partial x^2}\frac{\partial x}{\partial X}+\frac{\partial^2\Phi}{\partial x\partial y}\frac{\partial y}{\partial X}\right] + \frac{\partial\Phi}{\partial y}\frac{\partial^2 y}{\partial X^2} + \frac{\partial y}{\partial X}\left[\frac{\partial^2\Phi}{\partial x\partial y}\frac{\partial x}{\partial X}+\frac{\partial^2\Phi}{\partial y^2}\frac{\partial y}{\partial X}\right] \quad (4.1ก)$$

$$\frac{\partial^2\Phi}{\partial Y^2} = \frac{\partial\Phi}{\partial x}\frac{\partial^2 x}{\partial Y^2} + \frac{\partial x}{\partial Y}\left[\frac{\partial^2\Phi}{\partial x^2}\frac{\partial x}{\partial Y}+\frac{\partial^2\Phi}{\partial x\partial y}\frac{\partial y}{\partial Y}\right] + \frac{\partial\Phi}{\partial y}\frac{\partial^2 y}{\partial Y^2} + \frac{\partial y}{\partial Y}\left[\frac{\partial^2\Phi}{\partial x\partial y}\frac{\partial x}{\partial Y}+\frac{\partial^2\Phi}{\partial y^2}\frac{\partial y}{\partial Y}\right] \quad (4.1ข)$$

$$\frac{\partial^2\Phi}{\partial X\partial Y} = \frac{\partial\Phi}{\partial x}\frac{\partial^2 x}{\partial X\partial Y} + \frac{\partial x}{\partial X}\left[\frac{\partial^2\Phi}{\partial x^2}\frac{\partial x}{\partial Y}+\frac{\partial^2\Phi}{\partial x\partial y}\frac{\partial y}{\partial Y}\right] + \frac{\partial\Phi}{\partial y}\frac{\partial^2 y}{\partial X\partial Y} + \frac{\partial y}{\partial X}\left[\frac{\partial^2\Phi}{\partial x\partial y}\frac{\partial x}{\partial Y}+\frac{\partial^2\Phi}{\partial y^2}\frac{\partial y}{\partial Y}\right] \quad (4.1ค)$$



รูปที่ 4.1 พิกัด x-y และพิกัด X-Y

โดยพิกัด x-y และพิกัด X-Y มีความสัมพันธ์กันดังสมการ

$$x = X\cos\theta - Y\sin\theta \quad (4.2ก)$$

$$y = Y\cos\theta + X\sin\theta \quad (4.2ข)$$

ดังนั้นจะได้ค่าอนุพันธ์ของพิกัด x-y เทียบกับพิกัด X-Y เท่ากับ

$$\frac{\partial x}{\partial X} = \cos\theta \qquad \frac{\partial x}{\partial Y} = -\sin\theta \quad (4.3ก)$$

$$\frac{\partial y}{\partial X} = \sin\theta \qquad \frac{\partial y}{\partial Y} = \cos\theta \tag{4.3ข}$$

$$\frac{\partial^2 x}{\partial X^2} = 0 \qquad \frac{\partial^2 y}{\partial X^2} = 0 \tag{4.3ค}$$

แทนสมการ (4.3) ลงในสมการ (4.1) จะได้

$$\frac{\partial^2\Phi}{\partial X^2} = \frac{\partial^2\Phi}{\partial x^2}\cos^2\theta + 2\frac{\partial^2\Phi}{\partial x\partial y}\sin\theta\cos\theta + \frac{\partial^2\Phi}{\partial y^2}\sin^2\theta \tag{4.4ก}$$

$$\frac{\partial^2\Phi}{\partial Y^2} = \frac{\partial^2\Phi}{\partial x^2}\sin^2\theta - 2\frac{\partial^2\Phi}{\partial x\partial y}\sin\theta\cos\theta + \frac{\partial^2\Phi}{\partial y^2}\cos^2\theta \tag{4.4ข}$$

$$\frac{\partial^2\Phi}{\partial X\partial Y} = \left(\frac{\partial^2\Phi}{\partial x^2} - \frac{\partial^2\Phi}{\partial y^2}\right)\sin\theta\cos\theta + \frac{\partial^2\Phi}{\partial x\partial y}\left(\cos^2\theta - \sin^2\theta\right) \tag{4.4ค}$$

ค่าสูงสุดหรือค่าต่ำสุดของอนุพันธ์อันดับสองของผลเฉลยเทียบกับพิกัด X และ Y ที่มุม θ ใด ๆ สามารถหาได้จาก

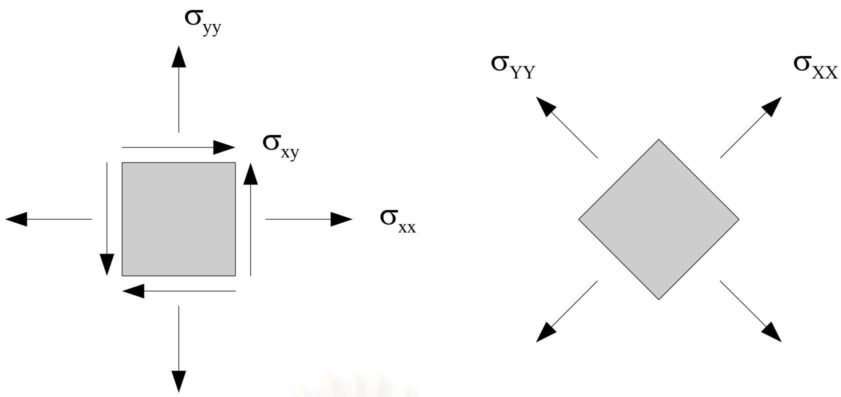
$$\frac{\partial}{\partial\theta}\left(\frac{\partial^2\Phi}{\partial X^2}\right) = -2\left(\frac{\partial^2\Phi}{\partial x^2} - \frac{\partial^2\Phi}{\partial y^2}\right)\sin\theta\cos\theta + 2\frac{\partial^2\Phi}{\partial x\partial y}\left(\cos^2\theta - \sin^2\theta\right) = 0 \tag{4.5ก}$$

$$\frac{\partial}{\partial\theta}\left(\frac{\partial^2\Phi}{\partial Y^2}\right) = 2\left(\frac{\partial^2\Phi}{\partial x^2} - \frac{\partial^2\Phi}{\partial y^2}\right)\sin\theta\cos\theta - 2\frac{\partial^2\Phi}{\partial x\partial y}\left(\cos^2\theta - \sin^2\theta\right) = 0 \tag{4.5ข}$$

สมการ (4.5) สามารถจัดรูปใหม่จากการประยุกต์ใช้สมการตรีโกณมิติได้เป็น

$$\tan 2\theta = \frac{\dfrac{\partial^2\Phi}{\partial x\partial y}}{\dfrac{1}{2}\left(\dfrac{\partial^2\Phi}{\partial x^2} - \dfrac{\partial^2\Phi}{\partial y^2}\right)} \tag{4.6}$$

สมการ (4.6) แสดงให้เห็นว่าค่าสูงสุดหรือค่าต่ำสุดของอนุพันธ์อันดับสองของผลเฉลยเทียบกับพิกัด X และ Y จะเกิดขึ้นที่มุม θ ค่าเดียวกันและจากสมการ (4.5) และ (4.4) ยังทำให้เราทราบว่าที่ค่าสูงสุดหรือค่าต่ำสุดของอนุพันธ์อันดับสองของผลเฉลยเทียบกับพิกัด X หรือ Y นั้นค่าอนุพันธ์อันดับสอง $\frac{\partial^2\Phi}{\partial X\partial Y}$ จะมีค่าเป็นศูนย์เสมอซึ่งเหมือนกับหลักการหาค่าความเค้นในแนวแกนหลัก (Principal Stress) ในวิชากลศาสตร์ของแข็ง (Solid Mechanics) ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 เปรียบเทียบค่าความเค้นกับอนุพันธ์อันดับสองของผลเฉลยเทียบกับแกนพิกัดต่าง ๆ

ดังนั้นในทำนองเดียวกันเราสามารถเขียนสมการความสัมพันธ์ได้เป็น

$$\begin{bmatrix} \dfrac{\partial^2\Phi}{\partial x^2} & \dfrac{\partial^2\Phi}{\partial x\partial y} \\ \dfrac{\partial^2\Phi}{\partial x\partial y} & \dfrac{\partial^2\Phi}{\partial y^2} \end{bmatrix} \rightarrow \begin{bmatrix} \dfrac{\partial^2\Phi}{\partial X^2} & 0 \\ 0 & \dfrac{\partial^2\Phi}{\partial Y^2} \end{bmatrix} \quad (4.7)$$

ค่าอนุพันธ์อันดับสอง $\dfrac{\partial^2\Phi}{\partial X^2}$ และ $\dfrac{\partial^2\Phi}{\partial Y^2}$ ในสมการ (4.4 ก) และ (4.4ข) สามารถจัดรูปใหม่ได้จากการประยุกต์ใช้สมการตรีโกณมิติได้เป็น

$$\frac{\partial^2\Phi}{\partial X^2} = \frac{\dfrac{\partial^2\Phi}{\partial x^2}+\dfrac{\partial^2\Phi}{\partial y^2}}{2} + \frac{1}{2}\left(\frac{\partial^2\Phi}{\partial x^2}-\frac{\partial^2\Phi}{\partial y^2}\right)\cos 2\theta + \frac{\partial^2\Phi}{\partial x\partial y}\sin 2\theta \quad (4.8\text{ ก})$$

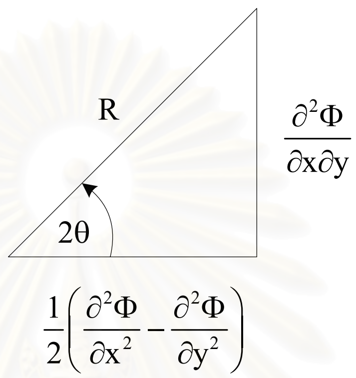
$$\frac{\partial^2\Phi}{\partial Y^2} = \frac{\dfrac{\partial^2\Phi}{\partial x^2}+\dfrac{\partial^2\Phi}{\partial y^2}}{2} - \frac{1}{2}\left(\frac{\partial^2\Phi}{\partial x^2}-\frac{\partial^2\Phi}{\partial y^2}\right)\cos 2\theta - \frac{\partial^2\Phi}{\partial x\partial y}\sin 2\theta \quad (4.8\text{ข})$$

พิจารณาสองพจน์สุดท้ายทางด้านขวาของสมการ (4.8) และรูปที่ 4.3 ซึ่งแสดงสมการ (4.6) ในรูปสามเหลี่ยมมุมฉาก สมการ (4.8) สามารถเขียนใหม่ได้เป็น

$$\frac{\partial^2\Phi}{\partial X^2} = \frac{\dfrac{\partial^2\Phi}{\partial x^2}+\dfrac{\partial^2\Phi}{\partial y^2}}{2} + R \quad (4.9\text{ ก})$$

$$\frac{\partial^2\Phi}{\partial Y^2} = \frac{\dfrac{\partial^2\Phi}{\partial x^2}+\dfrac{\partial^2\Phi}{\partial y^2}}{2} - R \qquad (4.9ข)$$

$$R = \frac{1}{2}\left(\frac{\partial^2\Phi}{\partial x^2}-\frac{\partial^2\Phi}{\partial y^2}\right)\cos 2\theta + \frac{\partial^2\Phi}{\partial x\partial y}\sin 2\theta \qquad (4.9ค)$$



รูปที่ 4.3 สมการ (4.6) ซึ่งถูกเขียนอยู่ในรูปความสัมพันธ์ของสามเหลี่ยมมุมฉาก

ดังนั้นค่าสูงสุดหรือค่าต่ำสุดของอนุพันธ์อันดับสอง $\frac{\partial^2\Phi}{\partial X^2}$ และ $\frac{\partial^2\Phi}{\partial Y^2}$ สามารถเขียนอยู่ในรูปค่าเจาะจงได้เป็น

$$\text{ค่าเจาะจง} = \frac{\dfrac{\partial^2\Phi}{\partial x^2}+\dfrac{\partial^2\Phi}{\partial y^2}}{2} \pm \sqrt{\frac{1}{4}\left(\frac{\partial^2\Phi}{\partial x^2}-\frac{\partial^2\Phi}{\partial y^2}\right)^2+\left(\frac{\partial^2\Phi}{\partial x\partial y}\right)^2} \qquad (4.10)$$

เนื่องจากเอลิเมนต์ที่ใช้ในการแก้ปัญหาในวิทยานิพนธ์นี้แบ่งออกเป็นสองประเภทคือเอลิเมนต์ที่ปลายรอยร้าวซึ่งเป็นเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อที่ด้านหนึ่งถูกยุบมารวมกันที่ปลายรอยร้าวโดยเอลิเมนต์เหล่านี้จะเรียงตัวอยู่รอบปลายรอยร้าว ถัดจากเอลิเมนต์ที่ปลายรอยร้าวจะเป็นเอลิเมนต์สามเหลี่ยมหกจุดต่อที่ไม่มีการเปลี่ยนแปลงใด ๆ กับเอลิเมนต์ ในวิทยานิพนธ์ฉบับนี้จะใช้ค่าอนุพันธ์อันดับสองของค่าความเค้นวอนมิสเซส (von Mises Stress) ซึ่งมีค่าเท่ากับค่าความเค้นประสิทธิผล (Effective Stress) ในการกำหนดขนาดของเอลิเมนต์ จากสมการ (ค.17) ในภาคผนวก ค ค่าความเค้นวอนมิสเซสจะแปรผันกับระยะทางจากปลายรอยร้าวเป็น

$$\sigma_e = \sigma_o\left(\frac{EJ}{\alpha\sigma_o^2 I_n r}\right)^{\frac{1}{n+1}}\tilde{\sigma}_e(n,\theta) \qquad (4.11)$$

ดังนั้นค่าอนุพันธ์อันดับสองของค่าความเค้นวอนมิสเซสเทียบกับระยะทางจากปลายรอยร้าวจะแปรผันกับระยะทางจากปลายรอยร้าวเป็น

$$\frac{\partial^2 \sigma_e}{\partial r^2} \rightarrow \left(\frac{1}{r}\right)^{\frac{2n+3}{n+1}} \quad (4.12)$$

ดังนั้นจากสมการ (4.12) ค่าอนุพันธ์อันดับสองของค่าความเค้นวอนมิสเซสจะมีค่าเข้าสู่อนันต์ที่บริเวณปลายรอบร้าวทำให้ในบริเวณนี้จะมีค่าความผิดพลาดสูงที่สุด ดังนั้นเอลิเมนต์ที่ใช้ในบริเวณนี้จะต้องมีขนาดเล็กที่สุดเสมอเมื่อเปรียบเทียบกับเอลิเมนต์ในบริเวณอื่นของปัญหา ดังนั้นในการปรับขนาดเอลิเมนต์ทั้งหมดเราจะพิจารณาเฉพาะเอลิเมนต์สามเหลี่ยมหกจุดต่อซึ่งอยู่ถัดจากเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อรอบปลายรอยร้าวเท่านั้นโดยกำหนดให้เอลิเมนต์ที่ปลายรอยร้าวเป็นเอลิเมนต์ที่มีขนาดเล็กที่สุด เพื่อให้การคำนวณค่าอนุพันธ์อันดับสองของค่าความเค้นวอนมิสเซสทำได้ง่ายขึ้นเราจะใช้เฉพาะค่าของผลเฉลยความเค้นวอนมิสเซสที่จุดต่อหลัก (Main Node) ซึ่งเป็นจุดต่อที่มุมของเอลิเมนต์สามเหลี่ยมหกจุดต่อและใช้การประมาณค่าความเค้นวอนมิสเซสบนเอลิเมนต์เป็นแบบระนาบเชิงเส้นสำหรับเอลิเมนต์สามเหลี่ยมสามจุดต่อ โดยค่าความเค้นที่จุดต่อทั้งสามจะได้จากค่าความเค้นที่จุดต่อหลักของเอลิเมนต์สามเหลี่ยมหกจุดต่อดังสมการ

$$\Phi(x, y) = N_1\Phi_1 + N_2\Phi_2 + N_3\Phi_3 \quad (4.12)$$

โดยที่ N_i แทนค่าฟังก์ชั่นการประมาณภายในเอลิเมนต์ของจุดต่อที่ i

สำหรับเอลิเมนต์สามเหลี่ยมสามจุดต่อฟังก์ชั่นการประมาณภายในเอลิเมนต์สามารถหาได้โดยง่ายจากความสัมพันธ์

$$N_i(x, y) = a_i + b_i x + c_i y \quad (4.13)$$

โดยที่

$$a_1 = (x_2y_3 - x_3y_2)/2A_e \quad b_1 = (y_2 - y_3)/2A_e \quad c_1 = (x_3 - x_2)/2A_e \quad (4.14\text{ก})$$

$$a_2 = (x_3y_1 - x_1y_3)/2A_e \quad b_2 = (y_3 - y_1)/2A_e \quad c_2 = (x_1 - x_3)/2A_e \quad (4.14\text{ข})$$

$$a_3 = (x_1y_2 - x_2y_1)/2A_e \quad b_3 = (y_1 - y_2)/2A_e \quad c_3 = (x_2 - x_1)/2A_e \quad (4.14\text{ค})$$

และ A_e คือพื้นที่ของเอลิเมนต์ที่พิจารณาโดยมีค่าเท่ากับ

$$A_e = \frac{1}{2}\left| x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \right| \tag{4.15}$$

เนื่องจากฟังก์ชั่นการประมาณภายในเอลิเมนต์ดังแสดงในสมการ (4.13) เป็นแบบระนาบเชิงเส้น ทำให้เราไม่สามารถหาค่าอนุพันธ์อันดับสองได้โดยตรง ดังนั้นในการหาค่าต่าง ๆ ของสมการที่ (4.9) ซึ่งอยู่ในรูปอนุพันธ์อันดับสองนั้นสามารถหาได้โดยทางอ้อม [16] โดยเริ่มจากการหาค่าอนุพันธ์อันดับหนึ่งของผลเฉลยเทียบกับแกน x และ y บนเอลิเมนต์ใด ๆ ได้เป็น

$$\frac{\partial \Phi_e}{\partial x} = \frac{\partial N_1}{\partial x}\Phi_1 + \frac{\partial N_2}{\partial x}\Phi_2 + \frac{\partial N_3}{\partial x}\Phi_3 \tag{4.16ก}$$

$$\frac{\partial \Phi_e}{\partial y} = \frac{\partial N_1}{\partial y}\Phi_1 + \frac{\partial N_2}{\partial y}\Phi_2 + \frac{\partial N_3}{\partial y}\Phi_3 \tag{4.16ข}$$

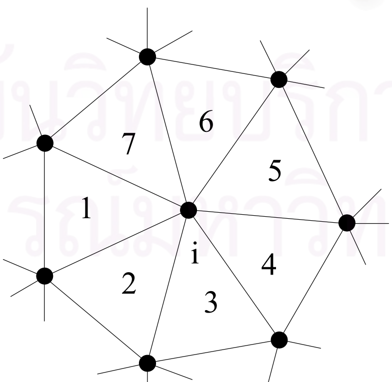
ซึ่งสำหรับเอลิเมนต์สามเหลี่ยมสามจุดต่อค่าที่ได้จะเท่ากันทั้งเอลิเมนต์ซึ่งทำให้ค่าที่ได้ที่จุดต่อร่วมของเอลิเมนต์ต่าง ๆ มีค่าไม่เท่ากัน ดังนั้นเราจึงต้องทำการหาค่าอนุพันธ์นี้บนเอลิเมนต์ต่าง ๆ แล้วทำการเฉลี่ยค่าที่จุดต่อร่วมโดยใช้สมการ

$$\bar{U}_i = \frac{\sum_{j=1}^{m} U_e^j}{m} \tag{4.17}$$

โดยที่ $\bar{U}_i$ แทนค่าเฉลี่ยของค่าความเค้นวอนมิสเซสที่จุดต่อ i

U_e^j แทนค่าความเค้นวอนมิสเซสที่จุดต่อ i ของเอลิเมนต์ที่ j ซึ่งมีจุดต่อ i เป็นจุดต่อร่วม

m แทนจำนวนเอลิเมนต์ทั้งหมดที่มีจุดต่อ i เป็นจุดต่อร่วม



รูปที่ 4.4 การเรียงตัวของเอลิเมนต์ล้อมรอบจุดต่อ i

ตัวอย่างเช่น ผลเฉลยของค่าอนุพันธ์อันดับหนึ่งที่จุดต่อ i ซึ่งเป็นจุดต่อร่วมของเอลิเมนต์ทั้งหมด 6 เอลิเมนต์ดังแสดงในรูปที่ 4.4 สามารถหาค่าได้จาก

$$\frac{\partial \bar{\Phi}_i}{\partial x} = \frac{\left(\frac{\partial \Phi_e}{\partial x}\right)_1 + \left(\frac{\partial \Phi_e}{\partial x}\right)_2 + \cdots + \left(\frac{\partial \Phi_e}{\partial x}\right)_6}{6} \quad (4.18)$$

เมื่อทำการเฉลี่ยค่าทั่วทั้งโดเมนแล้วเราจะได้ค่าอนุพันธ์อันดับหนึ่งที่จุดต่อทั่วทั้งโดเมนจากนั้นเราจะเริ่มทำการหาค่าอนุพันธ์อันดับสองโดยทำการประยุกต์การประมาณแบบระนาบเชิงเส้นเข้ากับค่าอนุพันธ์อันดับหนึ่งที่ได้จากการเฉลี่ยบนจุดต่อดังสมการเป็น

$$\frac{\partial \Phi}{\partial x}(x, y) = N_1 \frac{\partial \bar{\Phi}_1}{\partial x} + N_2 \frac{\partial \bar{\Phi}_2}{\partial x} + N_3 \frac{\partial \bar{\Phi}_3}{\partial x} \quad \text{(4.19ก)}$$

$$\frac{\partial \Phi}{\partial y}(x, y) = N_1 \frac{\partial \bar{\Phi}_1}{\partial y} + N_2 \frac{\partial \bar{\Phi}_2}{\partial y} + N_3 \frac{\partial \bar{\Phi}_3}{\partial y} \quad \text{(4.19ข)}$$

จะทำให้ได้ค่าอนุพันธ์อันดับที่สองเทียบกับแกน x และ y เป็น

$$\frac{\partial^2 \Phi_e}{\partial x^2} = \frac{\partial N_1}{\partial x}\frac{\partial \bar{\Phi}_1}{\partial x} + \frac{\partial N_2}{\partial x}\frac{\partial \bar{\Phi}_2}{\partial x} + \frac{\partial N_3}{\partial x}\frac{\partial \bar{\Phi}_3}{\partial x} \quad \text{(4.20ก)}$$

$$\frac{\partial^2 \Phi_e}{\partial y^2} = \frac{\partial N_1}{\partial y}\frac{\partial \bar{\Phi}_1}{\partial y} + \frac{\partial N_2}{\partial y}\frac{\partial \bar{\Phi}_2}{\partial y} + \frac{\partial N_3}{\partial y}\frac{\partial \bar{\Phi}_3}{\partial y} \quad \text{(4.20ข)}$$

$$\begin{aligned}\frac{\partial^2 \Phi_e}{\partial x \partial y} &= \frac{\partial N_1}{\partial y}\frac{\partial \bar{\Phi}_1}{\partial x} + \frac{\partial N_2}{\partial y}\frac{\partial \bar{\Phi}_2}{\partial x} + \frac{\partial N_3}{\partial y}\frac{\partial \bar{\Phi}_3}{\partial x} \\ &= \frac{\partial N_1}{\partial x}\frac{\partial \bar{\Phi}_1}{\partial y} + \frac{\partial N_2}{\partial x}\frac{\partial \bar{\Phi}_2}{\partial y} + \frac{\partial N_3}{\partial x}\frac{\partial \bar{\Phi}_3}{\partial y} \quad \text{(4.20ค)}\end{aligned}$$

จากสมการที่ (4.13) ค่าอนุพันธ์อันดับหนึ่งของฟังก์ชั่นการประมาณภายในจะมีค่าเป็น

$$\frac{\partial N_i}{\partial x} = b_i \quad \text{(4.21ก)}$$

$$\frac{\partial N_i}{\partial y} = c_i \quad \text{(4.21ข)}$$

ดังนั้นสมการ (4.16) และ (4.20) สามารถเขียนอยู่ในรูปอย่างง่ายได้เป็น

$$\frac{\partial \Phi_e}{\partial x} = b_1\Phi_1 + b_3\Phi_2 + b_3\Phi_3 \quad (4.22ก)$$

$$\frac{\partial \Phi_e}{\partial y} = c_1\Phi_1 + c_2\Phi_2 + c_3\Phi_3 \quad (4.22ข)$$

$$\frac{\partial^2 \Phi_e}{\partial x^2} = b_1\frac{\partial \bar{\Phi}_1}{\partial x} + b_2\frac{\partial \bar{\Phi}_2}{\partial x} + b_3\frac{\partial \bar{\Phi}_3}{\partial x} \quad (4.22ค)$$

$$\frac{\partial^2 \Phi_e}{\partial y^2} = c_1\frac{\partial \bar{\Phi}_1}{\partial y} + c_2\frac{\partial \bar{\Phi}_2}{\partial y} + c_3\frac{\partial \bar{\Phi}_3}{\partial y} \quad (4.22ง)$$

$$\begin{aligned}\frac{\partial^2 \Phi_e}{\partial x \partial y} &= c_1\frac{\partial \bar{\Phi}_1}{\partial x} + c_2\frac{\partial \bar{\Phi}_2}{\partial x} + c_3\frac{\partial \bar{\Phi}_3}{\partial x} \\ &= b_1\frac{\partial \bar{\Phi}_1}{\partial y} + b_2\frac{\partial \bar{\Phi}_2}{\partial y} + b_3\frac{\partial \bar{\Phi}_3}{\partial y} \quad (4.22จ)\end{aligned}$$

จากนั้นทำการเฉลี่ยค่าที่จุดต่อร่วมต่าง ๆ โดยใช้สมการที่ (4.17) ก็จะได้ค่าอนุพันธ์อันดับสองเทียบกับแกน x และ y ที่จุดต่อต่าง ๆ แทนค่าอนุพันธ์อันดับสองที่ได้ลงในสมการ (4.9) ก็จะได้ค่าเจาะจงทั้งสองค่าที่แต่ละจุดต่อทั้งหมดบนโดเมน เลือกค่าเจาะจงที่มีขนาดมากที่สุดเพื่อใช้กำหนดขนาดเอลิเมนต์โดยกำหนดให้

$$\lambda = \max\left(\left|\frac{\partial^2 \Phi_i}{\partial X^2}\right|, \left|\frac{\partial^2 \Phi_i}{\partial Y^2}\right|\right) \quad (4.23)$$

จากอนุกรมของเทย์เลอร์เราทราบว่าค่าความผิดพลาดจะสัมพันธ์กับค่าอนุพันธ์อันดับสองและขนาดของเอลิเมนต์ดังสมการ

$$\begin{aligned}O(h^2) &= \frac{1}{2}\left(\frac{\partial^2 \Phi}{\partial x^2}\right)_i (x - x_i)^2 + \left(\frac{\partial^2 \Phi}{\partial x \partial y}\right)_i (x - x_i)(y - y_i) \\ &\quad + \frac{1}{2}\left(\frac{\partial^2 \Phi}{\partial y^2}\right)_i (y - y_i)^2 + O(h^3) \quad (4.24)\end{aligned}$$

เนื่องจาก $(x - x_i)$ และ $(y - y_i)$ แปรผันกับขนาดของเอลิเมนต์ h ดังนั้นจะได้

$$O(h^2) \rightarrow \lambda h^2 \quad (4.25)$$

กำหนดให้ค่าความผิดพลาดในแต่ละเอลิเมนต์มีค่าคงที่ ดังนั้นเราสามารถคำนวณขนาดของเอลิเมนต์เปรียบเทียบกันในแต่ละบริเวณของปัญหาได้ในทำนองเดียวกันกับสมการ (4.25) เป็น

$$\lambda h^2 = \lambda_{max} h_{min}^2 = \text{constant} \tag{4.26}$$

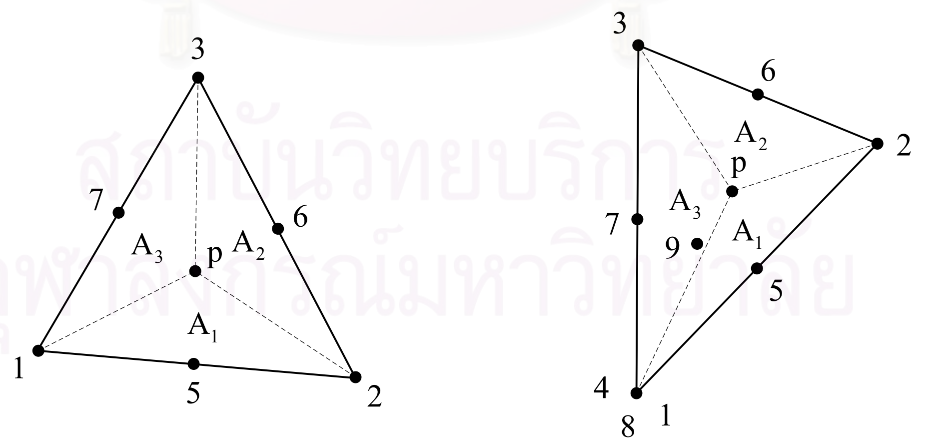
โดยที่ h_{min} แทนขนาดของเอลิเมนต์ที่เล็กที่สุดที่กำหนดโดยผู้ใช้

λ_{max} แทนขนาดของค่าเจาะจงที่มีค่ามากที่สุดในโดเมนปัญหา

4.2 การถ่ายทอดผลเฉลย

เพื่อให้การคำนวณสามารถดำเนินต่อไปได้ที่ค่าระดับภาระใดภาระหนึ่งหลังการปรับขนาดเอลิเมนต์เราจำเป็นต้องอาศัยการถ่ายทอดผลเฉลยการเคลื่อนตัวที่จุดต่อที่ได้จากโครงตาข่ายก่อนการปรับขนาดเอลิเมนต์มาสู่โครงตาข่ายใหม่หลังการปรับขนาดเอลิเมนต์ โดยในการถ่ายทอดผลเฉลยจะเริ่มจากการหาว่าจุดต่อทั้งหมดบนโครงตาข่ายใหม่อยู่บนเอลิเมนต์ใดในโครงตาข่ายเก่า จากนั้นจะใช้ค่าของผลเฉลยที่จุดต่อและฟังก์ชั่นการประมาณภายในเอลิเมนต์ของเอลิเมนต์นี้เพื่อถ่ายทอดค่าผลเฉลยที่ได้ไปไว้ที่จุดต่อบนโครงตาข่ายใหม่ โดยในวิทยานิพนธ์ฉบับนี้ได้แบ่งจุดต่อที่ต้องทำการถ่ายทอดผลเฉลยออกเป็นสองส่วนคือ จุดต่อที่ปลายรอยร้าวและจุดต่อที่บริเวณอื่นหรือจุดต่อที่ไม่ซ้อนกันโดยขั้นตอนการถ่ายทอดผลเฉลยมีรายละเอียดดังต่อไปนี้

ขั้นตอนแรกของการถ่ายทอดผลเฉลยจะเริ่มจากการหาว่าจุดต่อบนโครงตาข่ายใหม่อยู่บนเอลิเมนต์ใดในโครงตาข่ายเก่า [29] พิจารณาจุดต่อ p ใด ๆ ของโครงตาข่ายใหม่ที่ไม่ใช่จุดต่อที่ปลายรอยร้าว โดยจุดต่อ p นี้อยู่บนพื้นที่สามเหลี่ยมใด ๆ ซึ่งอาจเป็นเอลิเมนต์สามเหลี่ยมหกจุดต่อหรือเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อในโครงตาข่ายเก่าที่ด้าน ๆ หนึ่งของเอลิเมนต์ถูกยุบมารวมกันที่ปลายรอยร้าวดังแสดงในรูปที่ 4.5



(ก) เอลิเมนต์สามเหลี่ยมหกจุดต่อ (ข) เอลิเมนต์ที่ปลายรอยร้าว

รูปที่ 4.5 จุดต่อของโครงตาข่ายใหม่ p บนเอลิเมนต์ในโครงตาข่ายเก่า

พื้นที่สามเหลี่ยมย่อย A_1, A_2, A_3 ซึ่งเกิดจากจุดต่อ p กับด้านตรงข้ามที่เหลือของสามเหลี่ยม และพื้นที่สามเหลี่ยม A_e สามารถคำนวณได้จากค่าดีเทอร์มิแนนท์ (Determinant) ดังสมการ

$$A_1 = \frac{1}{2}\det\begin{vmatrix} 1 & x_p & y_p \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = 0.5\left\{x_2y_3 - x_3y_2 + \left(y_2 - y_3\right)x_p + \left(x_3 - x_2\right)y_p\right\} \quad (4.27\ ก)$$

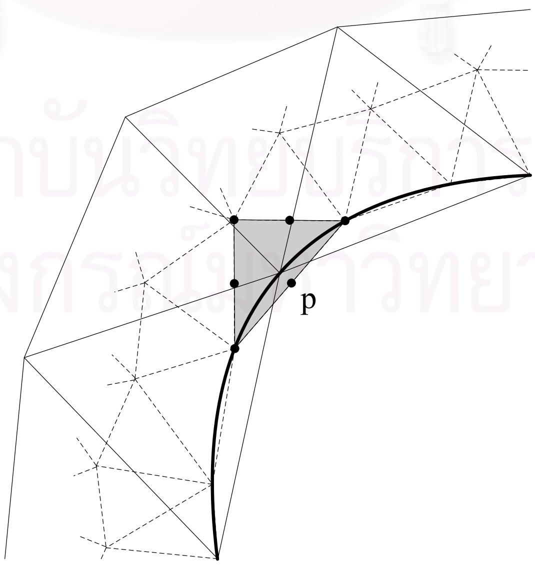
$$A_2 = \frac{1}{2}\det\begin{vmatrix} 1 & x_p & y_p \\ 1 & x_3 & y_3 \\ 1 & x_1 & y_1 \end{vmatrix} = 0.5\left\{x_3y_1 - x_1y_3 + \left(y_3 - y_1\right)x_p + \left(x_1 - x_3\right)y_p\right\} \quad (4.27\ ข)$$

$$A_3 = \frac{1}{2}\det\begin{vmatrix} 1 & x_p & y_p \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{vmatrix} = 0.5\left\{x_1y_2 - x_2y_1 + \left(y_1 - y_2\right)x_p + \left(x_2 - x_1\right)y_p\right\} \quad (4.27\ ค)$$

$$A_e = A_1 + A_2 + A_3 \quad (4.27\ ง)$$

โดยที่ $x_i, y_i,\ i = 1, 2, 3$ แทนค่าตำแหน่งพิกัดที่ปลายยอดสามเหลี่ยม

หากจุดต่อ p อยู่บนพื้นที่สามเหลี่ยมใดแล้วพื้นที่สามเหลี่ยมย่อย A_1, A_2 และ A_3 จะมีค่ามากกว่าหรือเท่ากับศูนย์เสมอ ในกรณีที่จุดต่อบนโครงตาข่ายใหม่ไม่ได้อยู่บนเอลิเมนต์ใดเลยในโครงตาข่ายเก่าซึ่งเป็นกรณีที่เกิดขึ้นได้เมื่อทำการสร้างโครงตาข่ายบนโดเมนปัญหาที่มีขอบเขต (Boundary) เป็นเส้นโค้งดังแสดงในรูปที่ 4.6



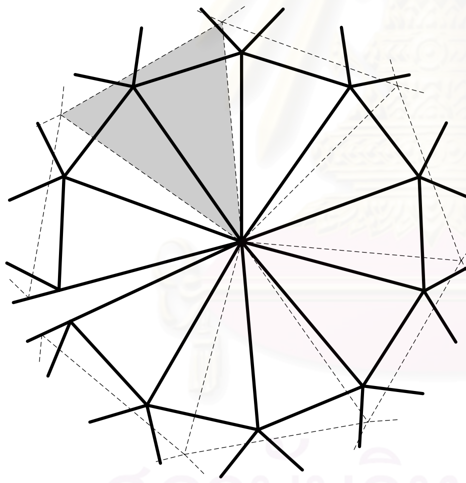
รูปที่ 4.6 การถ่ายทอดผลเฉลยที่ขอบเขตปัญหาเป็นเส้นโค้ง

การถ่ายทอดผลเฉลยสำหรับกรณีนี้จะใช้ผลเฉลยที่ได้จากเอลิเมนต์ในโครงตาข่ายเก่าที่อยู่ใกล้กับจุดต่อบนโครงตาข่ายใหม่นี้มากที่สุด โดยใช้อัตราส่วนผลต่างของพื้นที่ต่อพื้นที่เอลิเมนต์ซึ่งนิยามโดย

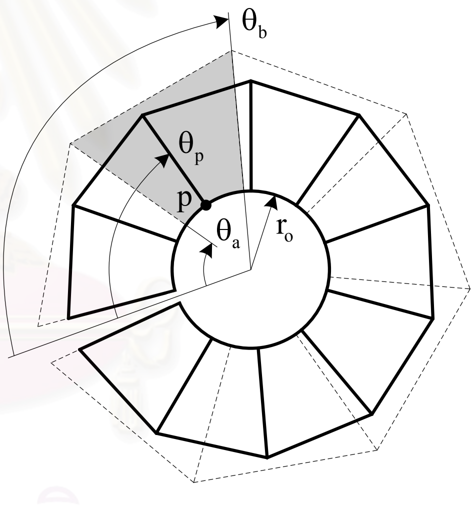
$$\frac{\Delta A}{A_e} = \frac{|A_1| + |A_2| + |A_3| - A_e}{2A_e} \quad (4.28)$$

ในการตรวจสอบ หากเอลิเมนต์ใดในโครงตาข่ายเก่ามีค่าอัตราส่วนพื้นที่นี้ต่ำที่สุดแล้วเอลิเมนต์นั้นก็จะเป็นเอลิเมนต์ที่อยู่ใกล้จุดต่อ p มากที่สุด

ในกรณีที่จุดต่อ p เป็นจุดต่อที่ปลายรอยร้าวนั้น วิธีการหาว่าจุดต่อ p อยู่บนเอลิเมนต์ใดในโครงตาข่ายเก่าอย่างที่ผ่านมานั้นจะไม่สามารถนำมาใช้ได้ในกรณีนี้เนื่องจากมีจุดต่อซ้อนกันอยู่หลายจุดที่ตำแหน่งเดียวกันที่ปลายรอยร้าว ดังนั้นจึงต้องหาวิธีใหม่ในการระบุเอลิเมนต์ที่จุดต่อ p นั้นอยู่ พิจารณาเอลิเมนต์ที่ปลายรอยร้าวของโครงตาข่ายใหม่และเก่าดังแสดงในรูปที่ 4.7ก โดยที่เส้นประแสดงขอบของเอลิเมนต์ในโครงตาข่ายเก่า ขณะที่เส้นเต็มแสดงขอบของเอลิเมนต์ในโครงตาข่ายใหม่



(ก)



(ข)

รูปที่ 4.7 การหาว่าจุดต่อที่ปลายรอยร้าวของเอลิเมนต์ในโครงต่ข่ายใหม่อยู่บนเอลิเมนต์ที่ปลายรอยร้าวใดในโครงตาข่ายเก่า (ก) เอลิเมนต์ที่ปลายรอยร้าวของโครงตาข่ายใหม่และโครงตาข่ายเก่าและ (ข) นิยามของมุมต่าง ๆ ที่ใช้ในการตรวจสอบ

จากรูปหากจุดต่อทั้งหมดที่ถูกยุบมารวมกันที่ปลายรอยร้าวของเอลิเมนต์ทั้งจากโครงตาข่ายใหม่และเก่าถูกเลื่อนออกจากตำแหน่งปลายรอยร้าวเป็นระยะ r_o ตามแนวรัศมีแล้ว เราสามารถตรวจสอบได้ว่าจุดต่อ p อยู่บนเอลิเมนต์ใดในโครงตาข่ายเก่าโดยพิจารณามุมของจุดต่อบนโครงตา

ข่ายใหม่ที่กระทำกับระนาบรอยร้าว θ_p ดังแสดงในรูปที่ 4.7ข ว่าอยู่ระหว่างมุมที่จุดต่อหลักสองจุดของด้านที่ถูกยุบมารวมกันที่ปลายรอยร้าวของเอลิเมนต์ใดในโครงตาข่ายเก่าโดยใช้สมการ

$$\left[\theta_a \ \le \ \theta_p \ \le \ \theta_b\right] \quad (4.29)$$

โดยที่ θ_p แทนมุมที่จุดต่อ p บนโครงตาข่ายใหม่วัดจากระนาบรอยร้าว

θ_a แทนมุมที่จุดต่อหลักจุดแรกของด้านที่ถูกยุบมารวมกันที่ปลายรอยร้าวของเอลิเมนต์ที่ปลายรอยร้าวในโครงตาข่ายเก่า

θ_b แทนมุมที่จุดต่อหลักถัดไปของด้านที่ถูกยุบมารวมกันที่ปลายรอยร้าวของเอลิเมนต์ที่ปลายรอยร้าวในโครงตาข่ายเก่า

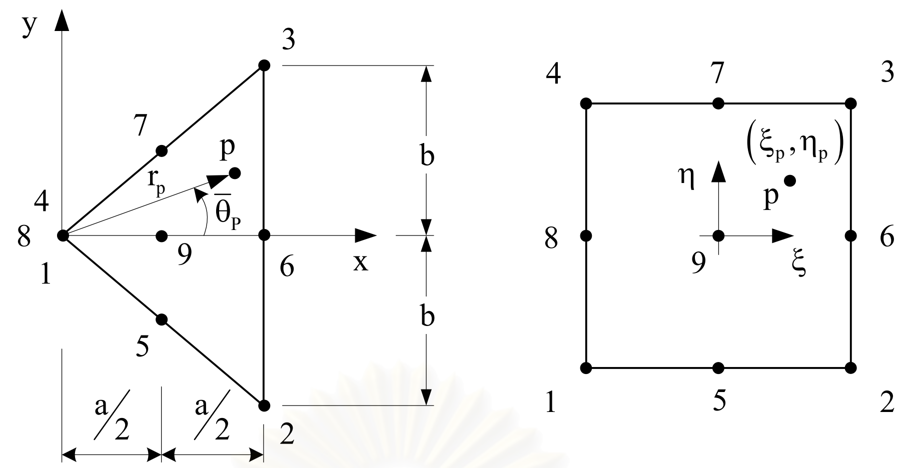
เมื่อได้เอลิเมนต์ที่อยู่ใกล้จุดต่อ p มากที่สุดหรือเอลิเมนต์ที่จุดต่อ p อยู่ข้างในแล้วขั้นตอนต่อไปของการถ่ายทอดผลเฉลยจะเป็นการหาฟังก์ชั่นการประมาณภายในเอลิเมนต์ของเอลิเมนต์ที่ได้ที่สอดคล้องกับจุดต่อ p โดยหากเอลิเมนต์ที่ได้นั้นเป็นเอลิเมนต์สามเหลี่ยมหกจุดต่อแล้ว ฟังก์ชั่นการประมาณภายในเอลิเมนต์ที่สอดคล้องกับจุดต่อ p ของเอลิเมนต์ชนิดนี้จะสามารถหาได้โดยง่ายหากฟังก์ชั่นการประมาณภายในเอลิเมนต์ถูกเขียนอยู่ในรูปพิกัดพื้นที่ (Area Coordinates) เป็น [25]

$$\begin{aligned} N_1 &= (2L_1-1)L_1 \\ N_2 &= (2L_2-1)L_2 \\ N_3 &= (2L_3-1)L_3 \\ N_4 &= 4L_1L_2 \\ N_5 &= 4L_2L_3 \\ N_6 &= 4L_3L_1 \end{aligned} \quad (4.30)$$

โดยนิยามของพิกัดพื้นที่มีดังนี้

$$L_1 = \frac{A_1}{A_e}, \quad L_2 = \frac{A_2}{A_e}, \quad L_3 = \frac{A_3}{A_e} \quad (4.31)$$

ในกรณีที่เอลิเมนต์ที่ได้นั้นเป็นเอลิเมนต์ที่ปลายรอยร้าวเราจะเริ่มจากการคำนวณหาค่าพิกัดธรรมชาติของเอลิเมนต์ในโครงตาข่ายเก่านี้ที่สอดคล้องกับจุดต่อ p พิจารณารูปที่ 4.8 ซึ่งแสดงความสัมพันธ์ระหว่างค่ามุมและระยะจากปลายรอยร้าวของจุดต่อ p ซึ่งวัดเทียบกับเอลิเมนต์ที่ปลายรอยร้าวในโครงตาข่ายเก่า



รูปที่ 4.8 เอลิเมนต์ที่ปลายรอยร้าวในโครงตาข่ายเก่าที่บรรจุจุดต่อ p (ก) บนพิกัด x-y และ (ข) บนพิกัดธรรมชาติ ξ-η

ในทำนองเดียวกันกับสมการ (ง.6) ในภาคผนวก ง ความสัมพันธ์ระหว่างระยะในแนวรัศมีจากปลายรอยร้าว r_p และมุมที่วัดเทียบกับแกนกลางของเอลิเมนต์ที่ปลายรอยร้าวในโครงตาข่ายเก่า $\bar{\theta}_p$ ดังแสดงในรูปที่ 4.8 กับพิกัดธรรมชาติ (ξ_p, η_p) ที่สอดคล้องกับจุดต่อ p สำหรับเอลิเมนต์ที่ปลายรอยร้าวซึ่งจุดต่อตรงกลางด้านและจุดต่อตรงกลางเอลิเมนต์ยังคงอยู่ที่ตำแหน่งกึ่งกลางนั้นจะเท่ากับ

$$\begin{aligned} \xi_p &= \frac{r_p}{\frac{a}{2}\sqrt{1+\tan^2\bar{\theta}_p}} - 1 \\ \eta_p &= \frac{a}{b}\tan\bar{\theta}_p \end{aligned} \qquad (4.32)$$

ในกรณีที่จุดต่อ p เป็นจุดต่อที่ปลายรอยร้าวซึ่งค่าระยะในแนวรัศมีจากปลายรอยร้าว r_p มีค่าเท่ากับศูนย์แล้ว พิกัดธรรมชาติที่สอดคล้องกับจุดต่อ p จะสามารถหาได้จากการพิจารณารูปที่ 4.7ข และสมการ (4.32) เป็น

$$\begin{aligned} \xi_p &= -1 \\ \eta_p &= \frac{\tan\left[\theta_p - \left(\frac{\theta_b+\theta_a}{2}\right)\right]}{\tan\left(\frac{\theta_b-\theta_a}{2}\right)} \end{aligned} \qquad (4.33)$$

เมื่อได้ค่าพิกัดธรรมชาติที่สอดคล้องกับจุดต่อ p สำหรับเอลิเมนต์ที่ปลายรอยร้าวในโครงตาข่ายเก่าแล้ว ฟังก์ชั่นการประมาณภายในเอลิเมนต์ที่สอดคล้องกับจุดต่อนี้จะสามารถหาได้โดยการแทนค่าพิกัดธรรมชาติที่ได้นี้ลงในสมการ (3.30)

เมื่อได้ค่าฟังก์ชั่นการประมาณภายในของเอลิเมนต์ที่ต้องการแล้ว ขั้นตอนต่อไปจะเป็นการคำนวณค่าการเคลื่อนตัวของเอลิเมนต์ในโครงตาข่ายเก่าที่สอดคล้องกับจุดต่อ p เพื่อใช้ถ่ายทอดกลับไปยังจุดต่อ p ในโครงตาข่ายใหม่โดยใช้สมการ

$$\Phi_p = \sum_{i=1}^{n} N_i \Phi_i \quad (4.34)$$

โดยที่ $n = 6$ สำหรับเอลิเมนต์ในโครงตาข่ายเก่าแบบสามเหลี่ยมหกจุดต่อ

$n = 9$ สำหรับเอลิเมนต์ที่ปลายรอยร้าวในโครงตาข่ายเก่า

Φ_p แทนค่าการเคลื่อนตัวของจุดต่อ p บนโครงตาข่ายใหม่

Φ_i แทนค่าการเคลื่อนตัวของจุดต่อต่าง ๆ สำหรับเอลิเมนต์ในโครงตาข่ายเก่า

N_i แทนค่าฟังก์ชั่นการประมาณภายในเอลิเมนต์ที่สอดคล้องกับจุดต่อ p ของเอลิเมนต์ในโครงตาข่ายเก่า

4.3 บทสรุป

ในบทนี้หลักการของเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติและเทคนิคการถ่ายทอดผลเฉลยระยะการเคลื่อนตัวจากโครงตาข่ายก่อนการปรับขนาดเอลิเมนต์ไปสู่โครงตาข่ายที่ทำการปรับขนาดเอลิเมนต์แล้วพร้อมทั้งการนำไปประยุกต์เข้ากับระเบียบวิธีไฟไนต์เอลิเมนต์ได้ถูกอธิบายอย่างละเอียดโดยเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติจะใช้ค่าอนุพันธ์อันดับสองของค่าความเค้นวอนมิสเซสในการปรับขนาด เทคนิคการถ่ายทอดผลเฉลยระยะการเคลื่อนตัวใช้การถ่ายทอดผลเฉลยโดยใช้ฟังก์ชันการประมาณภายในเอลิเมนต์และผลเฉลยที่จุดต่อจากโครงตาข่ายก่อนการปรับขนาดเอลิเมนต์ไปสู่จุดต่อบนโครงตาข่ายที่ทำการปรับขนาดเอลิเมนต์โดยอัตโนมัติแล้ว ขั้นตอนการทำงานและลักษณะโครงสร้างของไฟล์นำเข้าเพื่อใช้กับโปรแกรม JFACTOR ที่ได้ประดิษฐ์ขึ้นได้ถูกอธิบายอย่างละเอียดในบทต่อไป

บทที่ 5

โปรแกรมคอมพิวเตอร์สำหรับคำนวณค่าพารามิเตอร์เจอินทิกรัล

ในบทนี้จะกล่าวถึงขั้นตอนการทำงานของโปรแกรมคอมพิวเตอร์ที่ได้ประดิษฐ์ขึ้นจากสมการไฟไนต์เอลิเมนต์ดังที่ได้อธิบายไปแล้วในบทที่ผ่านมา โดยตัวโปรแกรมคอมพิวเตอร์จะถูกเขียนขึ้นจากภาษาฟอร์แทรน (FORTRAN) โดยสามารถนำไปใช้งานได้โดยตรงกับเครื่องคอมพิวเตอร์ส่วนบุคคล โปรแกรมที่ประดิษฐ์ขึ้นนี้มีชื่อว่า JFACTOR โดยมีรายละเอียดดังต่อไปนี้

5.1 ลักษณะของโปรแกรม JFACTOR

โปรแกรมคอมพิวเตอร์ JFACTOR ประกอบด้วยโปรแกรมหลัก (Main Program) และ 27 โปรแกรมย่อย (Subroutine) โดยมีขั้นตอนการทำงานของโปรแกรมดังนี้

5.1.1 เริ่มต้นการทำงานโดยอ่านไฟล์ข้อมูลนำเข้าของปัญหา (Input File) ซึ่งประกอบด้วยข้อมูลที่จำเป็นต่าง ๆ ที่ต้องใช้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลโดยเรียกโปรแกรมย่อย [SUBROUTINE INPUT] ซึ่งในโปรแกรมย่อยนี้ยังประกอบด้วยโปรแกรมย่อย [SUBROUTINE CRACKFACE] ซึ่งทำหน้าที่แปลงเอลิเมนต์รอบปลายรอยร้าวจากเอลิเมนต์สามเหลี่ยมหกจุดต่อไปเป็นเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อ โดยในโปรแกรมย่อยนี้ยังประกอบด้วยโปรแกรมย่อย [SUBROUTINE CENTER] ซึ่งทำหน้าที่หาการเรียงตัวของเอลิเมนต์รอบปลายรอยร้าวเพื่อใช้ในการถ่ายทอดผลเฉลยของจุดต่อที่ปลายรอยร้าวและหาการเรียงตัวของจุดต่อที่ผิวรอยร้าวเพื่อใช้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลซึ่งเกิดจากพจน์ค่าความเค้นดึงที่ผิวรอยร้าว นอกจากนี้ในโปรแกรมย่อย [SUBROUTINE INPUT] ยังประกอบด้วยโปรแกรมย่อย [SUBROUTINE FTHETA] และ [SUBROUTINE XYLOCAL] ซึ่งทำหน้าที่คำนวณค่ามุมของจุดต่อที่ขอบนอกของเอลิเมนต์ที่ปลายรอยร้าวทั้งหมดเทียบกับระนาบรอยร้าวเพื่อใช้ในการถ่ายทอดผลเฉลยค่าการเคลื่อนตัวของจุดต่อที่ปลายรอยร้าว และ [SUBROUTINE TemBDfFUNC] ซึ่งใช้ในการกำหนดค่าอุณหภูมิและค่าแรงวัตถุที่จุดต่อในกรณีที่ผู้ใช้ไม่ต้องการจะกำหนดโดยตรงลงในไฟล์ข้อมูลนำเข้า

5.1.2 คำนวณเอลิเมนต์เมตริกซ์ต่าง ๆ โดยเรียกโปรแกรมย่อย [SUBROUTINE LST] ซึ่งประกอบไปด้วยโปรแกรมย่อย [SUBROUTINE VOLUMETRIC] และ [SUBROUTINE DEVIATORIC] ซึ่งทำหน้าที่คำนวณเอลิเมนต์เมตริกซ์ต่าง ๆ ที่เกี่ยวข้องกับพจน์ความเค้นตั้งฉากเฉลี่ยและพจน์ความเค้นดิเวียทอริก ตามลำดับโดยใช้จำนวนจุดเกาส์ที่แตกต่างกัน โปรแกรมย่อยทั้งสองนี้ยังประกอบไปด้วยโปรแกรมย่อยต่าง ๆ ซึ่งมีหน้าที่ดังนี้ โปรแกรมย่อย [SUBROUTINE BJ9] ทำหน้าที่คำนวณเมตริกซ์ความสัมพันธ์ระหว่างค่าความเครียดและค่าการเคลื่อนตัวที่จุดต่อและค่าดีเทอร์มิแนนท์ของจาโคเบียนเมตริกซ์ โปรแกรมย่อย [SUBROUTINE GVALUE] ทำหน้าที่คำนวณค่าของตัวแปรใด ๆ ณ ตำแหน่งที่กำหนดในเอลิเมนต์ โปรแกรมย่อย [SUBROUTINE FINDSTRSS] ทำหน้าที่คำนวณค่าความเค้นประสิทธิผลจากค่าความเครียดและค่าอุณหภูมิที่จุดเกาส์ และโปรแกรมย่อย [SUBROUTINE ASSMBLE] ทำหน้าที่ประกอบเมตริกซ์ต่าง ๆ ที่คำนวณได้ในแต่ละเอลิเมนต์เข้าเป็นเมตริกซ์ระบบรวม

5.1.3 กำหนดเงื่อนไขขอบเขตของปัญหา เช่น บางจุดต่ออาจถูกกำหนดให้ไม่มีการเคลื่อนที่หรือถูกกำหนดให้มีค่าระยะการเคลื่อนตัวเท่ากับค่าใดค่าหนึ่ง หรือบางจุดต่ออาจถูกกำหนดให้มีแรงภายนอกมากระทำ โดยเรียกโปรแกรมย่อย [SUBROUTINE APPLYBC]

5.1.4 แก้ระบบสมการรวมเพื่อหาค่าระยะการเคลื่อนตัวที่เพิ่มขึ้นจากค่าภาระที่เพิ่มขึ้น โดยเริ่มจากการประยุกต์การแยกตัวประกอบแบบเคราท์ (Crout Factorization) เข้ากับระบบสมการเพื่อให้ได้เมตริกซ์สามเหลี่ยมด้านล่างและเมตริกซ์ในแนวทแยงมุมโดยใช้โปรแกรมย่อย [SUBROUTINE CROUT] จากนั้นแก้ระบบสมการโดยเรียกโปรแกรมย่อย [SUBROUTINE SOLVE]

5.1.5 ประยุกต์วิธีการทำซ้ำเข้ากับระบบสมการจนกระทั่งได้ค่าการเคลื่อนตัวที่จุดต่อซึ่งมีความถูกต้องอยู่ในระดับที่กำหนดด้วยโปรแกรมย่อย [SUBROUTINE ITER]

5.1.6 คำนวณค่าความเค้นที่จุดต่อโดยเรียกโปรแกรมย่อย [SUBROUTINE GAUSSNODE] ซึ่งค่าความเค้นที่ได้จะแยกการคำนวณออกเป็นสองส่วนโดยค่าความเค้นตั้งฉากเฉลี่ยจะคำนวณด้วยโปรแกรมย่อย [SUBROUTINE VOLSTRESS] และค่าความเค้นดิเวียทอริกจะคำนวณด้วยโปรแกรมย่อย [SUBROUTINE DEVSTRESS] โดยเริ่มคำนวณค่าความเค้นบนเอลิเมนต์ที่จุดเกาส์ต่าง ๆ จากนั้นจึงแปลงค่าความเค้นที่ได้ไปอยู่ที่จุดต่อโดยใช้โปรแกรมย่อย

[SUBROUTINE TRMAT] ในการกำหนดเมตริกซ์ที่ใช้แปลงค่าความเค้นจากจุดเกาส์ไปที่จุดต่อ

5.1.7 คำนวณค่าพารามิเตอร์เจอินทิกรัลบนโดเมนต่าง ๆ โดยเรียกโปรแกรมย่อย [SUBROUTINE CJINT] โดยแยกการคำนวณออกเป็นสามส่วนคือคำนวณค่าพารามิเตอร์เจอินทิกรัลจากพจน์ที่เกี่ยวข้องกับค่าความเค้นตั้งฉากเฉลี่ยด้วยโปรแกรมย่อย [SUBROUTINE CJVOL] คำนวณค่าพารามิเตอร์เจอินทิกรัลจากพจน์ที่เกี่ยวข้องกับค่าความเค้นดิเวียทอริกด้วยโปรแกรมย่อย [SUBROUTINE CJDEV] และคำนวณค่าพารามิเตอร์เจอินทิกรัลจากพจน์ที่เกี่ยวข้องกับค่าความเค้นดึงที่ผิวด้วยโปรแกรมย่อย [SUBROUTINE CFACE]

5.1.8 พิมพ์ค่าระยะการเคลื่อนตัวที่จุดต่อ ค่าความเค้นที่จุดต่อ และค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้ ณ สถานะการคำนวณต่าง ๆ ลงในไฟล์ผลลัพธ์เพื่อใช้ในการปรับขนาดเอลิเมนต์หรือแสดงผลด้วยโปรแกรมกราฟฟิกต่าง ๆ บนจอคอมพิวเตอร์ [MAIN PROGRAM] โดยโปรแกรมย่อย [SUBROUTINE GETSTRING] ทำหน้าที่แปลงตัวเลขแสดงครั้งที่ทำการคำนวณเป็นตัวอักษรเพื่อใช้ในการตั้งชื่อไฟล์ผลลัพธ์

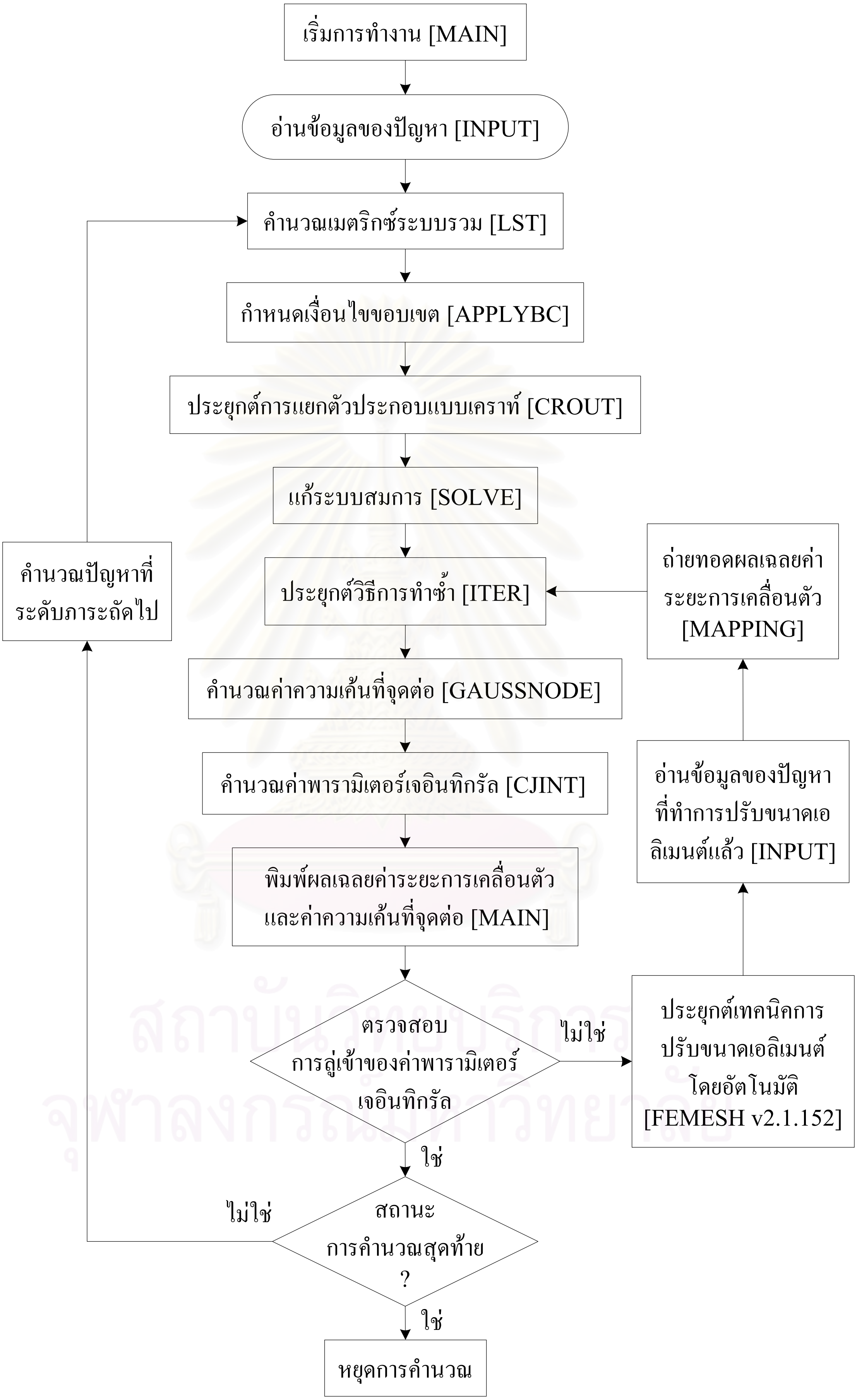
5.1.9 ในกรณีที่มีการปรับขนาดเอลิเมนต์โปรแกรมจะอ่านข้อมูลต่าง ๆ ของแบบจำลองไฟไนต์เอลิเมนต์ที่ทำการปรับขนาดเอลิเมนต์แล้วด้วยโปรแกรม FEMESH v2.1.152 โดยเรียกโปรแกรมย่อย [SUBROUTINE INPUT] ขึ้นมาอีกครั้งหนึ่ง จากนั้นทำการถ่ายทอดผลเฉลยของค่าการเคลื่อนตัวที่จุดต่อต่าง ๆ ที่ได้จากการคำนวณในโครงตาข่ายเก่าไปสู่จุดต่อต่าง ๆ ในโครงตาข่ายใหม่โดยเรียกโปรแกรมย่อย [SUBROUTINE MAPPING] ซึ่งในโปรแกรมย่อยนี้ยังประกอบไปด้วยโปรแกรมย่อย

5.1.10 ประยุกต์วิธีการทำซ้ำเข้ากับผลเฉลยค่าการเคลื่อนตัวที่จุดต่อที่ถูกถ่ายทอดมาเพื่อให้ได้ผลเฉลยใหม่ที่ทำให้ระบบสมการของแบบจำลองไฟไนต์เอลิเมนต์ที่ทำการปรับขนาดแล้วนี้สมดุลย์โดยเรียกโปรแกรมย่อย [SUBROUTINE ITER]

5.1.11 คำนวณค่าความเค้นที่จุดต่อและค่าพารามิเตอร์เจอินทิกรัลบนโดเมนต่าง ๆ สำหรับแบบจำลองไฟไนต์เอลิเมนต์ที่ทำการปรับขนาดเอลิเมนต์แล้วนี้โดยเรียกโปรแกรมย่อย [SUBROUTINE GAUSSNODE] และ [SUBROUTINE CJINT] ตามลำดับ

5.1.12 เมื่อได้ผลเฉลยค่าพารามิเตอร์เจอินทิกรัลที่ลู่เข้าแล้วจึงเริ่มกระบวนการทั้งหมดใหม่ที่ค่าระดับภาระหรือค่าระยะการเคลื่อนตัวที่ถูกกำหนดในสถานะการคำนวณถัดไป

ลำดับขั้นตอนการทำงานทั้งหมดของโปรแกรม JFACTOR สามารถสรุปได้ดังแสดงในรูปที่ 5.1



รูปที่ 5.1 ลำดับขั้นตอนการทำงานของโปรแกรม JFACTOR

5.2 รายละเอียดของโปรแกรม JFACTOR

รายละเอียดทั้งหมดของโปรแกรมคอมพิวเตอร์ JFACTOR ได้แสดงไว้ในภาคผนวก จ ตอนท้ายของวิทยานิพนธ์นี้

5.3 ลักษณะของไฟล์ข้อมูลนำเข้า

ลักษณะของไฟล์ข้อมูลนำเข้าที่ใช้กับโปรแกรม JFACTOR จะประกอบไปด้วยส่วนต่าง ๆ ทั้งหมด 6 ส่วนดังนี้

ส่วนที่ 1 ประโยคอธิบายกำกับลักษณะของไฟล์

บรรทัดแรก ตัวเลขระบุจำนวนบรรทัดที่ใช้อธิบายลักษณะของไฟล์

บรรทัดต่อไป ประโยคอธิบายลักษณะของไฟล์ข้อมูลนำเข้าจำนวนบรรทัดเท่ากับที่ระบุไว้ในบรรทัดแรก

ตัวอย่างเช่น 3

Single Edge Notched Tension specimen, SENT
Crack length per width ratio is 0.5
Incremental load factor is 0.25

ส่วนที่ 2 ความต้องการการปรับขนาดเอลิเมนต์และชนิดของปัญหา

บรรทัดแรก คำระบุรหัสความต้องการการปรับขนาดเอลิเมนต์ ชนิดของปัญหา และตำแหน่งแกนสมมาตร

บรรทัดที่สอง รหัสตัวเลขแสดงความต้องการการปรับขนาดเอลิเมนต์สำหรับไฟล์ข้อมูลนำเข้านี้ รหัสตัวเลขระบุชนิดของปัญหา และตำแหน่งที่แกนสมมาตรตัดกับแกน x เฉพาะในกรณีชนิดของปัญหาเป็นปัญหาสมมาตรรอบแกน

ตัวอย่างเช่น	INOADAPT	IPLANE	XSHIFT
1		3	30.0

หมายเหตุ : รหัสตัวเลขต่าง ๆ มีความหมายดังนี้

INOADAPT = 1 ไม่มีการปรับขนาดเอลิเมนต์ในการวิเคราะห์ปัญหานี้ ดังนั้นใช้เฉพาะไฟล์ข้อมูลนำเข้านี้ตลอดการคำนวณในทุกสถานะการคำนวณ

INOADAPT = 0 มีการปรับขนาดเอลิเมนต์ในการวิเคราะห์ปัญหานี้ ดังนั้นในระหว่างการวิเคราะห์ปัญหานี้อาจต้องมี

การอ่านไฟล์ข้อมูลนำเข้าที่ทำการปรับขนาดเอลิเมนต์แล้วใหม่

IPLANE = 1 ปัญหาความเค้นในระนาบ

IPLANE = 2 ปัญหาความเครียดในระนาบ

IPLANE = 3 ปัญหาสมมาตรรอบแกน ในกรณีนี้ต้องกำหนดค่า XSHIFT ซึ่งเป็นค่าพิกัดในแนวแกน x ที่แกนสมมาตรตัดกับแกน x ด้วย

ส่วนที่ 3 ขนาดของปัญหา

บรรทัดแรก คำระบุจำนวนจุดต่อ เอลิเมนต์และจุดต่อที่มีการกำหนดเงื่อนไขขอบเขต

บรรทัดที่สอง ตัวเลขระบุจำนวนจุดต่อ เอลิเมนต์และจุดต่อที่มีการกำหนดเงื่อนไขขอบเขต

ตัวอย่างเช่น NPOIN NELEM NPBC
720 331 25

ส่วนที่ 4 วิธีการทำซ้ำและค่าความผิดพลาดที่ยอมรับได้

บรรทัดแรก คำระบุรหัสวิธีการทำซ้ำและค่าความผิดพลาดที่ยอมรับได้

บรรทัดที่สอง รหัสตัวเลขระบุวิธีการทำซ้ำและตัวเลขระบุค่าความผิดพลาดที่ยอมรับได้

ตัวอย่างเช่น ITERTY BETOK
1 5.E-8

หมายเหตุ : รหัสตัวเลขระบุวิธีการทำซ้ำมีความหมายดังนี้

ITERTY = 1 ใช้วิธีการทำซ้ำแบบนิวตัน-ราฟสัน (Full Newton-Raphson Method)

IT ERTY = 2 ใช้วิธีการทำซ้ำแบบนิวตัน-ราฟสันที่ถูกดัดแปลง (Modified Newton-Raphson Method)

ส่วนที่ 5 ลักษณะของรอยร้าวและเอลิเมนต์ที่รอบปลายรอยร้าว

บรรทัดแรก คำระบุลักษณะของรอยร้าวและเอลิเมนต์ที่ปลายรอยร้าว

บรรทัดที่สอง หมายเลขจุดต่อที่ปลายรอยร้าว ตัวเลขระบุค่ามุมที่ระนาบรอยร้าวกระทำกับแกน x รหัสตัวเลขระบุความต้องการให้โปรแกรมเปลี่ยนข้อมูลนำเข้าของเอลิเมนต์ที่ปลายรอยร้าวจากเอลิเมนต์สามเหลี่ยมหกจุดต่อไปเป็นเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อ รหัสตัวเลขระบุว่าที่ผิวรอยร้าวมีค่าความเค้นดึงมากระทำที่ผิวหรือไม่และจำนวนผิวรอยร้าวที่มีในแบบจำลอง

บรรทัดที่สาม คำระบุลักษณะของแรงที่กระทำกับจุดต่อบนผิวรอยร้าว (เฉพาะกรณีที่ผิวของรอยร้าวมีค่าความเค้นดึงมากระทำเท่านั้น)

บรรทัดต่อไป (เฉพาะกรณีที่ผิวของรอยร้าวมีค่าความเค้นดึงมากระทำเท่านั้น) รหัสตัวเลขผิวรอยร้าว แรงที่กระทำกับจุดต่อที่ปลายรอยร้าวในทิศ x และ y ซึ่งได้จากการอินทิเกรตค่าความเค้นดึงที่ผิวซึ่งกระทำกับผิวรอยร้าวของเอลิเมนต์ที่ปลายรอยร้าวที่สอดคล้องกับผิวรอยร้าวนั้น ๆ และแรงที่กระทำกับจุดต่อที่อยู่ไกลที่สุดบนผิวรอยร้าวในทิศ x และ y ซึ่งได้จากการอินทิเกรตค่าความเค้นดึงที่ผิวซึ่งกระทำกับเอลิเมนต์ที่ผิวรอยร้าวที่อยู่ไกลที่สุดซึ่งอยู่ในโดเมนที่ทำการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัล (แรงที่ได้นี้ไม่ใช่แรงลัพธ์ที่ได้จากการรวมแรงทั้งหมดที่กระทำกับจุดต่อทั้งสองนี้จากเอลิเมนต์ต่าง ๆ ที่มีจุดต่อทั้งสองเป็นจุดต่อร่วม แต่เป็นแรงที่คิดจากค่าความเค้นดึงที่ผิวเฉพาะที่กระทำกับเอลิเมนต์ที่ผิวรอยร้าวทั้งสองที่กล่าวถึงข้างต้นนี้เท่านั้น โดยแรงที่ต้องกำหนดในไฟล์ข้อมูลนำเข้านี้จะถูกนำมาใช้ในการแปลงแรงที่จุดต่อบนผิวรอยร้าวกลับเป็นค่าความเค้นดึงที่ผิวเพื่อใช้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลที่เกิดจากพจน์ค่าความเค้นดึงที่ผิวอีกทีหนึ่ง)

ตัวอย่างเช่น	NODEK1	CANGLE	ICTETRAN	ICFLOAD	NFACE
287		25.0	1	1	2

	IFACE	CTQFIX	CTQFIY	FDQFIX	FDQFIY
1		0.0	+10.0	0.0	+10.0
2		0.0	–10.0	0.0	–10.0

หมายเหตุ : รหัสตัวเลขต่าง ๆ มีความหมายดังนี้

ICTETRAN = 1 แปลงข้อมูลนำเข้าของเอลิเมนต์ที่ปลายรอยร้าวจากเอลิเมนต์สามเหลี่ยมหกจุดต่อไปเป็นเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อ

ICTETRAN = 0 ไม่ต้องแปลงข้อมูลนำเข้า ในกรณีนี้จะใช้ข้อมูลนำเข้าเดิมสำหรับเอลิเมนต์สามเหลี่ยมหกจุดต่อ

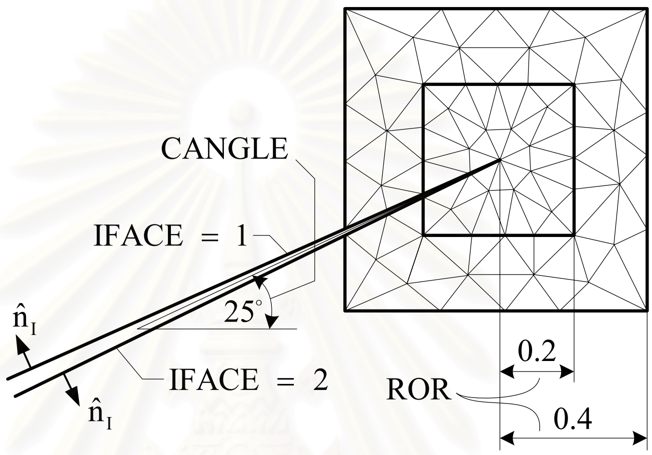
ICFLOAD = 1 ผิวรอยร้าวมีค่าความเค้นดึงมากระทำ

ICFLOAD = 0 ไม่มีค่าความเค้นดึงมากระทำที่ผิวรอยร้าว

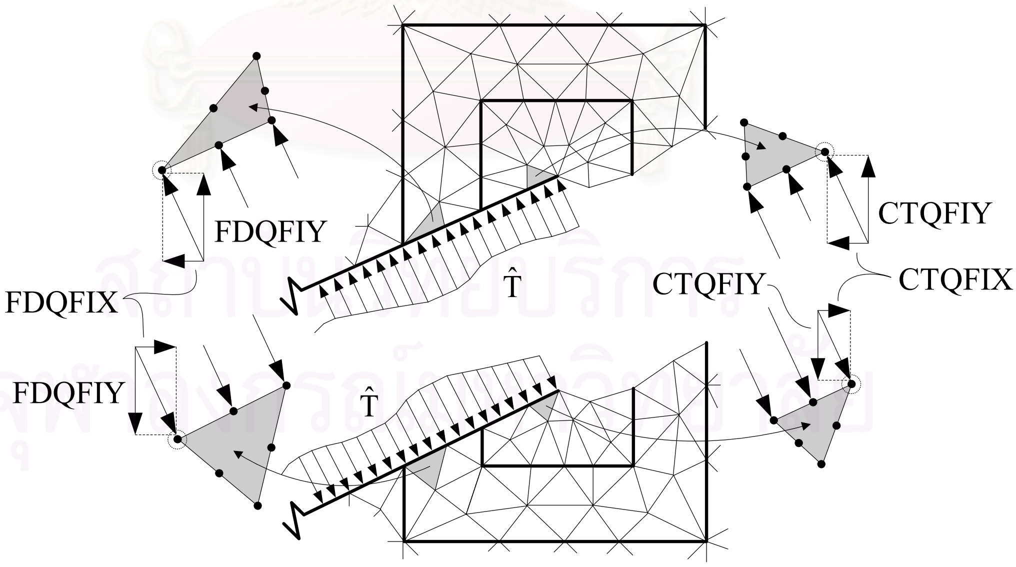
IFACE = 1 ผิวรอยร้าวที่มีเวกตอร์ตั้งฉากกับผิวรอยร้าวในทิศทางเปิดรอยร้าว $\hat{n}_I$ ดังแสดงในรูปที่ 5.3 กระทำกับระยะจากปลายรอยร้าวถึงเวกเตอร์นั้นในทิศทางตามเข็มนาฬิกา

IFACE = 2 ผิวรอยร้าวที่มีเวกตอร์ $\hat{n}_I$ กระทำกับระยะจากปลายรอยร้าวถึงเวกเตอร์นั้นในทิศทางทวนเข็มนาฬิกา

โดยค่ามุมที่ระนาบรอยร้าวกระทำกับแกน x และรหัสตัวเลขผิวรอยร้าวได้ถูกนิยามไว้เป็นตัวอย่างดังแสดงในรูปที่ 5.2 ในทำนองเดียวกันรูปที่ 5.3 แสดงการกำหนดแรงที่กระทำกับจุดต่อที่ปลายรอยร้าวและจุดต่อที่อยู่ไกลที่สุดบนผิวรอยร้าวในทิศ x และ y เนื่องจากค่าความเค้นดึงที่ผิว $\hat{T}$ สำหรับเอลิเมนต์ที่ผิวรอยร้าวซึ่งอยู่ในโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัล



รูปที่ 5.2 การกำหนดมุมของระนาบรอยร้าว รหัสตัวเลขผิวรอยร้าว และขนาดของโดเมน



รูปที่ 5.3 การกำหนดแรงที่กระทำกับจุดต่อที่ปลายรอยร้าวและจุดต่อที่อยู่ไกลที่สุดบนผิวรอยร้าวในโดเมนที่ทำการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัล

<u>ส่วนที่ 6</u> ชนิด จำนวนและขนาดของโดเมนที่ใช้อินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัล

บรรทัดแรก	คำระบุชนิดและจำนวนโดเมน
บรรทัดที่สอง	รหัสตัวเลขระบุชนิดของโดเมนและตัวเลขแสดงจำนวนโดเมน
บรรทัดที่สาม	คำระบุขนาดของโดเมน
บรรทัดต่อไป	ตัวเลขแสดงขนาดของโดเมน ในกรณีโดเมนรูปสี่เหลี่ยมผืนผ้าดังแสดงในตัวอย่างข้างล่างนี้ ผู้ใช้จะต้องกำหนดค่าสี่ค่าเพื่อใช้ในการกำหนดขนาดของโดเมนแต่ละโดเมน ขณะที่ในกรณีโดเมนรูปสี่เหลี่ยมจัตุรัสและโดเมนรูปวงกลมค่าขนาดที่ต้องกำหนดจะมีเพียงค่าเดียวต่อโดเมน

ตัวอย่างเช่น

```
IDOMTY NDOM
 3         2
ROR1  ROR2  ROR3  ROR4
 0.20  0.25  0.18  -1.00
 0.20  0.45 -1.00  -1.00
```

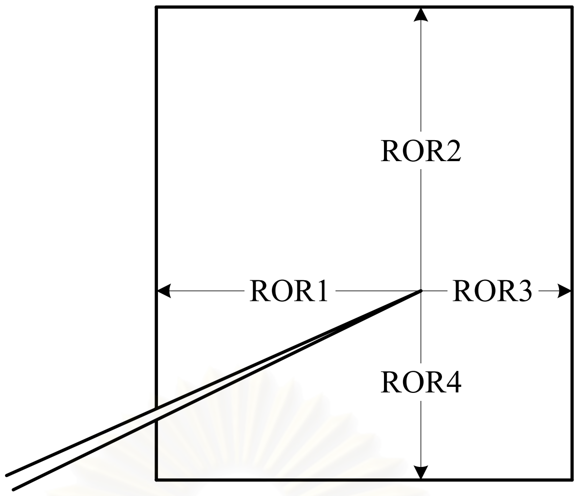
หมายเหตุ : รหัสตัวเลขระบุชนิดของโดเมนมีความหมายดังนี้

IDOMTY = 1 โดเมนรูปสี่เหลี่ยมจตุรัส

IDOMTY = 2 โดเมนรูปวงกลม

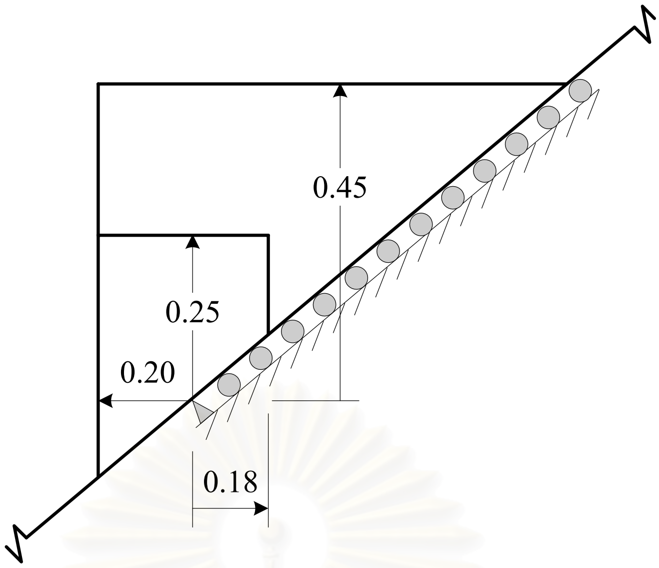
IDOMTY = 3 โดเมนรูปสี่เหลี่ยมผืนผ้า

โดยขนาดของโดเมนรูปสี่เหลี่ยมจัตุรัสที่ต้องกำหนดจะมีค่าเท่ากับความยาวครึ่งหนึ่งของด้านสี่เหลี่ยมจัตุรัสดังแสดงในรูปที่ 5.2 ในทำนองเดียวกันขนาดของโดเมนรูปวงกลมที่ต้องกำหนดจะมีค่าเท่ากับระยะรัศมีของรูปวงกลม สำหรับโดเมนรูปสี่เหลี่ยมผืนผ้านั้นการกำหนดขนาดในแต่ละโดเมนจะเริ่มจากระยะจากปลายรอยร้าวถึงขอบทางด้านซ้าย (ROR1), ด้านบน (ROR2), ด้านขวา (ROR3) และด้านล่าง (ROR4) ของโดเมนดังแสดงในรูปที่ 5.4



รูปที่ 5.4 การกำหนดขนาดของโดเมนรูปสี่เหลี่ยมผืนผ้า

ในกรณีที่ปัญหามีลักษณะสมมาตรบนระนาบรอยร้าวดังแสดงในรูปที่ 5.5 การกำหนดขนาดของโดเมนจะแตกต่างจากกรณีที่ผ่านมาโดยหากขอบของโดเมนด้านใดไม่อยู่ในแบบจำลองไฟไนต์เอลิเมนต์แล้ว ระยะจากปลายรอยร้าวถึงขอบโดเมนด้านนั้นจะถูกกำหนดให้มีค่าเท่ากับลบหนึ่ง โดยโดเมนในรูปที่ 5.5 นี้จะใช้การกำหนดขนาดดังแสดงในตัวอย่างการกำหนดขนาดของโดเมนในส่วนที่ 6 นี้ ในการกำหนดขอบของโดเมนต่าง ๆ ให้ถูกต้องนั้นมีความสำคัญกับการคำนวณค่าพารามิเตอร์เจอินทิกรัลให้ถูกต้องเป็นอย่างมากเนื่องจากที่ขอบของโดเมนต่าง ๆ นั้นค่าเวกเตอร์ q_k ดังแสดงในสมการ (2.46) และ (2.55) ต้องถูกกำหนดให้มีค่าเท่ากับศูนย์เสมอ ในการกำหนดขนาดของโดเมนต่าง ๆ นั้นยังมีเงื่อนไขเพิ่มเติมอีกข้อหนึ่งว่าขนาดของโดเมนจะต้องเรียงลำดับจากน้อยไปหามากเสมอซึ่งหมายความว่าโดเมนสุดท้ายจะต้องเป็นโดเมนที่บรรจุโดเมนต่าง ๆ ไว้ทั้งหมดและมีขนาดใหญ่ที่สุดเสมอ



รูปที่ 5.5 การกำหนดขนาดโดเมนรูปสี่เหลี่ยมผืนผ้าสำหรับปัญหาสมมาตรบนระนาบรอยร้าว

<u>ส่วนที่ 7</u> สถานะการคำนวณทั้งหมดและตัวคูณที่สอดคล้องกับสถานะการคำนวณต่าง ๆ

บรรทัดแรก คำระบุจำนวนสถานะการคำนวณ

บรรทัดที่สอง ตัวเลขระบุจำนวนสถานะการคำนวณทั้งหมด

บรรทัดที่สาม คำระบุค่าตัวคูณภาระ ระยะเคลื่อนตัว อุณหภูมิ และแรงวัตถุที่สอดคล้องกับสถานะการคำนวณต่าง ๆ

บรรทัดต่อไป ตัวเลขระบุค่าตัวคูณภาระ ระยะเคลื่อนตัว อุณหภูมิ และแรงวัตถุที่สอดคล้องกับสถานะการคำนวณต่าง ๆ

ตัวอย่างเช่น NSTATE

6

ForcFAC	DispFAC	TempFAC	BodyFAC
0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00
0.25	0.00	1.00	0.00
0.50	0.00	1.00	0.00
0.75	0.00	1.00	0.00
1.00	0.00	1.00	0.00

หมายเหตุ : ตัวเลขระบุค่าตัวคูณภาระ ระยะเคลื่อนตัว อุณหภูมิ และแรงวัตถุที่สถานะการคำนวณแรกต้องมีค่าเป็นศูนย์ทั้งหมดเสมอ

<u>ส่วนที่ 8</u> จำนวนวัสดุที่ใช้และคุณสมบัติต่าง ๆ

บรรทัดแรก คำระบุจำนวนวัสดุที่ใช้ในแบบจำลองไฟไนต์เอลิเมนต์

บรรทัดที่สอง ตัวเลขแสดงจำนวนวัสดุที่ใช้

บรรทัดที่สาม คำระบุคุณสมบัติต่าง ๆ ของวัสดุ

บรรทัดต่อไป ตัวเลขแสดงค่าโมดูลัสของความยืดหยุ่น อัตราส่วนปัวส์ซง ความเค้นที่จุดคราก ตัวเลขยกกำลังของความเครียด ค่าคงที่ของวัสดุ สัมประสิทธิ์การขยายตัวเนื่องจากอุณหภูมิ และค่าความหนาของชิ้นงาน (เฉพาะกรณีปัญหาความเค้นหรือความเครียดในระนาบ) เรียงตามลำดับจากวัสดุที่หนึ่งถึงวัสดุสุดท้ายที่กำหนดไว้ในบรรทัดที่สอง “ตัวเลขแสดงจำนวนวัสดุที่ใช้”

ตัวอย่างเช่น NMAT

2

ELAS	PR	YSTRSS	AHARD	ALPHA	COTHR	THICK
3.0E+4	0.3	60.0	5.0	0.5	7.3E-6	
3.0E+4	0.3	60.0	5.0	0.1E-60	7.3E-6	

<u>ส่วนที่ 9</u> ตำแหน่งพิกัดของจุดต่อ

บรรทัดแรก คำระบุหมายเลขจุดต่อ พิกัด x และ y

บรรทัดต่อไป ตัวเลขแสดงหมายเลขจุดต่อเรียงตามลำดับ พิกัดในแนวแกน x และ y

ตัวอย่างเช่น

NODE	X	Y
1	0.0	0.0
2	0.212657	0.002507
3	0.5	0.5
4	1.0	1.0
⋮	⋮	⋮
720	42.5	50.0

<u>ส่วนที่ 10</u> ลักษณะการเรียงตัวของจุดต่อภายในเอลิเมนต์และวัสดุที่ใช้ในเอลิเมนต์

บรรทัดแรก คำระบุลักษณะการเรียงตัวของจุดต่อภายในเอลิเมนต์และรหัสหมายเลขวัสดุที่ใช้ในเอลิเมนต์ที่สอดคล้องกับคุณสมบัติของวัสดุดังแสดงในส่วนที่ 8

บรรทัดต่อไป ตัวเลขแสดงหมายเลขของเอลิเมนต์เรียงตามลำดับ หมายเลขของจุดต่อทั้งหกที่ประกอบกันขึ้นเป็นเอลิเมนต์และหมายเลขวัสดุที่ใช้ในเอลิเมนต์

ตัวอย่างเช่น

ELEMENT	NODAL CONNECTIVITY						IEMAT
1	1	2	3	4	5	6	2
2	2	31	4	32	204	10	1
3	10	30	40	35	210	20	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
331	147	146	149	616	617	619	1

<u>ส่วนที่ 11</u> การกำหนดเงื่อนไขขอบเขตที่จุดต่อ

บรรทัดแรก คำระบุการกำหนดเงื่อนไขขอบเขตที่จุดต่อ

บรรทัดต่อไป ตัวเลขแสดงหมายเลขจุดต่อที่มีการกำหนดเงื่อนไขขอบเขต รหัสเงื่อนไขขอบเขตในทิศทาง x และ y และค่าเงื่อนไขขอบเขตที่สอดคล้องกับรหัสเงื่อนไขขอบเขตในแนวแกน x และ y ที่สถานะการคำนวณสุดท้าย

ตัวอย่างเช่น

NODE	IBCX	IBCY	QFIX	QFIY
1	0	+1	0.0	0.0
3	−1	+1	0.01	0.0
7	0	0	50.0	100.0
⋮	⋮	⋮	⋮	⋮
10	+1	−1	0.0	0.005

หมายเหตุ : รหัสเงื่อนไขขอบเขตทั้งหมดมีความหมายดังนี้

IBC = +1 จุดต่อนั้นถูกตรึงให้ไม่มีการเคลื่อนที่ในทิศนั้น
ในกรณีนี้ QFI แทนค่าการเคลื่อนตัวในทิศนั้น ดังนั้นจึงต้องกำหนดให้ QFI มีค่าเท่ากัยศูนย์

IBC = 0 มีการกำหนดค่าแรงภายนอกที่มากระทำกับจุดต่อในทิศนั้น
ในกรณีนี้ QFI แทนค่าแรงภายนอกสุทธิที่มากระทำโดยอาจมีค่าเท่ากับศูนย์ก็ได้

IBC = −1 มีการกำหนดค่าการเคลื่อนตัวในทิศนั้น
ในกรณีนี้ QFI แทนค่าการเคลื่อนตัวสุทธิที่สถานะการคำนวณสุดท้าย

<u>ส่วนที่ 12</u> วิธีการอ่านค่าอุณหภูมิและแรงวัตถุที่จุดต่อ

บรรทัดแรก คำระบุวิธีการอ่านค่าอุณหภูมิและแรงวัตถุที่จุดต่อ

บรรทัดที่สอง รหัสตัวเลขแสดงวิธีการอ่านค่าอุณหภูมิและแรงวัตถุที่จุดต่อ

บรรทัดที่สาม คำระบุหมายเลขจุดต่อ ค่าอุณหภูมิและแรงวัตถุที่จุดต่อ

บรรทัดต่อไป ตัวเลขแสดงหมายเลขจุดต่อเรียงตามลำดับ ค่าอุณหภูมิและค่าแรงวัตถุในทิศทาง x และ y

ตัวอย่างเช่น IRTemp IRBody
+1 0

NODE	TEMP	BDFX	BDFY
1	0.0000		
2	20.000		
3	21.125		
⋮	⋮		
720	100.00		

หมายเหตุ : รหัสตัวเลขทั้งหมดของ IRTemp และ IRBody มีความหมายดังนี้

+1 หมายถึงอ่านค่าทั้งหมดจากไฟล์นำเข้านี้

0 หมายถึงให้โปรแกรมกำหนดค่าเป็นศูนย์ทั้งหมด

–1 หมายถึงผู้ใช้กำหนดค่าเองเป็นสมการโดยตรงลงในโปรแกรมย่อย [SUBROUTINE TemBDfFUNC] ของโปรแกรม JFACTOR

5.4 ลักษณะของไฟล์ผลลัพธ์

ไฟล์ผลลัพธ์ที่ได้จากโปรแกรมคอมพิวเตอร์ JFACTOR จะถูกบรรจุอยู่ในไฟล์ชื่อเดียวกันกับไฟล์ข้อมูลนำเข้าตามด้วย "_**" โดยที่ ** แทนเลขจำนวนเต็มหนึ่งหรือสองตำแหน่งแสดงครั้งที่ทำการคำนวณซึ่งสอดคล้องกับสถานะการคำนวณของไฟล์ผลลัพธ์ที่ได้ยกตัวอย่างเช่น SingleEPS_1 แสดงผลการคำนวณครั้งแรกซึ่งสอดคล้องกับค่าภาระต่าง ๆ ที่สถานะการคำนวณที่สอง (สถานะการคำนวณแรกค่าภาระต่าง ๆ ต้องเป็นศูนย์เสมอสอดคล้องกับค่าตัวคูณภาระต่าง ๆ ที่สถานะการคำนวณนี้ที่มีค่าเท่ากับศูนย์ทั้งหมด) โดยไฟล์ผลลัพธ์ที่ได้นี้ยังแบ่งออกเป็น 3 ชนิดขึ้นอยู่กับลักษณะการใช้งานดังนี้

5.4.1 ไฟล์ที่นำไปใช้สำหรับการปรับขนาดเอลิเมนต์

ไฟล์ชนิดนี้จะมีนามสกุล ".out" โดยจะนำไปใช้ร่วมกับส่วนหัวของไฟล์ผลลัพธ์ที่ได้จากการวิเคราะห์ปัญหาเดียวกันของโปรแกรม FEMESH v2.1.152 (ซึ่งมีนามสกุล ".out" เช่นเดียวกัน) เพื่อให้โปรแกรม FEMESH v2.1.152 ยอมรับผลการคำนวณค่าความเค้นวอนมิสเซสที่จุดต่อซึ่งคำนวณได้จากโปรแกรม JFACTOR เพื่อนำไปปรับขนาดเอลิเมนต์ โดยรายละเอียดของข้อมูลผลลัพธ์ที่ได้ประกอบไปด้วย หมายเลขจุดต่อ ค่าการเคลื่อนตัวในแนวแกน x และ y ของจุดต่อต่าง ๆ ค่าความเค้นตั้งฉากในแนวแกน x และ y ค่าความเค้นเฉือนและค่าความเค้นวอนมิสซิสและขนาดของเอลิเมนต์ที่จุดต่อ H ซึ่งกำหนดให้มีค่าเป็นศูนย์เนื่องจากโปรแกรม FEMESH v2.1.152 จะใช้เฉพาะค่าความเค้นวอนมิสเซสที่จุดต่อเท่านั้นในการปรับขนาดเอลิเมนต์ โดยขนาดของเอลิเมนต์ที่เล็กที่สุดผู้ใช้จะต้องกำหนดผ่านโปรแกรม FEMESH v2.1.152 ด้วยตัวเอง นอกจากนี้ยังพิมพ์ผลการคำนวณค่าพารามิเตอร์เจอินทิกรัลจากโดเมนต่าง ๆ และค่าเฉลี่ยของทุกโดเมน พร้อมกับค่าเปอร์เซ็นต์ความแตกต่างระหว่างค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้เมื่อเปรียบเทียบกับโดเมนถัดไป และค่าเปอร์เซ็นต์ความแตกต่างระหว่างค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้เมื่อเปรียบเทียบกับค่าเฉลี่ยดังแสดงในตัวอย่างต่อไปนี้

```
NODE           U                V                 Sxx          … H
1         -.32039880E-02   .00000000E+00   -.83045497E+00   … .00   000000E+00
2         -.28969745E-02   .00000000E+00    .83712706E+01   … .00   000000E+00
3         -.24634279E-02   .00000000E+00    .29093107E+02   … .00   000000E+00
4         -.12257961E-02   .00000000E+00    .96909592E+02   … .00   000000E+00
5         -.13524844E-02    .36024768E-03   .72654596E+02   … .00   000000E+00
    ⋮            ⋮                ⋮                ⋮                      ⋮
7    19   -.47451004E-03    .17200855E-02   .11552974E+03   … .0   0000000E+00
7    20   -.98853774E-05    .18630842E-02   .35488095E+02   … .00   000000E+00

[DOMAIN]  [J-INTEGRAL]    [ER. from pDOM]   [ER. from AVG]
   1       .17440980E+03   .00000000E+00    .71652313E-02
   2       .17441759E+03   .44644754E-02    .27008764E-02
   3       .17443951E+03   .12565744E-01    .98661077E-02
[AVERAGE J-INTEGRAL]  =  .17442230E+03
```

รูปที่ 5.6 รูปแบบของไฟล์ผลลัพธ์ที่ได้จากโปรแกรม JFACTOR เพื่อนำไปใช้สำหรับการปรับขนาดเอลิเมนต์

5.4.2 ไฟล์ที่นำไปใช้สำหรับการแสดงผลกราฟฟิกบนจอคอมพิวเตอร์

ไฟล์ชนิดนี้จะมีนามสกุล ".plt" โดยจะนำไปใช้แสดงผลด้วยโปรแกรมกราฟฟิก โดยรายละเอียดของข้อมูลผลลัพธ์ที่ได้จะประกอบไปด้วยค่าพิกัดในแนวแกน x และ y ของจุดต่อบวกกับค่าระยะการเคลื่อนตัวที่สอดคล้องกันคูณด้วยค่าตัวคูณขนาด (Scaling Factor) เพื่อให้ได้ค่าพิกัดในแนวแกน x และ y ของจุดต่อที่เกิดการเสียรูปแล้วที่สามารถมองเห็นได้ชัดเจนขึ้น ค่าความเค้นตั้งฉากในแนวแกน x และ y ค่าความเค้นเฉือนและค่าความเค้นวอนมิสซิสที่จุดต่อ การเรียงตัวของจุดต่อหลักภายในเอลิเมนต์สามเหลี่ยมและสี่เหลี่ยมดังแสดงในตัวอย่างต่อไปนี้

```
VARIABLES = "X-CO","Y-CO","SXX","SYY","SXY","SVM"
ZONE N=720, E=331, F=FEPOINT, ET=QUADRILATERAL
-.411  39880E-02     .00000000E+00     -.83045497E+00    … .45   809938E+02
-.3   7059745E-02    .00000000E+00      .83712706E+01    … .99   231532E+01
-.3   3724279E-02    .00000000E+00      .29093107E+02    … .29   641552E+02
              ⋮                ⋮                 ⋮                      ⋮
-.5   8551004E-03    .18320065E-02      .11552974E+03    … .14   189829E+03
-.9   8443784E-05    .19680942E-02      .35488095E+02    … .41   717930E+02
              ⋮                ⋮                 ⋮                      ⋮
-.9   2755984E-03    .88898923E-03      .99076349E+02    … .32   785618E+03
1 2 3                      1
2           31 4           2
10     30      40          10
    ⋮       ⋮       ⋮       ⋮
1    01 1    02 7    23    724
```

รูปที่ 5.7 รูปแบบของไฟล์ผลลัพธ์ที่ได้จากโปรแกรม JFACTOR เพื่อนำไปใช้สำหรับการแสดงผลกราฟฟิกบนจอคอมพิวเตอร์

5.4.3 ไฟล์ที่นำไปใช้เพื่อเริ่มต้นการคำนวณที่สถานะการคำนวณที่ได้ทำการวิเคราะห์ไปแล้ว

ไฟล์ชนิดนี้จะมีนามสกุล ".res" โดยจะนำไปใช้เพื่อให้สามารถเริ่มต้นการคำนวณที่สถานะการคำนวณก่อนหน้านี้ที่ได้ทำการวิเคราะห์เสร็จเรียบร้อยไปแล้ว โดยรายละเอียดข้อมูลของไฟล์ชนิดนี้จะประกอบไปด้วย หมายเลขสมการ ค่าการเคลื่อนตัวที่จุดต่อและค่าเวกเตอร์ความไม่สมดุลย์ของแรง (Out of Balance Force Vector) ที่สอดคล้องกับหมายเลขสมการ ดังแสดงในตัวอย่างต่อไปนี้

1	2.309754444693246E-003	3.568269036175319E-013
2	0.000000000000000E+000	0.000000000000000E+000
3	2.488180544459391E-003	-2.398081733190338E-014
4	0.000000000000000E+000	0.000000000000000E+000
5	2.698702326306068E-003	-1.612261092526706E-013
6	0.000000000000000E+000	0.000000000000000E+000
7	2.959227277028109E-003	2.947531543712411E-013
8	0.000000000000000E+000	0.000000000000000E+000
9	3.111815926301099E-003	-3.801403636316536E-013
10	0.000000000000000E+000	0.000000000000000E+000

รูปที่ 5.8 รูปแบบของไฟล์ที่นำไปใช้เพื่อเริ่มต้นการคำนวณที่สถานะการคำนวณก่อนหน้านี้ที่ได้ทำการวิเคราะห์เสร็จเรียบร้อยไปแล้ว

5.5 บทสรุป

ในบทนี้ขั้นตอนการทำงานและลักษณะโครงสร้างของไฟล์นำเข้าเพื่อใช้กับโปรแกรม JFACTOR ได้ถูกอธิบายโดยละเอียด ลักษณะและหน้าที่การทำงานของไฟล์ผลลัพธ์ซึ่งได้จากโปรแกรม JFACTOR ซึ่งประกอบไปด้วยไฟล์ที่นำไปใช้สำหรับการปรับขนาดเอลิเมนต์ ไฟล์ที่นำไปใช้สำหรับการแสดงผลกราฟฟิกบนจอคอมพิวเตอร์และไฟล์ที่นำไปใช้เพื่อเริ่มต้นการคำนวณที่สถานะการคำนวณที่ได้ทำการวิเคราะห์ไปแล้วได้ถูกอธิบายอย่างละเอียด ในบทต่อไปโปรแกรม JFACTOR ที่ได้ประดิษฐ์ขึ้นได้ถูกตรวจสอบโดยนำไปวิเคราะห์ปัญหาชิ้นงานที่มีรอยร้าวมาตรฐานต่าง ๆ โดยทำการเปรียบเทียบค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้จากโปรแกรม JFACTOR กับผลการคำนวณที่มีในงานวิจัยอื่น ๆ

บทที่ 6

การตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์

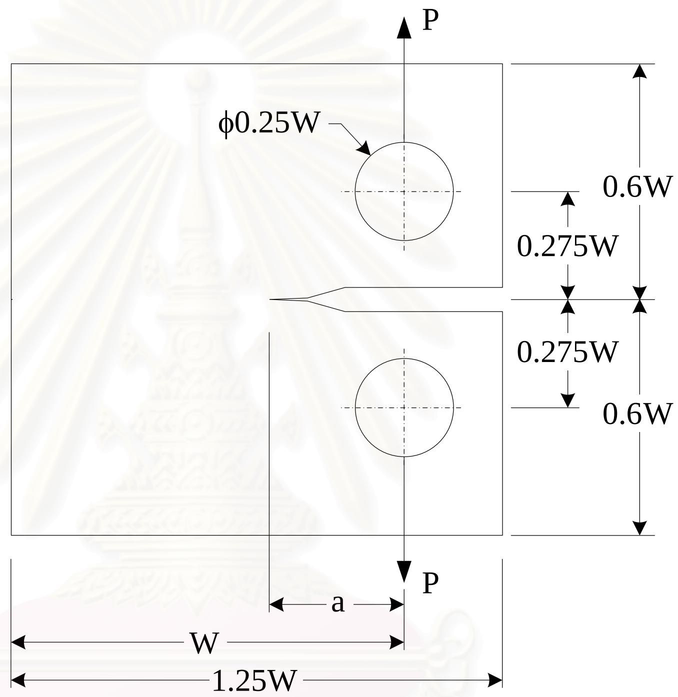
ในบทนี้จะเป็นการตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์ JFACTOR ที่ได้ประดิษฐ์ขึ้นโดยนำโปรแกรมคอมพิวเตอร์นี้ไปแก้ปัญหาพื้นฐานต่าง ๆ แล้วนำผลลัพธ์ค่าพารามิเตอร์เจอินทิกรัลที่ได้ไปเปรียบเทียบกับผลลัพธ์จากการคำนวณด้วยวิธีอื่น ๆ ที่มีอยู่สำหรับปัญหาเดียวกัน โดยปัญหาพื้นฐานที่นำมาใช้ในการตรวจสอบความถูกต้องจะเป็นปัญหาของชิ้นงานที่มีรอยร้าวในรูปแบบต่าง ๆ ดังนี้

(1) ชิ้นทดสอบมาตรฐานแบบ CT (Compact Tension Specimen)
(2) ชิ้นทดสอบมาตรฐานแบบ DENT (Double Edge Notched Tension Specimen)
(3) ท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้แรงดันภายใน (Axially Internal Cracked Cylinder under Internal Pressure)
(4) ท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความหนาของท่อ (Circumferentially Cracked Cylinder under Remote Uniform Tension and Thermal Gradient across Cylinder Wall)
(5) แผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความกว้างของแผ่น (Single-Edge Cracked Plate subject to Remote Uniform Tension and Thermal Gradient across the Plate Width)

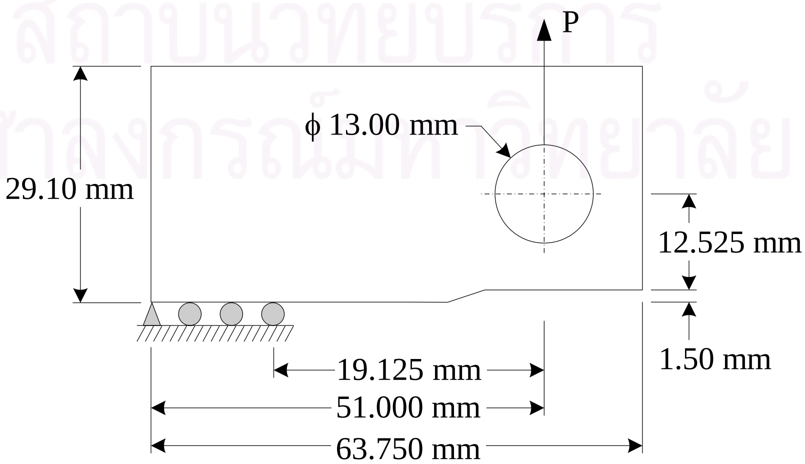
6.1 ชิ้นทดสอบมาตรฐานแบบ CT

ปัญหาที่นำมาใช้ตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์นี้เป็นปัญหาชิ้นทดสอบมาตรฐานแบบ CT ภายใต้เงื่อนไขความเครียดในระนาบ (Plane Strain) ซึ่งถูกกระทำด้วยแรงดึง P โดยมีรายละเอียดดังแสดงในรูปที่ 6.1 กำหนดให้ชิ้นงานมีความกว้าง $W = 51\ \text{mm}$ อัตราส่วนความยาวรอยร้าวต่อความกว้าง $a/W = 0.75$ และความหนา $t = 1\ \text{mm}$ ค่าโมดูลัสของความยืดหยุ่น $E = 202\ \text{GPa}$ อัตราส่วนปัวซงส์ $\nu = 0.3$ ค่าความเค้นที่จุดคราก $\sigma_o = 414\ \text{MPa}$ ค่าคงที่ของวัสดุ $\alpha = 0.05$ และค่ายกกำลังของความเครียด $n = 10$ ใช้จำนวนจุดเกาส์เท่ากับ 3 และ 2×2 จุดในการอินทิเกรตเชิงตัวเลขพจน์ที่เกี่ยวข้องกับค่าความเค้นตั้งฉากเฉลี่ยสำหรับเอลิเมนต์สามเหลี่ยมหกจุดต่อและเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อตามลำดับ และใช้จำนวนจุดเกาส์เท่ากับ 7 และ 3×3 จุดในการอินทิเกรตพจน์ที่เกี่ยวข้องกับค่าเค้นดิเวียทอริกสำหรับเอลิเมนต์สามเหลี่ยม

และสี่เหลี่ยมตามลำดับ (Reduced and Selective Integration) เอลิเมนต์ที่ปลายรอยร้าวที่ใช้ในการคำนวณเป็นเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อที่จุดต่อทั้งสามของด้านที่ถูกยุบมารวมกันที่ตำแหน่งปลายรอยร้าวและสามารถเคลื่อนตัวได้เป็นอิสระต่อกัน ขณะที่จุดต่อกลางด้านทั้งสองที่อยู่ติดกับปลายรอยร้าวและจุดต่อกลางเอลิเมนต์ยังคงอยู่ที่ตำแหน่งกึ่งกลาง โดยขนาดของเอลิเมนต์ที่ปลายรอยร้าวที่ใช้มีค่าเท่ากับ $0.01(W-a)=0.1275$ mm พื้นที่โดเมนรูปสี่เหลี่ยมจัตุรัส 3 โดเมนขนาดกำหนดในไฟล์ข้อมูลนำเข้า (ROR1) เท่ากับ 2.125, 6.375 และ 12.75 mm ตามลำดับถูกใช้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัล



รูปที่ 6.1 รูปร่างของปัญหาชิ้นทดสอบมาตรฐานแบบ CT



รูปที่ 6.2 รูปร่างและรายละเอียดของปัญหาชิ้นทดสอบมาตรฐานแบบ CT ที่นำมาพิจารณา

เนื่องจากปัญหามีลักษณะสมมาตรดังนั้นเราจึงใช้เฉพาะครึ่งหนึ่งของชิ้นทดสอบด้านบนดังแสดงในรูปที่ 6.2 เป็นแบบจำลองไฟไนต์เอลิเมนต์ ตารางที่ 6.1 แสดงค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมน (Domain Dependence) ของค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้จากโปรแกรม JFACTOR ซึ่งถูกนิยามให้มีค่าเท่ากับเปอร์เซ็นต์ผลต่างของค่าพารามิเตอร์เจอินทิกรัลสูงสุดที่คำนวณได้จากโดเมนต่าง ๆ เปรียบเทียบกับค่าเฉลี่ย โดยเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลนี้จะถูกใช้เพื่อบ่งบอกถึงความถูกต้องของผลลัพธ์ที่คำนวณได้เนื่องจากเราทราบจากสมการ (2.46) และ (2.56) ว่าค่าพารามิเตอร์เจอินทิกรัลเมื่อเขียนอยู่ในรูปของการอินทิเกรตบนพื้นที่แล้วจะไม่ขึ้นกับขนาดของพื้นที่ที่ใช้ทำการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัลตราบใดที่ขนาดของเวกเตอร์ q_k มีค่าเท่ากับหนึ่งที่ตำแหน่งปลายรอยร้าวและมีค่าเท่ากับศูนย์ที่ขอบของพื้นที่ที่ใช้ทำการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัลและมีค่าอยู่ระหว่างหนึ่งถึงศูนย์บนพื้นที่นี้ โดยในวิทยานิพนธ์นี้ใช้จำนวนพื้นที่โดเมนทั้งหมด 3 โดเมนในการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัล ดังนั้นค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลสามารถเขียนอยู่ในรูปสมการได้เป็น

$$\text{Domain Dependence }(\%) = \text{Max}\left(\frac{\left|J^{i\text{th}} - J_{\text{avg}}\right|}{J_{\text{avg}}}\times 100\right),\ i = 1, 2, 3 \qquad (8.1)$$

โดยที่ $J^{i\text{th}},\ i = 1,\ 2,\ 3$ แทนค่าพารามิเตอร์เจอินทิกรัลที่คำนวณจากโดเมนที่ *i*th

$J_{\text{avg}} = \dfrac{J^{1\text{th}} + J^{2\text{th}} + J^{3\text{th}}}{3}$ แทนค่าพารามิเตอร์เจอินทิกรัลเฉลี่ย

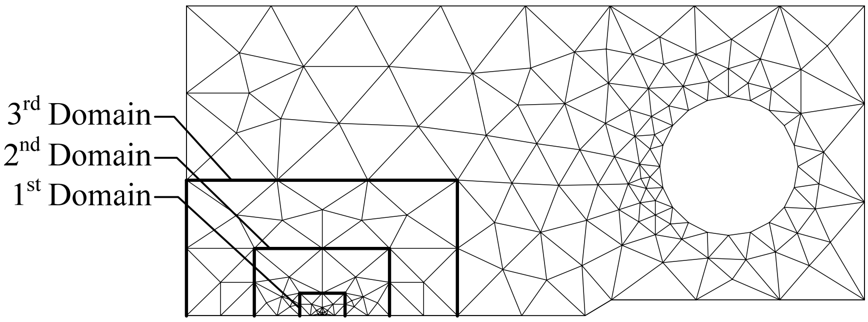
จากตารางพบว่าผลการคำนวณที่ได้จากโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัตินั้นจะให้ผลลัพธ์ที่ขึ้นกับโดเมนน้อยกว่าผลการคำนวณที่ได้จากโครงตาข่ายเริ่มต้น (Initial Mesh) ดังนั้นค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้จากโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัตินั้นจะให้ผลการคำนวณที่มีความถูกต้องสูงกว่าผลการคำนวณที่ได้จากโครงตาข่ายเริ่มต้น เนื่องจากค่าอนุพันธ์อันดับสองของค่าความเค้นวอนมิสเซสที่ได้จากการวิเคราะห์โดยใช้โครงตาข่ายเริ่มต้นที่ค่าแรงดึง $P = 55$ N นั้นมีค่ายังไม่สูงมากนักและเพื่อไม่ให้ต้องทำการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติบ่อยครั้งจนเกินไป ดังนั้นโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โครงตาข่ายแรกที่ทำการวิเคราะห์ที่ค่าแรงดึง $P = 55$ N ซึ่งประกอบด้วยจำนวนจุดต่อและเอลิเมนต์ทั้งหมดเท่ากับ 1940 จุดต่อและ 909 เอลิเมนต์ตามลำดับนั้นผู้วิจัยจึงได้จากการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติเข้ากับโครงตาข่ายเริ่มต้นโดยใช้ผลการคำนวณที่ค่าแรงดึง $P = 550$ N ซึ่งได้จากการวิเคราะห์โดยใช้โครงตาข่ายเริ่มต้นนี้แทน จากนั้นทำการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์

โดยอัตโนมัติอีกในการวิเคราะห์ที่ค่าแรงดึงเท่ากับ 880 และ 1100 N ตามลำดับโดยในการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์แต่ละครั้งจะพยายามปรับปรุงให้ผลการคำนวณค่าการกระจายตัวของสนามความเค้นวอนมิสเซสมีความต่อเนื่องมากขึ้นและค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลมีค่าต่ำกว่า 0.1%

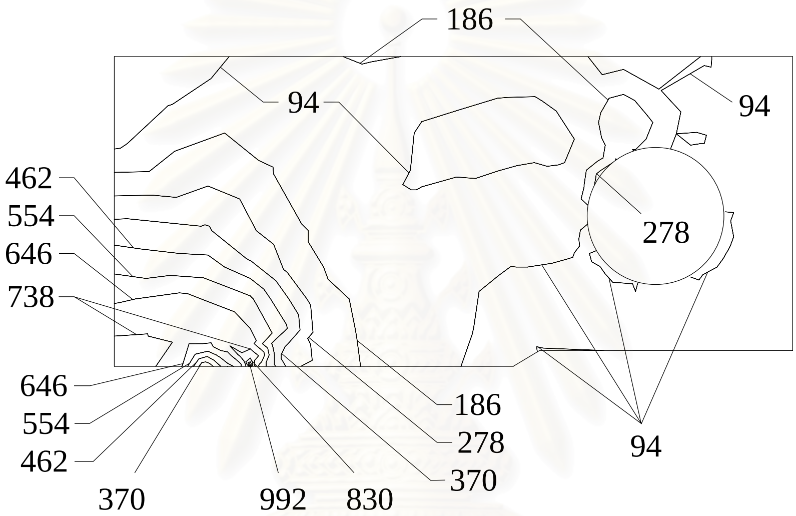
ตารางที่ 6.1 ค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่ค่าแรงดึงต่าง ๆ สำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ CT

Load (N)	Initial Mesh			Adaptive Mesh		
	Nodes	Elements	Domain Dependence (%)	Nodes	Elements	Domain Dependence (%)
55	653	293	2.55E-01	1940	909	6.41E-02
110			2.44E-01			3.36E-02
165			2.39E-01			1.89E-02
220			2.35E-01			1.31E-02
275			2.42E-01			1.09E-02
330			2.43E-01			9.98E-03
385			2.34E-01			9.22E-03
440			2.23E-01			8.36E-03
495			2.12E-01			7.34E-03
550			1.97E-01			6.43E-03
605			1.78E-01			5.90E-03
660			1.53E-01			6.16E-03
715			1.17E-01			7.81E-03
770			5.59E-02			8.57E-03
825			5.92E-02			7.75E-03
880			2.44E-01	2435	1150	4.76E-03
935			4.83E-01			6.59E-03
990			7.38E-01			8.71E-03
1045			9.74E-01			1.10E-02
1100			1.17E+00	3383	1610	2.18E-02

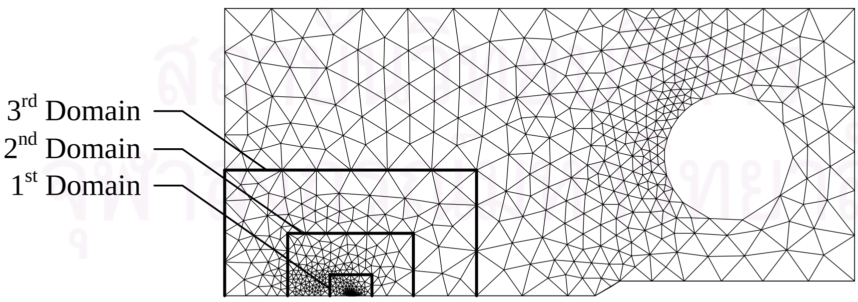
รูปที่ 6.3 และ 6.4 แสดงลักษณะของโครงตาข่ายเริ่มต้นพร้อมกับโดเมนที่ใช้ในการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัลทั้ง 3 โดเมน โดยโครงตาข่ายเริ่มต้นนี้ประกอบด้วย 653 จุดต่อ 293 เอลิเมนต์และลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ค่าแรงดึง $p = 1100$ N ตามลำดับเปรียบเทียบกับรูปที่ 6.5 และ 6.6 ซึ่งแสดงลักษณะของโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติพร้อมกับโดเมนที่ใช้ในการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัลโดยโครงตาข่ายนี้ประกอบไปด้วย 3383 จุดต่อ 1610 เอลิเมนต์และลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ค่าแรงดึงเดียวกัน



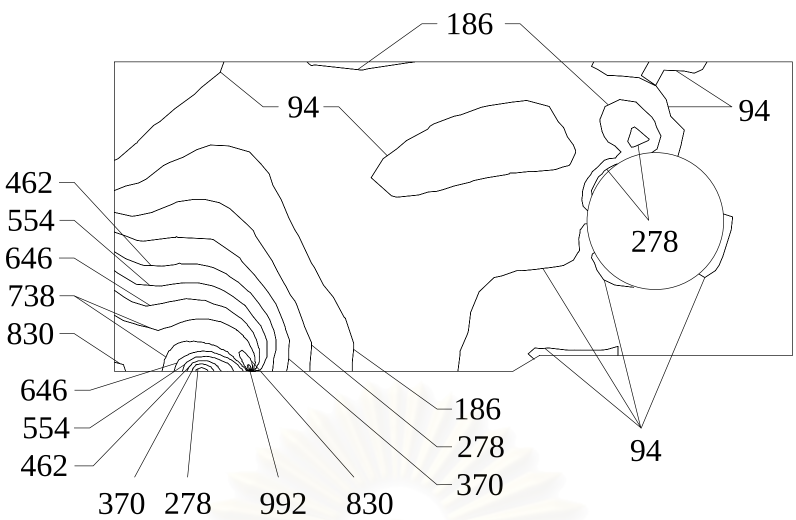
รูปที่ 6.3 ลักษณะโครงตาข่ายเริ่มต้นและโดเมนที่ใช้ในการคำนวณสำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ CT



รูปที่ 6.4 ลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซส, MPa บนโครงตาข่ายเริ่มต้นสำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ CT ที่ค่าแรงดึง $p = 1100$ N

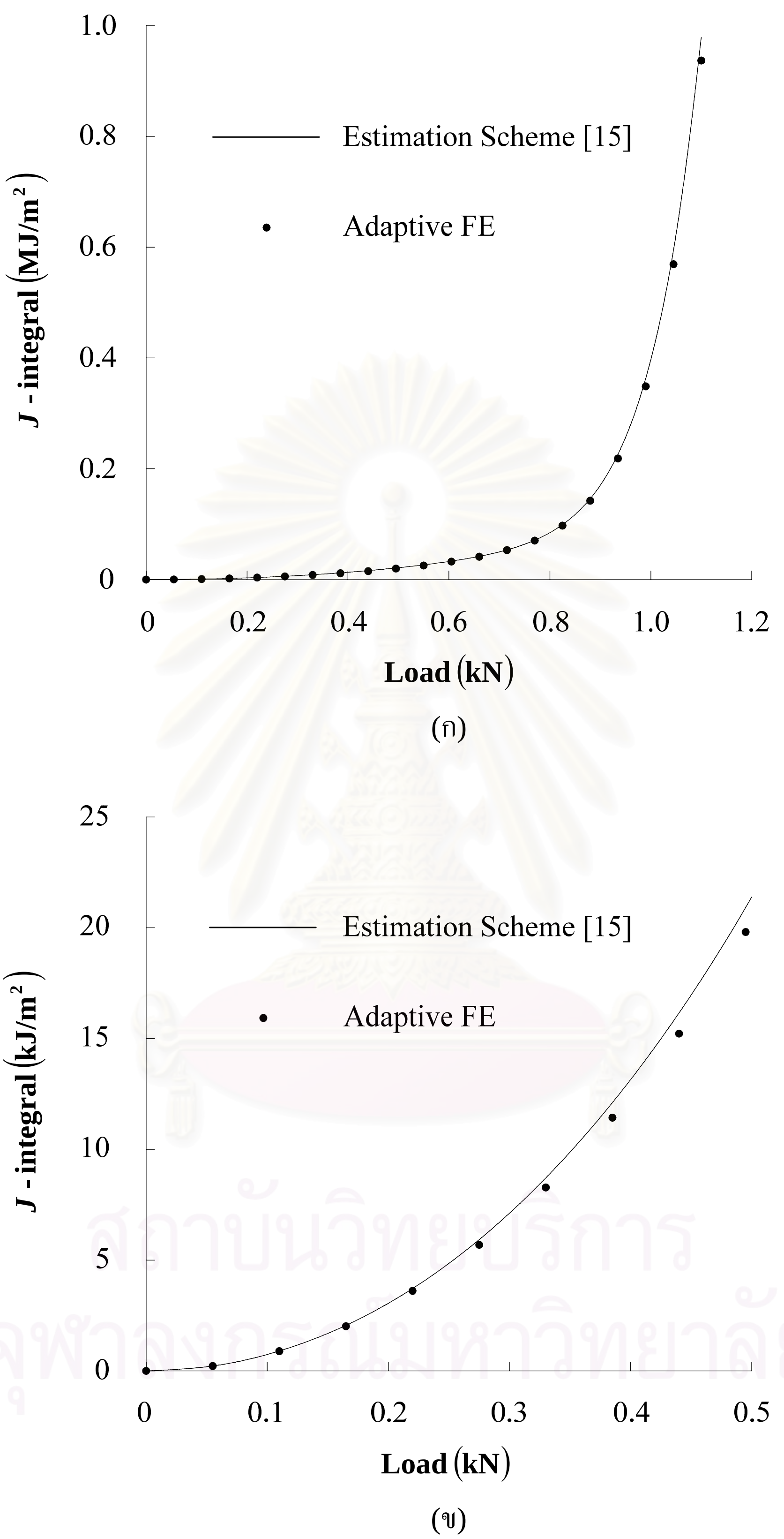


รูปที่ 6.5 ลักษณะโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติและโดเมนที่ใช้ในการคำนวณสำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ CT



รูปที่ 6.6 ลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซส, MPa สำหรับโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติสำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ CT ที่ค่าแรงดึง $p = 1100$ N

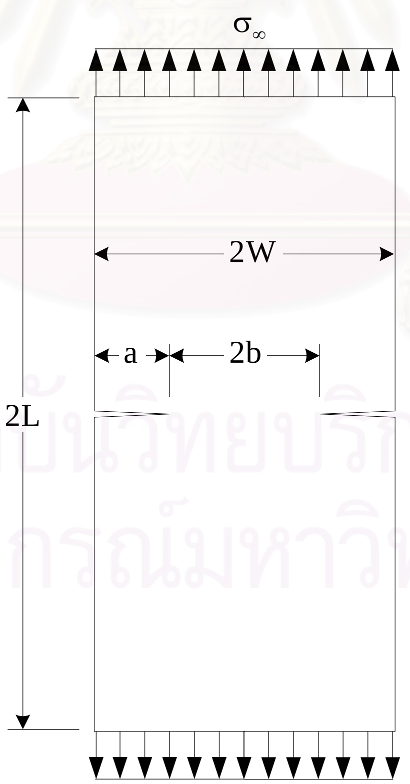
จากรูปจะพบว่าโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติจะมีเอลิเมนต์ขนาดเล็กจำนวนมากเรียงตัวกันอยู่ที่บริเวณปลายรอยร้าวซึ่งสอดคล้องกับผลลัพธ์ที่ได้จากวิธีเชิงวิเคราะห์ว่าที่บริเวณปลายรอยร้าวจะมีค่าการเปลี่ยนแปลงความชันของสนามความเค้นวอนมิสเซสเข้าสู่อนันต์ซึ่งทำให้ลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ได้จากการคำนวณโดยใช้โครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัตินี้มีความต่อเนื่องมากกว่าโครงตาข่ายเริ่มต้น ในการตรวจสอบความถูกต้องของโปรแกรม JFACTOR นั้นจะทำการเปรียบเทียบค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้จากโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติกับผลลัพธ์ที่ได้จากวิธีการคำนวณของ Electric Power Research Institute (EPRI) ซึ่งวิธีการคำนวณนี้ถูกเรียกว่า EPRI J Estimation Scheme [15] รูปที่ 6.7 แสดงผลการคำนวณค่าพารามิเตอร์เจอินทิกรัลเฉลี่ยจากทั้ง 3 โดเมนที่ค่าแรงดึงระดับต่าง ๆ เปรียบเทียบกับผลการคำนวณของ EPRI โดยผลลัพธ์ที่ได้นั้นแสดงให้เห็นถึงความสอดคล้องกันเป็นอย่างดี



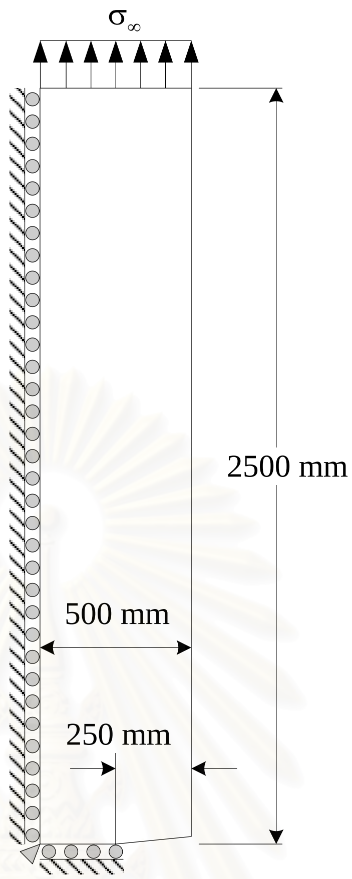
รูปที่ 6.7 ค่าพารามิเตอร์เจอินทิกรัลของชิ้นงานมาตรฐานแบบ CT สำหรับปัญหาความเครียดในระนาบ (ก) ที่ค่าระดับภาระสูงและ (ข) ที่ค่าระดับภาระต่ำถึงปานกลาง

6.2 ชิ้นทดสอบมาตรฐานแบบ DENT

ปัญหาที่นำมาตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์นี้เป็นปัญหาชิ้นทดสอบมาตรฐานแบบ DENT ภายใต้เงื่อนไขความเค้นในระนาบ (Plane Stress) ซึ่งถูกกระทำด้วยค่าความเค้นดึง σ_∞ ที่ปลายทั้งสองข้างดังแสดงในรูปที่ 6.8 กำหนดให้ชิ้นงานมีความกว้าง $2W = 1000$ mm อัตราส่วนความยาวรอยร้าวต่อความกว้าง $a/W = 0.5$ และความหนา $t = 1$ mm โดยค่าคุณสมบัติของวัสดุมีค่าเท่ากับกรณีชิ้นทดสอบมาตรฐานแบบ CT ทุกประการ เนื่องจากความซับซ้อนของสมการไฟไนต์เอลิเมนต์ภายใต้เงื่อนไขความเค้นในระนาบ ดังนั้นจึงใช้ทั้งจำนวนจุดเกาส์เท่ากับ 7 และ 3×3 จุดในการอินทิเกรตเชิงตัวเลขพจน์ที่เกี่ยวข้องกับค่าความเค้นตั้งฉากเฉลี่ยและค่าเค้นดิเวียทอริกสำหรับเอลิเมนต์สามเหลี่ยมหกจุดต่อและเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อตามลำดับ (Full Integration) กำหนดให้ขนาดของเอลิเมนต์ที่ปลายรอยร้าวมีค่าเท่ากับ $0.01b = 2.5$ mm โดยใช้โดเมนรูปสี่เหลี่ยมผืนผ้า 3 โดเมนในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลโดยโดเมนแรกมีขนาด (ROR1, ROR2, ROR3, ROR4) เท่ากับ 31.25, 31.25, 31.25, -1.0 mm โดเมนที่สองมีขนาดเท่ากับ 31.25, 156.25, 125.0, -1.0 mm และโดเมนสุดท้ายมีขนาดเท่ากับ 31.25, 375.0, 250.0, -1.0 mm



รูปที่ 6.8 รูปร่างของปัญหาชิ้นทดสอบมาตรฐานแบบ DENT



รูปที่ 6.9 รูปร่างและรายละเอียดของปัญหาชิ้นทดสอบมาตรฐานแบบ DENT ที่นำมาพิจารณา

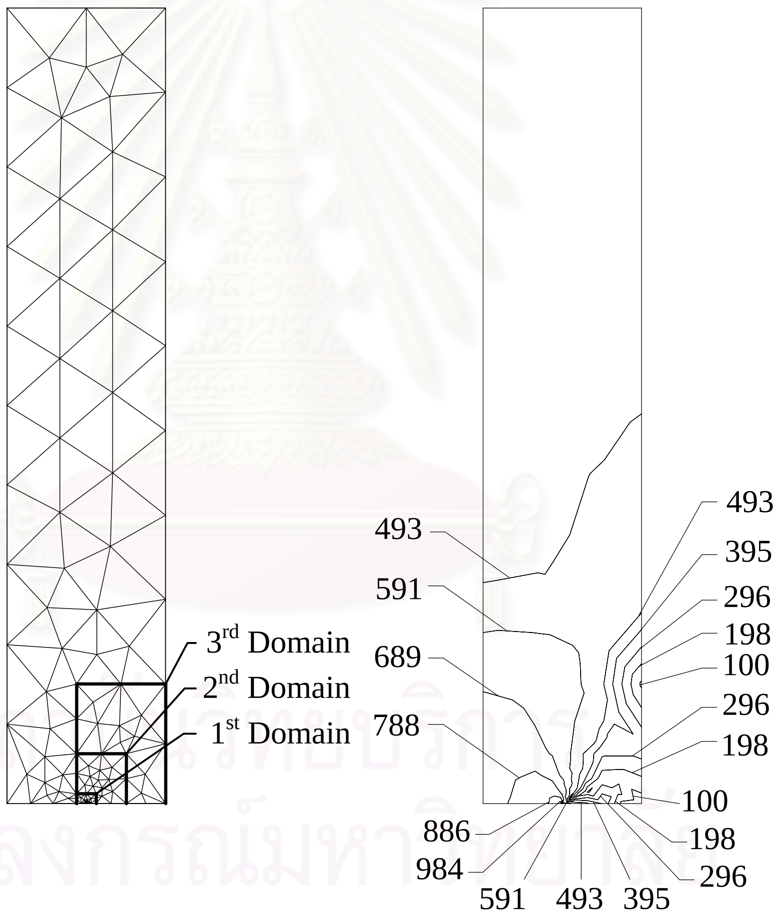
เนื่องจากปัญหามีลักษณะสมมาตรเราจึงใช้เฉพาะหนึ่งในสี่ของชิ้นงานด้านบนขวาดังแสดงในรูปที่ 6.9 เป็นแบบจำลองไฟไนต์เอลิเมนต์ ตารางที่ 6.2 แสดงค่าเปอร์เซ็นต์การขึ้นกับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้จากโปรแกรม JFACTOR เปรียบเทียบกันระหว่างผลลัพธ์จากโครงตาข่ายเริ่มต้นกับผลลัพธ์จากโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติที่ค่าความเค้นดึงระดับต่าง ๆ โดยเอลิเมนต์ที่ปลายรอยร้าวที่ใช้ในโครงตาข่ายทั้งสองยังคงเป็นเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อซึ่งจุดต่อทั้งสามของด้านที่ถูกยุบมารวมกันที่ตำแหน่งปลายรอยร้าวสามารถเคลื่อนตัวได้เป็นอิสระต่อกันขณะที่จุดต่อกลางด้านทั้งสองที่อยู่ติดกับปลายรอยร้าวและจุดต่อกลางเอลิเมนต์ยังคงอยู่ที่ตำแหน่งกึ่งกลาง

ตารางที่ 6.2 ค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่ค่าความเค้นดึงระดับต่าง ๆ สำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ DENT

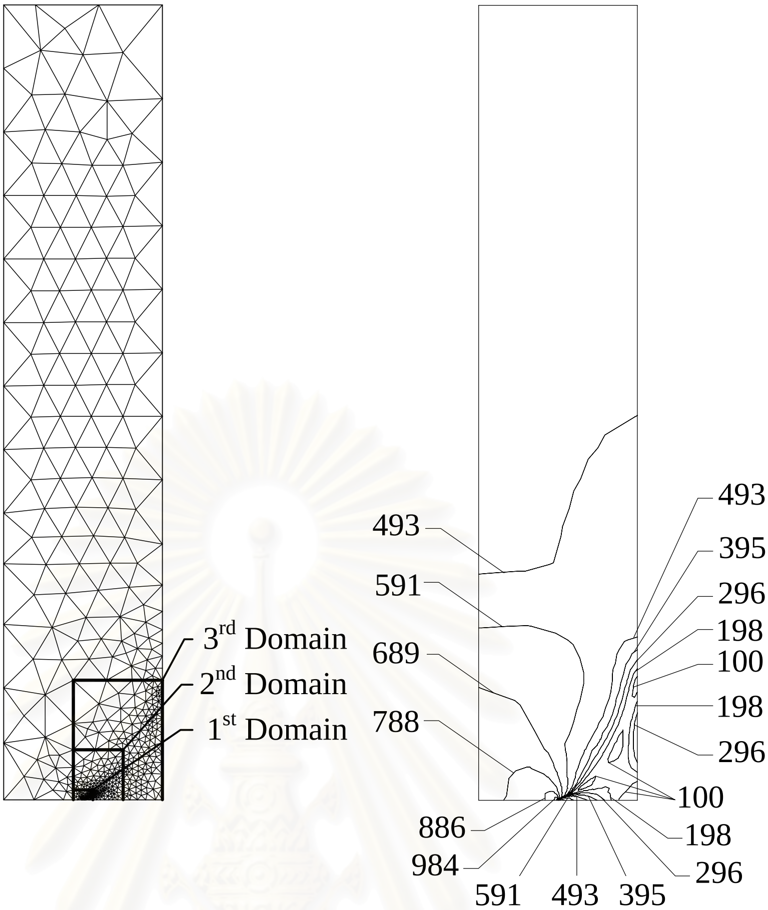
Applied Stress (MPa)	Initial Mesh			Adaptive Mesh		
	Nodes	Elements	Domain Dependence (%)	Nodes	Elements	Domain Dependence (%)
24	447	195	1.91E-01	1362	635	1.47E-02
48			1.80E-01			8.83E-03
72			1.64E-01			8.62E-03
96			1.42E-01			8.50E-03
120			1.23E-01			8.45E-03
144			1.14E-01			8.46E-03
168			1.16E-01			8.47E-03
192			1.28E-01			8.48E-03
216			1.46E-01			8.49E-03
240			1.65E-01			8.63E-03
264			1.88E-01	1750	825	1.15E-02
288			2.17E-01			1.43E-02
312			2.58E-01			1.75E-02
336			3.13E-01			2.08E-02
360			3.78E-01	2114	1007	3.10E-02
384			4.35E-01			3.49E-02
408			4.74E-01			3.86E-02
432			4.99E-01	2204	1046	7.77E-02
456			5.22E-01			8.17E-02
480			5.38E-01	2649	1268	7.74E-02

จากตาราง 6.2 พบว่าโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติให้ผลลัพธ์ที่ขึ้นกับโดเมนน้อยกว่าผลลัพธ์ที่ได้จากโครงตาข่ายเริ่มต้นซึ่งแสดงให้เห็นว่าค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้จากโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัตินั้นจะมีความถูกต้องมากกว่า ในทำนองเดียวกันกับปัญหาที่ผ่านมาโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โครงตาข่ายแรกที่ทำการวิเคราะห์ที่ค่าความเค้นดึง $\sigma_\infty = 24$ MPa ซึ่งประกอบด้วยจำนวนจุดต่อและเอลิเมนต์ทั้งหมดเท่ากับ 1362 จุดต่อและ 635 เอลิเมนต์ตามลำดับนั้นได้จากการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติเข้ากับโครงตาข่ายเริ่มต้นโดยใช้ผลการคำนวณที่ค่าความเค้นดึง $\sigma_\infty = 144$ MPa ซึ่งได้จากการวิเคราะห์โดยใช้โครงตาข่ายเริ่มต้นนี้ จากนั้นทำการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติเข้ากับการวิเคราะห์ที่ค่าความเค้นดึงเท่ากับ 264, 360, 432 และ 480 MPa ตามลำดับเพื่อให้การกระจายตัวของสนามความเค้นวอนมิสเซสที่ได้มีความต่อเนื่องมากขึ้นและผลการคำนวณที่ได้ขึ้นอยู่กับโดเมนน้อยลง รูปที่ 6.10 แสดงลักษณะของโครงตาข่ายเริ่มต้นที่ได้ทำการคำนวณไว้ในตาราง 6.2

ซึ่งประกอบไปด้วย 447 จุดต่อ 195 เอลิเมนต์พร้อมกับโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลทั้ง 3 โดเมนและลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ค่าความเค้นดึง $\sigma_\infty = 480$ MPa เปรียบเทียบกับรูปที่ 6.11 ซึ่งแสดงลักษณะของโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติซึ่งประกอบไปด้วย 2649 จุดต่อ 1268 เอลิเมนต์พร้อมกับลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ค่าความเค้นดึงเดียวกัน โดยโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติจะมีเอลิเมนต์ขนาดเล็กจำนวนมากเรียงตัวกันอยู่ที่บริเวณปลายรอยร้าวเช่นเดียวกับในปัญหาที่ผ่านมาและลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ได้ยังมีความต่อเนื่องมากกว่าผลที่ได้จากการคำนวณโดยใช้โครงตาข่ายเริ่มต้นอีกด้วย

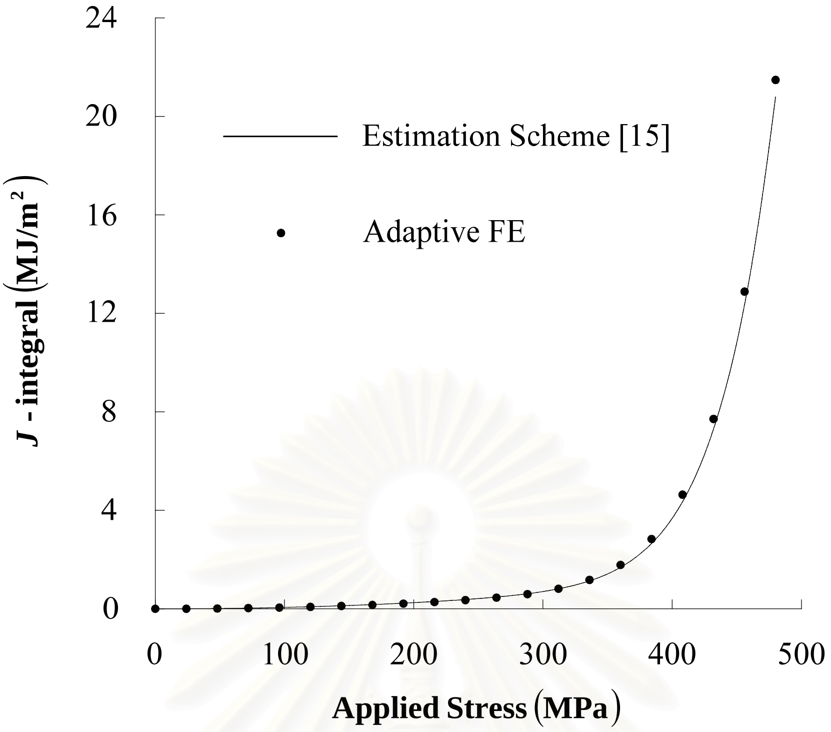


รูปที่ 6.10 ลักษณะโครงตาข่ายเริ่มต้นของปัญหาชิ้นทดสอบมาตรฐานแบบ DENT พร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลและการกระจายตัวของสนามความเค้นวอนมิสเซส, MPa ที่ $\sigma_\infty = 480$ MPa

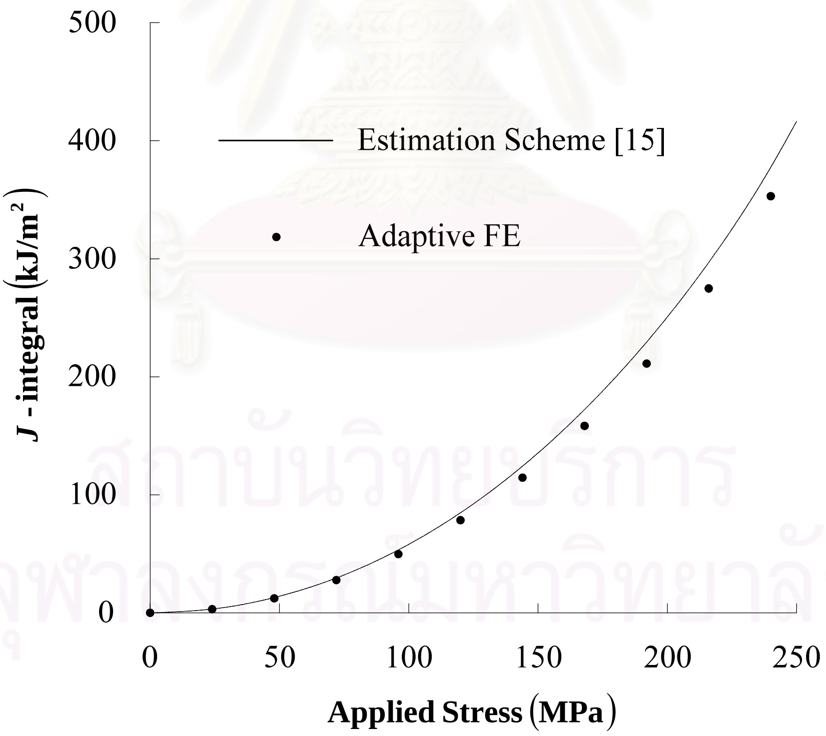


รูปที่ 6.11 ลักษณะโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลและการกระจายตัวของสนามความเค้นวอนมิสเซส, MPa สำหรับปัญหาชิ้นทดสอบมาตรฐานแบบ DENT ที่ $\sigma_{\infty} = 480$ MPa

รูปที่ 6.12 แสดงผลการคำนวณค่าพารามิเตอร์เจอินทิกรัลเฉลี่ยที่ค่าระดับภาระความเค้นดึงต่าง ๆ เปรียบเทียบกันระหว่างผลลัพธ์ที่ได้จากโปรแกรม JFACTOR ที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติกับผลลัพธ์ของ EPRI ซึ่งผลลัพธ์ที่ได้แสดงให้เห็นถึงความสอดคล้องกันเป็นอย่างดี



(ก)

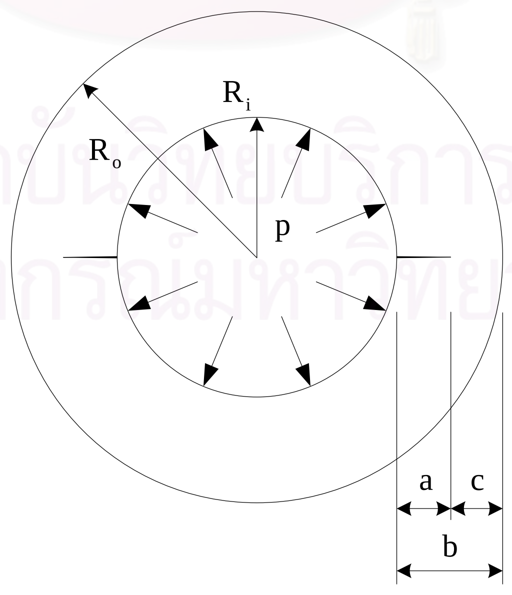


(ข)

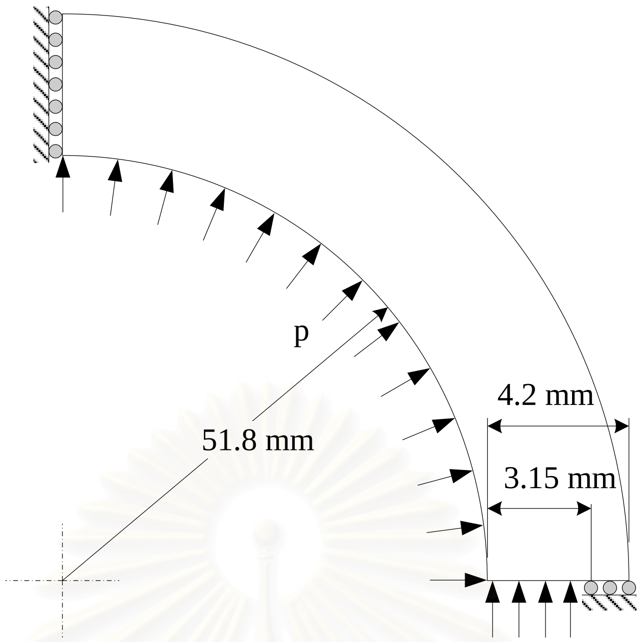
รูปที่ 6.12 ค่าพารามิเตอร์เจอินทิกรัลของชิ้นงานมาตรฐานแบบ DENT สำหรับปัญหาความเค้นในระนาบ (ก) ที่ค่าระดับภาระสูงและ (ข) ที่ค่าระดับภาระต่ำถึงปานกลาง

6.3 ท่อทรงกระบอกที่มีรอยร้าววางตัวตามแนวแกนที่ผิวด้านในภายใต้ความดัน

ปัญหาที่นำมาใช้ตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์ปัญหาต่อไปเป็นปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้เงื่อนไขความเครียดระนาบซึ่งถูกกระทำด้วยแรงดันภายใน p ที่ผิวด้านในของท่อและที่ผิวรอยร้าวดังแสดงในรูปที่ 6.13 กำหนดให้ท่อมีความหนา $b = 4.2$ mm อัตราส่วนความยาวรอยร้าวต่อความหนา $a/b = 0.75$ และอัตราส่วนระยะรัศมีภายในท่อต่อระยะรัศมีภายนอกท่อเท่ากับ $R_i/R_o = 0.925$ ค่าโมดูลัสของความยืดหยุ่น $E = 121.955$ GPa อัตราส่วนปัวซงส์ $\nu = 0.3$ ค่าความเค้นที่จุดคราก $\sigma_o = 463.367$ MPa ค่าคงที่ของวัสดุ $\alpha = 0.06$ และค่ายกกำลังของความเครียด $n = 10.4$ จำนวนจุดเกาส์ที่ใช้ในการอินทิเกรตเชิงตัวเลขพจน์ที่เกี่ยวข้องกับค่าความเค้นตั้งฉากเฉลี่ยสำหรับเอลิเมนต์สามเหลี่ยมและเอลิเมนต์สี่เหลี่ยมที่ปลายรอยร้าวจะเท่ากับ 3 และ 2×2 จุดตามลำดับ ขณะที่การอินทิเกรตพจน์ที่เกี่ยวข้องกับค่าความเค้นดิเวียทอริกสำหรับเอลิเมนต์สามเหลี่ยมและเอลิเมนต์ที่ปลายรอยร้าวจะใช้จำนวนจุดเกาส์เท่ากับ 7 และ 3×3 จุดตามลำดับ กำหนดให้ขนาดของเอลิเมนต์รอบปลายรอยร้าวมีค่าเท่ากับ $0.01c = 0.0105$ mm โดยใช้โดเมนสี่เหลี่ยมผืนผ้าทั้งหมด 3 โดเมนในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัล ขนาดของโดเมน (ROR1, ROR2, ROR3, ROR4) ที่หนึ่งมีค่าเท่ากับ 0.2625, 0.2625, 0.2625, -1.0 mm โดเมนที่สองมีขนาดเท่ากับ 1.05, 0.7875, 0.2625, -1.0 mm โดเมนสุดท้ายมีขนาดเท่ากับ 2.1, 1.8375, 0.2625, -1.0 mm เนื่องจากปัญหามีความสมมาตรเราจึงพิจารณาเฉพาะพื้นที่หนึ่งในสี่ของท่อด้านบนขวาดังแสดงในรูปที่ 6.14 เป็นแบบจำลองไฟไนต์เอลิเมนต์



รูปที่ 6.13 รูปร่างของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดัน



รูปที่ 6.14 รูปร่างและรายละเอียดของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดันที่นำมาพิจารณา

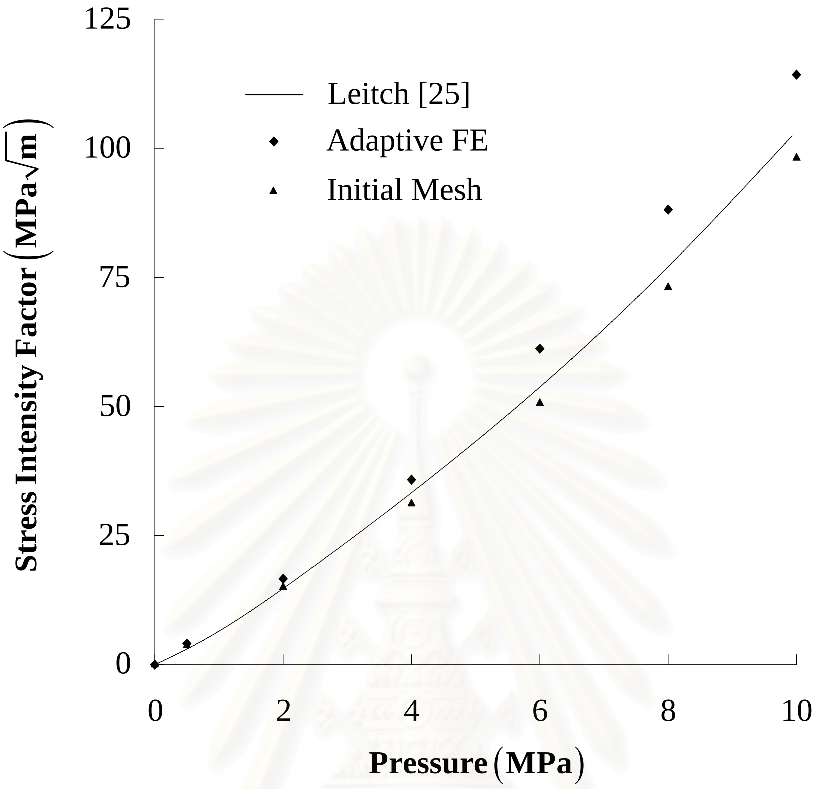
สำหรับปัญหานี้ Leitch [30] ได้ทำการวิเคราะห์ไว้ โดยในการคำนวณของ Leitch นั้นผลลัพธ์ที่ได้จะถูกเขียนอยู่ในรูปค่าพารามิเตอร์ตัวประกอบความเข้มของความเค้น (Stress Intensity Factor) ซึ่งเขียนอยู่ในรูปความสัมพันธ์กับค่าพารามิเตอร์เจอินทิกรัลสำหรับปัญหาความเครียดระนาบดังสมการ

$$K_I = \sqrt{\frac{EJ}{1-\nu^2}} \qquad (8.2)$$

โดยที่ K_I แทนค่าพารามิเตอร์ตัวประกอบความเข้มของความเค้น
J แทนค่าพารามิเตอร์เจอินทิกรัล
E แทนค่าโมดูลัสของความยืดหยุ่น
ν แทนค่าอัตราส่วนปัวซงส์

รูปที่ 6.15 แสดงผลการคำนวณที่ค่าความดันภายในระดับต่าง ๆ เปรียบเทียบกันระหว่างผลการคำนวณของ Leitch กับผลการคำนวณที่ได้จากโปรแกรม JFACTOR ซึ่งประกอบไปด้วยผลลัพธ์ที่ได้จากโครงตาข่ายเริ่มต้นกับผลลัพธ์ที่ได้จากการคำนวณโดยใช้โครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติ เอลิเมนต์ที่ปลายรอยร้าวที่ใช้ในการคำนวณด้วยโปรแกรม JFACTOR เป็นเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อที่จุดต่อทั้งสามของด้านที่ถูกยุบมารวมกันที่

ปลายรอยร้าวสามารถเคลื่อนตัวได้เป็นอิสระต่อกันขณะที่จุดต่อกลางด้านทั้งสองที่อยู่ติดกับปลายรอยร้าวและจุดต่อกลางเอลิเมนต์ยังคงอยู่ที่ตำแหน่งกึ่งกลาง



รูปที่ 6.15 ค่าพารามิเตอร์ตัวประกอบความเข้มของความเค้นที่ระดับความดันต่าง ๆ สำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านใน

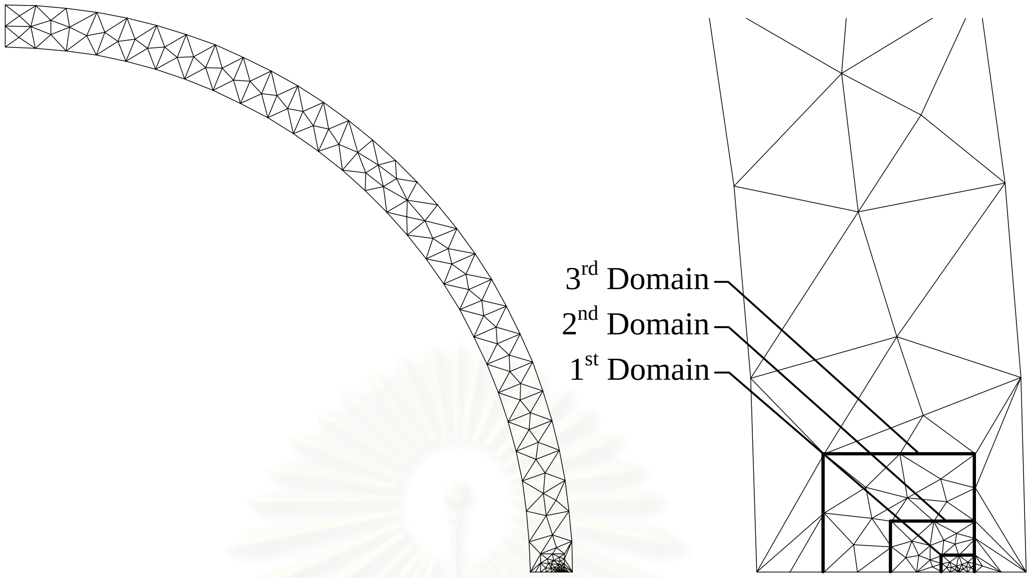
โดยผลการคำนวณของ Leitch จะใกล้เคียงกับผลการคำนวณที่ได้จากโครงตาข่ายเริ่มต้นมากกว่าผลการคำนวณที่ได้จากโครงตาข่ายที่มีการปรับขนาดเอลิเมนต์ เนื่องจากโครงตาข่ายที่ใช้ในการคำนวณของ Leitch นั้นมีลักษณะที่ค่อนข้างหยาบเมื่อเปรียบเทียบกับโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติ นอกจากนี้ค่าคุณสมบัติของวัสดุที่ผู้วิจัยใช้ในบางค่านั้นได้จากการทำ Curve Fitting จากกราฟความสัมพันธ์ระหว่างค่าความเค้นและค่าความเครียดที่ให้ไว้ในงานวิจัยของ Leitch ซึ่งอาจทำให้ค่าคุณสมบัติที่ได้มีความแตกต่างจากค่าที่ Leitch ใช้ในการคำนวณอยู่บ้าง ตารางที่ 6.3 แสดงค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้จากโปรแกรม JFACTOR ที่ค่าระดับความดันต่าง ๆ จากตารางพบว่าผลการคำนวณค่าพารามิเตอร์เจอินทิกรัลที่ได้จากโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติจะให้ผลการคำนวณที่ขึ้นอยู่กับโดเมนน้อยกว่าแม้ว่าผลการคำนวณที่ได้จากโครงตาข่ายเริ่มต้นนั้นจะให้เปอร์เซ็นต์การขึ้นอยู่กับโดเมนที่ไม่มากนักก็ตาม สิ่งที่ผู้วิจัยพบในการวิเคราะห์ปัญหานี้อีกอย่างหนึ่งก็คือ แม้ว่าโครงตาข่ายเริ่มต้นจะให้ผลการคำนวณที่มี

ค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนที่ค่อนข้างต่ำแต่ผลลัพธ์ค่าตัวประกอบความเข้มของความเค้นที่ได้เมื่อเปรียบเทียบกับผลลัพธ์ที่ได้จากโครงตาข่ายที่มีการปรับขนาดเอลิเมนต์โดยอัติโนมัติแล้วมีความแตกต่างกันค่อนข้างมากที่ค่าระดับภาระสูง ๆ ดังนั้นนอกจากจะต้องตรวจสอบค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของผลลัพธ์ค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้ให้มีค่าที่ต่ำแล้วเพื่อให้ได้ผลการคำนวณที่มีความถูกต้องแม่นยำเราจึงจำเป็นต้องทำการตรวจสอบลักษณะการกระจายตัวของค่าความเค้นวอนมิสเซสว่ามีลักษณะต่อเนื่องดีหรือไม่ควบคู่กันไปอีกด้วย (หากโดเมนที่ใช้ในการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัลมีขนาดใกล้เคียงกันมากไม่ว่าโครงตาข่ายที่ใช้ในการคำนวณจะมีความละเอียดหรือหยาบค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนที่ได้จะมีค่าค่อนข้างต่ำเสมอ)

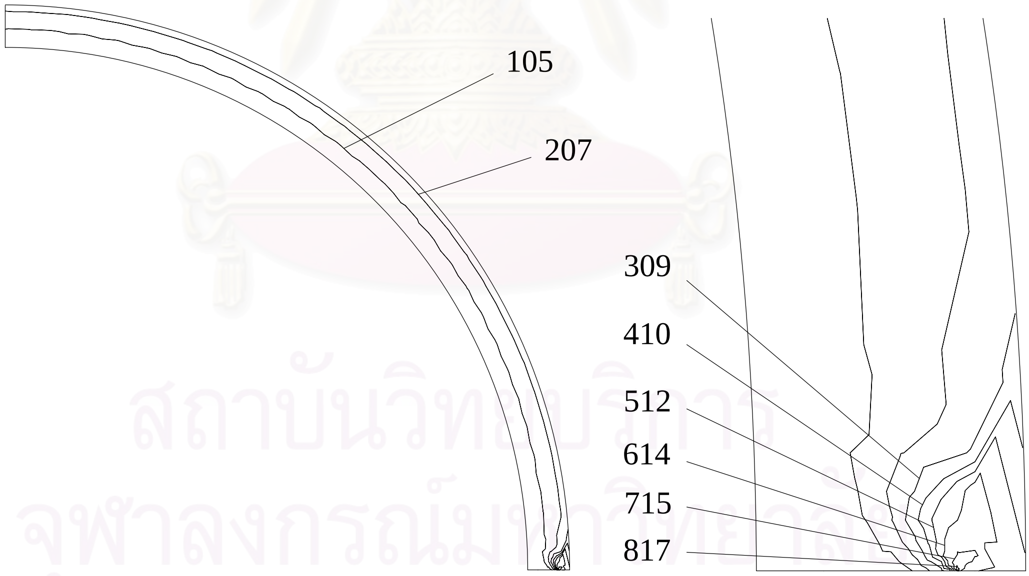
ตารางที่ 6.3 ค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลคำนวณที่ค่าความดันระดับต่าง ๆ สำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าววางตัวตามแนวแกนที่ผิวด้านในภายใต้ความดัน

Internal Pressure (MPa)	Initial Mesh			Adaptive Mesh		
	Nodes	Elements	Domain Dependence (%)	Nodes	Elements	Domain Dependence (%)
0.50	622	270	2.76E-01	1788	810	1.80E-02
2.00			2.50E-01			1.42E-02
4.00			4.11E-01			2.91E-02
6.00			5.57E-01	2606	1210	3.91E-02
8.00			7.51E-01			5.56E-02
10.00			9.15E-01			6.51E-02

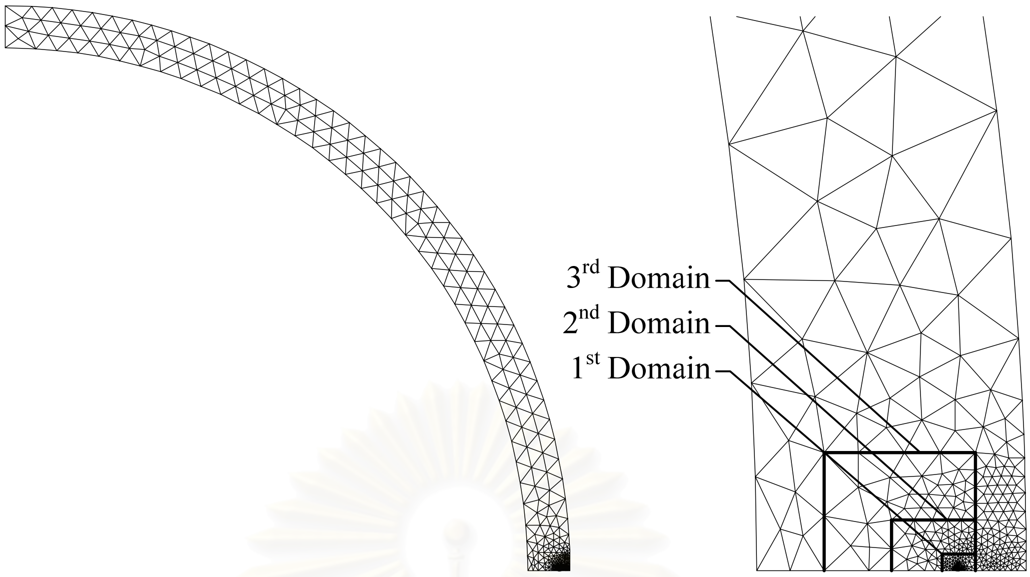
รูปที่ 6.16 และ 6.17 แสดงลักษณะของโครงตาข่ายเริ่มต้นซึ่งประกอบด้วย 622 จุดต่อ 270 เอลิเมนต์พร้อมกับโดเมนที่ใช้ในการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัลและลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ได้จากการคำนวณที่ค่าความดันภายใน $p = 10$ MPa ตามลำดับเปรียบเทียบกับรูปที่ 6.18 และ 6.19 ซึ่งแสดงลักษณะของโครงตาขายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติซึ่งประกอบด้วย 2606 จุดต่อ 1210 เอลิเมนต์และลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสตามลำดับที่ระดับความดันเดียวกัน ในทำนองเดียวกันรูปที่ 6.20 แสดงการเปรียบเทียบสนามความเค้นตามแนวเส้นรอบวง (Hoop Stress) ที่เกิดขึ้นบนโครงตาข่ายทั้งสอง โดยเอลิเมนต์ขนาดเล็กจะเรียงตัวกันอยู่ที่บริเวณปลายรอยร้าวเช่นเดียวกับในปัญหาที่ผ่านมา ในขณะเดียวกันการกระจายตัวของสนามความเค้นที่เกิดขึ้นบนโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติจะมีความต่อเนื่องมากกว่าโครงตาข่ายเริ่มต้นอีกด้วย



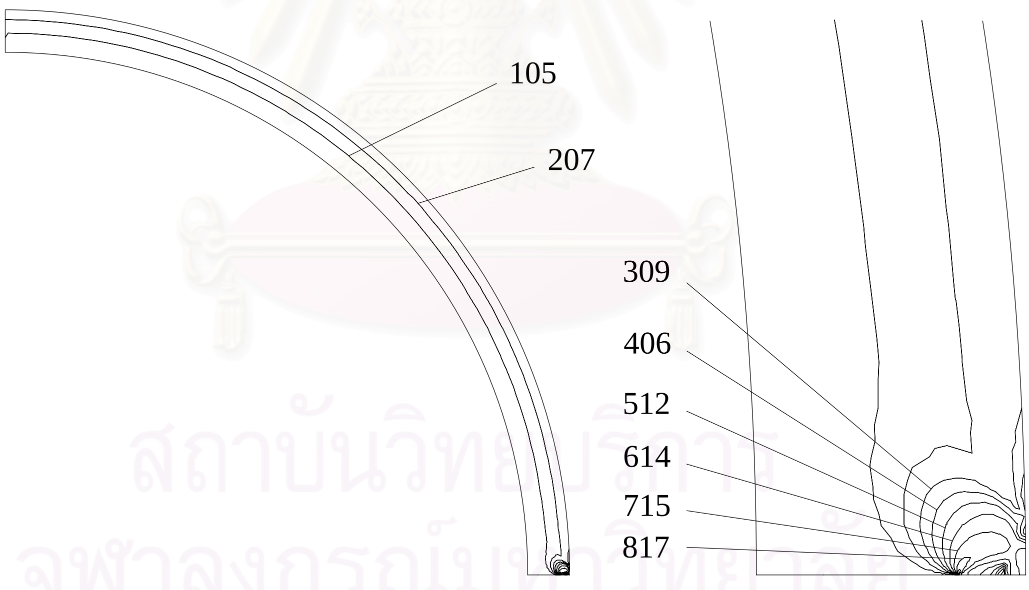
รูปที่ 6.16 ลักษณะของโครงตาข่ายเริ่มต้นพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลสำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดันภายใน



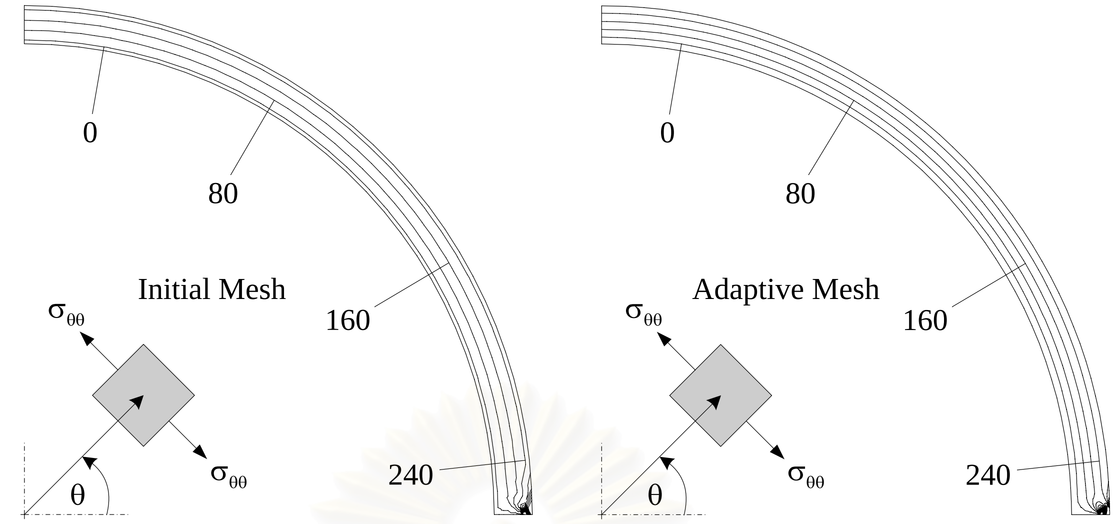
รูปที่ 6.17 ลักษณะการกระจายตัวของค่าความเค้นวอนมิสเซส, MPa สำหรับโครงตาข่ายเริ่มต้นที่ค่าความดัน $p = 10$ MPa ของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดันภายใน



รูปที่ 6.18 ลักษณะของโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติ พร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลสำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดันภายใน



รูปที่ 6.19 ลักษณะการกระจายตัวของค่าความเค้นวอนมิสเซส, MPa สำหรับโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติที่ค่าความดัน $p = 10$ MPa สำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวแกนที่ผิวด้านในภายใต้ความดันภายใน



รูปที่ 6.20 ลักษณะการกระจายตัวของค่าความเค้นตามแนวเส้นรอบวง $\sigma_{\theta\theta}$, MPa สำหรับโครงตาข่ายเริ่มต้นและโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติที่ค่าความดัน $p = 10$ MPa

6.4 ท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความหนาของท่อ

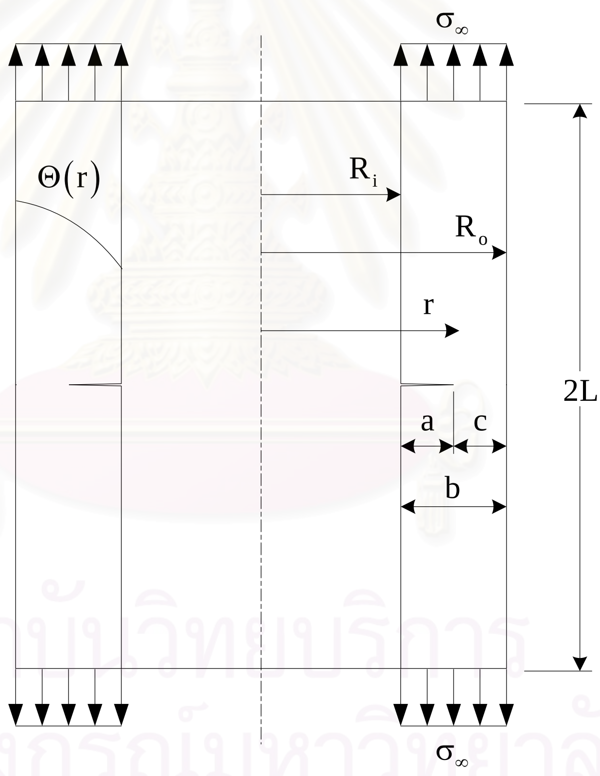
รูปที่ 6.21 แสดงรายละเอียดของปัญหาที่นำมาตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์ซึ่งเป็นท่อทรงกระบอกกลวงที่มีรอยร้าวตามแนวเส้นรอบวงถูกกระทำด้วยภาระความเค้นดึง σ_∞ และมีค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิดังสมการ

$$\Theta(r) \;=\; \left[125 + 100(r - R_i) - 6.25(r - R_i)^2\right] \qquad (8.3)$$

โดยที่ Θ แทนค่าสนามอุณหภูมิ
r แทนค่าระยะตามแนวรัศมีจากแกนสมมาตร
R_i แทนค่าระยะรัศมีภายในของท่อทรงกระบอก

กำหนดให้ท่อมีความหนา $b = 8$ in อัตราส่วนความยาวรอยร้าวต่อความหนา $a/b = 0.25$ อัตราส่วนระยะรัศมีภายในของท่อต่อความหนา $R_i/b = 10$ และอัตราส่วนครึ่งความยาวของท่อต่อความหนา $L/b = 15$ ค่าโมดูลัสของความยืดหยุ่น $E = 30\times10^3$ ksi อัตราส่วนปัวซงส์ $\nu = 0.3$ ค่าความเค้นที่จุดคราก $\sigma_o = 60$ ksi ค่าคงที่ของวัสดุ $\alpha = 0.5$ ค่าสัมประสิทธิ์การขยายตัวเนื่องจากสนามอุณหภูมิ $\kappa = 7.3\times10^{-6}$ in/in/$^\circ$F และค่ายกกำลังของความเครียด $n = 5$ เนื่องจากการกระจายตัวของสนามอุณหภูมิในท่อเป็นแบบอันดับสอง (Quadratic) ดังนั้นจึงใช้ทั้งจำนวนจุด

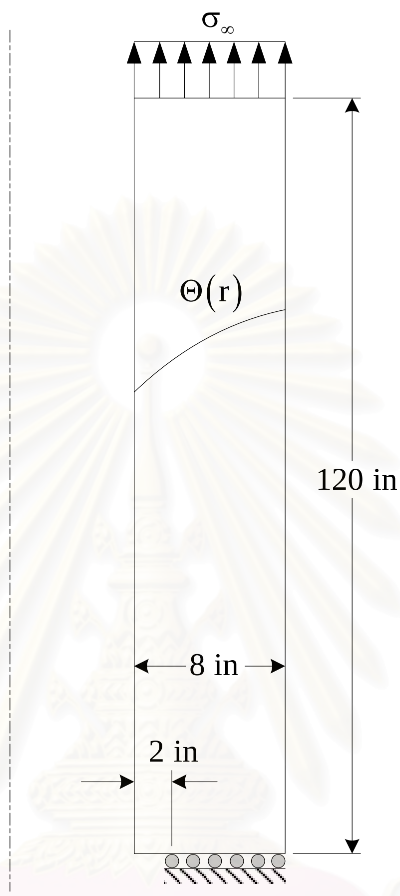
เกาส์เท่ากับ 7 และ 3×3 จุดในการอินทิเกรตเชิงตัวเลขพจน์ที่เกี่ยวข้องกับค่าความเค้นตั้งฉากเฉลี่ยและค่าเค้นดิเวียทอริกสำหรับเอลิเมนต์สามเหลี่ยมหกจุดต่อและเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อตามลำดับ เอลิเมนต์ที่ปลายรอยร้าวที่ใช้ในการคำนวณเป็นเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อที่จุดต่อทั้งสามของด้านที่ถูกยุบมารวมกันที่ตำแหน่งปลายรอยร้าวสามารถเคลื่อนตัวได้เป็นอิสระต่อกัน ขณะที่จุดต่อกลางด้านทั้งสองที่อยู่ติดกับปลายรอยร้าวและจุดต่อกลางเอลิเมนต์ยังคงอยู่ที่ตำแหน่งกึ่งกลาง โดยขนาดของเอลิเมนต์ที่ปลายรอยร้าวที่ใช้มีค่าเท่ากับ $0.01c = 0.06$ in โดยที่ c แทนความหนาที่เหลือของท่อ (Ligament Length) โดเมนรูปสี่เหลี่ยมผืนผ้า 3 โดเมนซึ่งโดเมนแรกมีขนาด (ROR1, ROR2, ROR3, ROR4) เท่ากับ 31.25, 31.25, 31.25, -1.0 in โดเมนที่สองมีขนาดเท่ากับ 31.25, 156.25, 125.0, -1.0 in และโดเมนสุดท้ายมีขนาดเท่ากับ 31.25, 375.0, 250.0, -1.0 in ถูกใช้ในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัล



รูปที่ 6.21 รูปร่างของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ

เนื่องจากปัญหามีลักษณะสมมาตรรอบแกน (Axisymmetric Condition) ดังนั้นเราจึงใช้เฉพาะหนึ่งในสี่ของพื้นที่หน้าตัดด้านบนขวาของท่อดังแสดงในรูปที่ 6.22 เป็นแบบจำลองไฟไนต์เอลิเมนต์ ตารางที่ 6.4 แสดงค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่คำนวณ

ได้จากโปรแกรม JFACTOR เปรียบเทียบกันระหว่างผลลัพธ์ที่ได้จากโครงตาข่ายเริ่มต้นกับผลลัพธ์ที่ได้จากโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติที่ค่าความเค้นดึงระดับต่าง ๆ



รูปที่ 6.22 รูปร่างและรายละเอียดของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิที่นำมาพิจารณา

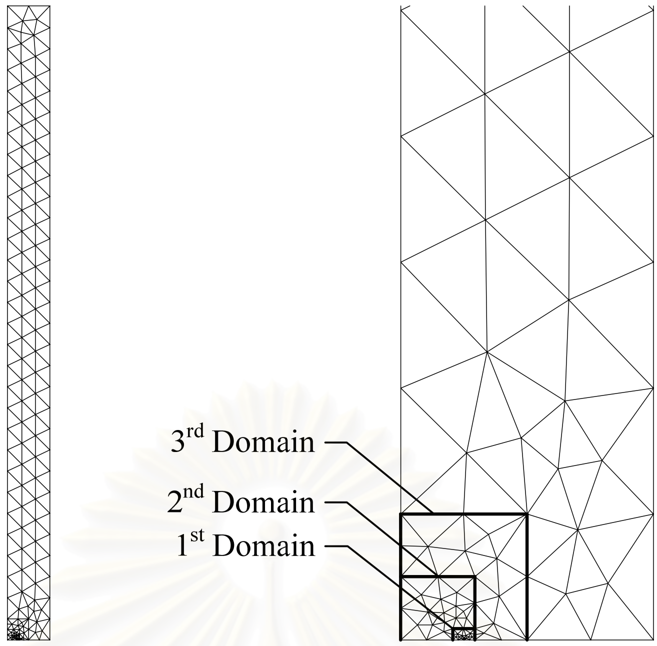
จากตารางพบว่าโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัตินั้นจะให้ผลลัพธ์ที่ขึ้นกับโดเมนน้อยกว่าโครงตาข่ายเริ่มต้น โครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โครงตาข่ายแรกซึ่งทำการวิเคราะห์ที่ค่าความเค้นดึง $\sigma_\infty = 0$ ksi สอดคล้องกับภาระที่มากระทำกับชิ้นงานเฉพาะค่าความเค้นเนื่องมาจากการกระจายตัวของสนามอุณหภูมิตามความหนาของท่อเท่านั้นได้จากการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติเข้ากับโครงตาข่ายเริ่มต้นโดยใช้ผลการคำนวณที่ค่าความเค้นดึง $\sigma_\infty = 9$ ksi จากนั้นทำการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติในการคำนวณที่ค่าความเค้นดึงเท่ากับ 18, 40.5, 63 และ 90 ksi ตามลำดับโดยในการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์แต่ละครั้งจะพยายาม

ปรับปรุงให้การกระจายตัวของสนามความเค้นวอนมิสเซสมีความต่อเนื่องมากขึ้นและค่าเปอร์เซ็นต์การขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลมีค่าไม่เกิน 0.1%

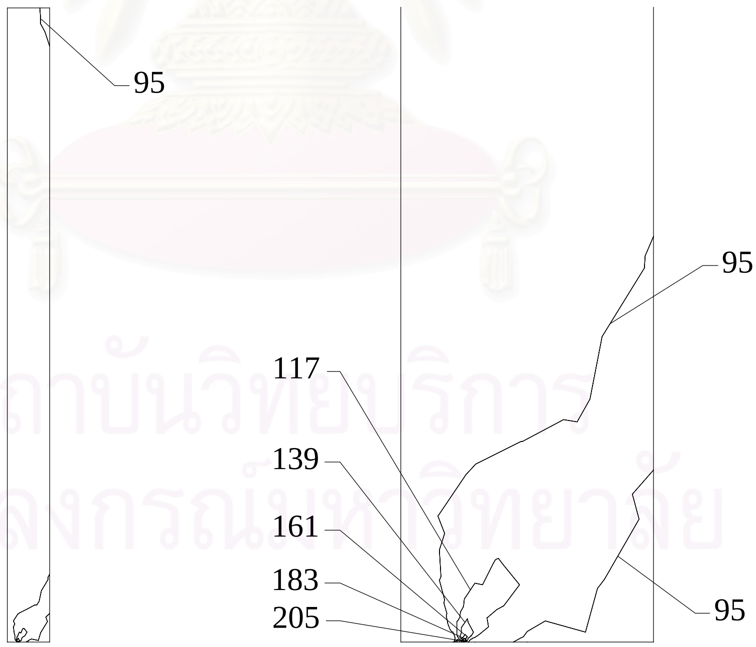
ตารางที่ 6.4 ค่าเปอร์เซ็นต์การขึ้นกับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่ค่าความเค้นดึงระดับต่าง ๆ สำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ

Applied Stress (ksi)	Initial Mesh			Adaptive Mesh		
	Nodes	Elements	Domain Dependence (%)	Nodes	Elements	Domain Dependence (%)
0.00	677	295	2.96E-01	1481	670	6.04E-02
4.50			3.15E-01			6.49E-02
9.00			3.38E-01			6.93E-02
13.50			3.64E-01			7.35E-02
18.00			3.92E-01	1971	908	7.04E-02
22.50			4.22E-01			7.58E-02
27.00			4.54E-01			8.13E-02
31.50			4.87E-01			8.68E-02
36.00			5.21E-01			9.24E-02
40.50			5.58E-01	2223	1029	4.49E-02
45.00			5.97E-01			4.73E-02
49.50			6.37E-01			4.98E-02
54.00			7.00E-01			5.24E-02
58.50			7.77E-01			5.51E-02
63.00			8.59E-01	2712	1269	2.96E-02
67.50			9.45E-01			3.10E-02
72.00			1.03E+00			3.25E-02
76.50			1.13E+00			3.39E-02
81.00			1.22E+00			3.52E-02
85.50			1.30E+00			2.96E-02
90.00			1.39E+00	3021	1426	1.44E-02

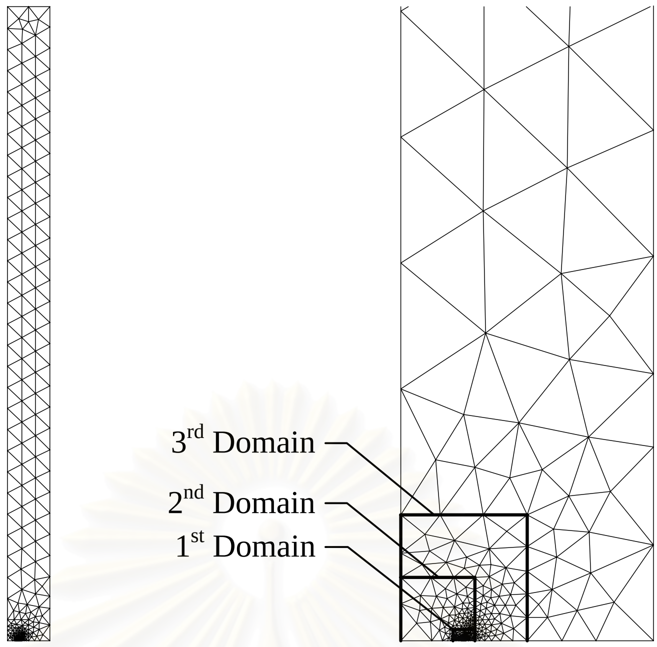
รูปที่ 6.23 และ 6.24 แสดงลักษณะของโครงตาข่ายเริ่มต้นซึ่งประกอบด้วย 677 จุดต่อ 295 เอลิเมนต์พร้อมทั้งโดเมนที่ใช้ในการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัลและลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ค่าภาระความเค้นดึง $\sigma^{\infty} = 90$ ksi ตามลำดับเปรียบเทียบกับรูปที่ 6.25 และ 6.26 ซึ่งแสดงลักษณะของโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติซึ่งประกอบด้วย 3021 จุดต่อ 1426 เอลิเมนต์และลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ค่าภาระความเค้นดึงเดียวกันตามลำดับ



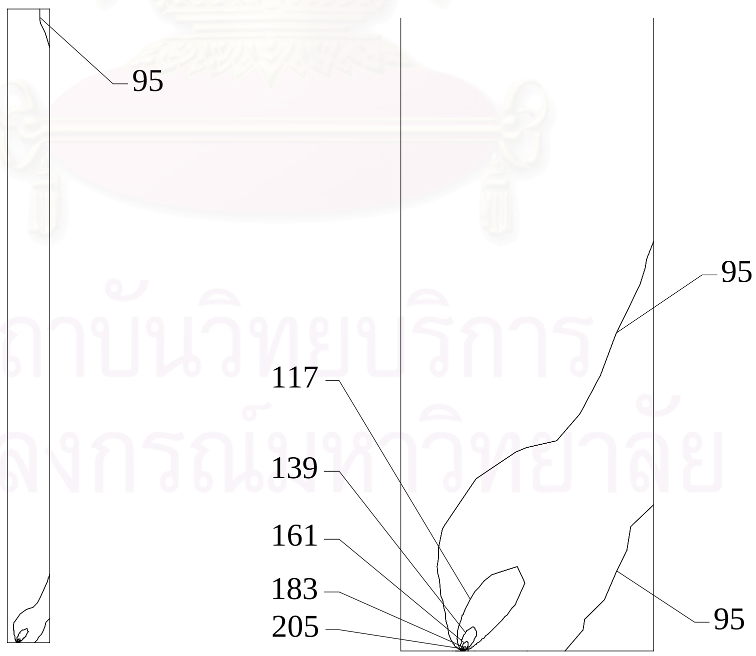
รูปที่ 6.23 ลักษณะโครงตาข่ายเริ่มต้นพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัล สำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึง และค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ



รูปที่ 6.24 ลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซส, ksi สำหรับโครงตาข่ายเริ่มต้น ที่ค่าภาระความเค้นดึง $\sigma^{\infty} = 90$ ksi ของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวง

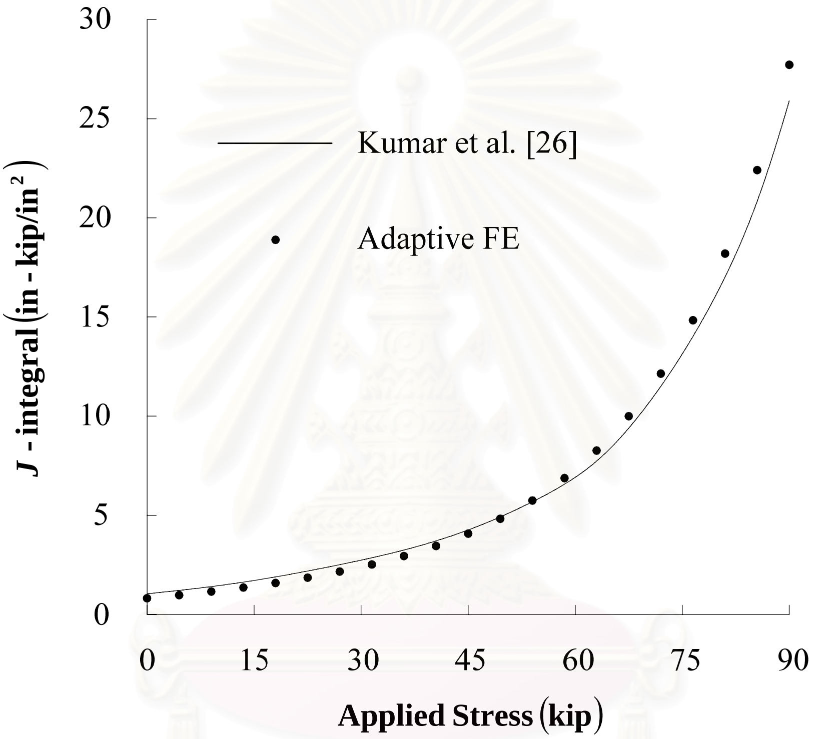


รูปที่ 6.25 ลักษณะโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลสำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ



รูปที่ 6.26 ลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซส, ksi สำหรับโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติที่ค่าภาระความเค้นดึง $\sigma^{\infty} = 90$ ksi ของปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวง

โครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติจะมีเอลิเมนต์ขนาดเล็กจำนวนมากเรียงตัวกันอยู่ที่บริเวณปลายรอยร้าวเช่นเดียวกับในปัญหาที่ผ่านมาและลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ได้ก็มีความต่อเนื่องมากกว่าโครงตาข่ายเริ่มต้นอีกด้วย รูปที่ 6.27 แสดงผลการคำนวณค่าพารามิเตอร์เจอินทิกรัลเฉลี่ยที่ค่าระดับภาระความเค้นดึงต่าง ๆ เปรียบเทียบระหว่างผลลัพธ์ที่คำนวณได้จากโปรแกรม JFACTOR ที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติกับผลลัพธ์ที่ได้จากการคำนวณของ Kumar et al. [31] ซึ่งผลลัพธ์ที่ได้นั้นมีความสอดคล้องกันเป็นอย่างดี



รูปที่ 6.27 ค่าพารามิเตอร์เจอินทิกรัลสำหรับปัญหาท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ

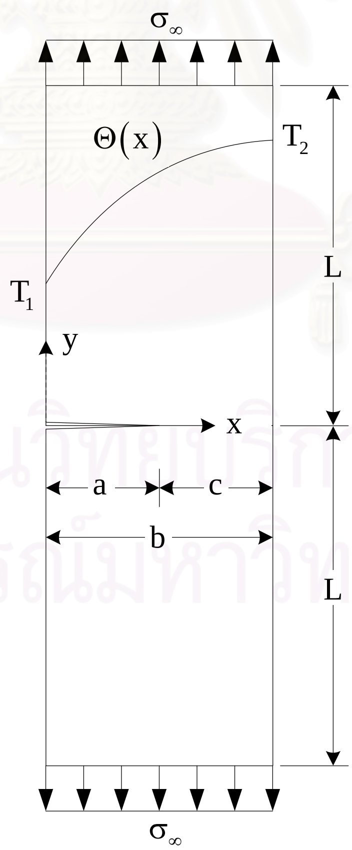
6.5 แผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความกว้างของแผ่น

รูปที่ 6.28 แสดงรายละเอียดของปัญหาที่นำมาตรวจสอบความถูกต้องของโปรแกรมคอมพิวเตอร์ซึ่งเป็นแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวภายใต้เงื่อนไขความเค้นในระนาบ

ซึ่งถูกกระทำด้วยภาระความเค้นดึง σ_∞ ที่ปลายทั้งสองข้างและมีค่าความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิดังสมการ

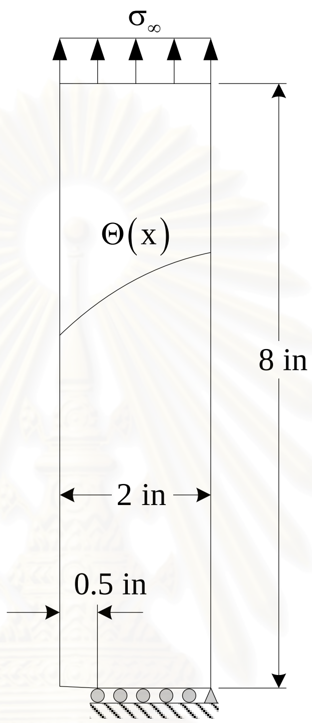
$$\Theta(x) = \left[250 + 800x - 200x^2\right] \tag{8.4}$$

กำหนดให้ชิ้นงานมีความกว้าง $b = 2$ in อัตราส่วนความยาวรอยร้าวต่อความกว้าง $a/b = 0.25$ อัตราส่วนครึ่งความยาวของชิ้นงานต่อความกว้าง $L/b = 4$ และความหนาของแผ่น $t = 1$ in โดยค่าคุณสมบัติของวัสดุทั้งหมดมีค่าเท่ากับกรณีท่อทรงกระบอกที่มีรอยร้าวตามแนวเส้นรอบวงภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความหนาของท่อทุกประการ เนื่องจากความซับซ้อนของสมการไฟไนต์เอลิเมนต์ภายใต้เงื่อนไขความเค้นในระนาบและความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิแบบอันดับสองเกิดขึ้นในชิ้นงาน ดังนั้นจึงใช้ทั้งจำนวนจุดเกาส์เท่ากับ 7 และ 3×3 จุดในการอินทิเกรตเชิงตัวเลขพจน์ที่เกี่ยวข้องกับค่าความเค้นตั้งฉากเฉลี่ยและค่าเค้นดิเวียทอริกสำหรับเอลิเมนต์สามเหลี่ยมหกจุดต่อและเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อตามลำดับ กำหนดให้ขนาดของเอลิเมนต์ที่ปลายรอยร้าวมีค่าเท่ากับ $0.01c = 0.015$ in โดยที่ c แทนความกว้างที่เหลือของแผ่นสี่เหลี่ยม



รูปที่ 6.28 รูปร่างของปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความกว้างของแผ่น

โดเมนรูปสี่เหลี่ยมผืนผ้า 3 โดเมนถูกใช้ในการคำนวณค่าพารามิเตอร์เจอินทิกรัล โดยโดเมนแรกมีขนาด (ROR1, ROR2, ROR3, ROR4) เท่ากับ 0.25, 0.25, 0.25, -1.0 in โดเมนที่สองมีขนาดเท่ากับ 0.5, 0.5, 0.25, -1.0 in และโดเมนสุดท้ายมีขนาดเท่ากับ 0.5, 0.75, 0.5, -1.0 in เนื่องจากปัญหามีลักษณะสมมาตรจึงใช้เฉพาะครึ่งหนึ่งของชิ้นงานด้านบนดังแสดงในรูปที่ 6.29 เป็นแบบจำลองไฟไนต์เอลิเมนต์



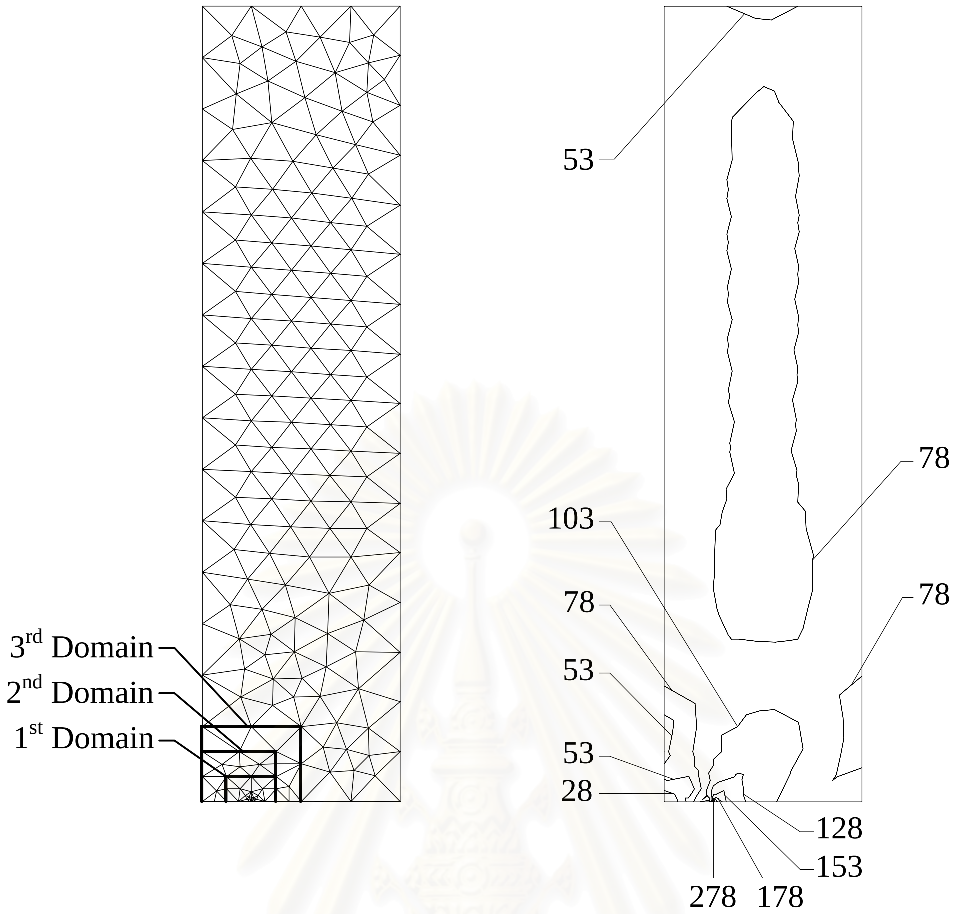
รูปที่ 6.29 รูปร่างและรายละเอียดของปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความกว้างของแผ่นที่นำมาพิจารณา

ตารางที่ 6.5 แสดงค่าเปอร์เซ็นต์การขึ้นกับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้จากโปรแกรม JFACTOR เปรียบเทียบกันระหว่างผลลัพธ์จากโครงตาข่ายเริ่มต้นกับผลลัพธ์จากโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติที่ค่าความเค้นดึงระดับต่าง ๆ โดยเอลิเมนต์ที่ปลายรอยร้าวที่ใช้ในโครงตาข่ายทั้งสองยังคงเป็นเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อซึ่งจุดต่อทั้งสามของด้านที่ถูกยุบมารวมกันที่ตำแหน่งปลายรอยร้าวสามารถเคลื่อนตัวได้เป็นอิสระต่อกันขณะที่จุดต่อกลางด้านทั้งสองที่อยู่ติดกับปลายรอยร้าวและจุดต่อกลางเอลิเมนต์ยังคงอยู่ที่ตำแหน่งกึ่งกลาง

ตารางที่ 6.5 ค่าเปอร์เซ็นต์การขึ้นกับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่ค่าความเค้นดึงระดับต่าง ๆ สำหรับปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความกว้างของแผ่น

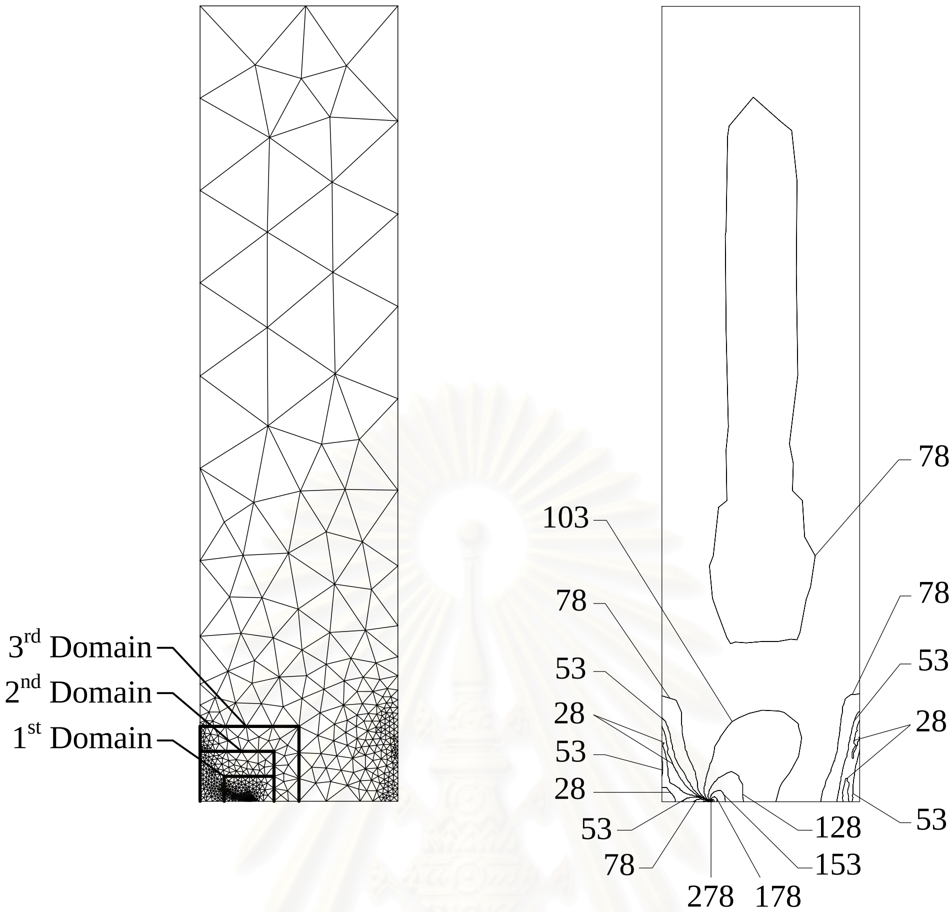
Applied Stress (ksi)	Initial Mesh			Adaptive Mesh		
	Nodes	Elements	Domain Dependence (%)	Nodes	Elements	Domain Dependence (%)
0.00	736	337	6.35E-01	1749	815	1.08E-02
7.90			3.54E-01			6.75E-03
15.80			2.32E-01			1.04E-02
23.70			1.61E-01			1.25E-02
31.60			1.11E-01			1.40E-02
39.50			1.04E-01			1.53E-02
47.40			9.82E-02			1.64E-02
55.30			9.39E-02			1.73E-02
63.20			9.09E-02			1.78E-02
71.10			9.78E-02			1.82E-02
79.00			1.11E-01			1.85E-02
86.90			1.23E-01			1.86E-02
94.80			1.35E-01			1.88E-02
102.70			1.45E-01			1.89E-02
110.60			1.54E-01			1.91E-02
118.50			1.63E-01	2427	1149	1.92E-02
126.40			1.70E-01			1.93E-02
134.30			1.76E-01			1.93E-02
142.20			1.82E-01			1.94E-02
150.10			1.87E-01			1.94E-02
158.00			1.91E-01	3304	1577	2.07E-02

จากตารางที่ 6.5 เราพบว่าโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติจะให้ผลลัพธ์ค่าพารามิเตอร์เจอินทิกรัลที่ขึ้นกับโดเมนน้อยกว่าผลลัพธ์ที่ได้จากโครงตาข่ายเริ่มต้นซึ่งแสดงให้เห็นว่าผลลัพธ์ค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้นั้นมีความถูกต้องมากกว่าตามไปด้วย



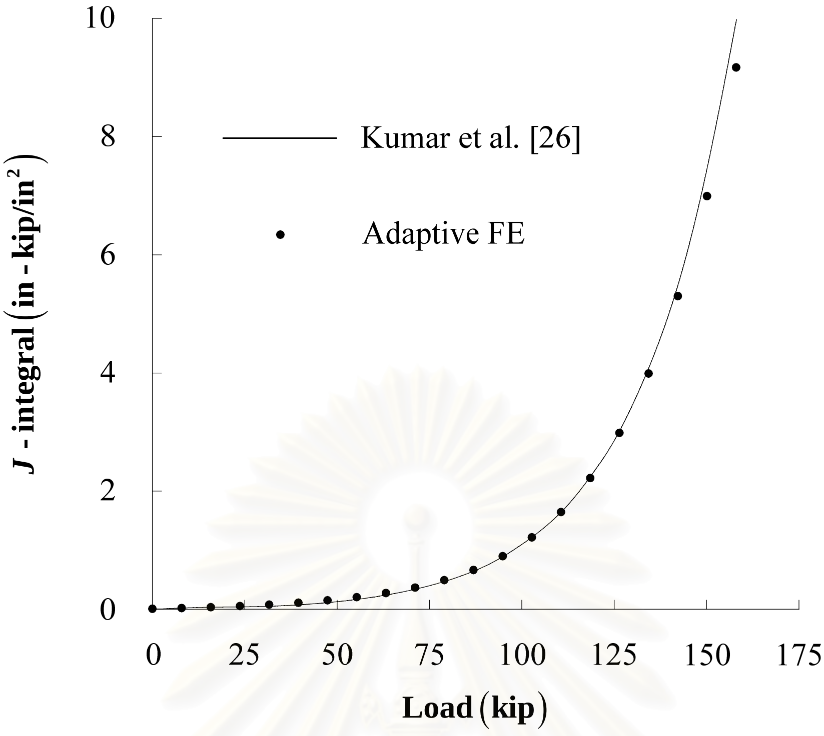
รูปที่ 6.30 ลักษณะโครงตาข่ายเริ่มต้นพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลสำหรับปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียวและการกระจายตัวของสนามความเค้นวอนมิสเซส, ksi ที่ค่าความเค้นดึง $\sigma_{\infty} = 79$ ksi

รูปที่ 6.30 แสดงลักษณะของโครงตาข่ายเริ่มต้นที่ได้ทำการคำนวณไว้ในตารางที่ 6.5 ซึ่งประกอบไปด้วย 736 จุดต่อ 337 เอลิเมนต์พร้อมกับโดเมนที่ใช้ในการอินทิเกรตหาค่าพารามิเตอร์เจอินทิกรัลและลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ค่าความเค้นดึง $\sigma_{\infty} = 79$ ksi เปรียบเทียบกับรูปที่ 6.31 ซึ่งแสดงลักษณะของโครงตาข่ายที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติซึ่งประกอบไปด้วย 3304 จุดต่อ 1577 เอลิเมนต์พร้อมกับลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ค่าความเค้นดึงระดับเดียวกัน

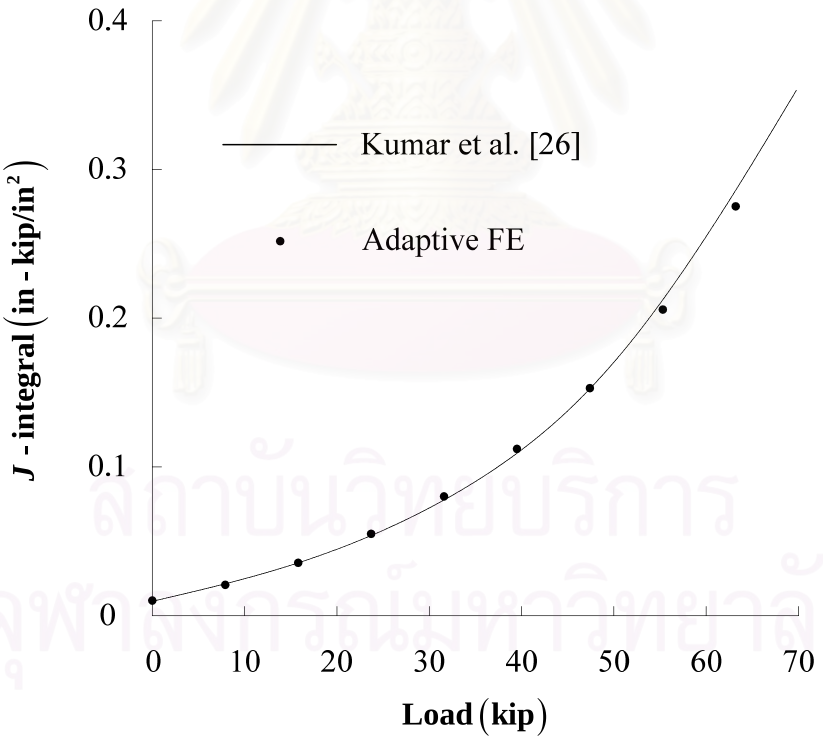


รูปที่ 6.31 ลักษณะโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติพร้อมโดเมนที่ใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลและการกระจายตัวของสนามความเค้นวอนมิสเซส, ksi ที่ค่าความเค้นดึง $\sigma_\infty = 79$ ksi ของปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียว

จากรูปพบว่าโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติจะมีเอลิเมนต์ขนาดเล็กจำนวนมากเรียงตัวกันอยู่ที่บริเวณปลายรอยร้าวและบริเวณอื่นที่มีค่าอนุพันธ์อันดับสองของค่าความเค้นวอนมิสเซสขนาดสูงเช่นเดียวกับปัญหาที่ผ่านมา นอกจากนั้นลักษณะการกระจายตัวของสนามความเค้นวอนมิสเซสที่ได้ยังมีความต่อเนื่องมากกว่าผลลัพธ์ที่ได้จากโครงตาข่ายเริ่มต้นอีกด้วย รูปที่ 6.32 แสดงผลการคำนวณค่าพารามิเตอร์เจอินทิกรัลเฉลี่ยที่ระดับภาระความเค้นดึงต่าง ๆ เปรียบเทียบกันระหว่างผลลัพธ์ที่คำนวณได้จากโปรแกรม JFACTOR ที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัติโนมัติกับผลลัพธ์ที่ได้จากการคำนวณของ Kumar et al. [31] ซึ่งผลลัพธ์ที่ได้นั้นมีความสอดคล้องกันเป็นอย่างดี



(ก)



(ข)

รูปที่ 6.32 ค่าพารามิเตอร์เจอินทิกรัลของปัญหาแผ่นสี่เหลี่ยมแบนที่มีรอยร้าวที่ขอบข้างเดียว ภายใต้ภาระความเค้นดึงและการกระจายตัวของอุณหภูมิตามความกว้างของแผ่น ภายใต้เงื่อนไขความเค้นในระนาบ (ก) ที่ค่าแรงดึงระดับสูง และ (ข) ที่ค่าแรงดึงระดับต่ำถึงปานกลาง

6.6 บทสรุป

ในบทนี้โปรแกรม JFACTOR ที่ได้ประดิษฐ์ขึ้นได้ถูกตรวจสอบโดยนำไปวิเคราะห์ปัญหาชิ้นงานที่มีรอยร้าวมาตรฐานต่าง ๆ ภายใต้เงื่อนไขความเค้นระนาบ ความเครียดระนาบและสมมาตรรอบแกนโดยภาระที่มากระทำกับชิ้นงานนั้นประกอบด้วยภาระความเค้นดึง ภาระความเค้นที่ผิวรอยร้าวและภาระความเค้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ โดยเริ่มจากการเปรียบเทียบความถูกต้องของผลการค่าพารามิเตอร์เจอินทิกรัลจากโครงตาข่ายเริ่มต้นและโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติโดยอาศัยคุณสมบัติความไม่ขึ้นกับโดเมนของค่าพารามิเตอร์เจอินทิกรัลที่เขียนอยู่ในรูปอินทิเกรตบนพื้นที่โดเมน จากนั้นจึงเปรียบเทียบผลการคำนวณค่าพารามิเตอร์เจอินทิกรัลที่ได้จากโครงตาข่ายที่มีการประยุกต์เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติกับผลการคำนวณที่มีในงานวิจัยอื่น ๆ ซึ่งผลการเปรียบเทียบแสดงให้เห็นถึงความสอดคล้องกันเป็นอย่างดี

บทที่ 7

บทสรุป ปัญหาที่พบและข้อเสนอแนะ

7.1 บทสรุปรวม

วิทยานิพนธ์นี้ได้แสดงวิธีการประยุกต์ใช้ระเบียบวิธีไฟไนต์เอลิเมนต์ในการแก้ปัญหารอยร้าวในวัตถุแบบอิลาสติก-พลาสติกสำหรับปัญหาสองมิติและสมมาตรรอบแกนซึ่งรวมผลที่เกิดจากภาระทั้งความเค้นดึง ความเครียดเริ่มต้นเนื่องจากอุณหภูมิและแรงวัตถุ โดยในบทที่ 2 ได้กล่าวถึงความรู้พื้นฐานและทฤษฎีกลศาสตร์การแตกหักแบบอิลาสติก-พลาสติกซึ่งเน้นไปที่การคำนวณหาค่าพารามิเตอร์เจอินทิกรัลเป็นพารามิเตอร์หลักที่ใช้ในการตรวจสอบความรุนแรงของสนามความเค้นและสนามความเครียดที่บริเวณปลายรอยร้าว โดยภายในบทนี้ได้อธิบายถึงความหมายทางกายภาพ ความเป็นมา ตลอดจนวิธีการประดิษฐ์สมการที่ใช้หาค่าพารามิเตอร์เจอินทิกรัลอย่างละเอียด บทที่ 3 ได้อธิบายถึงการประดิษฐ์สมการไฟไนต์เอลิเมนต์เพื่อใช้วิเคราะห์ความรุนแรงของสนามความเค้นและความเครียดที่เกิดขึ้นกับชิ้นงานที่มีรอยร้าวโดยใช้ทฤษฎีงานเสมือน จากนั้นได้แสดงการสร้างสมการเพื่อใช้คำนวณหาค่าพารามิเตอร์เจอินทิกรัลจากสมการไฟไนต์เอลิเมนต์ที่ได้ประดิษฐ์ขึ้นโดยแบ่งการคำนวณออกเป็นพจน์ต่าง ๆ ที่เกี่ยวข้องกับค่าความเค้นตั้งฉากเฉลี่ย ค่าความเค้นดิเวียทอริกและค่าความเค้นดึงที่ผิวรอยร้าว บทที่ 4 ได้อธิบายถึงหลักการพื้นฐานของเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติตลอดจนการนำไปประยุกต์ใช้ร่วมกับระเบียบวิธีไฟไนต์เอลิเมนต์ที่ได้ประดิษฐ์ขึ้นเพื่อปรับปรุงผลการคำนวณให้มีความถูกต้องมากขึ้นโดยใช้เอลิเมนต์ขนาดเล็กเฉพาะในบริเวณที่ผลเฉลยมีความผิดพลาดสูงในขณะเดียวกันจะใช้เอลิเมนต์ขนาดใหญ่ในบริเวณที่ผลเฉลยมีความผิดพลาดต่ำทำให้ประหยัดหน่วยความจำและระยะเวลาในการคำนวณลงเนื่องจากไม่จำเป็นต้องใช้เอลิเมนต์ขนาดเล็กเป็นจำนวนมากในการแก้ปัญหา นอกจากนั้นหลักการของเทคนิคการถ่ายทอดผลเฉลยระยะการเคลื่อนตัวจากโครงตาข่ายก่อนการปรับขนาดเอลิเมนต์ไปสู่โครงตาข่ายที่ทำการปรับขนาดเอลิเมนต์แล้วเพื่อให้การคำนวณสามารถดำเนินต่อเนื่องไปได้โดยไม่ต้องเริ่มต้นการคำนวณใหม่ที่ค่าภาระเริ่มต้นทุกครั้งหลังการปรับขนาดเอลิเมนต์ยังได้ถูกอธิบายอย่างละเอียดในบทนี้ บทที่ 5 ได้อธิบายหลักการทำงานของโปรแกรม JFACTOR ที่ได้ถูกประดิษฐ์ขึ้นตลอดจนหน้าที่การทำงานของโปรแกรมย่อยต่าง ๆ ที่ประกอบกันขึ้นเป็นโปรแกรม JFACTOR นอกจากนี้ยังได้อธิบายการสร้างไฟล์ข้อมูลนำเข้าเพื่อใช้งานกับโปรแกรม JFACTOR ตลอดจนลักษณะของไฟล์ที่ได้หลังการคำนวณซึ่งประกอบไปด้วยไฟล์ที่นำไปใช้สำหรับการปรับขนาดเอลิเมนต์ด้วยโปรแกรม FEMESH v2.1.152 ซึ่งมีนามสกุล ".out" ไฟล์ที่นำไปใช้สำหรับการแสดงผลกราฟฟิกบนจอคอมพิวเตอร์ซึ่งมีนามสกุล ".plt" และไฟล์ที่นำไปใช้เพื่อเริ่มต้นการ

คำนวณที่สถานะการคำนวณที่ได้ทำการวิเคราะห์ไปแล้วซึ่งมีนามสกุล ".res" ในบทที่ 6 ได้ทำการตรวจสอบความถูกต้องของโปรแกรม JFACTOR ที่ได้ประดิษฐ์ขึ้น ในขั้นแรกได้ทำการเปรียบเทียบความถูกต้องของค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้จากโปรแกรม JFACTOR นี้ระหว่างการคำนวณปกติซึ่งไม่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติกับการคำนวณที่มีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติโดยใช้ค่าเปอร์เซ็นการขึ้นอยู่กับโดเมนของค่าพารามิเตอร์เจอินทิกรัลในการเปรียบเทียบความผิดพลาดระหว่างการคำนวณทั้งสองแบบ โดยผลการคำนวณที่ได้แสดงให้เห็นว่าเทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติสามารถช่วยปรับปรุงผลการคำนวณให้มีความถูกต้องเพิ่มสูงขึ้นได้ ในขั้นตอนต่อไปได้ทำการเปรียบเทียบค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้จากโปรแกรม JFACTOR โดยประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติกับผลการคำนวณที่มีในงานวิจัยอื่น ๆ โดยปัญหาที่ใช้ในการตรวจสอบความถูกต้องนั้นครอบคลุมถึงปัญหาความเค้นในระนาบ ปัญหาความเครียดในระนาบและปัญหาสมมาตรรอบแกนซึ่งภาระที่มากระทำนั้นมีความหลากหลายทั้ง ภาระความเครียดเริ่มต้นเนื่องจากการกระจายตัวของสนามอุณหภูมิ ภาระความเค้นดึงที่ผิวรอยร้าวและแรงวัตถุ โดยผลการเปรียบเทียบนั้นแสดงให้เห็นถึงความสอดคล้องกันเป็นอย่างดี

7.2 ปัญหาที่พบในขณะทำวิทยานิพนธ์

(1) ปัญหาการเรียงตัวของเอลิเมนต์ในโดเมนที่ใช้อินทิเกรตค่าพารามิเตอร์เจอินทิกรัล

เนื่องจากในการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลนั้นเราจำเป็นต้องสร้างโดเมนขนาดต่าง ๆ ล้อมรอบปลายรอยร้าวไว้ ซึ่งในการสร้างโครงตาข่ายที่ประกอบด้วยโดเมนต่าง ๆ เหล่านี้หลาย ๆ โดเมนรวมกันจะทำให้การเรียงตัวของเอลิเมนต์ในโดเมนเหล่านั้นมีการเรียงตัวที่ไม่เป็นระเบียบมากนัก โดยการเรียงตัวของเอลิเมนต์ในโดเมนสามารถถูกปรับปรุงให้ดีขึ้นได้หากขนาดของโดเมนที่ใช้มีความแตกต่างกันมาก ด้วยเหตุนี้ผู้วิจัยจึงได้ปรับปรุงให้โปรแกรม JFACTOR สามารถใช้งานได้กับโดเมนรูปสี่เหลี่ยมผืนผ้านอกเหนือจากโดเมนรูปสี่เหลี่ยมจัตุรัสและโดเมนรูปวงกลมซึ่งขนาดของโดเมนจะถูกจำกัดด้วยขนาดของรอยร้าวหรือขนาดความกว้างของชิ้นงานส่วนที่เหลือจากรอยร้าว (Uncracked Ligament)

(2) ปัญหาความคลาดเคลื่อนของผลลัพธ์ที่ได้จากงานวิจับอื่น

ในวิทยานิพนธ์นี้ผลลัพธ์ค่าพารามิเตอร์เจอินทิกรัลที่ใช้เปรียบเทียบความถูกต้องกับผลลัพธ์ที่คำนวณได้จากโปรแกรม JFACTOR นั้นส่วนใหญ่แล้วเป็นผลลัพธ์ที่ได้จากวิธีการคำนวณของ EPRI ซึ่งเขียนอยู่ในรูปสมการผลบวกของค่าพารามิเตอร์เจอินทิกรัลซึ่งคำนวณได้ในช่วงยืดหยุ่นเชิงเส้นและช่วง Fully Plastic โดยผู้วิจัยได้นำมาจากหนังสือของ Anderson [15]

ซึ่งเป็นฉบับปรับปรุงแก้ไขใหม่ในปี 1995 อีกทีหนึ่ง ผู้วิจัยได้ค้นพบในภายหลังว่าผลลัพธ์ที่ได้สำหรับบางปัญหานั้น Shih and Needleman [32, 33] ได้นำมาวิเคราะห์ใหม่และพบว่าผลลัพธ์ที่ได้ในช่วง Fully Plastic ของ EPRI นั้นมีความคลาดเคลื่อนมากถึง 30% และสำหรับในบางปัญหานั้นผู้วิจัยยังพบอีกด้วยว่าผลลัพธ์ที่ได้ในช่วง Fully Plastic ซึ่งพิมพ์อยู่ในหนังสือของ Anderson นั้นมีความถูกต้องดีอยู่แล้วเพียงแต่สมการที่พิมพ์ในหนังสือนั้นไม่ถูกต้องซึ่งกว่าที่ผู้วิจัยจะค้นพบความจริงต่าง ๆ เหล่านี้นั้นผู้วิจัยต้องเสียเวลาในการตรวจสอบความถูกต้องของโปรแกรม JFACTOR อยู่นานพร้อมกับความสงสัยว่าทำไมผลลัพธ์ที่ได้ถึงมีความคลาดเคลื่อนจากผลลัพธ์ของ EPRI อยู่สูงถึงเกือบ 30% ทั้ง ๆ ที่โปรแกรม JFACTOR เองนั้นให้ผลลัพธ์ที่มีความถูกต้องดีอยู่แล้ว

(3) ปัญหาการเสียรูปที่บริเวณปลายรอบร้าวสำหรับเอลิเมนต์ที่ปลายรอยร้าวซึ่งด้านหนึ่งของเอลิเมนต์ถูกยุบมารวมกันที่ตำแหน่งปลายรอยร้าว

ในช่วงแรกของการคำนวณหาค่าพารามิเตอร์เจอินทิกรัลนั้น ผู้วิจัยได้ทดลองใช้เอลิเมนต์สี่เหลี่ยมเก้าจุดต่อที่ด้านหนึ่งของเอลิเมนต์ถูกยุบมารวมกันที่ตำแหน่งปลายรอยร้าวขณะที่จุดต่อกึ่งกลางเอลิเมนต์และกึ่งกลางด้านที่อยู่ถัดจากปลายรอยร้าวถูกเลื่อนมาที่ตำแหน่งหนึ่งในสี่เป็นเอลิเมนต์ที่ปลายรอยร้าวเปรียบเทียบกับเอลิเมนต์รอบปลายรอยร้าวชนิดเดียวกันที่จุดต่อกลางเอลิเมนต์และกึ่งกลางด้านยังคงอยู่ที่ตำแหน่งกึ่งกลาง จากการทดลองผู้วิจัยพบว่าค่าพารามิเตอร์เจอินทิกรัลที่คำนวณได้โดยใช้เอลิเมนต์รอบปลายรอยร้าวที่ถูกเลื่อนจุดต่อมาที่ตำแหน่งหนึ่งในสี่นั้นจะให้ผลการคำนวณที่ขึ้นกับโดเมนน้อยกว่าเล็กน้อยเนื่องจากเอลิเมนต์ชนิดนี้มีผลเฉลยที่ปลายรอยร้าวสอดคล้องกับวัสดุแบบพลาสติกสมบูรณ์ (Perfectly Plastic Material) และวัสดุแบบยืดหยุ่นเชิงเส้นพร้อมกัน ขณะที่เอลิเมนต์ที่ไม่ถูกเลื่อนจุดต่อมาที่ตำแหน่งหนึ่งในสี่นั้นให้ผลเฉลยที่ปลายรอยร้าวสอดคล้องกับวัสดุแบบพลาสติกสมบูรณ์เพียงอย่างเดียว เมื่อผู้วิจัยตรวจสอบการเสียรูปที่บริเวณปลายรอยร้าวของเอลิเมนต์ทั้งสอง ผู้วิจัยพบว่าการเสียรูปที่บริเวณปลายรอยร้าวของเอลิเมนต์ที่ไม่ถูกเลื่อนจุดต่อมาที่ตำแหน่งหนึ่งในสี่นั้นมีการเสียรูปที่เรียกว่า Crack Tip Blunting ได้ดีกว่าเอลิเมนต์ที่ถูกเลื่อนจุดต่อมาที่ตำแหน่งหนึ่งในสี่เนื่องจากการเสียรูปที่บริเวณปลายรอยร้าวของเอลิเมนต์ชนิดนี้สอดคล้องกับวัสดุแบบพลาสติกสมบูรณ์เพียงอย่างเดียวซึ่งต่างจากเอลิเมนต์รอบปลายรอยร้าวที่ถูกเลื่อนจุดต่อมาที่ตำแหน่งหนึ่งในสี่ที่มีการเสียรูปที่ปลายรอยร้าวผสมกันระหว่างวัสดุแบบพลาสติกสมบูรณ์กับวัสดุแบบยืดหยุ่นเชิงเส้น ด้วยเหตุนี้ผู้วิจัยจึงได้เลือกใช้เอลิเมนต์ที่จุดต่อกึ่งกลางเอลิเมนต์และกึ่งกลางด้านทั้งหมดยังคงอยู่ที่ตำแหน่งกึ่งกลางเป็นเอลิเมนต์ที่ปลายรอยร้าวนอกเหนือจากเหตุผลเรื่องความสะดวกที่ไม่ต้องเลื่อนจุดต่อมาที่ตำแหน่งหนึ่งในสี่ของด้านดังที่ได้กล่าวไว้ในภาคผนวก ง

(4) ปัญหาระยะเวลาที่ใช้ในการคำนวณเมื่อแบบจำลองมีจำนวนจุดต่อเพิ่มสูงขึ้น

ในช่วงหลังของการทำวิทยานิพนธ์ เมื่อผู้วิจัยได้ทำการตรวจสอบความถูกต้องของโปรแกรม JFACTOR เข้ากับปัญหาต่าง ๆ แล้ว ผู้วิจัยพบว่าเมื่อแบบจำลองไฟไนต์เอลิเมนต์มีจำนวนจุดต่อมากกว่า 2,000 จุดขึ้นไปแล้ว โปรแกรม JFACTOR จะใช้เวลาในการแก้ปัญหาแต่ละครั้งเป็นระยะเวลาที่ค่อนข้างสูงมาก ในตอนแรกผู้วิจัยคาดว่าปัญหาอาจเกิดขึ้นจากวิธีที่ผู้วิจัยใช้ในขั้นตอนการแก้ระบบสมการ แต่เมื่อผู้วิจัยได้ตรวจสอบสาเหตุของปัญหาอย่างละเอียดแล้วกลับพบว่าสาเหตุที่แท้จริงของปัญหานั้นเกิดขึ้นจากการสร้างและการเตรียมเมตริกซ์สมการระบบรวม (System Matrix) ต่าง ๆ ที่ใช้ในโปรแกรมย่อย [SUBROUTINE LST], [SUBROUTINE VOLUMETRIC] และ [SUBROUTINE DEVIATORIC] ซึ่งถูกเขียนขึ้นอย่างไม่มีประสิทธิภาพ ดังนั้นผู้วิจัยจึงได้ปรับปรุงแก้ไขให้การสร้างและการเตรียมเมตริกซ์ระบบรวมต่าง ๆ นั้นมีประสิทธิภาพมากยิ่งขึ้น

(5) ปัญหาการหาเอลิเมนต์ในโครงตาข่ายเก่าที่อยู่ใกล้กับจุดต่อบนโครงตาข่ายใหม่

ในการหาเอลิเมนต์ในโครงตาข่ายเก่าที่อยู่ใกล้กับจุดต่อบนโครงตาข่ายใหม่มากที่สุดเพื่อใช้ในการถ่ายทอดผลเฉลยนั้นในตอนแรกผู้วิจัยได้ใช้ค่าความผิดพลาดของพื้นที่ซึ่งคำนวณจากผลต่างระหว่างผลบวกของค่าสัมบูรณ์ของพื้นที่สามเหลี่ยมย่อย A_1, A_2 และ A_3 ดังแสดงในสมการ (4.27ก-ค) กับพื้นที่สามเหลี่ยม A_e ซึ่งคำนวณจากสมการ (4.27ง) เมื่อผู้วิจัยได้นำไปประยุกต์ใช้กับปัญหาที่มีขอบเขต (Boundary) เป็นเส้นโค้งแล้ว เช่น ปัญหาท่อทรงกระบอกที่มีรอยร้าววางตัวตามแนวแกน การหาเอลิเมนต์ในโครงตาข่ายเก่าที่อยู่ใกล้กับจุดต่อบนโครงตาข่ายใหม่มากที่สุดนั้นจะให้ผลลัพธ์ที่ผิดไปมาก เมื่อผู้วิจัยได้ตรวจสอบดูแล้วพบว่าค่าความผิดพลาดของพื้นที่ที่ผู้วิจัยใช้ในตอนแรกนั้นจะต้องทำการเปรียบเทียบกับพื้นที่สามเหลี่ยม A_e ของเอลิเมนต์ที่กำลังทำการตรวจสอบอยู่นั้น ๆ ด้วยดังสมการ (4.28) เนื่องจากเมื่อมีการประยุกต์ใช้เทคนิคการปรับขนาดเอลิเมนต์โดยอัตโนมัติแล้วเอลิเมนต์ในบางพื้นที่ของปัญหาจะมีขนาดแตกต่างจากเอลิเมนต์ที่บริเวณปลายรอยร้าวมากทำให้เอลิเมนต์ที่หาได้อาจไม่ใช่เอลิเมนต์ที่เราต้องการนำไปใช้จริง ๆ

7.3 ข้อเสนอแนะสำหรับงานวิจัยในอนาคต

งานในวิทยานิพนธ์ฉบับนี้ถือเป็นงานวิจัยขั้นพื้นฐานในด้านการวิเคราะห์ปัญหารอยร้าวด้วยระเบียบวิธีไฟไนต์เอลิเมนต์ที่สำคัญต่อจากงานของ ธนวัช ศรีเจริญชัย [16] จากงานวิจัยที่ผ่านมาจนกระทั่งถึงงานวิจัยนี้นั้นการนำไปประยุกต์ใช้กับวัสดุและรอยร้าวที่มีพฤติกรรมต่าง ๆ ยังทำได้ค่อนข้างจำกัดอยู่มาก ดังนั้นนักวิจัยรุ่นใหม่ที่สนใจสามารถศึกษาและพัฒนาไปสู่งานวิจัยใหม่ที่ต่อเนื่องได้ในหลายแนวทางด้วยกันดังนี้

1. พัฒนาโปรแกรมคอมพิวเตอร์ให้สามารถประยุกต์ใช้ได้กับวัสดุที่มีความสัมพันธ์ระหว่างค่าความเค้นและค่าความเครียดอื่นนอกจากแบบ Ramberg-Osgood เพื่อให้สามารถจำลองพฤติกรรมของวัสดุต่าง ๆ ได้ถูกต้องมากขึ้น เช่น แบบ Bilinear หรือแบบ Piecewise Power Hardening

2. พัฒนาโปรแกรมคอมพิวเตอร์ที่สามารถหาค่าพารามิเตอร์ที่สำคัญอื่น ๆ ที่กำลังได้รับการยอมรับและมีความสำคัญในปัจจุบัน เช่น พารามิเตอร์ CTOA และพารามิเตอร์ T_{ε}^{*}-integral ซึ่งสามารถใช้ได้กับการวิเคราะห์แบบอิลาสติก-พลาสติกที่รอยร้าวมีการเติบโตภายใต้สถานะอยู่ตัวได้

3. พัฒนาโปรแกรมคอมพิวเตอร์เพื่อให้สามารถใช้ได้กับการวิเคราะห์ปัญหารอยร้าวที่รอยร้าวมีลักษณะไม่เป็นเส้นตรงอยู่ในระนาบรอบร้าว

4. พัฒนาโปรแกรมคอมพิวเตอร์ให้สามารถประยุกต์ใช้ได้กับปัญหาสามมิติเพื่อให้สามารถนำไปประยุกต์ใช้ได้กับปัญหาในการทำงานจริง

5. ปรับปรุงโปรแกรม JFACTOR ให้สามารถเก็บและแก้เมตริกซ์สมการระบบรวมต่าง ๆ ในแบบ Element by Elem ent [34] เพื่อประหยัดเนื้อที่หน่วยความจำและลดระยะเวลาในการคำนวณของเครื่องคอมพิวเตอร์ลงให้ได้มากที่สุด

รายการอ้างอิง

1. Parks, D. M. A Stiffness Derivative Finite Element Technique for Determination of Crack Tip Stress Intensity Factors. International Journal of Fracture 10 (1974): 487-502.
2. Parks, D. M. The Virtual Crack Extension Method for Nonlinear Material Behavior. Computer Methods in Applied Mechanics and Engineering 12 (1977): 353-364.
3. Hellen, T. K. On the Method of Virtual Crack Extensions. International Journal for Numerical Methods in Engineering 9 (1975): 187-207.
4. Henshell, R. D. and Shaw, K. G. Crack Tip Finite Elements Are Unnecessary. International Journal for Numerical Methods in Engineering 9 (1975): 495-507.
5. Barsoum, R. S. On the Use of Isoparametric Finite Elements in Linear Fracture Mechanics. International Journal for Numerical Methods in Engineering 10 (1976): 25-37.
6. Barsoum, R. S. Triangular Quarter-Point Elements as Elastic and Perfectly-Plastic Crack Tip Elements. International Journal for Numerical Methods in Engineering 11 (1977): 85-98.
7. deLorenzi, H. G. On the Energy Release Rate and the *J*-integral for 3-D Crack Configurations. International Journal of Fracture 19 (1982): 183-193.
8. deLorenzi, H. G. Energy Release Rate Calculations by the Finite Element Method. Engineering Fracture Mechanics 21 (1985): 129-143.
9. Li, F. Z., Shih, C. F. and Needleman, A. A Comparison of Methods for Calculating Energy Release Rates. Engineering Fracture Mechanics 21 (1985): 405-421.
10. Shih, C. F., Moran, B. and Nakamura, T. Energy Release Rate along a Three-Dimensional Crack Front in a Thermally Stressed Body. International Journal of Fracture 30 (1986): 79-102.
11. Moran, B. and Shih, C. F. A General Treatment of Crack Tip Contour Integrals. International Journal of Fracture 35 (1987): 295-310.

12. Moran, B. and Shih, C. F. Crack Tip and Associated Domain Integrals from Momentum and Energy Balance. Engineering Fracture Mechanics 27, 6 (1987): 615-642.

13. Omori, Y., Kobayashi, A. S., Okada, H., Atluri, S. N. and Tan, P. W. T_{ε}^{*} integral as a Crack Growth Criterion. Mechanics of Materials 28 (1998): 147-154.

14. Newman, J. C., James, M. A. and Zerbst, U. A Review of the CTOA/CTOD Fracture Criterion. Engineering Fracture Mechanics 70 (2003): 371-385.

15. Anderson, T. L. Fracture Mechanics: Fundamentals and Applications. Second Edition. Boca Raton: CRC Press Inc., 1995.

16. ธนวัช ศรีเจริญชัย. การศึกษาการทำนายอายุของชิ้นงานที่มีรอยร้าวภายใต้สภาวะความเครียดระนาบโดยระเบียบวิธีไฟไนต์เอลิเมนต์. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2544.

17. Mendelson, A. Plasticity: Theory and Application. Malabar: Robert Kreiger Publishing Co., 1986.

18. ปราโมทย์ เดชะอำไพ. ระเบียบวิธีไฟไนต์เอลิเมนต์เพื่อการคำนวณพลศาสตร์ของไหล. พิมพ์ครั้งที่ 2. กรุงเทพฯ: สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2545.

19. Shih, C. F. and German, M. D. Requirements for a One Parameter Characterization of Crack Tip Fields by the HRR Singularity. International Journal of Fracture 17 (1981): 27-43.

20. Roy, Y. A. and Narasimhan, R. *J*-Dominance in Mixed Mode Ductile Fracture Specimens. International Journal of Fracture 88 (1997): 259-279.

21. Chao, Y. J. and Zhu, X. K. J-A_2 Characterization of Crack-Tip Fields: Extent of J-A_2 Dominance and Size Requirements. International Journal of Fracture 89 (1998): 285-307.

22. Trädegård, A., Nilsson, F. and Östlund, S. *J-Q* Characterization of Propagating Cracks. International Journal of Fracture 94 (1998): 357-369.

23. Zhu, X. K., Jang, S. K. and Chen, Y. F. A Modification of *J-Q* Theory and Its Applications. International Journal of Fracture 111 (2001): L47-L52.

24. deLorenzi, H. G. and Shih, C. F. 3-D Elastic-Plastic Investigation of Fracture Parameters in Side-Grooved Compact Specimen. International Journal of Fracture 21 (1983): 195-220.

25. Zienkiewicz, O. C. and Taylor, R. L. The Finite Element Method. Fifth Edition. Oxford: McGraw-Hill, 2000.

26. Burnett, D. S. Finite Element Analysis from Concepts to Applications. New Jersey: Addison-Wesley, 1987.

27. Crisfield, M. A. Non-linear finite element analysis of solids and structures. volume 1: essentials. Chichester: John Wiley, 1991.

28. ปราโมทย์ เดชะอำไพ. ไฟไนต์เอลิเมนต์ในงานวิศวกรรม. พิมพ์ครั้งที่ 3. กรุงเทพฯ: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2544.

29. Nishioka, T., Tokudome, H. and Kinoshita, M. Dynamic Fracture-Path Prediction in Impact Fracture Phenomena Using Moving Finite Element Method Based on Delaunay Automatic Mesh Generation. International Journal of Solids and Structures 38 (2001): 5273-5301.

30. Leitch, B. W. Plane-Strain, Work-Hardening Response of an Internally Pressurized Cylinder Containing Surface Flaws. Engineering Fracture Mechanics 23, 5 (1986): 833-841.

31. Kumar, V., Schumacher, B. I. and German, M. D. Effect of Thermal and Residual Stresses on the *J*-integral Elastic-Plastic Fracture Analysis. Computers and Structures 40, 2 (1991): 487-501.

32. Shih, C. F. and Needleman, A. Fully Plastic Crack Problems Part 1: Solutions by a Penalty Method. ASME Journal of Applied Mechanics 51 (1984): 48-56.

33. Shih, C. F. and Needleman, A. Fully Plastic Crack Problems Part 2: Application of Consistency Checks. ASME Journal of Applied Mechanics 51 (1984): 57-64.

34. อธิพงษ์ มาลาทิพย์. ระเบียบวิธีไฟไนต์เอลิเมนต์เพื่อการวิเคราะห์การถ่ายเทความร้อนแบบคอนจูเกต. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2547.

35. Banthia, V. Singularity of Collapse Q-8 Finite Element. International Journal for Numerical Methods in Engineering 21 (1985): 959-965.

ภาคผนวก

ภาคผนวก ก

คุณสมบัติของวัสดุยืดหยุ่นและพลังงานความเครียดหนาแน่น

สำหรับวัสดุยืดหยุ่น (Elastic Solid) ค่าพลังงานความเครียดหนาแน่น (Strain Energy Density) จะมีค่าเท่ากับงานจากความเค้นซึ่งเขียนแทนได้ดังสมการ

$$W = \int_0^W dW = \int_0^{\varepsilon_{ij}^m} \sigma_{ij} d\varepsilon_{ij}^m \qquad (ก.1)$$

เนื่องจากปรากฏการณ์ทางกลทั้งหมดที่เกิดขึ้นในวัสดุยืดหยุ่นสามารถย้อนกลับได้ (Reversible Process) และความสัมพันธ์ระหว่างค่าความเค้นและค่าความเครียดถูกเขียนอยู่ในรูปสุทธิ ดังนั้น ในทำนองเดียวกันค่าพลังงานความเครียดหนาแน่นสามารถเขียนอยู่ในรูปฟังก์ชั่นของค่าความเค้น ค่าความเครียดทางกลและคุณสมบัติของวัสดุได้ หากเรากำหนดให้ค่าพลังงานความเครียดหนาแน่น ถูกเขียนอยู่ในรูปฟังก์ชั่นของค่าความเครียดทางกลและคุณสมบัติของวัสดุแล้ว ค่าพลังงานความเครียดหนาแน่นที่เพิ่มขึ้นเนื่องจากค่าความเครียดทางกลสามารถเขียนแทนได้ด้วยสมการ

$$dW = \frac{\partial W}{\partial \varepsilon_{ij}^m} d\varepsilon_{ij}^m \qquad (ก.2)$$

แทนสมการ (ก.2) ลงในสมการ (ก.1) จะได้สมการค่าพลังงานความเครียดหนาแน่นเป็น

$$W = \int_0^{\varepsilon_{ij}^m} \frac{\partial W}{\partial \varepsilon_{ij}^m} d\varepsilon_{ij}^m = \int_0^{\varepsilon_{ij}^m} \sigma_{ij} d\varepsilon_{ij}^m \qquad (ก.3)$$

จากสมการ (ก.3) เราสามารถสรุปได้ว่า

$$\frac{\partial W}{\partial \varepsilon_{ij}^m} = \sigma_{ij} \qquad (ก.4)$$

ซึ่งเป็นคุณสมบัติของวัสดุแบบยืดหยุ่น (Property of Elastic Potential)

พิจารณาวัสดุที่มีพฤติกรรมแบบ Ramberg-Osgood เราสามารถหาค่าพลังงานความเครียดหนาแน่นได้จากสมการ (ก.1) โดยแบ่งค่าความเครียดทางกลออกเป็นพจน์ยืดหยุ่นเชิงเส้นและพลาสติกดังสมการ

$$\varepsilon_{ij}^{m} = \varepsilon_{ij}^{e} + e_{ij}^{p} \tag{ก.5}$$

โดยที่ ε_{ij}^{e} แทนค่าความเครียดยืดหยุ่นเชิงเส้น (Linear Elastic Strain)

e_{ij}^{p} แทนค่าความเครียดพลาสติก (Plastic Strain)

ในทำนองเดียวกันจากสมการด้านบนเราสามารถแบ่งค่าพลังงานความเครียดหนาแน่นออกเป็นพจน์ที่เกิดจากความเครียดยืดหยุ่นเชิงเส้นและพจน์ที่เกิดจากความเครียดพลาสติกเป็น

$$W = W^{e} + W^{p} = \int_{0}^{\varepsilon_{ij}^{e}} \sigma_{ij} d\varepsilon_{ij}^{e} + \int_{0}^{e_{ij}^{p}} \sigma_{ij} de_{ij}^{p} \tag{ก.6}$$

โดยที่ W^{e} แทนค่าพลังงานความเครียดหนาแน่นเนื่องจากค่าความเครียดยืดหยุ่นเชิงเส้น

W^{p} แทนค่าพลังงานความเครียดหนาแน่นเนื่องจากค่าความเครียดพลาสติก

ค่าพลังงานความเครียดหนาแน่นเนื่องจากค่าความเครียดพลาสติกสามารถคำนวณหาได้โดยง่ายเมื่อเขียนสมการอยู่ในรูปประสิทธิผล [17] เป็น

$$W^{p} = \int_{0}^{\varepsilon_{p}} \sigma_{e} d\varepsilon_{p} \tag{ก.7}$$

โดยที่ σ_{e} แทนค่าความเค้นประสิทธิผล (Effective Stress)

ε_{p} แทนค่าความเครียดพลาสติกประสิทธิผล (Effective Plastic Strain)

ความสัมพันธ์ระหว่างค่าความเค้นประสิทธิผลและค่าความเครียดพลาสติกประสิทธิผลสามารถหาได้จากการทดสอบชิ้นงานภายใต้การทดลองในทิศทางเดียว (Uniaxial Test) โดยสำหรับวัสดุที่มีพฤติกรรมแบบ Ramberg-Osgood แล้วจะได้ความสัมพันธ์เป็น

$$\sigma_{e} = \sigma_{o} \left(\frac{\varepsilon_{p}}{\alpha \varepsilon_{o}} \right)^{\frac{1}{n}} \tag{ก.8}$$

โดยที่ σ_{o} แทนค่าความเค้นที่จุดคราก (Yield Stress)

ε_{o} แทนค่าความเครียดที่จุดคราก (Yield Strain)

α แทนค่าคงที่ของวัสดุ

n แทนค่ายกกำลังของความเครียด (Strain Hardening Exponent)

แทนสมการ (ก.8) ลงในสมการ (ก.7) จากนั้นทำการอินทิเกรตแล้วเปลี่ยนให้อยู่ในรูปของค่าความเค้นประสิทธิผลโดยใช้สมการ (ก.8) จะได้

$$W^p = \alpha\varepsilon_o\sigma_o\left(\frac{n}{n+1}\right)\left(\frac{\sigma_e}{\sigma_o}\right)^{n+1} \qquad (\text{ก}.9)$$

ค่าพลังงานความเครียดหนาแน่นเนื่องจากค่าความเครียดยืดหยุ่นเชิงเส้นจะมีค่าเท่ากับค่าพลังงานความเครียดหนาแน่นของวัตถุแบบยืดหยุ่นเชิงเส้นดังสมการ

$$W^e = \frac{1}{2}\sigma_{ij}\varepsilon_{ij}^e = \frac{3}{2}\left(\frac{1-2\nu}{E}\right)\bar{\sigma}^2 + \frac{1}{3}\left(\frac{1+\nu}{E}\right)\sigma_e^2 \qquad (\text{ก}.10)$$

โดยที่ $\bar{\sigma}$ แทนค่าความเค้นตั้งฉากเฉลี่ย (Mean Stress)

แทนสมการ (ก.9-10) ลงในสมการ (ก.6) จะได้ค่าพลังงานความเครียดหนาแน่นมีค่าเท่ากับ

$$W = \frac{3}{2}\left(\frac{1-2\nu}{E}\right)\bar{\sigma}^2 + \frac{1}{3}\left(\frac{1+\nu}{E}\right)\sigma_e^2 + \alpha\varepsilon_o\sigma_o\left(\frac{n}{n+1}\right)\left(\frac{\sigma_e}{\sigma_o}\right)^{n+1} \qquad (\text{ก}.11)$$

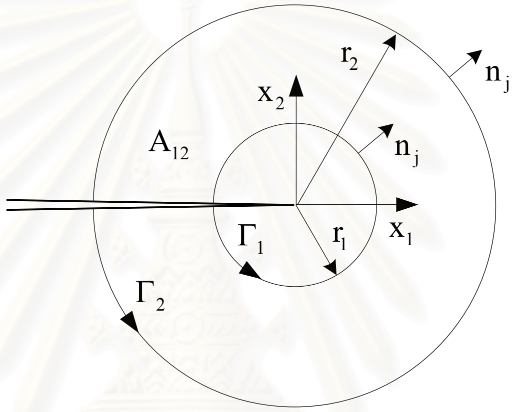
โดยสมการ (ก.11) ถูกเขียนอยู่ในรูปฟังก์ชั่นของค่าความเค้นต่าง ๆ และค่าคุณสมบัติของวัสดุเท่านั้นเพื่อให้สามารถประยุกต์ใช้ได้แม้ในกรณีที่วัตถุมีค่าความเครียดเริ่มต้นเนื่องจากอุณหภูมิ

ภาคผนวก ข

คุณสมบัติที่พารามิเตอร์เจอินทิกรัลไม่ขึ้นกับเส้นทางเดิน

พิจารณาสมการ (2.44) ที่เขียนขึ้นใหม่จากการแทนค่าสมการ (2.26) สำหรับปัญหาสองมิติที่มีรอยร้าววางตัวตามแนวแกน x_1 ดังแสดงในรูปที่ ข.1 เป็น

$$J = \lim_{\Gamma \to 0} \oint_{\Gamma} n_j \left(W\delta_{j1} - \sigma_{ji} u_{i,1} \right) d\Gamma \qquad (ข.1)$$



รูปที่ ข.1 รอยร้าวที่วางตัวตามแนวแกน x_1 บนชิ้นงานสองมิติ

พิจารณาเส้นทางเดินปิดซึ่งเกิดจากเส้นทางเดิน Γ_1 และ Γ_2 กับส่วนของผิวรอยร้าวที่เชื่อมเส้นทางเดินทั้งสองดังแสดงในรูปที่ ข.1 เนื่องจากไม่มีค่าความเค้นดึงมากระทำที่ผิวรอยร้าว ดังนั้นประยุกต์ทฤษฎีบทของเกาส์เข้ากับเส้นทางเดินปิดนี้จะได้

$$J(\Gamma_2) - J(\Gamma_1) = \int_{A_{12}} \left(W\delta_{j1} - \sigma_{ji} u_{i,1} \right)_{,j} dA \qquad (ข.2)$$

โดยที่ A_{12} แทนพื้นที่ที่ถูกปิดล้อมด้วยเส้นทางเดินปิดดังแสดงในรูปที่ ข.1

กระจายสมการ (ข.2) แล้วใช้สมการ (2.38) จะได้

$$J(\Gamma_2) - J(\Gamma_1) = \int_{A_{12}} \left(W_{,1} - \sigma_{ji,j} u_{i,1} - \sigma_{ij} \varepsilon_{ij,1} \right) dA \qquad (ข.3)$$

พจน์แรกในวงเล็บทางด้านขวาของสมการ (ข.3) สามารถกระจายออกได้ในทำนองเดียวกันกับสมการ (2.39) สำหรับวัตถุที่ไม่มีค่าความเครียดเริ่มต้นเนื่องจากอุณหภูมิและคุณสมบัติของวัตถุไม่ขึ้นอยู่กับตำแหน่งบนพิกัด x_1 เป็น

$$W_{,1} = \sigma_{ij}\varepsilon_{ij,1} \tag{ข.4}$$

จากสมการ (2.1) เมื่อไม่พิจารณาแรงวัตถุและพจน์ความเร่งจะได้

$$\sigma_{ji,j} = 0 \tag{ข.5}$$

แทนสมการ (ข.4-5) ลงในสมการ (ข.3) จะได้

$$J\left(\Gamma_2\right) = J\left(\Gamma_1\right) \tag{ข.6}$$

สมการ (ข.6) บอกเราว่าในกรณีนี้ค่าพารามิเตอร์เจอินทิกรัลจะไม่ขึ้นกับเส้นทางเดิน (Path-Independence) ดังนั้นสมการ (ข.1) สามารถเขียนใหม่ในกรณีนี้ได้เป็น

$$J = \int_{\Gamma}\left(W\delta_{j1} - \sigma_{ji}u_{i,1}\right)n_j d\Gamma \tag{ข.7}$$

สมการ (ข.7) และคุณสมบัติที่ค่าพารามิเตอร์เจอินทิกรัลจะไม่ขึ้นกับเส้นทางเดินภายใต้สมมุติฐานที่กล่าวถึงทั้งหมดข้างต้นนี้ได้ถูกนำเสนอเป็นครั้งโดย Rice [15] ซึ่งต่อมาได้ถูกนำไปประยุกต์ใช้เพื่อให้การคำนวณหาค่าพารามิเตอร์เจอินทิกรัลด้วยระเบียบวิธีไฟไนต์เอลิเมนต์ในระยะเริ่มแรกมีความแม่นยำมากขึ้นโดยใช้เส้นทางเดินที่อยู่ห่างจากปลายรอยร้าวแทนเส้นทางเดินที่อยู่ใกล้กับตำแหน่งปลายรอยร้าว

ภาคผนวก ค

พารามิเตอร์เจอินทิกรัลในฐานะพารามิเตอร์ตัวประกอบความเข้มของความเค้น

Hutchinson [15] ได้สมมุติให้ผลเฉลยของฟังก์ชั่นความเค้น Airy's Stress Function สำหรับปัญหารอยร้าวสองมิติสามารถเขียนอยู่ในรูปอนุกรมของฟังก์ชั่นที่ขึ้นอยู่กับมุม θ และระยะตามแนวรัศมีจากปลายรอยร้าว r ดังสมการ

$$\Phi(r,\theta) \;=\; C_1(\theta)r^s \;+\; C_2(\theta)r^t \;+\; \ldots \qquad (ค.1)$$

โดยกำหนดให้ $s < t$ และ t น้อยกว่าเลขยกกำลังที่เหลือทั้งหมดของ r ในอนุกรม ดังนั้นเมื่อ $r \to 0$ พจน์แรกทางด้านขวาของฟังก์ชั่นความเค้นจะมีค่ามากเมื่อเทียบกับพจน์ที่เหลือทั้งหมดเสมอ ดังนั้นเมื่อพิจารณาเฉพาะที่บริเวณปลายรอยร้าวฟังก์ชั่นความเค้นสามารถเขียนใหม่โดยพิจารณาเพียงพจน์แรกของสมการ (ค.1) ได้เป็น

$$\Phi(r,\theta) \;=\; k\sigma_o r^s \tilde{\Phi}(\theta) \qquad (ค.2)$$

โดยที่ k แทนขนาดของฟังก์ชั่นความเค้น

$\tilde{\Phi}$ แทนฟังก์ชันไร้หน่วยที่ขึ้นกับค่ามุม θ

σ_o แทนค่าความเค้นที่จุดคราก

โดยค่าความเค้นที่จุดคราก σ_o ถูกกำหนดลงในสมการ (ค.2) เพื่อให้สอดคล้องกับผลลัพธ์ที่จะได้ในภายหลัง ค่าความเค้นในระบบพิกัดทรงกระบอกสามารถหาได้จากฟังก์ชั่นความเค้นดังสมการ

$$\sigma_{rr} \;=\; \frac{1}{r^2}\frac{\partial^2\Phi}{\partial\theta^2} \;+\; \frac{1}{r}\frac{\partial\Phi}{\partial r} \;=\; k\sigma_o r^{s-2}\left(s\tilde{\Phi} \;+\; \tilde{\Phi}''\right) \;=\; k\sigma_o r^{s-2}\tilde{\sigma}_{rr}(s,\theta) \qquad (ค.3ก)$$

$$\sigma_{\theta\theta} \;=\; \frac{\partial^2\Phi}{\partial r^2} \;=\; k\sigma_o r^{s-2}s\left(s-1\right)\tilde{\Phi} \;=\; k\sigma_o r^{s-2}\tilde{\sigma}_{\theta\theta}(s,\theta) \qquad (ค.3ข)$$

$$\tau_{r\theta} \;=\; -\frac{1}{r}\frac{\partial^2\Phi}{\partial r\partial\theta} \;+\; \frac{1}{r^2}\frac{\partial\Phi}{\partial\theta} \;=\; k\sigma_o r^{s-2}\left(1-s\right)\tilde{\Phi}' \;=\; k\sigma_o r^{s-2}\tilde{\sigma}_{r\theta}(s,\theta) \qquad (ค.3ค)$$

$$\sigma_e \;=\; k\sigma_o r^{s-2}\left(\tilde{\sigma}_{rr}^2 + \tilde{\sigma}_{\theta\theta}^2 - \tilde{\sigma}_{rr}\tilde{\sigma}_{\theta\theta} + 3\tilde{\sigma}_{r\theta}^2\right)^{1/2} \;=\; k\sigma_o r^{s-2}\tilde{\sigma}_e(s,\theta) \qquad (ค.3ง)$$

จากภาคผนวก ข สมการ (ข.7) ค่าพารามิเตอร์เจอินทิกรัลเมื่อเขียนอยู่ในรูปการอินทิเกรตบนเส้นทางเดินวงกลมรอบปลายรอยร้าวและค่าความเค้นที่ผิวจากสมการ $t_i = n_j \sigma_{ji}$ จะเท่ากับ

$$J = \int_{-\pi}^{+\pi} \left(W\cos\theta - t_i u_{i,1} \right) r d\theta \qquad (ค.4)$$

จากภาคผนวก ก เนื่องจากค่าพลังงานความเครียดหนาแน่นที่เกิดจากค่าความเครียดยืดหยุ่นเชิงเส้นจะมีค่าน้อยกว่าค่าพลังงานความเครียดหนาแน่นที่เกิดจากค่าความเครียดพลาสติกที่บริเวณปลายรอยร้าว ดังนั้นเราจึงพิจารณาเฉพาะค่าพลังงานความเครียดหนาแน่นที่เกิดจากค่าความเครียดพลาสติกเท่านั้นโดยแทนค่าสมการ (ค.3ง) ลงในสมการ (ก.9) จะได้

$$W = \alpha \sigma_o \varepsilon_o k^{n+1} \frac{n}{n+1} r^{(n+1)(s-2)} \tilde{\sigma}_e^{n+1} \qquad (ค.5)$$

แทนค่าความเค้นที่ได้จากสมการ (ค.3) ทั้งหมดลงในสมการความสัมพันธ์ระหว่างค่าความเครียดและค่าความเค้น (3.13) โดยในกรณีปัญหาความเค้นในระนาบที่ไม่พิจารณาค่าความเครียดเริ่มต้นเนื่องจากอุณหภูมิจะได้ค่าความเครียดในระบบพิกัดทรงกระบอกที่บริเวณปลายรอยร้าวมีค่าเป็น

$$\varepsilon_{rr} = \frac{1}{2}\frac{\alpha}{E}\left(\frac{\sigma_e}{\sigma_o}\right)^{n-1} 2 \left(\sigma_{rr} - \sigma_{\theta\theta} \right) = \alpha \varepsilon_o k^n r^{n(s-2)} \tilde{\varepsilon}_{rr}(s,\theta) \qquad (ค.6ก)$$

$$\varepsilon_{\theta\theta} = \frac{1}{2}\frac{\alpha}{E}\left(\frac{\sigma_e}{\sigma_o}\right)^{n-1} 2 \left(\sigma_{\theta\theta} - \sigma_{rr} \right) = \alpha \varepsilon_o k^n r^{n(s-2)} \tilde{\varepsilon}_{\theta\theta}(s,\theta) \qquad (ค.6ข)$$

$$\gamma_{r\theta} = \frac{3\alpha}{E}\left(\frac{\sigma_e}{\sigma_o}\right)^{n-1} \left(\tau_{r\theta} \right) = \alpha \varepsilon_o k^n r^{n(s-2)} \tilde{\gamma}_{r\theta}(s,\theta) \qquad (ค.6ค)$$

โดยสมการความสัมพันธ์ระหว่างค่าความเครียดและการเคลื่อนตัวในพิกัดทรงกระบอกจะเท่ากับ

$$\varepsilon_{rr} = \frac{\partial u_r}{\partial r} \qquad (ค.7ก)$$

$$\varepsilon_{\theta\theta} = \frac{u_r}{r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} \qquad (ค.7ข)$$

$$\gamma_{r\theta} = \frac{u_r}{r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} \qquad (ค.7ค)$$

อินทิเกรตสมการ (ค.7) จะได้ค่าการเคลื่อนตัวที่บริเวณปลายรอยร้าวเป็น

$$u_r = \alpha\varepsilon_o k^n r^{n(s-2)+1}\tilde{u}_r(n,\theta) \qquad (ค.8ก)$$

$$u_\theta = \alpha\varepsilon_o k^n r^{n(s-2)+1}\tilde{u}_\theta(n,\theta) \qquad (ค.8ข)$$

จากนั้นแปลงค่าการเคลื่อนตัวในพิกัดทรงกระบอกมาอยู่ในพิกัด x_1 และ x_2 โดยใช้ความสัมพันธ์

$$u_1 = u_r\cos\theta - u_\theta\sin\theta \qquad (ค.9ก)$$

$$u_2 = u_r\sin\theta + u_\theta\cos\theta \qquad (ค.9ข)$$

ในทำนองเดียวกันค่าความเค้นดึงสามารถเปลี่ยนให้อยู่ในพิกัด $x_1 - x_2$ จะได้จากความสัมพันธ์

$$t_1 = t_r\cos\theta - t_{r\theta}\sin\theta = \sigma_{rr}\cos\theta - \sigma_{r\theta}\sin\theta \qquad (ค.10ก)$$

$$t_2 = t_r\sin\theta + t_{r\theta}\cos\theta = \sigma_{rr}\sin\theta + \sigma_{r\theta}\cos\theta \qquad (ค.10ข)$$

จากนั้นหาค่าอนุพันธ์ของการเคลื่อนตัวในระบบพิกัด $x_1 - x_2$ โดยใช้ความสัมพันธ์

$$\frac{\partial u_1}{\partial x_1} = \frac{\partial u_1}{\partial r}\frac{\partial r}{\partial x_1} + \frac{\partial u_1}{\partial \theta}\frac{\partial \theta}{\partial x_1} = \frac{\partial u_1}{\partial r}\cos\theta - \frac{\partial u_1}{\partial r}\frac{\sin\theta}{r} \qquad (ค.11ก)$$

$$\frac{\partial u_2}{\partial x_1} = \frac{\partial u_2}{\partial r}\frac{\partial r}{\partial x_1} + \frac{\partial u_2}{\partial \theta}\frac{\partial \theta}{\partial x_1} = \frac{\partial u_2}{\partial r}\cos\theta - \frac{\partial u_2}{\partial r}\frac{\sin\theta}{r} \qquad (ค.11ข)$$

แทนสมการ (ค.8) ลงในสมการ (ค.9) จากนั้นหาค่าอนุพันธ์ของการเคลื่อนตัวจากสมการ (ค.11) และใช้สมการ (ค.10) จะได้

$$\begin{aligned} t_i u_{i,1} &= t_1\frac{\partial u_1}{\partial x_1} + t_2\frac{\partial u_2}{\partial x_1} \\ &= \alpha\sigma_o k\varepsilon_o r^{n+1\ (n+1)(s-2)}\left\{\sin\theta\left[\tilde{\sigma}_{rr}\left(\tilde{u}_\theta - \tilde{u}'_r\right) - \tilde{\sigma}_{r\theta}\left(\tilde{u}_r - \tilde{u}'_\theta\right)\right]\right. \\ &\qquad \left. + \cos\theta\left[n\left(s-2\right)+1\right]\left(\tilde{\sigma}_{rr}\tilde{u}_r + \tilde{\sigma}_{r\theta}\tilde{u}_\theta\right)\right\} \qquad (ค.12) \end{aligned}$$

แทนสมการ (ค.5) และ (ค.12) ลงในสมการ (ค.4) เพื่อหาค่าพารามิเตอร์เจอินทิกรัลบนเส้นทางเดินวงกลมนี้โดยกำหนดให้รัศมีของเส้นทางเดินมีค่าเท่ากับ r_2 เป็น

$$J = \alpha\varepsilon_o\sigma_o k^{n+1} r_2^{(n+1)(s-2)+1} I_n \qquad (ค.13)$$

โดย I_n มีค่าดังนี้

$$I_n = \int_{-\pi}^{+\pi}\left\{\frac{n}{n+1}\tilde{\sigma}_e^{n+1}\cos\theta - \left[\sin\theta\left\{\tilde{\sigma}_{rr}\left(\tilde{u}_\theta - \tilde{u}'_r\right) - \tilde{\sigma}_{r\theta}\left(\tilde{u}_r - \tilde{u}'_\theta\right)\right\}\right.\right.$$
$$\left.\left. + \cos\theta\left\{n\left(s-2\right)+1\right\}\left(\tilde{\sigma}_{rr}\tilde{u}_r + \tilde{\sigma}_{r\theta}\tilde{u}_\theta\right)\right]\right\}d\theta \qquad \text{(ค.14)}$$

เนื่องจากค่าพารามิเตอร์เจอินทิกรัลไม่ขึ้นกับเส้นทางเดิน ดังนั้นจากสมการ (ค.13) ค่าพารามิเตอร์เจอินทิกรัลที่ได้จึงไม่ขึ้นกับค่าระยะรัศมีของเส้นทางเดิน r_2 ด้วย จากความเข้าใจนี้จะได้สมการความสัมพันธ์ระหว่างค่า n และ s เป็น

$$s = \frac{2n+1}{n+1} \qquad \text{(ค.15)}$$

แทนสมการ (ค.15) กลับลงไปในสมการ (ค.13) แล้วจัดรูปใหม่จะได้

$$k = \left(\frac{J}{\alpha\sigma_o\varepsilon_o I_n}\right)^{\frac{1}{n+1}} \qquad \text{(ค.16)}$$

แทนสมการ (ค.15-16) ลงในสมการ (ค.3) จะได้สมการความสัมพันธ์ของค่าความเค้นที่บริเวณปลายรอยร้าวเป็น

$$\sigma_{ij} = \sigma_o\left(\frac{EJ}{\alpha\sigma_o^2 I_n r}\right)^{\frac{1}{n+1}}\tilde{\sigma}_{ij}(n,\theta) \qquad \text{(ค.17)}$$

ในทำนองเดียวกันจะได้สมการความสัมพันธ์ของค่าความเครียดที่บริเวณปลายรอยร้าวเท่ากับ

$$\varepsilon_{ij} = \frac{\alpha\sigma_o}{E}\left(\frac{EJ}{\alpha\sigma_o^2 I_n r}\right)^{\frac{n}{n+1}}\tilde{\varepsilon}_{ij}(n,\theta) \qquad \text{(ค.18)}$$

สมการความสัมพันธ์ของค่าระยะเคลื่อนตัวที่บริเวณปลายรอยร้าวเป็น

$$u_i = \frac{\alpha\sigma_o}{E}\left(\frac{EJ}{\alpha\sigma_o^2 I_n r}\right)^{\frac{n}{n+1}} r\tilde{u}_i(n,\theta) \qquad \text{(ค.19)}$$

และค่าพลังงานความเครียดหนาแน่นเป็น

$$W = \frac{n}{n+1}\left(\frac{J}{I_n r}\right)\tilde{\sigma}_e^{n+1} \qquad \text{(ค.20)}$$

สมการ (ค.17-19) ถูกเรียกว่าสนามเอกพันธ์ของ HRR (HRR Singularity) ซึ่งตั้งตามชื่อของนักวิจัยผู้ค้นพบทั้งสามคนคือ Hutchinson, Rice และ Rosengren โดยค่าพารามิเตอร์เจอินทิกรัลจะทำหน้าที่เป็นตัวกำหนดขนาดของสนามเอกพันธ์นี้ ด้วยเหตุนี้ค่าพารามิเตอร์เจอินทิกรัลจึงสามารถใช้บอกความรุนแรงของสนามความเค้น ความเครียดและระยะการเคลื่อนตัวที่บริเวณปลายรอยร้าวได้ซึ่งเป็นหลักการเดียวกันกับค่าพารามิเตอร์ตัวประกอบความเข้มของความเค้น (Stress Intensity Factor) ที่ใช้ในการวิเคราะห์รอยร้าวสำหรับวัสดุยืดหยุ่นเชิงเส้นนั่นเอง

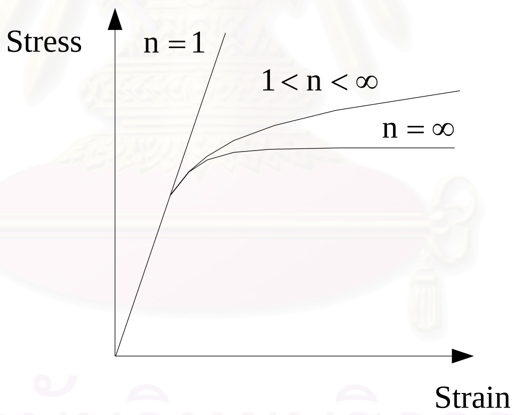
ภาคผนวก ง

เอลิเมนต์ที่ปลายรอยร้าว

จากภาคผนวก ค สมการ (ค.18) เราทราบว่าการกระจายของผลเฉลยค่าความเครียดที่บริเวณปลายรอยร้าวแปรผันกับระยะจากปลายรอยร้าวด้วยความสัมพันธ์

$$\varepsilon_{ij} \to \left(\frac{1}{r}\right)^{\frac{n}{n+1}} \qquad (ง.1)$$

ในกรณีวัสดุยืดหยุ่นเชิงเส้น $n = 1$ จะได้ค่าความเครียดที่บริเวณปลายรอยร้าวแปรผันกับระยะจากปลายรอยร้าวเป็น $1/\sqrt{r}$ ส่วนกรณีวัสดุแบบพลาสติกสมบูรณ์ (Perfectly Plastic Material) $n = \infty$ ค่าความเครียดจะแปรผันกับระยะจากปลายรอยร้าวเป็น $1/r$ ซึ่งพฤติกรรมของวัสดุทั้งสองชนิดนี้จะทำหน้าที่เป็นเหมือนขีดจำกัดบนและล่างของวัสดุที่มีค่า $1 < n < \infty$ ดังแสดงในกราฟความสัมพันธ์ระหว่างค่าความเค้นและค่าความเครียดในรูปที่ ง.1



รูปที่ ง.1 กราฟค่าความเค้นและค่าความเครียดของวัสดุที่ค่า $0 \le n \le \infty$

เพื่อให้ผลเฉลยค่าความเครียดในเอลิเมนต์สอดคล้องกับผลเฉลยที่ได้จากวิธีเชิงวิเคราะห์ของ HRR ดังแสดงในสมการ (ง.1) แล้ว Barsoum [6] ได้แสดงให้เห็นว่าการแปรผันแบบ $1/r$ ของค่าความเครียดที่บริเวณปลายรอยร้าวนั้นสามารถเกิดขึ้นได้กับเอลิเมนต์สี่เหลี่ยมแปดจุดต่อแบบไอโซพาราเมตริกซ์เมื่อจุดต่อทั้งสามจุดของด้านใดด้านหนึ่งถูกยุบมารวมกันที่ตำแหน่งปลายรอยร้าวโดยที่จุดต่อทั้งสามจุดนี้ยังคงสามารถเคลื่อนตัวได้เป็นอิสระต่อกัน ขณะที่การแปรผันแบบ $1/\sqrt{r}$ นั้นจะเกิดขึ้นเมื่อจุดต่อตรงกลางด้านทั้งสองที่อยู่ติดกับปลายรอยร้าวถูกเลื่อนมาที่ตำแหน่งหนึ่งในสี่ของด้าน Shih and Needleman [32, 33] และ Li et al. [9] ได้ประยุกต์เอลิเมนต์สี่เหลี่ยมเก้าจุด

ต่อซึ่งมีคุณสมบัติเช่นเดียวกับเอลิเมนต์สี่เหลี่ยมแปดจุดต่อของ Barsoum เข้ากับปัญหารอยร้าวสำหรับวัสดุที่พิจารณาเฉพาะความสัมพันธ์แบบพลาสติกเพียงอย่างเดียว (Fully Plastic Material) เนื่องจากเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อให้ผลเฉลยที่ดีกว่าเอลิเมนต์สี่เหลี่ยมแปดจุดต่อเมื่อใช้กับปัญหาแบบอัดตัวไม่ได้ (Incompressible Solid) [25] เนื่องจากเอลิเมนต์สี่เหลี่ยมถูกใช้เฉพาะเป็นเอลิเมนต์ที่ปลายรอยร้าวเท่านั้น ดังนั้นในการใช้เอลิเมนต์สี่เหลี่ยมเก้าจุดต่อแทนเอลิเมนต์สี่เหลี่ยมแปดจุดต่อจึงไม่ทำให้จำนวนจุดต่อเพิ่มขึ้นมากนัก ด้วยเหตุผลต่าง ๆ เหล่านี้ผู้วิจัยจึงได้เลือกใช้เอลิเมนต์สี่เหลี่ยมเก้าจุดต่อเป็นเอลิเมนต์ที่ปลายรอยร้าว ในการพิสูจน์คุณสมบัติต่าง ๆ ที่ได้กล่าวถึงของต้นนั้นผู้วิจัยจะดำเนินตามงานวิจัยของ Barsoum [6] ทุกอย่างแต่จะใช้เอลิเมนต์สี่เหลี่ยมเก้าจุดต่อแทนเอลิเมนต์สี่เหลี่ยมแปดจุดต่อโดยมีรายละเอียดดังต่อไปนี้

พิจารณาเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อ (Lagrange Element) แบบไอโซพาราเมตริกซ์ซึ่งความสัมพันธ์ระหว่างพิกัด x-y กับพิกัดธรรมชาติ (Natural Coordinate) ξ-η สามารถเขียนแทนได้ด้วยสมการ

$$x = \sum_{i=1}^{9} N_i x_i \qquad (ง.2ก)$$

$$y = \sum_{i=1}^{9} N_i y_i \qquad (ง.2ข)$$

โดยที่ N_i แทนฟังก์ชันการประมาณภายในเอลิเมนต์ที่จุดต่อ i

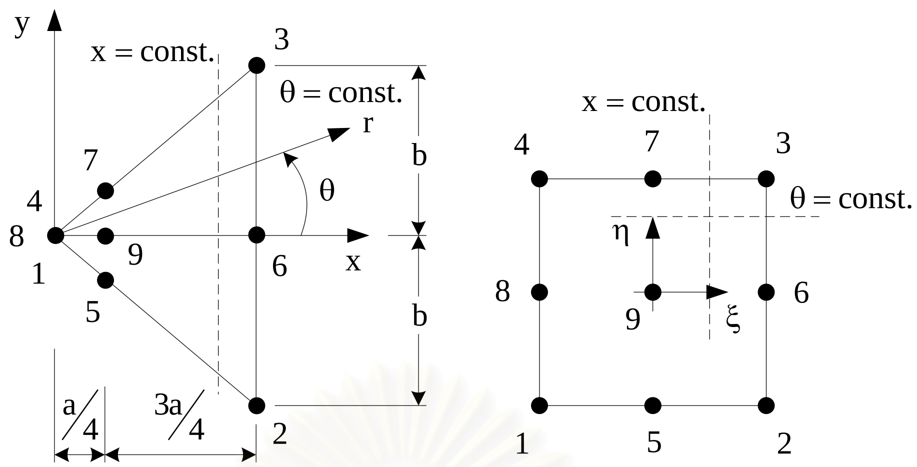
x_i แทนพิกัดในแนวแกน x ของจุดต่อ i

y_i แทนพิกัดในแนวแกน y ของจุดต่อ i

โดยฟังก์ชันการประมาณภายในเอลิเมนต์สามารถเขียนอยู่ในรูปฟังก์ชั่นของพิกัดธรรมชาติได้ดังสมการ (3.30) พิจารณาเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อที่จุดต่อ 1, 4 และ 8 ถูกยุบมารวมกันอยู่ที่ตำแหน่งปลายรอยร้าวและจุดต่อ 7, 5 และ 9 ถูกเลื่อนมาอยู่ที่ตำแหน่งหนึ่งในสี่ของด้านใกล้กับปลายรอยร้าวดังแสดงในรูปที่ ง.2 โดยที่ตำแหน่งบนพิกัด x-y ของจุดต่อต่าง ๆ มีค่าดังนี้

$$x_1 = x_4 = x_8 = 0,\ x_5 = x_7 = x_9 = \frac{a}{4},\ x_2 = x_3 = x_6 = a \qquad (ง.3ก)$$

$$y_1 = y_4 = y_6 = y_8 = y_9 = 0,\ y_5 = -y_7 = -\frac{b}{4},\ y_2 = -y_3 = -b \qquad (ง.3ข)$$



รูปที่ ง.2 แสดงเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อที่ด้านหนึ่งถูกยุบมารวมกันเป็นเอลิเมนต์สามเหลี่ยมและจุดต่อกึ่งกลางด้านและกึ่งกลางเอลิเมนต์ถูกเลื่อนมาอยู่ที่ตำแหน่งหนึ่งในสี่

แทนสมการ (ง.3ก) และ (ง.3ข) ลงในสมการ (ง.2ก) และ (ง.2ข) จะได้พิกัด x และ y ซึ่งเขียนอยู่ในรูปฟังก์ชั่นพิกัดธรรมชาติเป็น

$$x = \xi(1+\xi)\left(1-\eta^2\right)\frac{a}{2} + \left[\xi(1+\xi)\eta(1+\eta) - \xi(1+\xi)\eta(1-\eta) + \left(1-\xi^2\right)\left(1-\eta^2\right)\right]\frac{a}{4} + \left[\left(1-\xi^2\right)\eta(1+\eta) - \left(1-\xi^2\right)\eta(1-\eta)\right]\frac{a}{8} \qquad \text{(ง.4ก)}$$

$$y = \left[\left(1-\xi^2\right)\eta(1-\eta) + \left(1-\xi^2\right)\eta(1+\eta)\right]\frac{b}{8} + \left[\xi(1+\xi)\eta(1-\eta) + \xi(1+\xi)\eta(1+\eta)\right]\frac{b}{4} \qquad \text{(ง.4ข)}$$

ซึ่งเมื่อกระจายแล้วรวมพจน์ต่าง ๆ แล้วจะได้

$$x = (1+\xi)^2\frac{a}{4} \qquad \text{(ง.5ก)}$$

$$y = (1+\xi)^2\eta\frac{b}{4} \qquad \text{(ง.5ข)}$$

โดยสัมพันธ์กับระยะในแนวรัศมีจากปลายรอยร้าวและมุมดังแสดงในรูป ง.3 เป็น

$$r = \sqrt{x^2+y^2} = \frac{b}{4}(1+\xi)^2\sqrt{\left(\frac{a}{b}\right)^2+\eta^2} \qquad \text{(ง.6ก)}$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) = \tan^{-1}\left(\frac{b\eta}{a}\right) \qquad (ง.6ข)$$

จากสมการ (ง.5) เราสามารถคำนวณหาเมตริกซ์ยาโคบี (Jacobian Matrix) ได้เท่ากับ

$$[J] = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \dfrac{a}{2}(1+\xi) & \dfrac{b}{2}\eta(1+\xi) \\ 0 & \dfrac{b}{4}(1+\xi)^2 \end{bmatrix} \qquad (ง.7)$$

ดังนั้นค่าดีเทอร์มิแนนท์ (Determinant) และอินเวอร์ส (Inverse) ของเมตริกซ์ยาโคบีจะเท่ากับ

$$\det|J| = \frac{ab}{8}(1+\xi)^3 \qquad (ง.8ก)$$

$$[J]^{-1} = \begin{bmatrix} \dfrac{\partial \xi}{\partial x} & \dfrac{\partial \eta}{\partial x} \\ \dfrac{\partial \xi}{\partial y} & \dfrac{\partial \eta}{\partial y} \end{bmatrix} = \begin{bmatrix} \dfrac{2}{a(1+\xi)} & \dfrac{-4\eta}{a(1+\xi)^2} \\ 0 & \dfrac{4}{b(1+\xi)^2} \end{bmatrix} \qquad (ง.8ข)$$

โดยค่าการเคลื่อนตัวที่เกิดในเอลิเมนต์สามารถหาได้จากสมการ

$$u = \sum_{i=1}^{9} N_i u_i \qquad (ง.9ก)$$

$$v = \sum_{i=1}^{9} N_i v_i \qquad (ง.9ข)$$

โดยที่ u_i แทนการเคลื่อนตัวในแนวแกน x ของจุดต่อ i

v_i แทนการเคลื่อนตัวในแนวแกน y ของจุดต่อ i

ในทำนองเดียวกันค่าอนุพันธ์ของการเคลื่อนตัวเทียบกับพิกัดธรรมชาติสามารถหาได้จากสมการ

$$\frac{\partial u}{\partial \xi} = \sum_{i=1}^{9} \frac{\partial N_i}{\partial \xi} u_i, \quad \frac{\partial u}{\partial \eta} = \sum_{i=1}^{9} \frac{\partial N_i}{\partial \eta} u_i \qquad (ง.10ก)$$

$$\frac{\partial v}{\partial \xi} = \sum_{i=1}^{9} \frac{\partial N_i}{\partial \xi} v_i, \quad \frac{\partial v}{\partial \eta} = \sum_{i=1}^{9} \frac{\partial N_i}{\partial \eta} v_i \qquad (ง.10ข)$$

โดยค่าอนุพันธ์ของฟังก์ชั่นการประมาณภายในเทียบกับพิกัดธรรมชาติมีค่าเท่ากับ

$$
\begin{aligned}
\frac{\partial N_1}{\partial \xi} &= +\frac{1}{4}(1-2\xi)\eta(1-\eta) & \frac{\partial N_1}{\partial \eta} &= +\frac{1}{4}\xi(1-\xi)(1-2\eta) \\
\frac{\partial N_2}{\partial \xi} &= -\frac{1}{4}(1+2\xi)\eta(1-\eta) & \frac{\partial N_2}{\partial \eta} &= -\frac{1}{4}\xi(1+\xi)(1-2\eta) \\
\frac{\partial N_3}{\partial \xi} &= +\frac{1}{4}(1+2\xi)\eta(1+\eta) & \frac{\partial N_3}{\partial \eta} &= +\frac{1}{4}\xi(1+\xi)(1+2\eta) \\
\frac{\partial N_4}{\partial \xi} &= -\frac{1}{4}(1-2\xi)\eta(1+\eta) & \frac{\partial N_4}{\partial \eta} &= -\frac{1}{4}\xi(1-\xi)(1+2\eta) \\
\frac{\partial N_5}{\partial \xi} &= +\xi\eta(1-\eta) & \frac{\partial N_5}{\partial \eta} &= -\frac{1}{2}(1-\xi^2)(1-2\eta) \\
\frac{\partial N_6}{\partial \xi} &= +\frac{1}{2}(1+2\xi)(1-\eta^2) & \frac{\partial N_6}{\partial \eta} &= -\xi(1+\xi)\eta \\
\frac{\partial N_7}{\partial \xi} &= -\xi\eta(1+\eta) & \frac{\partial N_7}{\partial \eta} &= +\frac{1}{2}(1-\xi^2)(1+2\eta) \\
\frac{\partial N_8}{\partial \xi} &= -\frac{1}{2}(1-2\xi)(1-\eta^2) & \frac{\partial N_8}{\partial \eta} &= +\xi(1-\xi)\eta \\
\frac{\partial N_9}{\partial \xi} &= -2\xi(1-\eta^2) & \frac{\partial N_9}{\partial \eta} &= -2\eta(1-\xi^2)
\end{aligned}
\tag{ง.11}
$$

แทนค่าสมการ (ง.11) ลงในสมการ (ง.10) จะได้

$$
\begin{aligned}
\frac{\partial u}{\partial \xi} = & \left[\frac{3u_1}{4}(\eta-\eta^2)+\frac{u_2}{4}(\eta-\eta^2)-\frac{u_3}{4}(\eta+\eta^2)\right. \\
& -\frac{3u_4}{4}(\eta+\eta^2)-u_5(\eta-\eta^2)-\frac{u_6}{2}(1-\eta^2) \\
& \left.-u_7(-\eta-\eta^2)-\frac{3u_8}{2}(1-\eta^2)-2u_9(-1+\eta^2)\right] \\
& +(1+\xi)\left[\frac{u_1}{2}(-\eta+\eta^2)+\frac{u_2}{2}(-\eta+\eta^2)+\frac{u_3}{2}(\eta+\eta^2)\right. \\
& +\frac{u_4}{2}(\eta+\eta^2)+u_5(\eta-\eta^2)+u_6(1-\eta^2) \\
& \left.+u_7(-\eta-\eta^2)+u_8(1-\eta^2)+2u_9(-1+\eta^2)\right]
\end{aligned}
\tag{ง.12ก}
$$

$$
\begin{aligned}
\frac{\partial u}{\partial \eta} = & \left[\frac{u_1}{2}(-1+2\eta)+\frac{u_4}{2}(1+2\eta)-2u_8\eta\right] \\
& +(1+\xi)\left[\frac{3u_1}{4}(1-2\eta)+\frac{u_2}{4}(1-2\eta)+\frac{u_3}{4}(-1-2\eta)+\frac{3u_4}{4}(-1-2\eta)\right. \\
& \left.+u_5(-1+2\eta)+u_6\eta+u_7(1+2\eta)+3u_8\eta-4u_9\eta\right]
\end{aligned}
$$

$$+\left(1+\xi\right)^{2}\left[\frac{u_1}{4}\left(2\eta-1\right)+\frac{u_2}{4}\left(-1+2\eta\right)+\frac{u_3}{4}\left(1+2\eta\right)+\frac{u_4}{4}\left(1+2\eta\right)\right.$$
$$\left.+\frac{u_5}{2}\left(1-2\eta\right)-u_6\eta+\frac{u_7}{2}\left(-1-2\eta\right)-u_8\eta+2u_9\eta\right] \quad (ง.12ข)$$

ในทำนองเดียวกันค่าอนุพันธ์ $\frac{\partial v}{\partial \xi}$ และ $\frac{\partial v}{\partial \eta}$ สามารถหาได้โดยแทนค่าการเคลื่อนตัวที่จุดต่อ u_i ในสมการ (ง.12) ด้วยค่าการเคลื่อนตัวที่จุดต่อ v_i จากการสังเกตเราพบว่าหากกำหนดให้จุดต่อทั้งสามที่ถูกยุบมารวมกันที่ตำแหน่งปลายรอยร้าวเคลื่อนที่ไปพร้อมกัน

$$\begin{aligned} u_1 &= u_4 = u_8 \\ v_1 &= v_4 = v_8 \end{aligned} \quad (ง.13)$$

จะทำให้พจน์แรกทางด้านขวาของสมการ (ง.12ข) มีค่าเป็นศูนย์ จากสมการ (ง.6ข) ที่มุม θ คงที่ใด ๆ จะได้ η มีค่าคงที่ ดังนั้นค่าอนุพันธ์ของฟังก์ชั่นการประมาณภายในเทียบกับพิกัดธรรมชาติที่ตำแหน่งใด ๆ ตามแนวรัศมีจะมีค่าเท่ากับ

$$\frac{\partial u}{\partial \xi} = a_o + a_1\left(1+\xi\right) \quad (ง.14ก)$$

$$\frac{\partial u}{\partial \eta} = b_o + b_1\left(1+\xi\right) + b_2\left(1+\xi\right)^2 \quad (ง.14ข)$$

$$\frac{\partial v}{\partial \xi} = c_o + c_1\left(1+\xi\right) \quad (ง.14ค)$$

$$\frac{\partial v}{\partial \eta} = d_o + d_1\left(1+\xi\right) + d_2\left(1+\xi\right)^2 \quad (ง.14ง)$$

โดยที่ a_o, a_1, b_o, b_1, b_2, c_o, c_1, d_o, d_1, d_2 แทนค่าคงที่ใด ๆ ขึ้นกับค่าการเคลื่อนตัวที่จุดต่อและค่าพิกัดธรรมชาติ η ที่สอดคล้องกับมุม θ นั้น ๆ เนื่องจากสมการความสัมพันธ์ระหว่างค่าความเครียดกับการเคลื่อนตัวมีค่าเป็น

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y}+\frac{\partial v}{\partial x} \end{Bmatrix} \quad (ง.15)$$

จากสมการ (ง.14) และสมการ (ง.8ข) จะได้ค่าอนุพันธ์ของการเคลื่อนตัวเทียบกับพิกัด x-y ที่ตำแหน่งใด ๆ ตามแนวรัศมีเป็น

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial u}{\partial \eta}\frac{\partial \eta}{\partial x}$$
$$= \frac{2a_o}{a(1+\xi)} + \frac{2a_1}{a} - \frac{4\eta b_o}{a(1+\xi)^2} - \frac{4\eta b_1}{a(1+\xi)} - \frac{4\eta b_2}{a} \qquad \text{(ง.16ก)}$$

$$\frac{\partial u}{\partial y} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial u}{\partial \eta}\frac{\partial \eta}{\partial y}$$
$$= \frac{4b_o}{b(1+\xi)^2} + \frac{4b_1}{b(1+\xi)} + \frac{4b_2}{b} \qquad \text{(ง.16ข)}$$

$$\frac{\partial v}{\partial x} = \frac{\partial v}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial v}{\partial \eta}\frac{\partial \eta}{\partial x}$$
$$= \frac{2c_o}{a(1+\xi)} + \frac{2c_1}{a} - \frac{4\eta d_o}{a(1+\xi)^2} - \frac{4\eta d_1}{a(1+\xi)} - \frac{4\eta d_2}{a} \qquad \text{(ง.16ค)}$$

$$\frac{\partial v}{\partial y} = \frac{\partial v}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial v}{\partial \eta}\frac{\partial \eta}{\partial y}$$
$$= \frac{4d_o}{b(1+\xi)^2} + \frac{4d_1}{b(1+\xi)} + \frac{4d_2}{b} \qquad \text{(ง.16ง)}$$

โดยสมการด้านบนสามารถเขียนอยู่ในรูประยะตามแนวรัศมีจากปลายรอยร้าว r ที่มุม θ ใด ๆ ได้โดยแทนความสัมพันธ์ระหว่างค่าระยะตามแนวรัศมี r และ $(1+\xi)$ ดังแสดงในสมการ (ง.6ก) ลงในสมการ (ง.16) เป็น

$$\frac{\partial u}{\partial x} = \frac{A_o}{\sqrt{r}} + \frac{b'_o}{r} + A_1 \qquad \text{(ง.17ก)}$$

$$\frac{\partial u}{\partial y} = \frac{B_o}{\sqrt{r}} + \frac{b''_o}{r} + B_1 \qquad \text{(ง.17ข)}$$

$$\frac{\partial v}{\partial x} = \frac{C_o}{\sqrt{r}} + \frac{d'_o}{r} + C_1 \qquad \text{(ง.17ค)}$$

$$\frac{\partial v}{\partial y} = \frac{D_o}{\sqrt{r}} + \frac{d''_o}{r} + D_1 \qquad \text{(ง.17ง)}$$

โดยที่ A_o, A_1, b'_o, b''_o, B_o, B_1, C_o, C_1, d'_o, d''_o, D_o, D_1 แทนค่าคงที่ใด ๆ ที่ไม่ขึ้นกับระยะตามแนวรัศมี r ดังนั้นจากสมการ (ง.15) และสมการ (ง.17) จะได้ค่าความเครียดแปรผันกับระยะตามแนวรัศมีจากปลายรอยร้าวเป็น $1/\sqrt{r}$ และ $1/r$ ซึ่งเป็นผลเฉลยของค่าความเครียดที่ปลายรอยร้าวสำหรับวัตถุแบบยืดหยุ่นเชิงเส้นและแบบพลาสติกสมบูรณ์ตามลำดับ นอกจากนี้ค่าคงที่ A_1, B_1, C_1, D_1 ยังแสดงให้เห็นว่าเอลิเมนต์ชนิดนี้สามารถใช้ได้กับการวิเคราะห์ซึ่งรวมผลของค่าความเครียดเริ่มต้นเนื่องจากอุณหภูมิอีกด้วย ในกรณีที่จุดต่อที่ปลายรอยร้าวทั้งสามเคลื่อนที่ไปพร้อมกันดังแสดงในสมการ (ง.13) ค่าคงที่ b_o และ d_o จะมีค่าเป็นศูนย์ซึ่งทำให้ค่าคงที่ b'_o, b''_o, d'_o, d''_o มีค่าเป็นศูนย์ ดังนั้นค่าความเครียดจะแปรผันกับระยะตามแนวรัศมีเป็น $1/\sqrt{r}$ ซึ่งใช้ได้กับการวิเคราะห์รอยร้าวสำหรับวัสดุแบบยืดหยุ่นเชิงเส้นเท่านั้น พิจารณาสมการ (ง.17) เมื่อ $r \rightarrow 0$ เราพบว่าพจน์ที่แปรผันตาม $1/r$ จะมีค่ามากเมื่อเปรียบเทียบกับพจน์อื่น ๆ ดังนั้นในการวิเคราะห์รอยร้าวสำหรับวัสดุแบบอิลาสติก-พลาสติกนั้นเราอาจใช้เฉพาะเอลิเมนต์สี่เหลี่ยมเก้าจุดต่อแบบไอโซพาราเมตริกซ์ (หรือเอลิเมนต์สี่เหลี่ยมแปดจุดต่อ) ซึ่งจุดต่อทั้งสามของด้านที่ถูกยุบมารวมกันที่ปลายรอยร้าวสามารถเคลื่อนตัวได้เป็นอิสระต่อกันและจุดต่อกลางด้านทั้งสองที่อยู่ติดกับปลายรอยร้าวและจุดต่อกลางเอลิเมนต์ยังคงอยู่ที่ตำแหน่งกึ่งกลางเป็นเอลิเมนต์ที่ปลายรอยร้าวก็ได้ โดยค่าความเครียดของเอลิเมนต์ชนิดนี้ยังคงแปรผันกับระยะทางตามแนวรัศมีจากปลายรอยร้าวเป็น $1/r$ Li et al. [9] และ Banthia [35] เนื่องจากจุดต่อที่ตำแหน่งกึ่งกลางด้านและกึ่งกลางเอลิเมนต์ไม่ได้ถูกเลื่อนมาที่ตำแหน่งหนึ่งในสี่ของด้านเช่นเดียวกับเอลิเมนต์ที่ใช้ในบริเวณอื่นของปัญหา ดังนั้นเอลิเมนต์รอบปลายรอยร้าวชนิดนี้จึงใช้งานได้สะดวกกว่าเพราะไม่จำเป็นต้องเลื่อนตำแหน่งจุดต่อมาที่ตำแหน่งหนึ่งในสี่

ภาคผนวก จ

รายละเอียดของโปรแกรมคอมพิวเตอร์ JFACTOR

โปรแกรมคอมพิวเตอร์ JFACTOR ที่ได้ประดิษฐ์ขึ้นมีรายละเอียดดังต่อไปนี้

```
C*******************************************************************************
C     A NON-LINEAR ELASTIC FINITE ELEMENT CRACK ANALYSIS PROGRAM               *
C     FOR TWO-DIMENSIONAL PLANE PROBLEMS AND AXISYMMETRIC CASE                 *
C*******************************************************************************
C     This program computes a parameter, J-integral for Ramberg-Osgood         *
C     material behavior including thermal strains, crack face tractions        *
C     and body forces. The solver used is a full-matrix Crout factorisation    *
C     in which a system matrix is seperated into L, U, and transversed L       *
C     matrices. This program must be compiled on Microsoft Developer Studio    *
C     only.                                                                    *
C                                                 Kobsak Potjananapasiri  *
C                                                 Date:29/AUGUST/2005     *
C                                                                              *
C*******************************************************************************
C     The values declared in the parameter statement below should be assigned  *
C     according to the size of the problem.                                    *
C*******************************************************************************
C       MXPOI    = maximum number of nodes in your model.                      *
C       MXELE    = maximum number of elements.                                 *
C       MXDOM    = maximum number of J-integrated domains.                     *
C       MXEDOM   = maximum number of elements in the last integrated domain.   *
C       MXCTE    = maximum number of crack tip elements.                       *
C       MXCFNODE = maximum number of crack face nodes on the largest integrated*
C                  domain(odd number only).                                    *
C       MXNAME   = maximum number of input file names used in this model.      *
C       MXMAT    = maximum number of materials used in this model.             *
C       MXSTATE  = maximum number of analysis states.                          *
C       MXFIG    = maximum significant numbers of analysis state increment for *
C                  file names.                                                 *
C       SCALE    = scaling factor for output results.                          *
C       SCALEmap = scaling factor for mapping results.                         *
C*******************************************************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (PI = 3.14159265358979323846264338327950_8)
      PARAMETER (SCALE=1._8,SCALEmap=1._8)
      PARAMETER (MXPOI=3500,MXELE=1650,MXEDOM=1200,MXDOM=6)
      PARAMETER (MXSTATE=30,MXCTE=50,MXCFNODE=201,MXNAME=40)
      PARAMETER (MXMAT=5,MXFIG=6)
C
      DIMENSION ROR(MXDOM,4)
      DIMENSION STSM(MXPOI*2,MXPOI*2)
      DIMENSION QFI(MXPOI*2),QEX(MXPOI*2)
      DIMENSION PIVOT(MXPOI*2),GM(MXPOI*2)
      DIMENSION FIMECH(MXPOI*2),FITHER(MXPOI*2)
      DIMENSION FIBODY(MXPOI*2),FISURF(MXPOI*2)
      DIMENSION DFITHER(MXPOI*2),DFIBODY(MXPOI*2)
      DIMENSION QFINC(MXPOI*2),QINC(MXPOI*2)
      DIMENSION BDfF(MXPOI*2),TEMPF(MXPOI)
      DIMENSION BDf(MXPOI*2,MXSTATE),TEMP(MXPOI,MXSTATE)
      DIMENSION ForcFAC(MXSTATE),DispFAC(MXSTATE)
      DIMENSION TempFAC(MXSTATE),BodyFAC(MXSTATE)
      DIMENSION ForcFACp(MXSTATE),DispFACp(MXSTATE)
      DIMENSION TempFACp(MXSTATE),BodyFACp(MXSTATE)
      DIMENSION PROP(MXMAT,7),PROPp(MXMAT,7)
      DIMENSION CTQFI(4),CTQEX(4),FDQFI(4),FDQEX(4)
      DIMENSION COORD(MXPOI,2),COORDp(MXPOI,2)
      DIMENSION PT(MXPOI*2),PTp(MXPOI*2)
      DIMENSION SXX(MXPOI),SXXp(MXPOI)
      DIMENSION SXY(MXPOI),SXYp(MXPOI)
      DIMENSION SYY(MXPOI),SYYp(MXPOI)
      DIMENSION SVM(MXPOI),SVMp(MXPOI)
      DIMENSION RTHETA(MXCTE*2+1),RTHETAp(MXCTE*2+1)
```

```fortran
      DIMENSION AJdom(MXDOM),AJface(MXDOM),AJint(MXDOM)
      DIMENSION ErrJ(MXDOM),ErrAVJ(MXDOM)
C
      INTEGER IBC(MXPOI*2)
      INTEGER IFACEN(2,MXCFNODE,MXDOM),NFACEN(2,MXDOM)
      INTEGER IFACEE(2,(MXCFNODE-1)/2,MXDOM)
      INTEGER NEIND(MXDOM),IEIND(MXEDOM,MXDOM)
      INTEGER INTMAT9(MXELE,9),INTMAT9p(MXELE,9),INTMAT(MXELE,6)
      INTEGER IETIP(MXELE),IETIPp(MXELE),IEMAT(MXELE)
      INTEGER ICTEBN(MXCTE*2+1),ICTEBNp(MXCTE*2+1)
      INTEGER ICTN(MXCTE*2+1),ICTNp(MXCTE*2+1)
      INTEGER IOCTE(MXCTE),IOCTEp(MXCTE)
C
      CHARACTER(30) NAME(MXNAME)
      CHARACTER(MXFIG) INCstr,LINCstr,INCRESstr

C     Input file name index.
      INAME  = 1  !First input file name.
C     Adaptive number index.
      IADAPT = 0  !Mesh in this input file is not the adaptive remeshing one.
C=============================================================================
C     Print the title and description of the program on screen.
C=============================================================================
      WRITE(*,15)
   15 FORMAT(' ====================================================',
     *       '===',/,' A NON-LINEAR ELASTIC FINITE ELEMENT CRACK ',
     *       'ANALYSIS PROGRAM',/,' FOR TWO-DIMENSIONAL PLANE PROBLEMS',
     *       ' AND AXISYMMETRIC CASE',/,' ==========================',
     *       '============================',/,1X,'This program ',
     *       'computes the parameter J-integral for',/,1X,
     *       'Ramberg-Osgood material behavior including thermal',/,1X,
     *       'strains, crack face tractions and body forces.',2/,37X,
     *       'Kobsak Potjananapasiri',/,' ===========================',
     *       '=============================')
C=============================================================================
C     Read input data from the first input file.
C=============================================================================
      CALL INPUT(CANGLE,MXNAME,MXPOI,MXELE,MXDOM,MXEDOM,MXCTE,
     *           MXCFNODE,NAME,INAME,IPLANE,NODEK1old,NPOIN,
     *           COORD,NELEM,INTMAT9,IBC,QFI,NDOM,ROR,NEIND,
     *           IEIND,IFACEN,IFACEE,NFACEN,NFACE,IFACE,IETIP,
     *           PROP,TEMPF,BDfF,NODEK1new,ICTEBN,ICTN,NCTN,
     *           RTHETA,IOCTE,NCTELE,ICFLOAD,NSTATE,MXSTATE,
     *           ForcFAC,DispFAC,TempFAC,BodyFAC,ICTETRAN,
     *           INTMAT,NPOINold,CTQFI,FDQFI,INOADAPT,XSHIFT,
     *           ITERTY,BETOK,IDOMTY,IREDSEL,THETA0,IEMAT,
     *           MXMAT,NMAT,IRTemp,IRBody)
C     Number of degrees of freedom.
      NDF = 2
C     Number of equations after crack tip element transformation.
      NEQ = NPOIN*NDF
C     In case of axisymmetric problem, then edit new X coordinates.
      IF(IPLANE.EQ.3) THEN
      DO 220 IP = 1, NPOIN
      COORD(IP,1) = COORD(IP,1) - XSHIFT
  220 CONTINUE
      ENDIF
C=============================================================================
C     Resume analysis if required.
C=============================================================================
      IRESUME = 0
      INCstart = 1
      CALL GETSTRING(NSTATE-1,LINCstr,MXFIG)
      L1 = LEN_TRIM(NAME(INAME))
      L4 = LEN_TRIM(LINCstr)
      OPEN(UNIT=18, FILE=NAME(INAME)(1:L1)//'_'//LINCstr(1:L4)//'.res',
     *     ERR=3005, STATUS='OLD')
      CLOSE(UNIT=18 ,STATUS='KEEP')
      GOTO 3300
 3005 WRITE(*,3010)
 3010 FORMAT(/,' THE PROGRAM HAS DETECTED THAT ALL THE ANALYZES USING'
     *        ,' THIS',/,' INPUT FILE HAS NOT COMPLETED YET',/,
     *       /,' WHAT DO YOU WANT TO DO NEXT ?',
     *       /,' 1 = RESUME ANALYSIS FROM THE PREVIOUS INCREMENT',
     *       /,' 0 = START NEW ANALYSIS FROM THE 1ST INCREMENT')
      READ(*,*,ERR=3005) IRESUME
      IF(IRESUME.EQ.1) THEN
```

```
 3030 WRITE(*,3020)
 3020 FORMAT(/,' PLEASE ENTER THE INCREMENT NUMBER OF THE PREVIOUS',
     *         ' ANALYSIS:')
      READ(*,*,ERR=3030) INCRES
      INCstart = INCRES + 1
      CALL GETSTRING(INCRES,INCRESstr,MXFIG)
      L5 = LEN_TRIM(INCRESstr)
      OPEN(UNIT=19, FILE=NAME(INAME)(1:L1)//'_'//INCRESstr(1:L5)//'.res'
     *            , ERR=3030, STATUS='OLD')
      ENDIF
C     Store iterative solution type for use again after solution mapping.
 3300 ITERTYp = ITERTY
C     Set total nodal displacement and out of balance force vector according to
C     analysis state.
      IF(IRESUME.NE.1) THEN
      DO 250 IEQ = 1, NEQ
      PT(IEQ) = 0._8
      GM(IEQ) = 0._8
  250 CONTINUE
      ELSE
      DO 255 IEQ = 1, NEQ
      READ(19,*) I, PT(I), GM(I)
      IF(I.NE.IEQ) WRITE(*,265) IEQ
  265 FORMAT(/,' EQUATION NO. ',I5,' IN RESUMED FILE IS MISSING')
      IF(I.NE.IEQ) GOTO 3030
  255 CONTINUE
      CLOSE(UNIT=19 ,STATUS='KEEP')
      ENDIF
C================================================================================
C     Begin loop through all increments.
C================================================================================
      DO 1000 INC = INCstart, NSTATE-1
C     Print increment number that is being solved.
      WRITE(*,270) INC, IADAPT
  270 FORMAT(/,' INCREMENT NUMBER =',I3,
     *       /,' ADAPTIVE  NUMBER =',I3)
C     Print input file name that is being solved.
      L1 = LEN_TRIM(NAME(INAME))
      WRITE(*,*)
      WRITE(*,*)'INPUT FILE  = [',NAME(INAME)(1:L1),'.dat]'
C     Print number of nodes, elements and equations being solved.
      WRITE(*,261) NPOIN, NELEM, NEQ
  261 FORMAT(  ' THE FINITE ELEMENT MODEL CONSISTS OF',
     *       /,' NUMBER OF NODES          =',I5,
     *       /,' NUMBER OF ELEMENTS       =',I5,
     *       /,' NUMBER OF EQUATIONS      =',I5)
C     Set total and incremental load and displacement vectors ,and set body
C     force vector at present and next analysis state.
      DO 280 IEQ = 1, NEQ
      IF(IBC(IEQ).EQ.0) THEN   !Nodal force is known.
      QEX(IEQ)  = QFI(IEQ)*ForcFAC(INC+1)
      QINC(IEQ) = QFI(IEQ)*(ForcFAC(INC+1)-ForcFAC(INC)) + GM(IEQ)
      ELSE                     !Nodal displacement is known.
      QEX(IEQ)  = QFI(IEQ)*DispFAC(INC+1)
      QINC(IEQ) = QFI(IEQ)*(DispFAC(INC+1)-DispFAC(INC)) + GM(IEQ)
      ENDIF
      BDf(IEQ,INC)   = BDfF(IEQ)*BodyFAC(INC)
      BDf(IEQ,INC+1) = BDfF(IEQ)*BodyFAC(INC+1)
  280 CONTINUE
C     Set total temperature vector at present and next analysis state.
      DO 285 IN = 1, NPOIN
      TEMP(IN,INC)   = TEMPF(IN)*TempFAC(INC)
      TEMP(IN,INC+1) = TEMPF(IN)*TempFAC(INC+1)
  285 CONTINUE
C     Set contributions of total nodal forces for transforming nodal crack face
C     forces to nodal crack face tractions.
      IF(ICFLOAD.EQ.1) THEN
       DO 286 ICFACE = 1, NFACE
        IF(NFACE.EQ.2) IFACE = ICFACE
        CTQEX(2*IFACE-1) = CTQFI(2*IFACE-1)*ForcFAC(INC+1)
        CTQEX(2*IFACE)   = CTQFI(2*IFACE)*ForcFAC(INC+1)
        FDQEX(2*IFACE-1) = FDQFI(2*IFACE-1)*ForcFAC(INC+1)
        FDQEX(2*IFACE)   = FDQFI(2*IFACE)*ForcFAC(INC+1)
  286  CONTINUE
      ENDIF
C================================================================================
C     Compute system tangent stiffness matrix, incremental thermal and body
```

```
C     force vectors (in the incremental part).
C     IMOD = 1, compute all load vectors.
C          = 2, compute tangent stiffness matrix and incremental load vectors.
C          = 3, compute both load vectors and tangent stiffness matrix.
C===============================================================================
      IMOD = 2       !Incremental part.
      WRITE(*,290)
  290 FORMAT(/,' *** ESTABLISHING ELEMENT TANGENT',
     *         ' STIFFNESS MATRICES AND',
     *       /,'     ASSEMBLING THEM FOR SYSTEM',
     *         ' TANGENT STIFFNESS MATRIX')
      CALL LST(NPOIN,MXPOI,NELEM,MXELE,INC,MXSTATE,IPLANE,
     *         NDF,PI,IMOD,INTMAT9,IETIP,COORD,PT,TEMP,BDf,
     *         STSM,DFITHER,DFIBODY,FIMECH,FITHER,FIBODY,
     *         PROP,IREDSEL,IEMAT,MXMAT,IRTemp,IRBody)
C     Sum all incremental load vectors.
      DO 300 IEQ = 1, NEQ
       IF(IBC(IEQ).EQ.0) THEN
        QFINC(IEQ) = QINC(IEQ) + DFITHER(IEQ) + DFIBODY(IEQ)
       ELSE
        QFINC(IEQ) = QINC(IEQ)
       ENDIF
  300 CONTINUE
C===============================================================================
C     Apply boundary conditions.
C===============================================================================
      WRITE(*,310)
  310 FORMAT(' *** APPLYING BOUNDARY CONDITIONS')
      CALL APPLYBC(NEQ,IBC,STSM,QFINC,MXPOI)
C===============================================================================
C     Change the system tangent stiffness matrix to multiplication of L, U and
C     transveresed L matrices.
C===============================================================================
      WRITE(*,320)
  320 FORMAT(' *** APPLYING CROUT FACTORISATION')
      CALL CROUT(STSM,PIVOT,NEQ,MXPOI)
C===============================================================================
C     Solve a set of simultaneous equations.
C===============================================================================
      WRITE(*,330)
  330 FORMAT(' *** SOLVING A SET OF SIMULTANEOUS EQUATIONS')
      CALL SOLVE(NEQ,STSM,PIVOT,QFINC,MXPOI)
C===============================================================================
C     Obtain new total displacement predictors.
C===============================================================================
      DO 340 IEQ = 1, NEQ
       IF(IBC(IEQ).EQ.0) THEN
        PT(IEQ) = PT(IEQ) + QFINC(IEQ)
       ELSE
        PT(IEQ) = QEX(IEQ)
       ENDIF
  340 CONTINUE
C     Change iterative solution type back after solution mapping.
      ITERTY = ITERTYp
      IREMESH = 0
 9000 CONTINUE    !This 9000 CONTINUE line is from mapping solution scheme.
C===============================================================================
C     Iterates solutions to equilibrium.
C===============================================================================
      WRITE(*,331)
  331 FORMAT(' *** ITERATING SOLUTIONS TO EQUILIBRIUM')
      CALL ITER(PT,BETOK,QEX,IBC,STSM,ITERTY,GM,NPOIN,MXPOI,
     *          MXELE,COORD,INTMAT9,PIVOT,NDF,NELEM,IETIP,
     *          TEMP,MXSTATE,INC,IPLANE,PI,BDf,PROP,FISURF,
     *          BET,ICFLOAD,IREDSEL,IREMESH,BAS,IEMAT,MXMAT,
     *          IRTemp,IRBody)
C===============================================================================
C     Compute nodal stresses.
C===============================================================================
      WRITE(*,332)
  332 FORMAT(/,' *** COMPUTING NODAL STRESSES')
      CALL GAUSSNODE(IPLANE,NDF,NPOIN,MXPOI,INC,MXSTATE,NELEM,
     *               MXELE,IETIP,INTMAT9,COORD,PT,TEMP,SXX,SYY,
     *               SXY,SVM,PROP,MXCTE,ICTN,NCTN,SIGXXMAX,
     *               SIGYYMAX,SIGXYMAX,SIGVMMAX,IREDSEL,IEMAT,
     *               MXMAT)
C===============================================================================
```

```
C     Compute J-integral from domain integral method.
C===============================================================================
      WRITE(*,333)
  333 FORMAT(' *** COMPUTING J-INTEGRAL')
      CALL CJINT(IPLANE,NDF,NDOM,MXDOM,MXEDOM,NEIND,IEIND,
     *           MXCFNODE,IFACEN,IFACEE,NFACEN,NFACE,IFACE,
     *           NPOIN,MXPOI,INC,MXSTATE,MXELE,IETIP,INTMAT9,
     *           COORD,PT,TEMP,BDf,NODEK1new,CANGLE,ROR,PI,
     *           PROP,AJdom,AJface,AJint,FISURF,CTQEX,FDQEX,
     *           ICFLOAD,IDOMTY,IREDSEL,IEMAT,MXMAT)
C===============================================================================
C     Print output results for showing in Tecplot.
C===============================================================================
      CALL GETSTRING(INC,INCstr,MXFIG)
      L1 = LEN_TRIM(NAME(INAME))
      L3 = LEN_TRIM(INCstr)
C     Open Tecplot's output file.
      OPEN(UNIT=16, FILE=NAME(INAME)(1:L1)//'_'//INCstr(1:L3)//'.plt',
     *     STATUS='REPLACE')
C     Write deformed model and nodal stresses.
      WRITE(16,5190)
 5190 FORMAT('VARIABLES = "X-CO","Y-CO","SXX","SYY","SXY","SVM"')
      WRITE(16,5200) NPOIN, NELEM
 5200 FORMAT('ZONE N=',I6,', E=',I6,', F=FEPOINT, ET=QUADRILATERAL')
      DO 5210 IP = 1, NPOIN
      Xnew = COORD(IP,1) + SCALE*PT(2*IP-1)
      Ynew = COORD(IP,2) + SCALE*PT(2*IP)
      IF(IPLANE.EQ.3) Xnew = Xnew + XSHIFT
      WRITE(16,5220)  Xnew,Ynew,SXX(IP),SYY(IP),SXY(IP),SVM(IP)
 5220 FORMAT(6E16.8)
 5210 CONTINUE
C     Write nodal connectivities for each element type.
      DO 5230 IE = 1, NELEM
      IF(IETIP(IE).EQ.1)THEN
      WRITE(16,5240) (INTMAT9(IE,J),J=1,4)
      ELSE
      WRITE(16,5240) (INTMAT9(IE,J),J=1,3), INTMAT9(IE,3)
      ENDIF
 5240 FORMAT(4I6)
 5230 CONTINUE
C     Close this output file.
      CLOSE(UNIT=16 ,STATUS='KEEP')
C===============================================================================
C     Print resuming file.
C===============================================================================
      OPEN(UNIT=20, FILE=NAME(INAME)(1:L1)//'_'//INCstr(1:L3)//'.res',
     *     STATUS='REPLACE')
      DO 3500 IEQ = 1, NEQ
      WRITE(20,*) IEQ, PT(IEQ), GM(IEQ)
 3500 CONTINUE
      CLOSE(UNIT=20 ,STATUS='KEEP')
C===============================================================================
C     Print results on output file for adaptive remeshing in FEMESH v2.1.152
C===============================================================================
      WRITE(*,*)
      WRITE(*,*)'OUTPUT FILE = [',
     *           NAME(INAME)(1:L1)//'_'//INCstr(1:L3)//'.out',']'
C     Open output file.
      OPEN(UNIT=8, FILE=NAME(INAME)(1:L1)//'_'//INCstr(1:L3)//'.out',
     *     STATUS='REPLACE')
C     Write titles of each result.
      WRITE(8,4092)
 4092 FORMAT(2X,'NODE',15X,'U',15X,'V',13X,'Sxx',13X,'Syy'
     *               ,13X,'Sxy',13X,'Svm',15X,'H')
C     Transform new crack tip nodal quantities to the old one for showing and
C     refining in 6-node element mesh of FEMESH v2.1.
      DO 4000 IP = 1, NPOINold
      IF(ICTETRAN.EQ.1 .AND. IP.EQ.NODEK1old) THEN
      WRITE(8,4100) IP,PT(NODEK1new*2-1),PT(NODEK1new*2),
     *              SIGXXMAX,SIGYYMAX,SIGXYMAX,SIGVMMAX,0._8
      ELSE
      WRITE(8,4100) IP,PT(IP*2-1),PT(IP*2),SXX(IP),SYY(IP),SXY(IP),
     *              SVM(IP),0._8
      ENDIF
 4100 FORMAT(I6,7E16.8)
 4000 CONTINUE
C     Compute average J-integral.
```

```
      AvgJ = 0._8
      DO 4120 IDOM = 1, NDOM
      AvgJ = AvgJ + AJint(IDOM)
 4120 CONTINUE
      AvgJ = AvgJ/NDOM
C     Compute relative error between two adjacent domains.
      ErrJ(1) = 0._8
      DO 4140 IDOM = 2, NDOM
      ErrJ(IDOM) = DABS((AJint(IDOM)-AJint(IDOM-1))/AJint(IDOM)*100._8)
 4140 CONTINUE
C     Compute relative errors compared with average J-integral.
      DO 4145 IDOM = 1, NDOM
      ErrAVJ(IDOM) = DABS((AJint(IDOM)-AvgJ)/AvgJ*100._8)
 4145 CONTINUE
C     Write increment number and adaptive remeshing number.
      WRITE(8,4090) INC, IADAPT
 4090 FORMAT(/,' INCREMENT NUMBER   =',I3,3X,'ADAPTIVE NUMBER   =',I3)
C     Print convergence factor and the title of J-integral result.
      WRITE(8,*) 'CONVERGENCE FACTOR =',BET
      WRITE(*,4110)
      WRITE(8,4110)
 4110 FORMAT(/,1X,'[DOMAIN]',3X,'[J-INTEGRAL]',
     *         3X,'[ER. from pDOM]',3X,'[ER. from AVG]')
C     Print J-integral and relative error on each integrated domain.
      DO 4500 IDOM = 1, NDOM
      WRITE(*,4130) IDOM, AJint(IDOM), ErrJ(IDOM), ErrAVJ(IDOM)
      WRITE(8,4130) IDOM, AJint(IDOM), ErrJ(IDOM), ErrAVJ(IDOM)
 4130 FORMAT(2X,I3,3X,E16.8,2(1X,E16.8))
 4500 CONTINUE
C     Print average J-integral.
      WRITE(*,4150) AvgJ
      WRITE(8,4150) AvgJ
 4150 FORMAT(1X,'[AVERAGE J-INTEGRAL]   =',E16.8)
C     Close output file.
      CLOSE(UNIT=8, STATUS='KEEP')
C================================================================================
C     Print average J-integral results
C================================================================================
      OPEN(UNIT=17, FILE=NAME(INAME)(1:L1)//'.Jint', STATUS='REPLACE')
      WRITE(17,4091) INC, IADAPT
 4091 FORMAT(/,' INCREMENT NUMBER   =',I3,3X,'ADAPTIVE NUMBER   =',I3)
      WRITE(17,4151) AvgJ
 4151 FORMAT(1X,'[AVERAGE J-INTEGRAL]   =',E16.8)
C================================================================================
C     Map displacement solutions from previous mesh to the new refined one.
C================================================================================
      IF(INOADAPT.NE.1) THEN !Your model needs to be refined at some increments.
   10  WRITE(*,334)
  334  FORMAT(/,' DO YOU WANT TO READ NEW REFINED MESH INPUT FILE',
     *          ' FOR THIS INCREMENT ?',
     *        /,' 1 = YES',
     *        /,' 0 = NO')
       READ(*,*,ERR=10) IREMESH
       IF(IREMESH.EQ.1) THEN
        IADAPT = IADAPT + 1
C================================================================================
C       Store old mesh input and output necessary variables for mapping labeled
C       after as 'p' which means 'from previous mesh'.
C================================================================================
        IPLANEp = IPLANE
        CANGLEp = CANGLE
        ICTETRANp = ICTETRAN
        ICFLOADp  = ICFLOAD
        NFACEp  = NFACE
        XSHIFTp = XSHIFT
        NPOINp  = NPOIN
        NELEMp  = NELEM
        NCTNp   = NCTN
        NCTELEp = NCTELE
        NMATp   = NMAT
        NSTATEp = NSTATE
        THETA0p = THETA0
        NODEK1oldP = NODEK1old
        NODEK1newP = NODEK1new
C       Store crack tip element index and nodal connectivites.
        DO 1010 IE = 1, NELEM
         IETIPp(IE) = IETIP(IE)
```

```
        DO 1010 IN = 1, 9
         INTMAT9p(IE,IN) = INTMAT9(IE,IN)
 1010   CONTINUE
C       Store nodal stresses, coordinates and displacements.
        DO 1020 IP = 1, NPOIN
         SXXp(IP) = SXX(IP)
         SXYp(IP) = SXY(IP)
         SYYp(IP) = SYY(IP)
         SVMp(IP) = SVM(IP)
         PTp(2*IP-1) = PT(2*IP-1)
         PTp(2*IP)   = PT(2*IP)
         COORDp(IP,1) = COORD(IP,1)
         COORDp(IP,2) = COORD(IP,2)
 1020   CONTINUE
C       Store all crack tip element ordered in C.W. direction.
        DO 1025 I  = 1, NCTELE
         IOCTEp(I) = IOCTE(I)
 1025   CONTINUE
C       Store all crack tip nodes, crack tip element boundary nodes
C       and their relative angles w.r.t. the 1st one.
        DO 1030 I = 1, NCTN
         ICTNp(I)   = ICTN(I)
         ICTEBNp(I) = ICTEBN(I)
         RTHETAp(I) = RTHETA(I)
 1030   CONTINUE
C       Store all material properties.
        NPROP = 7
        IF(IPLANE.EQ.3) NPROP = 6
        DO 1031 IMAT  = 1, NMAT
        DO 1031 IPROP = 1, NPROP
        PROPp(IMAT,IPROP) = PROP(IMAT,IPROP)
 1031   CONTINUE
C       Store factors at each analysis state.
        DO 1032  ISTATE  = 1, NSTATE
        ForcFACp(ISTATE) = ForcFAC(ISTATE)
        DispFACp(ISTATE) = DispFAC(ISTATE)
        TempFACp(ISTATE) = TempFAC(ISTATE)
        BodyFACp(ISTATE) = BodyFAC(ISTATE)
 1032   CONTINUE
C       Store new crack tip boundary conditions.
        IBCCTXp = IBC(2*NODEK1new-1)
        IBCCTYp = IBC(2*NODEK1new)
C================================================================================
C       Read new refined mesh input file.
C================================================================================
        CALL INPUT(CANGLE,MXNAME,MXPOI,MXELE,MXDOM,MXEDOM,MXCTE,
     *             MXCFNODE,NAME,INAME+1,IPLANE,NODEK1old,NPOIN,
     *             COORD,NELEM,INTMAT9,IBC,QFI,NDOM,ROR,NEIND,
     *             IEIND,IFACEN,IFACEE,NFACEN,NFACE,IFACE,IETIP,
     *             PROP,TEMPF,BDfF,NODEK1new,ICTEBN,ICTN,NCTN,
     *             RTHETA,IOCTE,NCTELE,ICFLOAD,NSTATE,MXSTATE,
     *             ForcFAC,DispFAC,TempFAC,BodyFAC,ICTETRAN,
     *             INTMAT,NPOINold,CTQFI,FDQFI,INOADAPT,XSHIFT,
     *             ITERTY,BETOK,IDOMTY,IREDSEL,THETA0,IEMAT,
     *             MXMAT,NMAT,IRTemp,IRBody)
C       Set new number of equations for new refined mesh input data.
        NEQ = NPOIN*NDF
C       In case of axisymmetric problem, then edit new X coordinates.
        IF(IPLANE.EQ.3) THEN
         DO 372 IP = 1, NPOIN
          COORD(IP,1) = COORD(IP,1) - XSHIFT
  372    CONTINUE
        ENDIF
C================================================================================
C       Check consistency between two input files.
C================================================================================
C       Check problem case.
        IF(IPLANEp.NE.IPLANE) THEN
        WRITE(*,370)
  370   FORMAT(/,' PROBLEM CASE OF THE NEW INPUT FILE'
     *         ,' IS NOT EQUAL WITH THAT OF THE PREVIOUS ONE')
        ENDIF
C       Check axis of rotation.
        IF(IPLANEp.EQ.3 .AND. XSHIFTp.NE.XSHIFT) THEN
        WRITE(*,371)
  371   FORMAT(/,' AXIS OF ROTATION OF BOTH INPUT FILES'
     *         ,' DO NOT PASS THE SAME POINT')
```

```
      ENDIF
C     Check crack angle.
      IF(CANGLEp.NE.CANGLE) THEN
      WRITE(*,375)
  375 FORMAT(/,' CRACK ANGLE OF THE NEW INPUT FILE'
     *        ,' IS NOT EQUAL WITH THAT OF THE PREVIOUS ONE')
      ENDIF
C     Check crack tip element type.
      IF(ICTETRANp.NE.ICTETRAN) THEN
      WRITE(*,380)
  380 FORMAT(/,' THE CRACK TIP ELEMENT TYPE OF BOTH INPUT FILES'
     *        ,' ARE NOT EQUAL')
      ENDIF
C     Check crack face traction.
      IF(ICFLOADp.NE.ICFLOAD) THEN
      WRITE(*,385)
  385 FORMAT(/,' THE CRACK FACE TRACTION OF BOTH INPUT FILES'
     *        ,' ARE NOT CONSISTENCY')
      ENDIF
C     Check number of crack faces.
      IF(NFACEp.NE.NFACE) THEN
      WRITE(*,390)
  390 FORMAT(/,' NUMBER OF CRACK FACES OF BOTH INPUT FILES'
     *        ,' ARE NOT EQUAL')
      ENDIF
C     Check material properties.
      IF(NMAT.NE.NMATp) WRITE(*,353) NMATp, NMAT
  353 FORMAT(/,' NUMBER OF MATERIALS IN PREVIOUS INPUT FILE =',I3,
     *       /,' NUMBER OF MATERIALS IN THIS NEW MESH FILE  =',I3,
     *       /,' PLEASE MAKE THEM EQUAL')
C     Check material properties.
      NPROP = 7
      IF(IPLANE.EQ.3) NPROP = 6
      DO 354 IMAT  = 1, NMAT
      DO 354 IPROP = 1, NPROP
      IF(PROP(IMAT,IPROP).NE.PROPp(IMAT,IPROP)) THEN
      WRITE(*,356) IPROP, IMAT
  356 FORMAT(/,' PROPERTY NUMBER',I3,' OF MATERIAL NUMBER',I3,
     *         ' IS NOT EQUAL WITH THAT OF THE PREVIOUS INPUT FILE')
      ENDIF
  354 CONTINUE
C     Check number of analysis states.
      IF(NSTATE.NE.NSTATEp) WRITE(*,357) NSTATEp, NSTATE
  357 FORMAT(/,' NUMBER OF ANALYSIS STATES IN PREVIOUS INPUT FILE ='
     *        ,I3,
     *       /,' NUMBER OF ANALYSIS STATES IN THIS NEW MESH FILE  ='
     *        ,I3,
     *       /,' PLEASE MAKE THEM EQUAL')
C     Check factors at each analysis state.
      DO 358 ISTATE = 1, NSTATE
      IF(ForcFAC(ISTATE).NE.ForcFACp(ISTATE)) WRITE(*,359) ISTATE
      IF(DispFAC(ISTATE).NE.DispFACp(ISTATE)) WRITE(*,361) ISTATE
      IF(TempFAC(ISTATE).NE.TempFACp(ISTATE)) WRITE(*,362) ISTATE
      IF(BodyFAC(ISTATE).NE.BodyFACp(ISTATE)) WRITE(*,363) ISTATE
  359 FORMAT(/,' LOAD FACTOR AT ANALYSIS STATE NUMBER',I3,
     *         ' IS NOT EQUAL WITH THAT OF THE PREVIOUS INPUT FILE')
  361 FORMAT(/,' DISPLACEMENT FACTOR AT ANALYSIS STATE NUMBER',I3,
     *         ' IS NOT EQUAL WITH THAT OF THE PREVIOUS INPUT FILE')
  362 FORMAT(/,' TEMPERATURE FACTOR AT ANALYSIS STATE NUMBER',I3,
     *         ' IS NOT EQUAL WITH THAT OF THE PREVIOUS INPUT FILE')
  363 FORMAT(/,' BODY FORCE FACTOR AT ANALYSIS STATE NUMBER',I3,
     *         ' IS NOT EQUAL WITH THAT OF THE PREVIOUS INPUT FILE')
  358 CONTINUE
C     Check crack tip coordinates and B.C.s.
      IF(COORD(NODEK1new,1).NE.COORDp(NODEK1newP,1) .OR.
     *        COORD(NODEK1new,2).NE.COORDp(NODEK1newP,2)) WRITE(*,364)
  364 FORMAT(/,' CRACK TIP COORDINATES OF TWO MODELS MUST BE EQUAL')
      IF(IBC(2*NODEK1new-1).NE.IBCCTXp
     *   .OR. IBC(2*NODEK1new).NE.IBCCTYp) WRITE(*,365)
  365 FORMAT(/,' CRACK TIP B.C.s OF TWO MODELS MUST BE EQUAL')
C================================================================================
C     Print increment number that is being solved.
C================================================================================
      WRITE(*,351) INC, IADAPT
  351 FORMAT(/,' INCREMENT NUMBER =',I3,
     *       /,' ADAPTIVE  NUMBER =',I3)
C     Print file name that is being solved.
```

```
      L2 = LEN_TRIM(NAME(INAME+1))
      WRITE(*,*)
      WRITE(*,*)'INPUT FILE  = [',NAME(INAME+1)(1:L2),'.dat]'
C     Print number of nodes and elements being solved on screen.
      WRITE(*,352) NPOIN, NELEM, NEQ
  352 FORMAT(  ' THE FINITE ELEMENT MODEL CONSISTS OF',
     *       /,' NUMBER OF NODES          =',I5,
     *       /,' NUMBER OF ELEMENTS       =',I5,
     *       /,' NUMBER OF EQUATIONS      =',I5)
C================================================================================
C     Set total load and displacement vector ,and set body and temperature
C     force vector at present and next state for this new input file.
C================================================================================
      DO 1040 IEQ = 1, NEQ
      IF(IBC(IEQ).EQ.0) THEN !Nodal force is known.
      QEX(IEQ) = ForcFAC(INC+1)*QFI(IEQ)
      ELSE                  !Nodal displacement is known.
      QEX(IEQ) = DispFAC(INC+1)*QFI(IEQ)
      ENDIF
      BDf(IEQ,INC)   = BodyFAC(INC)*BDfF(IEQ)
      BDf(IEQ,INC+1) = BodyFAC(INC+1)*BDfF(IEQ)
 1040 CONTINUE
      DO 1045 IN = 1, NPOIN
      TEMP(IN,INC)   = TempFAC(INC)*TEMPF(IN)
      TEMP(IN,INC+1) = TempFAC(INC+1)*TEMPF(IN)
 1045 CONTINUE
C     Set contributions of total nodal forces for transforming nodal crack
C     face forces to nodal crack face tractions.
      IF(ICFLOAD.EQ.1) THEN
       DO 1050 ICFACE = 1, NFACE
        IF(NFACE.EQ.2) IFACE = ICFACE
        CTQEX(2*IFACE-1) = CTQFI(2*IFACE-1)*ForcFAC(INC+1)
        CTQEX(2*IFACE)   = CTQFI(2*IFACE)*ForcFAC(INC+1)
        FDQEX(2*IFACE-1) = FDQFI(2*IFACE-1)*ForcFAC(INC+1)
        FDQEX(2*IFACE)   = FDQFI(2*IFACE)*ForcFAC(INC+1)
 1050  CONTINUE
      ENDIF
C     Compute tangent stiffness matrix for this new model
      ITERTY = 1
C================================================================================
C     Map all nodal displacements from the old mesh into the new refined one.
C================================================================================
      CALL MAPPING(MXPOI,MXELE,MXCTE,NDF,IBC,QEX,NPOIN,NELEMp,
     *             PT,PTp,COORD,COORDp,INTMAT9p,IETIPp,ICTN,
     *             ICTNp,NCTN,RTHETA,RTHETAp,SXX,SXXp,SXY,SXYp,
     *             SYY,SYYp,SVM,SVMp,IOCTEp,NCTELEp,ICTETRAN,
     *             THETA0p,ICTEBNp)
C================================================================================
C     Print results after mapping for showing in Tecplot.
C================================================================================
      WRITE(*,*) NAME(INAME)
      L1 = LEN_TRIM(NAME(INAME))
      WRITE(*,*) NAME(INAME+1)
      L2 = LEN_TRIM(NAME(INAME+1))
      WRITE(*,*)
      WRITE(*,*)'MAPPING FILE = [',
     *           NAME(INAME)(1:L1)//'_to_'//NAME(INAME+1)(1:L2),
     *          '.plt]'
C     Open Tecplot's mapping file.
      OPEN(UNIT=17,
     *     FILE=NAME(INAME)(1:L1)//'_to_'//NAME(INAME+1)(1:L2)//
     *     '.plt',STATUS='REPLACE')
C     Write deformed model and nodal stresses after mapping.
      WRITE(17,5190)
      WRITE(17,5200)  NPOIN, NELEM
      DO 6210 IP = 1, NPOIN
      Xnew = COORD(IP,1) + SCALEmap*PT(2*IP-1)
      Ynew = COORD(IP,2) + SCALEmap*PT(2*IP)
      IF(IPLANE.EQ.3) Xnew = Xnew + XSHIFT
      WRITE(17,5220)  Xnew,Ynew,SXX(IP),SYY(IP),SXY(IP),SVM(IP)
 6210 CONTINUE
C     Write nodal connectivities according to element type.
      DO 6230 IE = 1, NELEM
      IF(IETIP(IE).EQ.1)THEN
      WRITE(17,5240) (INTMAT9(IE,J),J=1,4)
      ELSE
      WRITE(17,5240) (INTMAT9(IE,J),J=1,3), INTMAT9(IE,3)
```

```
        ENDIF
 6230   CONTINUE
C       Close this Tecplot's mapping file.
        CLOSE(UNIT=17, STATUS='KEEP')
C       Change input file name.
        INAME = INAME + 1
        WRITE(*,*)
C       Iterate new mesh solution to equilibrium.
        GOTO 9000
       ELSE
        IADAPT = 0
       ENDIF
      ENDIF
C     End each increment.
 1000 CONTINUE
C
      STOP
      END
C ============ THE END OF MAIN PROGRAM ========================================


C=============================================================================
C         SUBROUTINES USED IN THIS PROGRAM.
C=============================================================================
C          1. SUBROUTINE INPUT                        !
C          2. SUBROUTINE CRACKFACE                    !
C          3. SUBROUTINE CENTER                       !
C          4. SUBROUTINE FTHETA                       !
C          5. SUBROUTINE XYLOCAL                      !
C          6. SUBROUTINE LST                          !
C          7. SUBROUTINE VOLUMETRIC                   !
C          8. SUBROUTINE DEVIATORIC                   !
C          9. SUBROUTINE BJ9                          !
C         10. SUBROUTINE GVALUE                       !
C         11. SUBROUTINE FINDSTRSS                    !
C         12. SUBROUTINE ASSEMBLE                     !
C         13. SUBROUTINE APPLYBC                      !
C         14. SUBROUTINE CROUT                        !
C         15. SUBROUTINE SOLVE                        !
C         16. SUBROUTINE ITER                         !
C         17. SUBROUTINE GAUSSNODE                    !
C         18. SUBROUTINE VOLSTRESS                    !
C         19. SUBROUTINE DEVSTRESS                    !
C         20. SUBROUTINE TRMAT                        !
C         21. SUBROUTINE CJINT                        !
C         22. SUBROUTINE CJVOL                        !
C         23. SUBROUTINE CJDEV                        !
C         24. SUBROUTINE CJFACE                       !
C         25. SUBROUTINE GETSTRING                    !
C         26. SUBROUTINE MAPPING                      !
C         27. SUBROUTINE TemBDfFUNC                   !
C=============================================================================


      SUBROUTINE INPUT(CANGLE,MXNAME,MXPOI,MXELE,MXDOM,MXEDOM,MXCTE,
     *                 MXCFNODE,NAME,INAME,IPLANE,NODEK1old,NPOIN,
     *                 COORD,NELEM,INTMAT9,IBC,QFI,NDOM,ROR,NEIND,
     *                 IEIND,IFACEN,IFACEE,NFACEN,NFACE,IFACE,IETIP,
     *                 PROP,TEMPF,BDfF,NODEK1new,ICTEBN,ICTN,NCTN,
     *                 RTHETA,IOCTE,NCTELE,ICFLOAD,NSTATE,MXSTATE,
     *                 ForcFAC,DispFAC,TempFAC,BodyFAC,ICTETRAN,
     *                 INTMAT,NPOINold,CTQFI,FDQFI,INOADAPT,XSHIFT,
     *                 ITERTY,BETOK,IDOMTY,IREDSEL,THETA0,IEMAT,
     *                 MXMAT,NMAT,IRTemp,IRBody)
C=============================================================================
C     THIS SUBROUTINE READS THE INPUT FILE AND GENERATES ALL ADDITIONAL DATA
C     NECESSARY FOR ANALYZING MODEL.
C=============================================================================
C     ICASE = 1, SINGLE EDGE CRACKED PANEL (SECP) WITH T0 = 1 AND
C                T(X) = T0*[125+400*X-100X*X]*2.
C     ICASE = 2, AXIALLY CRACKED CYLINDER (ACC) WITH T0 = 0.25, Ri = 20, AND
C                T(r) = T0*[250+800*(r-Ri)-200(r-Ri)**2].
C     ICASE = 3, CIRCUMFERENTIALLY CRACKED CYLINDER (CCC) WITH T0 = 1, Ri = 80,
C                AND T(r) = T0*[125+100*(r-Ri)-6.25(r-Ri)**2].
C     ICASE = 4, CENTER CRACKED PANEL (CCP) WITH T0 = 1 AND T(X) = T0*[100X*X].
C     ICASE = 5, DISC ROTATING WITH A CENTRIFUGAL FORCE, OMEGA = 0.25,
C                BDfFX = 10*OMEGA*OMEGA*X, AND BDfFY = 10*OMEGA*OMEGA*Y.
```

```
C===============================================================================

      PARAMETER (ICASETemp=3, ICASEBDf=5)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION COORD(MXPOI,2),ROR(MXDOM,4)
      DIMENSION QFI(MXPOI*2),PROP(MXMAT,7)
      DIMENSION BDfF(MXPOI*2),TEMPF(MXPOI)
      DIMENSION RTHETA(MXCTE*2+1)
      DIMENSION ForcFAC(MXSTATE),DispFAC(MXSTATE)
      DIMENSION TempFAC(MXSTATE),BodyFAC(MXSTATE)
      DIMENSION ANVALTemp(6),ANVALBDfX(6),ANVALBDfY(6)
      DIMENSION CTQFI(4),FDQFI(4),Xp(3),Yp(3)
      INTEGER INTMAT(MXELE,6),INTMAT9(MXELE,9)
      INTEGER IETIP(MXELE),IEMAT(MXELE),IBC(MXPOI*2)
      INTEGER IFACEN(2,MXCFNODE,MXDOM),NFACEN(2,MXDOM)
      INTEGER IFACEE(2,(MXCFNODE-1)/2,MXDOM)
      INTEGER NEIND(MXDOM),IEIND(MXEDOM,MXDOM)
      INTEGER ICTEBN(MXCTE*2+1),ICTN(MXCTE*2+1)
      INTEGER IOCTE(MXCTE)
      CHARACTER*30 NAME(MXNAME),TEXT,XORY

C===============================================================================
C     Read input data from the input file.
C===============================================================================
      GOTO 8
C     Close this input file after finding an error before open it again.
    9 CLOSE(UNIT=7 , STATUS='KEEP')
      WRITE(*,*)
      WRITE(*,*)'!!!YOUR INPUT FILE HAS AN ERROR, '
      WRITE(*,*)'   PLEASE CORRECT IT BEFORE PROCEEDING!!!'
    8 CONTINUE
C     Read input file name.
    5 WRITE(*,15)
   15 FORMAT(/, ' PLEASE ENTER THE INPUT FILE NAME:')
      READ(*,'(A)', ERR=5) NAME(INAME)
      L1 = LEN_TRIM(NAME(INAME))
C     Open input file.
      OPEN(UNIT=7 , FILE=NAME(INAME)(1:L1)//'.dat', STATUS='OLD')
C===============================================================================
C     PART  1: Input file notes.
C===============================================================================
      READ(7,*,ERR=9) NLINE
      DO 2 ILINE = 1, NLINE
      READ(7,*,ERR=9) TEXT
    2 CONTINUE
C===============================================================================
C     PART  2: The need for adaptive remeshing and problem case.
C===============================================================================
      READ(7,*,ERR=9) TEXT
      IREDSEL = 0
      READ(7,*,ERR=9) INOADAPT, IPLANE,
     *                (IREDSEL,I=1,IPLANE,3), (XSHIFT,I=3,IPLANE,1)
C===============================================================================
C     PART  3: How large the model is ?
C===============================================================================
      READ(7,*,ERR=9) TEXT
      READ(7,*,ERR=9) NPOIN, NELEM, NPBC
      IF(NPOIN.GT.MXPOI) WRITE(*,10) NPOIN
   10 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXPOI TO ', I5)
      IF(NPOIN.GT.MXPOI) STOP
      IF(NELEM.GT.MXELE) WRITE(*,40) NELEM
   40 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXELE TO ', I5)
      IF(NELEM.GT.MXELE) STOP
C     Store number of nodes before element transforming.
      NPOINold = NPOIN
C     Degrees of freedom on a node (u,v).
      NDF = 2
C     Number of equations in 6-node element mesh.
      NEQ = NPOIN*NDF
C===============================================================================
C     PART  4: Iterative type and convergence tolerance factor.
C===============================================================================
      READ(7,*,ERR=9) TEXT
      READ(7,*,ERR=9) ITERTY, BETOK
C===============================================================================
C     PART  5: Description of the crack in your FEM model.
C===============================================================================
```

```
      READ(7,*,ERR=9) TEXT
      READ(7,*,ERR=9) NODEK1old, CANGLE, ICTETRAN, ICFLOAD, NFACE
      IF(ICFLOAD.EQ.1) THEN
      READ(7,*,ERR=9) TEXT
      DO 11 I = 1, NFACE
      READ(7,*,ERR=9) IFACE, CTQFI(2*IFACE-1), CTQFI(2*IFACE)
     *                     , FDQFI(2*IFACE-1), FDQFI(2*IFACE)
   11 CONTINUE
      ENDIF
C================================================================================
C     PART  6: Domain type, number of domain integrals and its characteristic
C              length.
C================================================================================
      READ(7,*,ERR=9) TEXT
      READ(7,*,ERR=9) IDOMTY, NDOM
      IF(IDOMTY.NE.1 .AND. IDOMTY.NE.2 .AND. IDOMTY.NE.3) WRITE(*,123)
  123 FORMAT(/,' INCORRECT DOMAIN TYPE !')
      IF(NDOM.GT.MXDOM) WRITE(*,122) NDOM
  122 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXDOM TO ', I5)
      IF(NDOM.GT.MXDOM) STOP
      READ(7,*,ERR=9) TEXT
      DO 124 ID = 1, NDOM
      IF(IDOMTY.NE.3) THEN
       READ(7,*,ERR=9)  ROR(ID,1)
      ELSE
       READ(7,*,ERR=9) (ROR(ID,I),I=1,4)
      ENDIF
  124 CONTINUE
      IF(IDOMTY.NE.3) THEN
      DO 120 ID = 1, NDOM
      IF(ROR(ID,1).LE.0._8) WRITE(*,121) ID
      IF(ID.GE.2 .AND. ROR(ID,1).LE.ROR(ID-1,1)) WRITE(*,121) ID
  121 FORMAT(/,' CHARACTERISTIC LENGTH OF DOMAIN NO. ',I5,' IS WRONG',
     *       /,' FOR THIS DOMAIN TYPE, IT MUST BE POSITIVE RANGING',
     *       /,' FROM MINIMUM TO MAXIMUM')
  120 CONTINUE
      ENDIF
C================================================================================
C     PART  7: Analysis states and its corresponding factors.
C================================================================================
      READ(7,*,ERR=9) TEXT
      READ(7,*,ERR=9) NSTATE
      IF(NSTATE.GT.MXSTATE) WRITE(*,38) NSTATE
   38 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXSTATE TO ', I5)
      IF(NSTATE.GT.MXSTATE) STOP
      READ(7,*,ERR=9) TEXT
      DO 37 ISTATE = 1, NSTATE
      READ(7,*,ERR=9) ForcFAC(ISTATE), DispFAC(ISTATE),
     *                TempFAC(ISTATE), BodyFAC(ISTATE)
   37 CONTINUE
      IF(ForcFAC(1).NE.0._8 .OR. DispFAC(1).NE.0._8 .OR.
     *   TempFAC(1).NE.0._8 .OR. BodyFAC(1).NE.0._8) WRITE(*,39)
   39 FORMAT(/,' AT THE 1ST ANALYSIS STATE ALL FACTORS MUST BE ZEROS')
      IF(ForcFAC(1).NE.0._8 .OR. DispFAC(1).NE.0._8 .OR.
     *   TempFAC(1).NE.0._8 .OR. BodyFAC(1).NE.0._8) STOP
C================================================================================
C     PART  8: Number of materials and their properties.
C================================================================================
C     PROP(1) = ELAS   = Young's modulus.
C     PROP(2) = PR     = Poisson's ratio.
C     PROP(3) = YSTRSS = yield stress.
C     PROP(4) = AHARD  = strain hardening exponent.
C     PROP(5) = ALPHA  = yield offset.
C     PROP(6) = COTHR  = coeficient of thermal expansion.
C     PROP(7) = THICK  = thinkness.
C================================================================================
      READ(7,*,ERR=9) TEXT
      READ(7,*,ERR=9) NMAT
      IF(NMAT.GT.MXMAT) WRITE(*,115) NMAT
  115 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXMAT TO ', I5)
      IF(NMAT.GT.MXMAT) STOP
      READ(7,*,ERR=9) TEXT
      DO 116 IMAT = 1, NMAT
      IF(IPLANE.EQ.3) THEN
       READ(7,*,ERR=9) (PROP(IMAT,IPROP),IPROP=1,6)
      ELSE
       READ(7,*,ERR=9) (PROP(IMAT,IPROP),IPROP=1,7)
```

```
      ENDIF
  116 CONTINUE
C===============================================================================
C     PART  9: Nodal coordinates.
C===============================================================================
      READ(7,*,ERR=9) TEXT
      DO 20 IP = 1, NPOIN
      READ(7,*,ERR=9) I, COORD(I,1), COORD(I,2)
   20 CONTINUE
C===============================================================================
C     PART 10: Nodal connectivity and material code for each element..
C===============================================================================
      READ(7,*,ERR=9) TEXT
      DO 50 IE = 1, NELEM
      READ(7,*,ERR=9) I, (INTMAT(I,J),J=1,6), IEMAT(I)
      IF(IE.NE.I)  WRITE(*,60) IE
   60 FORMAT(/,' ELEMENT NO. ',I5,' IN DATA FILE IS MISSING')
      IF(IE.NE.I) STOP
      IF(IEMAT(I).GT.NMAT .OR. IEMAT(I).LT.1)  WRITE(*,65) I
   65 FORMAT(/,' ELEMENT NO. ',I5,' HAS WRONG MATERIAL CODE')
      IF(IEMAT(I).GT.NMAT .OR. IEMAT(I).LT.1)  STOP
   50 CONTINUE
C===============================================================================
C     PART 11: BC's.
C===============================================================================
C     IBC(I) = +1, constraint displacement to zero.
C     IBC(I) =  0, free (external force is known, possibly zero).
C     IBC(I) = -1, displacement is prescribed (to be increment).
C     QFI(I) =  0. (displacement is zero)          , if IBC(I) = +1
C     QFI(I) =  fixed nodal force                  , if IBC(I) =  0
C     QFI(I) =  fixed displacement(to be increment), if IBC(I) = -1
C===============================================================================
      DO 70 IEQ = 1, NEQ
      IBC(IEQ) = 0
      QFI(IEQ) = 0._8
   70 CONTINUE
      READ(7,*,ERR=9) TEXT
      DO 80 IPBC = 1, NPBC
      READ(7,*,ERR=9) I, IBC(2*I-1), IBC(2*I), QFI(2*I-1), QFI(2*I)
   80 CONTINUE
      DO 100 INODE = 1, NPOIN
      DO 100 IDF = 1, NDF
      IEQ = (INODE-1)*2 + IDF
      IF(IDF.EQ.1) XorY = 'X'
      IF(IDF.EQ.2) XorY = 'Y'
      IF(IBC(IEQ).NE.1 .AND. IBC(IEQ).NE.0 .AND. IBC(IEQ).NE.-1)
     *                                   WRITE(*,110) INODE, XorY
      IF(IBC(IEQ).EQ. 1 .AND. QFI(IEQ).NE.0._8) WRITE(*,110) INODE, XorY
      IF(IBC(IEQ).EQ.-1 .AND. QFI(IEQ).EQ.0._8) WRITE(*,110) INODE, XorY
  110 FORMAT(' !!B.C. AT NODE',I5,' IN ', A1,' DIRECTION IS INCORECT')
  100 CONTINUE
C===============================================================================
C     PART 12: Nodal temperatures and body forces.
C===============================================================================
C     IRTemp,IRBody = +1, read them through this input file.
C     IRTemp,IRBody =  0, set them all to zeros.
C     IRTemp,IRBody = -1, set them according to equations by creating them in
C                         the subroutine TemBDfFUNC by yourself.
C===============================================================================
      READ(7,*,ERR=9) TEXT
      READ(7,*,ERR=9) IRTemp, IRBody
      IF(IRTemp.EQ.1 .OR. IRBody.EQ.1) THEN
      READ(7,*,ERR=9) TEXT
      DO 160 IP = 1, NPOIN
      IF(IRTemp.EQ.1 .OR. IRBody.EQ.1)
     *READ(7,*,ERR=9) IN, TEMPF(IN), BDfF(2*IN-1), BDfF(2*IN)
      IF(IRTemp.EQ.1 .OR. IRBody.NE.1)
     *READ(7,*,ERR=9) IN, TEMPF(IN)
      IF(IRTemp.NE.1 .OR. IRBody.EQ.1)
     *READ(7,*,ERR=9) IN, BDfF(2*IN-1), BDfF(2*IN)
      IF(IP.NE.IN) WRITE(*,170) IP
  170 FORMAT(/,' TEMPERATURE OR BODY FORCES AT NODE NO. ',I5,
     *         ' IS MISSING')
      IF(IP.NE.IN) STOP
  160 CONTINUE
      ENDIF
C     In case of IRTemp = 0
```

```
      IF(IRTemp.EQ.0) THEN
      DO 165 I = 1, NPOIN
      TEMPF(I) = 0._8
  165 CONTINUE
      ENDIF
C     In case of IRBody = 0
      IF(IRBody.EQ.0) THEN
      DO 175 I = 1, NPOIN
      BDfF(2*I-1) = 0._8
      BDfF(2*I)   = 0._8
  175 CONTINUE
      ENDIF
C     In case of IRTemp = -1
      IF(IRTemp.EQ.-1) THEN
      ICASE = ICASETemp
      DO 166 INODE = 1, NPOIN
      X = COORD(INODE,1)
      Y = COORD(INODE,2)
      CALL TemBDfFUNC(ICASE,X,Y,TEMP,BDfFX,BDfFY)
      TEMPF(INODE) = TEMP
  166 CONTINUE
      ENDIF
C     In case of IRBody = -1
      IF(IRBody.EQ.-1) THEN
      ICASE = ICASEBDf
      DO 167 INODE = 1, NPOIN
      X = COORD(INODE,1)
      Y = COORD(INODE,2)
      CALL TemBDfFUNC(ICASE,X,Y,TEMP,BDfFX,BDfFY)
      BDfF(2*INODE-1) = BDfFX
      BDfF(2*INODE)   = BDfFY
  167 CONTINUE
      ENDIF
C     Close this input file.
      CLOSE(UNIT=7 , STATUS='KEEP')
C================================================================================
C     Find elements and number of elements in each integrated domain.
C================================================================================
      DO 180 ID = 1, NDOM
      NEIND(ID) = 0
  180 CONTINUE
      DO 190 IE = 1, NELEM
      DO 200 ID = 1, NDOM
      DO 210 IN = 1, 3
      NODE = INTMAT(IE,IN)
      DX   = COORD(NODE,1)-COORD(NODEK1old,1)
      DY   = COORD(NODE,2)-COORD(NODEK1old,2)
      IF(IDOMTY.EQ.1) THEN  !Square domain.
       AR = MAX(DABS(DX),DABS(DY))/ROR(ID,1)
      ENDIF
C
      IF(IDOMTY.EQ.2) THEN  !Circular domain.
       AR = DSQRT(DX*DX+DY*DY)/ROR(ID,1)
      ENDIF
C
      IF(IDOMTY.EQ.3) THEN  !Rectangular domain.
       IF(DX.LE.0._8) THEN
        IF(ROR(ID,1).GT.0._8) THEN
         AX = -DX/ROR(ID,1)
        ELSE
         AX = -1._8
        ENDIF
       ELSE
        IF(ROR(ID,3).GT.0._8) THEN
         AX =  DX/ROR(ID,3)
        ELSE
         AX = -1._8
        ENDIF
       ENDIF
C
       IF(DY.LE.0._8) THEN
        IF(ROR(ID,4).GT.0._8) THEN
         AY = -DY/ROR(ID,4)
        ELSE
         AY = -1._8
        ENDIF
       ELSE
```

```
        IF(ROR(ID,2).GT.0._8) THEN
         AY =  DY/ROR(ID,2)
        ELSE
         AY = -1._8
        ENDIF
       ENDIF
       AR = MAX(AX,AY)
      ENDIF
C     Protect round-off error from transforming FLOAT to EXPONENTIAL number.
      IF(AR.GT.1.000000000010000_8)  GOTO 200
  210 CONTINUE
      DO 195 IDHE = ID, NDOM
      NEIND(IDHE) = NEIND(IDHE) + 1
      IEIND(NEIND(IDHE),IDHE) = IE  !Element's number in each integrated domain.
  195 CONTINUE
      GOTO 190
  200 CONTINUE
  190 CONTINUE
C     Check whether number of elements in each integrated domain exceed the
C     maximum.
      DO 205 ID = 1, NDOM
      IF(NEIND(ID).GT.MXEDOM) WRITE(*,206) NEIND(ID)
  206 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXEDOM TO ', I5)
      IF(NEIND(ID).GT.MXEDOM) STOP
  205 CONTINUE
C===============================================================================
C     Find crack face nodes and elements on each integrated domain and transform
C     all crack tip elements to 9-node rectangular collapsed elements.
C===============================================================================
      CALL CRACKFACE(NPOIN,MXPOI,COORD,NELEM,MXELE,INTMAT,NDOM,
     *               MXDOM,ROR,NODEK1old,MXCTE,MXCFNODE,IFACEN,
     *               IFACEE,NFACEN,NFACE,IFACE,INTMAT9,IETIP,
     *               NCTELE,ICTEBN,IOCTE,ICTETRAN,ICFLOAD,CANGLE,
     *               IDOMTY)
C     Set number of crack tip element boundary nodes.
      NCTEBN = 2*NCTELE + 1
C     Set new nodes' coordinates, body forces and temperatures.
      IF(ICTETRAN.EQ.1) THEN
C     For new crack tip nodes, set all quantities as the old crack tip ones.
      DO 215 IP = NPOIN + 1, NPOIN + 2*NCTELE
      COORD(IP,1)  = COORD(NODEK1old,1)
      COORD(IP,2)  = COORD(NODEK1old,2)
      TEMPF(IP)    = TEMPF(NODEK1old)
      BDfF(IP*2-1) = BDfF(NODEK1old*2-1)
      BDfF(IP*2)   = BDfF(NODEK1old*2)
  215 CONTINUE
C     For new 9th nodes of 9-node rectangular crack tip elements.
      DO 218 IE = 1, NCTELE
      IP = NPOIN + 2*NCTELE + IE
      X5 = COORD(INTMAT9(IOCTE(IE),5),1)
      Y5 = COORD(INTMAT9(IOCTE(IE),5),2)
      X6 = COORD(INTMAT9(IOCTE(IE),6),1)
      Y6 = COORD(INTMAT9(IOCTE(IE),6),2)
      X7 = COORD(INTMAT9(IOCTE(IE),7),1)
      Y7 = COORD(INTMAT9(IOCTE(IE),7),2)
      X8 = COORD(INTMAT9(IOCTE(IE),8),1)
      Y8 = COORD(INTMAT9(IOCTE(IE),8),2)
      upX9 = (X7*Y5-X5*Y7)*(X8-X6) - (X8*Y6-X6*Y8)*(X7-X5)
      beX9 = (Y8-Y6)*(X7-X5) - (Y7-Y5)*(X8-X6)
      upY9 = (X8*Y6-X6*Y8)*(Y7-Y5) - (X7*Y5-X5*Y7)*(Y8-Y6)
      beY9 = (X8-X6)*(Y7-Y5) - (X7-X5)*(Y8-Y6)
      COORD(IP,1) = upX9/beX9
      COORD(IP,2) = upY9/beY9
C     Set X and Y coordinates of the 9th node of colapsed crack tip elements.
      X = COORD(IP,1)
      Y = COORD(IP,2)
      DO 140 I = 1, 3
      Xp(I) = COORD(INTMAT(IOCTE(IE),I),1)
      Yp(I) = COORD(INTMAT(IOCTE(IE),I),2)
  140 CONTINUE
C     Calculate each area for computing area coordinates.
      Atot = 0.5_8*( Xp(2)*Yp(3)-Xp(3)*Yp(2)+
     *               (Yp(2)-Yp(3))*Xp(1)+(Xp(3)-Xp(2))*Yp(1) )
      A1   = 0.5_8*( Xp(2)*Yp(3)-Xp(3)*Yp(2)+
     *               (Yp(2)-Yp(3))*X+(Xp(3)-Xp(2))*Y )
      A2   = 0.5_8*( Xp(3)*Yp(1)-Xp(1)*Yp(3)+
     *               (Yp(3)-Yp(1))*X+(Xp(1)-Xp(3))*Y )
```

```
      A3   = 0.5_8*( Xp(1)*Yp(2)-Xp(2)*Yp(1)+
     *               (Yp(1)-Yp(2))*X+(Xp(2)-Xp(1))*Y )
C     Compute area coordinates of the 9th node.
      AL1 = A1/Atot
      AL2 = A2/Atot
      AL3 = A3/Atot
C     Compute interpolation functions of triangular crack tip elements.
      AN1 = 2._8*AL1*AL1-AL1
      AN2 = 2._8*AL2*AL2-AL2
      AN3 = 2._8*AL3*AL3-AL3
      AN4 = 4._8*AL1*AL2
      AN5 = 4._8*AL2*AL3
      AN6 = 4._8*AL3*AL1
C     In case of IRTemp = +1,
C     Get approximate quantities from triangular element before transformation.
      IF(IRTemp.EQ.1) THEN
      DO 220 IN = 1, 6
      INODE = INTMAT(IOCTE(IE),IN)
      ANVALTemp(IN) = TEMPF(INODE)
  220 CONTINUE
      TEMPF(IP) = AN1*ANVALTemp(1) + AN2*ANVALTemp(2) +
     *            AN3*ANVALTemp(3) + AN4*ANVALTemp(4) +
     *            AN5*ANVALTemp(5) + AN6*ANVALTemp(6)
      ENDIF
C     In case of IRBody = +1,
C     Get approximate quantities from triangular element before transformation.
      IF(IRBody.EQ.1) THEN
      DO 221 IN = 1, 6
      INODE = INTMAT(IOCTE(IE),IN)
      ANVALBDfX(IN) = BDfF(INODE*2-1)
      ANVALBDfY(IN) = BDfF(INODE*2)
  221 CONTINUE
      BDfF(IP*2-1) = AN1*ANVALBDfX(1) + AN2*ANVALBDfX(2) +
     *               AN3*ANVALBDfX(3) + AN4*ANVALBDfX(4) +
     *               AN5*ANVALBDfX(5) + AN6*ANVALBDfX(6)
      BDfF(IP*2)   = AN1*ANVALBDfY(1) + AN2*ANVALBDfY(2) +
     *               AN3*ANVALBDfY(3) + AN4*ANVALBDfY(4) +
     *               AN5*ANVALBDfY(5) + AN6*ANVALBDfY(6)
      ENDIF
C     In case of IRTemp = 0,
C     Set 9th nodal temperatures to zeros.
      IF(IRTemp.EQ.0) THEN
      TEMPF(IP) = 0._8
      ENDIF
C     In case of IRBody = 0,
C     Set 9th nodal body forces to zeros.
      IF(IRBody.EQ.0) THEN
      BDfF(IP*2-1) = 0._8
      BDfF(IP*2)   = 0._8
      ENDIF
C     In case of IRTemp = -1,
C     Set 9th nodal temperatures according to equation.
      IF(IRTemp.EQ.-1) THEN
      X = COORD(IP,1)
      Y = COORD(IP,2)
      ICASE = ICASETemp
      CALL TemBDfFUNC(ICASE,X,Y,TEMP,BDfFX,BDfFY)
      TEMPF(IP) = TEMP
      ENDIF
C     In case of IRBody = -1,
C     Set 9th nodal body forces according to equation.
      IF(IRBody.EQ.-1) THEN
      X = COORD(IP,1)
      Y = COORD(IP,2)
      ICASE = ICASEBDf
      CALL TemBDfFUNC(ICASE,X,Y,TEMP,BDfFX,BDfFY)
      BDfF(2*IP-1) = BDfFX
      BDfF(2*IP)   = BDfFY
      ENDIF
C     End each 9th node.
  218 CONTINUE
C     Set their B.C.s free and have zero nodal forces.
      DO 219 IP = NPOIN+1, NPOIN+3*NCTELE
      IBC(IP*2-1)  = 0
      IBC(IP*2)    = 0
      QFI(IP*2-1)  = 0._8
      QFI(IP*2)    = 0._8
```

```
  219 CONTINUE
      ENDIF
C     Store old crack tip node B.C.s and its corresponding known values.
      IBCKX = IBC(NODEK1old*2-1)
      IBCKY = IBC(NODEK1old*2)
      QFIKX = QFI(NODEK1old*2-1)
      QFIKY = QFI(NODEK1old*2)
      IF(IBCKX.EQ.-1 .OR. IBCKY.EQ.-1) WRITE(*,216)
  216 FORMAT(/,' THE CRACK TIP NODE DISPLACEMENTS CANNOT BE PRESCRIBED',
     *         ' TO BE INCREMENT')
      IF(IBCKX.EQ.-1 .OR. IBCKY.EQ.-1) STOP
      IF(NFACE.EQ.1 .AND. IBCKX.EQ.0 .AND. IBCKY.EQ.0) WRITE(*,217)
  217 FORMAT(/,' YOU ARE TAKING ADVANTAGE OF MODEL SYMMETRY',
     *       /,' BOTH CRACK TIP NODE B.C.s CANNOT BE FREE')
      IF(NFACE.EQ.1 .AND. IBCKX.EQ.0 .AND. IBCKY.EQ.0) STOP
C     Transform only fixed displacement B.C.s of the old crack tip node to the
C     new equivalent 6-node element crack tip node of 9-node collapsed crack tip
C     elements.
      IF(ICTETRAN.EQ.1) THEN
       IF(NFACE.EQ.2) THEN
        NODEK1new = NPOIN + NCTELE
       ELSE
        IF(IFACE.EQ.1) THEN  !Model has only   C.W. face.
         NODEK1new = NPOIN + 2*NCTELE
        ELSE                 !Model has only C.C.W. face.
         NODEK1new = NODEK1old
        ENDIF
       ENDIF
C      In X direction.
       IF(IBCKX.EQ.1) THEN
        IBC(2*NODEK1new-1) = 1
        QFI(2*NODEK1new-1) = 0._8
       ELSE
        IBC(2*NODEK1new-1) = 0
        QFI(2*NODEK1new-1) = 0._8
       ENDIF
C      In Y direction.
       IF(IBCKY.EQ.1) THEN
        IBC(2*NODEK1new) = 1
        QFI(2*NODEK1new) = 0._8
       ELSE
        IBC(2*NODEK1new) = 0
        QFI(2*NODEK1new) = 0._8
       ENDIF
      ELSE                   !No crack tip element transformation.
      NODEK1new = NODEK1old
      ENDIF
C===============================================================================
C     Edit 9-node crack tip nodal forces
C     *Remember that this program read only the input file containing only
C      6-node element mesh input data and will automatically transform these
C      data to the mixed 6 and 9-node elemet mesh data (only 9-node elements as
C      crack tip elements), but this process cannot automatically generate
C      correct nodal forces at the tip because after crack tip element
C      transformation there are several crack tip nodes located at the same
C      position. Therefore user must manually input these nodal loads at crack
C      tip nodes ,but for very small crack tip elements these nodal loads can be
C      approximately negligible.
C===============================================================================
      IF(ICFLOAD.EQ.1 .AND. ICTETRAN.EQ.1) THEN
C      Edit loads on the crack tip node of the C.W. crack face.
       IF(NFACE.EQ.2 .OR. IFACE.EQ.1) THEN
        QFI(2*NODEK1old-1) = CTQFI(1)
        QFI(2*NODEK1old)   = CTQFI(2)
        IBC(2*NODEK1old-1) = 0
        IBC(2*NODEK1old)   = 0
       ENDIF
C      Edit loads on the crack tip node of the C.C.W. crack face.
       IF(NFACE.EQ.2 .OR. IFACE.EQ.2) THEN
        QFI(2*(NPOIN+2*NCTELE)-1) = CTQFI(3)
        QFI(2*(NPOIN+2*NCTELE))   = CTQFI(4)
        IBC(2*(NPOIN+2*NCTELE)-1) = 0
        IBC(2*(NPOIN+2*NCTELE))   = 0
       ENDIF
      ENDIF
C     Change old crack tip node B.C.s free only for C.W. crack face.
      IF(ICFLOAD.NE.1 .AND. ICTETRAN.EQ.1
```

```
     *                .AND. (NFACE.EQ.2 .OR. IFACE.EQ.1)) THEN
       QFI(2*NODEK1old-1) = 0._8
       QFI(2*NODEK1old)   = 0._8
       IBC(2*NODEK1old-1) = 0
       IBC(2*NODEK1old)   = 0
      ENDIF
C     Set number of crack tip nodes and their corresponding nodal numbers.
      IF(ICTETRAN.EQ.1) THEN
      NCTN = 2*NCTELE + 1
      ICTN(1) = NODEK1old
      DO 250 IC = 2, NCTN
      ICTN(IC) = NPOIN + IC - 1
  250 CONTINUE
      ELSE
      NCTN = 1
      ICTN(1) = NODEK1old
      ENDIF
C     Find the angle formed by X axis, old crack tip node and the 1st crack tip
C     element boundary node.
      DX = COORD(ICTEBN(1),1) - COORD(NODEK1old,1)
      DY = COORD(ICTEBN(1),2) - COORD(NODEK1old,2)
      CALL FTHETA(DX,DY,THETA0)
C     Find angle of each crack tip element boundary node with respect to the 1st
C     crack tip element boundary node one.
      RTHETA(1) = 0._8
      DO 260 I = 2, NCTEBN
      DX = COORD(ICTEBN(I),1) - COORD(NODEK1old,1)
      DY = COORD(ICTEBN(I),2) - COORD(NODEK1old,2)
      CALL XYLOCAL(DX,DY,DXL,DYL,THETA0)
      CALL FTHETA(DXL,DYL,THETA)
      RTHETA(I) = THETA
  260 CONTINUE
C     Show crack tip element boundary nodes and their relative angles.
      WRITE(*,320)
  320 FORMAT(/,' CRACK TIP ELEMENT BOUNDARY NODES AND',
     *         ' THEIR RELATIVE ANGLES')
      WRITE(*,330)
  330 FORMAT(4X,'NO.',4X,'NODE',8X,'RELATIVE ANGLE')
      DO 340 I = 1, NCTEBN
      WRITE(*,350) I, ICTEBN(I), RTHETA(I)
  350 FORMAT(2X,I4,4X,I5,4X,E21.16)
  340 CONTINUE
C     Edit new number of nodes after crack tip element transformation.
      IF(ICTETRAN.EQ.1) THEN
       NPOIN = NPOIN+3*NCTELE
       IF(NPOIN.GT.MXPOI) WRITE(*,10) NPOIN
       IF(NPOIN.GT.MXPOI) STOP
      ENDIF
C
      RETURN
      END
C================================================================================


      SUBROUTINE CRACKFACE(NPOIN,MXPOI,COORD,NELEM,MXELE,INTMAT,NDOM,
     *                     MXDOM,ROR,NODEK1old,MXCTE,MXCFNODE,IFACEN,
     *                     IFACEE,NFACEN,NFACE,IFACE,INTMAT9,IETIP,
     *                     NCTELE,ICTEBN,IOCTE,ICTETRAN,ICFLOAD,CANGLE,
     *                     IDOMTY)
C================================================================================
C     THIS SUBROUTINE SEARCHES CRACK FACE NODES AND ELEMENTS WITHIN EACH
C     INTEGRATED DOMAIN AND CREATES CRACK TIP ELEMENT INDEX AND NEW NODAL
C     CONNECTIVITY FOR MIXED 6 AND 9-NODE ELEMENT MESH.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ROR(MXDOM,4),COORD(MXPOI,2)
      INTEGER ICWCFN(MXCFNODE),ICCWCFN(MXCFNODE)
      INTEGER IFACEN(2,MXCFNODE,MXDOM),NFACEN(2,MXDOM)
      INTEGER IOCTE(MXCTE),IETIP(MXELE)
      INTEGER ICWCFE((MXCFNODE-1)/2),ICCWCFE((MXCFNODE-1)/2)
      INTEGER IFACEE(2,(MXCFNODE-1)/2,MXDOM)
      INTEGER INTMAT(MXELE,6),INTMAT9(MXELE,9)
      INTEGER ICTEBN(MXCTE*2+1)

C     Search 1st and 2nd C.W. and C.C.W. crack face nodes which are middle and
C     corner nodes respectively in 6-node element mesh. The clockwise and
C     counter clockwise crack face is defined below. The C.W. and C.C.W. nodes
```

```
C     are the nodes on that corresponding face.
C
C
C        *-*-* - * - * - * - * - * - * - * - *
C        I  / \    / \    / \    / \    /
C        * *   *   *   *   *   *   *   *   *
C        I/  E  \ /  E  \ /  E  \ /  E  \ /
C        * - * - * - * - * - * - * - * - *  <-- 2nd   C.W. crack face node
C                                          \  E
C                           C.W. crack face  *  <-- 1st   C.W. crack face node
C                                              \
C             crack growth direction ---->       *  <-- crack tip node
C                                              /
C                         C.C.W. crack face  *  <-- 1st C.C.W. crack face node
C                                          /  B
C        * - * - * - * - * - * - * - * - *  <-- 2nd C.C.W. crack face node
C        I\  B  / \  B  / \  B  / \  B  / \
C        * *   *   *   *   *   *   *   *   *
C        I  \ /    \ /    \ /    \ /    \
C        *-*-* - * - * - * - * - * - * - *
C
C     Note:  E =   C.W. crack face elements
C            B = C.C.W. crack face elements

C     Set the crack tip node in 6-node element mesh as a center node of
C     surrounding crack tip elements to find the other four nodes
C     (1st and 2nd C.W. and C.C.W. crack face nodes) on the crack faces.
      NODEC = NODEK1old
      CALL CENTER(NODEC,NODEK1old,NELEM,MXELE,INTMAT,MXCTE,ICWELEM,
     *            NODECW,MNODECW,ICCWELEM,NODECCW,MNODECCW,IOCTE,NEHCN)
C     Set number of elements having this center node in common as number of
C     crack tip elements.
      NCTELE = NEHCN
C     Store these corner and middle crack face nodes.
        NODECW1ST =  MNODECW
        NODECW2ND =   NODECW
       IELEMCW1ST =  ICWELEM
       NODECCW1ST = MNODECCW
       NODECCW2ND =  NODECCW
      IELEMCCW1ST = ICCWELEM
C     In case of no information of what crack face the model has, Search it by
C     using available information.
      IF(NFACE.NE.2 .AND. ICFLOAD.NE.1) THEN
       VX = COORD(NODECW2ND,1) - COORD(NODEK1old,1)
       VY = COORD(NODECW2ND,2) - COORD(NODEK1old,2)
       QX = DCOSD(CANGLE)
       QY = DSIND(CANGLE)
       DOT = VX*QX + VY*QY
       IF(DOT.GT.0._8) THEN
        IFACE = 2
       ELSE
        IFACE = 1
       ENDIF
      ENDIF
C===============================================================================
C     Transform all 6-node triangular crack tip elements into 9-node collapsed
C     rectangular elements and create crack tip element index, new nodal
C     connectivity and number of nodes.
C===============================================================================
      DO 90 IE = 1, NELEM
      DO 90 IN = 1, 9
      INTMAT9(IE,IN) = 0
   90 CONTINUE
C     Create new nodal connectivity for mixed 6-node and 9-node element mesh.
      DO 100 IE = 1, NELEM
      DO 100 IN = 1, 6
      INTMAT9(IE,IN) = INTMAT(IE,IN)
  100 CONTINUE
C     Set all elements as 6-node elements through the crack tip element index.
C     IETIP(IE) = 0, 6-node element.
C     IETIP(IE) = 1, 9-node crack tip element.
      DO 110 IE = 1, NELEM
      IETIP(IE) = 0
  110 CONTINUE
      IF(ICTETRAN.EQ.1) THEN
C     Add 2 new nodes in each old 6-node crack tip element and create new nodal
C     connectivity for these crack tip elements, also find crack tip element
```

```
C     boundary nodes which are nodes on the outer ring of crack tip elements.
C     *Note: IOCTE(I) are crack tip element's numbers ordered in C.W. direction
C            where IOCTE(1) is the 1st crack tip element's number on the C.W.
C            crack face.
      DO 120 I = 1, NCTELE
      IEM  = IOCTE(I)
      NADD = NPOIN + (I-1)*2
      DO 130 II = 1, 3
      IF(INTMAT(IEM,II).EQ.NODEK1old) THEN
       IF(I.EQ.1) THEN
        INTMAT9(IEM,II) = NODEK1old
C       Find the first crack tip element boundary node.
        I1 = II + 2
        IF(I1.GT.3) I1 = I1 - 3
        ICTEBN(1) = INTMAT(IEM,I1)
       ELSE
        INTMAT9(IEM,II) = NADD
       ENDIF
C      Find next (middle) crack tip element boundary node.
       I2 = II + 4
       IF(II.EQ.3) I2 = 4
       ICTEBN(I*2) = INTMAT(IEM,I2)
C      Find next (corner) crack tip element boundary node.
       I3 = II + 1
       IF(II.EQ.3) I3 = 1
       ICTEBN(I*2+1) = INTMAT(IEM,I3)
C      Arrange new middle nodes of new nodal connectivity
       INTMAT9(IEM,4+II) = NADD + 1
       DO 140 III = 4+II+1, 8
       INTMAT9(IEM,III) = INTMAT(IEM,III-2)
  140  CONTINUE
C      Arrange new corner nodes of new nodal connectivity
       INTMAT9(IEM,1+II) = NADD + 2
       DO 150 III = 1+II+1, 4, 1
       INTMAT9(IEM,III) = INTMAT(IEM,III-1)
  150  CONTINUE
       GOTO 160
      ELSE
       INTMAT9(IEM,II)   = INTMAT(IEM,II)
       INTMAT9(IEM,4+II) = INTMAT(IEM,3+II)
      ENDIF
  130 CONTINUE
C     Set this element as crack tip element.
  160 IETIP(IEM) = 1
  120 CONTINUE
C     Add a new node (9th node of 9-node quadrilateral crack tip element) on
C     each crack tip element nodal connectivity.
      DO 125 I = 1, NCTELE
      INTMAT9(IOCTE(I),9) = NPOIN + 2*NCTELE + I
  125 CONTINUE
      ENDIF
C================================================================================
C     Find crack face nodes and elements in each integrated domain.
C================================================================================
      IF(ICFLOAD.EQ.1) THEN
      DO 9000 IDOM = 1, NDOM
C================================================================================
C     In C.W. crack face.
C================================================================================
      IF( NFACE.EQ.2 .OR. IFACE.EQ.1 ) THEN
C     Set 1st and 2nd node and 1st element of C.W. crack face.
      NCWCFN = 2
      ICWCFE(1) = IELEMCW1ST
      ICWCFN(1) =  NODECW1ST
      ICWCFN(2) =  NODECW2ND
      DO 230 IH = 1, (MXCFNODE-1)/2-1
      NODEC = ICWCFN(IH*2)
      CALL CENTER(NODEC,NODEK1old,NELEM,MXELE,INTMAT,MXCTE,ICWELEM,
     *            NODECW,MNODECW,ICCWELEM,NODECCW,MNODECCW,IOCTE,NEHCN)
C
      DCWX  = COORD(NODECW,1)-COORD(NODEK1old,1)
      DCWY  = COORD(NODECW,2)-COORD(NODEK1old,2)
      IF(IDOMTY.EQ.1) THEN  !Square domain.
       ACWR = MAX(DABS(DCWX),DABS(DCWY))/ROR(IDOM,1)
      ENDIF
C
      IF(IDOMTY.EQ.2) THEN  !Circular domain.
```

```
       ACWR = DSQRT(DCWX*DCWX+DCWY*DCWY)/ROR(IDOM,1)
      ENDIF
C
      IF(IDOMTY.EQ.3) THEN  !Rectangular domain.
       IF(DCWX.LE.0._8) THEN
        IF(ROR(IDOM,1).GT.0._8) THEN
         ACWX = -DCWX/ROR(IDOM,1)
        ELSE
         ACWX = -1._8
        ENDIF
       ELSE
        IF(ROR(IDOM,3).GT.0._8) THEN
         ACWX =  DCWX/ROR(IDOM,3)
        ELSE
         ACWX = -1._8
        ENDIF
       ENDIF
C
       IF(DCWY.LE.0._8) THEN
        IF(ROR(IDOM,4).GT.0._8) THEN
         ACWY = -DCWY/ROR(IDOM,4)
        ELSE
         ACWY = -1._8
        ENDIF
       ELSE
        IF(ROR(IDOM,2).GT.0._8) THEN
         ACWY =  DCWY/ROR(IDOM,2)
        ELSE
         ACWY = -1._8
        ENDIF
       ENDIF
       ACWR = MAX(ACWX,ACWY)
      ENDIF
C     Protect round-off error when transforming FLOAT to EXPONENTIAL number.
      IF(ACWR.LE.1.000000000010000_8) THEN
      ICWCFN(IH*2+1) = MNODECW
      ICWCFN(IH*2+2) =  NODECW
      ICWCFE(IH+1)   = ICWELEM
      NCWCFN = NCWCFN + 2
      IF(DABS(ACWR-1._8).LE.1.E-12) GOTO 235
      ELSE
      GOTO 235
      ENDIF
  230 CONTINUE
      WRITE(*,*)'PLEASE INCREASE PARAMETER MXCFNODE'
      STOP
  235 CONTINUE
C     Arrange C.W. crack face nodes and elements according to crack face
C     expression of J-integral.
      IFACEN(1,NCWCFN+1,IDOM) = NODEK1old
      DO 250 I = 1, NCWCFN
      IFACEN(1,I,IDOM) = ICWCFN(NCWCFN+1-I)
  250 CONTINUE
      DO 255 I = 1, NCWCFN/2
      IFACEE(1,I,IDOM) = ICWCFE(NCWCFN/2+1-I)
  255 CONTINUE
      NFACEN(1,IDOM) = NCWCFN + 1
C     End C.W. crack face.
      ENDIF
C==============================================================================
C     In C.C.W. crack face.
C==============================================================================
      IF( NFACE.EQ.2 .OR. IFACE.EQ.2 ) THEN
C     Set 1st and 2nd node and 1st element of C.C.W. crack face.
      NCCWCFN = 2
      ICCWCFE(1) = IELEMCCW1ST
      ICCWCFN(1) =  NODECCW1ST
      ICCWCFN(2) =  NODECCW2ND
      DO 240  IH = 1, (MXCFNODE-1)/2-1
      NODEC = ICCWCFN(IH*2)
      CALL CENTER(NODEC,NODEK1old,NELEM,MXELE,INTMAT,MXCTE,ICWELEM,
     *             NODECW,MNODECW,ICCWELEM,NODECCW,MNODECCW,IOCTE,NEHCN)
C
      DCCWX  = COORD(NODECCW,1)-COORD(NODEK1old,1)
      DCCWY  = COORD(NODECCW,2)-COORD(NODEK1old,2)
      IF(IDOMTY.EQ.1) THEN  !Square domain.
       ACCWR = MAX(DABS(DCCWX),DABS(DCCWY))/ROR(IDOM,1)
```

```
      ENDIF
C
      IF(IDOMTY.EQ.2) THEN  !Circular domain.
       ACCWR = DSQRT(DCCWX*DCCWX+DCCWY*DCCWY)/ROR(IDOM,1)
      ENDIF
C
      IF(IDOMTY.EQ.3) THEN  !Rectangular domain.
       IF(DCCWX.LE.0._8) THEN
        IF(ROR(IDOM,1).GT.0._8) THEN
         ACCWX = -DCCWX/ROR(IDOM,1)
        ELSE
         ACCWX = -1._8
        ENDIF
       ELSE
        IF(ROR(IDOM,3).GT.0._8) THEN
         ACCWX =  DCCWX/ROR(IDOM,3)
        ELSE
         ACCWX = -1._8
        ENDIF
       ENDIF
C
       IF(DCCWY.LE.0._8) THEN
        IF(ROR(IDOM,4).GT.0._8) THEN
         ACCWY = -DCCWY/ROR(IDOM,4)
        ELSE
         ACCWY = -1._8
        ENDIF
       ELSE
        IF(ROR(IDOM,2).GT.0._8) THEN
         ACCWY =  DCCWY/ROR(IDOM,2)
        ELSE
         ACCWY = -1._8
        ENDIF
       ENDIF
       ACCWR = MAX(ACCWX,ACCWY)
      ENDIF
C     Protect round-off error when transforming FLOAT to EXPONENTIAL number.
      IF(ACCWR.LE.1.000000000010000_8) THEN
      ICCWCFN(IH*2+1) = MNODECCW
      ICCWCFN(IH*2+2) =  NODECCW
      ICCWCFE(IH+1)   = ICCWELEM
      NCCWCFN = NCCWCFN + 2
      IF(DABS(ACCWR-1._8).LE.1.E-12) GOTO 245
      ELSE
      GOTO 245
      ENDIF
  240 CONTINUE
      WRITE(*,*)'PLEASE INCREASE PARAMETER MXCFNODE'
      STOP
  245 CONTINUE
C     Arrange C.C.W. crack face nodes and elements according to crack face
C     expression of J-integral.
      IF(ICTETRAN.EQ.1) THEN
      IFACEN(2,1,IDOM) = NPOIN + 2*NCTELE
      ELSE
      IFACEN(2,1,IDOM) = NODEK1old
      ENDIF
      DO 280 I = 1, NCCWCFN
      IFACEN(2,I+1,IDOM) = ICCWCFN(I)
  280 CONTINUE
      DO 285 I = 1, NCCWCFN/2
      IFACEE(2,I,IDOM) = ICCWCFE(I)
  285 CONTINUE
      NFACEN(2,IDOM) = NCCWCFN + 1
C     End C.C.W. crack face.
      ENDIF
C=============================================================================
C     Check whether number of crack face nodes in each integrated domain exceeds
C     its maximum.
C=============================================================================
      DO 300 ICFACE = 1, NFACE
      IF(NFACE.EQ.2) IFACE = ICFACE
      IF(NFACEN(IFACE,IDOM).GT.MXCFNODE) WRITE(*,310) NFACEN(IFACE,IDOM)
  310 FORMAT(/,'PLEASE INCREASE PARAMETER MXCFNODE TO',I4)
      IF(NFACEN(IFACE,IDOM).GT.MXCFNODE) STOP
  300 CONTINUE
C=============================================================================
```

```
C     Show results on screen.
C===============================================================================
      WRITE(*,*)
      WRITE(*,1003) IDOM
 1003 FORMAT(' [INTEGRATED DOMAIN NUMBER ',I3,']')
      IF(NFACE.EQ.2) THEN    !The crack has two faces.
C      Show   C.W. crack face nodes on an integrated domain.
       WRITE(*,1004) NFACEN(1,IDOM)
 1004  FORMAT('   C.W. FACE HAS',I3,' NODES WHICH ARE')
       WRITE(*,*) (IFACEN(1,I,IDOM),I=1,NFACEN(1,IDOM))
C      Show   C.W. crack face elements on an integrated domain.
       WRITE(*,1020) (NFACEN(1,IDOM)-1)/2
 1020  FORMAT('   C.W. FACE HAS',I3,' ELEMENTS WHICH ARE')
       WRITE(*,*) (IFACEE(1,I,IDOM),I=1,(NFACEN(1,IDOM)-1)/2)
C      Show C.C.W. crack face nodes on an integrated domain.
       WRITE(*,1005) NFACEN(2,IDOM)
 1005  FORMAT(' C.C.W. FACE HAS',I3,' NODES WHICH ARE')
       WRITE(*,*) (IFACEN(2,I,IDOM),I=1,NFACEN(2,IDOM))
C      Show C.C.W. crack face elements on an integrated domain.
       WRITE(*,1030) (NFACEN(2,IDOM)-1)/2
 1030  FORMAT(' C.C.W. FACE HAS',I3,' ELEMENTS WHICH ARE')
       WRITE(*,*) (IFACEE(2,I,IDOM),I=1,(NFACEN(2,IDOM)-1)/2)
      ELSE    !The crack has only one face.
       IF(IFACE.EQ.1) THEN   !The crack has only   C.W. face.
        WRITE(*,1004) NFACEN(1,IDOM)
        WRITE(*,*) (IFACEN(1,I,IDOM),I=1,NFACEN(1,IDOM))
        WRITE(*,1020) (NFACEN(1,IDOM)-1)/2
        WRITE(*,*) (IFACEE(1,I,IDOM),I=1,(NFACEN(1,IDOM)-1)/2)
       ELSE                  !The crack has only C.C.W. face.
        WRITE(*,1005) NFACEN(2,IDOM)
        WRITE(*,*) (IFACEN(2,I,IDOM),I=1,NFACEN(2,IDOM))
        WRITE(*,1030) (NFACEN(2,IDOM)-1)/2
        WRITE(*,*) (IFACEE(2,I,IDOM),I=1,(NFACEN(2,IDOM)-1)/2)
       ENDIF
      ENDIF
C     End an integrated domain.
 9000 CONTINUE
      ENDIF
C
      RETURN
      END
C===============================================================================


      SUBROUTINE CENTER(NODEC,NODEK1old,NELEM,MXELE,INTMAT,MXCTE,
     *                  ICWELEM,NODECW,MNODECW,ICCWELEM,NODECCW,
     *                  MNODECCW,IOCTE,NEHCN)
C===============================================================================
C     THIS SUBROUTINE FINDS CRACK FACE NODES NEXT TO A SPECIFIC KNOWN CRACK FACE
C     NODE CALLED THE CENTER NODE AND CRACK FACE ELEMENTS CONTAINING THESE
C     NODES. IF A SPECIFIC KNOWN CRACK FACE NODE IS THE OLD CRACK TIP NODE, THE
C     SUBROUTINE ALSO FINDS CRACK TIP ELEMENTS ORDERED IN CLOCKWISE DIRECTION.
C===============================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER INTMAT(MXELE,6), IOCTE(MXCTE)
      INTEGER IEHCN(MXCTE), ICPOS(MXCTE)
      INTEGER ICWPOS(MXCTE) , ICWMPOS(MXCTE)
      INTEGER ICCWPOS(MXCTE), ICCWMPOS(MXCTE)
      INTEGER NCWNI(MXCTE), ICWEI(MXCTE)
      INTEGER NCCWNI(MXCTE), ICCWEI(MXCTE)

C     Number of elements having a center node in common.
      NEHCN = 0
C     Find all elements having this center node in common.
C     Find C.W. and C.C.W. position of corner and middle nodes in   INTMAT
      DO 10 IE  = 1, NELEM
      DO 20 ICN = 1, 3   !Loop over corner nodes.
C     NODEC is the center node.
      IF(INTMAT(IE,ICN).EQ.NODEC) THEN
      NEHCN = NEHCN + 1
C     IEHCN(I)    is the element having this center node.
      IEHCN(NEHCN) = IE
C     ICPOS(I)    is the position in INTMAT of this center node.
      ICPOS(NEHCN) = ICN
C     ICCWPOS(I)  is the position in INTMAT of the corner node next to this
C     center node in C.C.W. direction.
      ICCWPOS(NEHCN)  = ICN + 1
```

```
      IF(ICCWPOS(NEHCN).GT.3) ICCWPOS(NEHCN) = 1
C     ICWPOS(I)   is the position in INTMAT of the corner node next to this
C     center node in   C.W. direction.
      ICWPOS(NEHCN)   = ICN - 1
      IF(ICWPOS(NEHCN).LT.1)  ICWPOS(NEHCN)  = 3
C     ICCWMPOS(I) is the position in INTMAT of the middle node next to this
C     center node in C.C.W. direction.
      ICCWMPOS(NEHCN) = ICN + 3
C     ICWMPOS(I)  is the position in INTMAT of the middle node next to this
C     center node in   C.W. direction.
      ICWMPOS(NEHCN)  = ICN + 2
      IF(ICWMPOS(NEHCN).LT.4) ICWMPOS(NEHCN) = 6
      GOTO 10
      ENDIF
   20 CONTINUE
   10 CONTINUE
C===============================================================================
C     Find intersection between each element that has this center node.
C===============================================================================
      DO 40 IE = 1, NEHCN
C     NCWNI(I)  is number of   C.W. corner node intersections of IEHCN(I) element
C     with elements that have this center node in common.
      NCWNI(IE)  = 0
C     ICWEI(I)  is the element that intersects above   C.W. corner node.
      ICWEI(IE)  = 0
C     NCCWNI(I) is number of C.C.W. corner node intersections of IEHCN(I) element
C     with elements that have this center node in common.
      NCCWNI(IE) = 0
C     ICCWEI(I) is the element that intersects above C.C.W. corner node.
      ICCWEI(IE) = 0
      DO 50  ICHECK = 1, NEHCN
      IF(INTMAT(IEHCN(IE),ICWPOS(IE)).EQ.
     *   INTMAT(IEHCN(ICHECK),ICCWPOS(ICHECK))) THEN
      NCWNI(IE) = NCWNI(IE) + 1
      ICWEI(IE) = IEHCN(ICHECK)
      GOTO 50
      ENDIF
      IF(INTMAT(IEHCN(IE),ICCWPOS(IE)).EQ.
     *   INTMAT(IEHCN(ICHECK),ICWPOS(ICHECK))) THEN
      NCCWNI(IE) = NCCWNI(IE) + 1
      ICCWEI(IE) = IEHCN(ICHECK)
      GOTO 50
      ENDIF
   50 CONTINUE
   40 CONTINUE
C===============================================================================
C     Find C.W. and C.C.W. crack face elements and nodes.
C===============================================================================
      DO 60 IE = 1, NEHCN
      IF(NCWNI(IE).EQ.0) THEN !No intersection, so this is the edge element.
C     ICWELEM is the C.W. edge element.
      ICWELEM = IEHCN(IE)
       IF(NODEC.EQ.NODEK1old) THEN
C      IOCTE(I) is the crack tip elements ordered in C.W. direction.
       IOCTE(1) = IEHCN(IE)
       IOCTE(2) = ICCWEI(IE)
       ENDIF
C     MNODECW is the middle node of the above C.W. edge element.
      MNODECW = INTMAT(IEHCN(IE),ICWMPOS(IE))
C     NODECW  is the corner node of the above C.W. edge element.
      NODECW  = INTMAT(IEHCN(IE),ICWPOS(IE))
      ENDIF
      IF(NCCWNI(IE).EQ.0) THEN !No intersection, so this is the edge element.
C     ICCWELEM is the C.C.W. edge element.
      ICCWELEM = IEHCN(IE)
C     MNODECCW is the middle node of the above C.C.W. edge element.
      MNODECCW = INTMAT(IEHCN(IE),ICCWMPOS(IE))
C     NODECCW  is the corner node of the above C.C.W. edge element.
      NODECCW  = INTMAT(IEHCN(IE),ICCWPOS(IE))
      ENDIF
   60 CONTINUE
C===============================================================================
C     Arrange remaining crack tip elements in clockwise order.
C===============================================================================
      IF(NODEC.EQ.NODEK1old) THEN
      DO 70 IB = 3, NEHCN
      DO 80 IC = 1, NEHCN
```

```
       IF(IOCTE(IB-1).EQ.IEHCN(IC)) THEN
       IOCTE(IB) = ICCWEI(IC)
       GOTO 70
       ENDIF
   80 CONTINUE
   70 CONTINUE
      ENDIF
C
      RETURN
      END
C================================================================================


      SUBROUTINE FTHETA(DX,DY,THETA)
C================================================================================
C     THIS SUBROUTINE COMPUTES THE ANGLE IN (0.- 359.9999...) DEGREES OF A POINT
C     WITH RESPECT TO ANY CARTESIAN COORDINATE SYSTEM.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
C
      AL = DSQRT(DX*DX+DY*DY)
      THETA = DASIND(DY/AL)
C     2nd Quadrant.
      IF(THETA.GE.0._8 .AND. DX.LE.0._8) THETA = 180._8 - THETA
      IF(THETA.LT.0._8) THEN
C      3rd Quadrant.
       IF(DX.LE.0._8) THETA = 180._8 - THETA
C      4th Quadrant.
       IF(DX.GT.0._8) THETA = 360._8 + THETA
      ENDIF
C
      RETURN
      END
C================================================================================


      SUBROUTINE XYLOCAL(DX,DY,DXL,DYL,THETA0)
C================================================================================
C     THIS SUBROUTINE COMPUTES X'-Y' COORDINATE DIFFERENCES OF A POINT. THE
C     X'-Y' CARTESIAN COORDINATE SYSTEM HAS X' AND Y' AXIS MAKING AN ANGLE OF
C     THETA0 AND THETA0-90 DEGREES WITH RESPECT TO X AND Y AXIS RESPECTIVELY.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DXL = DX*DCOSD(THETA0) + DY*DSIND(THETA0)
      DYL = DX*DSIND(THETA0) - DY*DCOSD(THETA0)
C
      RETURN
      END
C================================================================================


      SUBROUTINE LST(NPOIN,MXPOI,NELEM,MXELE,INC,MXSTATE,IPLANE,
     *               NDF,PI,IMOD,INTMAT9,IETIP,COORD,PT,TEMP,BDf,
     *               STSM,DFITHER,DFIBODY,FIMECH,FITHER,FIBODY,
     *               PROP,IREDSEL,IEMAT,MXMAT,IRTemp,IRBody)
C================================================================================
C     THIS SUBROUTINE COMPUTES TANGENT STIFFNESS MATRIX, INTERNAL, THERMAL AND
C     BODY FORCE VECTORS ,AND INCREMENTAL THERMAL AND BODY FORCE VECTORS.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PROP(MXMAT,7)
      DIMENSION COORD(MXPOI,2)
      DIMENSION PT(MXPOI*2)
      DIMENSION TEMP(MXPOI,MXSTATE)
      DIMENSION BDf(MXPOI*2,MXSTATE)
      DIMENSION STSM(MXPOI*2,MXPOI*2),SKM(MXPOI*2,MXPOI*2)
      DIMENSION FIMECH(MXPOI*2),FITHER(MXPOI*2),DFITHER(MXPOI*2)
      DIMENSION FIBODY(MXPOI*2),DFIBODY(MXPOI*2)
      INTEGER INTMAT9(MXELE,9),IETIP(MXELE),IEMAT(MXELE)

C     Set number of equations.
      NEQ = NPOIN*NDF
C================================================================================
C     Compute matrices with lower number of Gauss's points.
C================================================================================
      CALL VOLUMETRIC(NPOIN,MXPOI,NELEM,MXELE,INC,MXSTATE,IPLANE,
```

```
     *                NDF,PI,IMOD,INTMAT9,IETIP,COORD,PT,TEMP,
     *                STSM,DFITHER,FITHER,PROP,IREDSEL,IEMAT,
     *                MXMAT,IRTemp,SKM)
C================================================================================
C     Compute matrices with higher number of Gauss's points.
C================================================================================
      CALL DEVIATORIC(NPOIN,MXPOI,NELEM,MXELE,INC,MXSTATE,IPLANE,
     *                NDF,PI,IMOD,INTMAT9,IETIP,COORD,PT,TEMP,
     *                BDf,STSM,DFITHER,DFIBODY,FITHER,FIBODY,
     *                PROP,IEMAT,MXMAT,IRTemp,IRBody,SKM)
C     Compute mechanical force vector.
      IF(IMOD.NE.2) THEN
      DO 380 I  = 1, NEQ
      FIMECH(I) = 0._8
      DO 380 J  = 1, NEQ
      FIMECH(I) = FIMECH(I) + SKM(I,J)*PT(J)
  380 CONTINUE
      ENDIF
C
      RETURN
      END
C================================================================================


      SUBROUTINE VOLUMETRIC(NPOIN,MXPOI,NELEM,MXELE,INC,MXSTATE,IPLANE,
     *                      NDF,PI,IMOD,INTMAT9,IETIP,COORD,PT,TEMP,
     *                      STSMv,DFIvolT,FIvolT,PROP,IREDSEL,IEMAT,
     *                      MXMAT,IRTemp,SKMv)
C================================================================================
C     THIS SUBROUTINE COMPUTES MATRICES AND VECTORS CORRESPONDING TO VOLUMETRIC
C     STRESS VECTORS BY USING LOWER NUMBER OF GAUSS'S POINTS.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COORD(MXPOI,2)
      DIMENSION X(9),Y(9),XG(9),YG(9),WG(9)
      DIMENSION BMAT(4,9*2),DNDA(9),DNDB(9),AJ(2,2),AJI(2,2)
      DIMENSION PT(MXPOI*2),EDISP(9*2)
      DIMENSION EMAT(4,4),DMAT(4,4),CVEC(4),AVEC(4)
      DIMENSION STSMv(MXPOI*2,MXPOI*2),ETSMv(9*2,9*2)
      DIMENSION SKMv(MXPOI*2,MXPOI*2),EKMv(9*2,9*2)
      DIMENSION EKBG(9*2,9*2)
      DIMENSION STRAIN(4),THETA(4)
      DIMENSION TEMP(MXPOI,MXSTATE),ETEMP(9),DETEMP(9)
      DIMENSION FIvolT(MXPOI*2),DFIvolT(MXPOI*2)
      DIMENSION EFIvolT(9*2),DEFIvolT(9*2)
      DIMENSION PROP(MXMAT,7)
      INTEGER INTMAT9(MXELE,9),IETIP(MXELE),IEMAT(MXELE)

C     Number of equations.
      NEQ = NPOIN*NDF
C     Set element matrix dimensions for analyzed problem.
      IF(IPLANE.EQ.3) THEN
      NDIM = 4  !Axisymmetric case.
      ELSE
      NDIM = 3  !Plane stress or plane strain case.
      ENDIF
C     Set consistent analysis state.
C     IMOD = 1 ,compute load vectors.
C          = 2 ,compute tangent stiffness matrix and incremental load vectors.
C          = 3 ,compute both load vectors and tangent stiffness matrix.
      IF(IMOD.EQ.2) IPOS = INC      !Incremental part.
      IF(IMOD.NE.2) IPOS = INC + 1  !Iterative part.
C     Set initial system tangent stiffness matrix to zero.
      IF(IMOD.NE.1) THEN
      DO 299  I  = 1, NEQ
      DO 299  J  = 1, NEQ
      STSMv(I,J) = 0._8
  299 CONTINUE
      ENDIF
C     Set initial system stiffness matrix to zero.
      IF(IMOD.NE.2) THEN
      DO 298 I  = 1, NEQ
      DO 298 J  = 1, NEQ
      SKMv(I,J) = 0._8
  298 CONTINUE
      ENDIF
C     Set initial system load vectors to zero.
```

```
      IF(IMOD.EQ.2) THEN
      DO 297  I  = 1, NEQ
      DFIvolT(I) = 0._8   !Incremental thermal force vector.
  297 CONTINUE
      ELSE
      DO 296 I  = 1, NEQ
      FIvolT(I) = 0._8    !Thermal force vector.
  296 CONTINUE
      ENDIF
C================================================================================
C     Loop over all elements.
C================================================================================
      DO 5000 IE = 1, NELEM
C     Read material properties of each element.
      ELAS   = PROP(IEMAT(IE),1)
      PR     = PROP(IEMAT(IE),2)
      YSTRSS = PROP(IEMAT(IE),3)
      AHARD  = PROP(IEMAT(IE),4)
      ALPHA  = PROP(IEMAT(IE),5)
      COTHR  = PROP(IEMAT(IE),6)
      IF(IPLANE.NE.3) THICK = PROP(IEMAT(IE),7)
C     9-node rectangular crack tip element.
      IF(IETIP(IE).EQ.1) THEN
      NN   =  9         !Number of nodes.
      NDOF =  NDF*NN    !Number of element degrees of freedom.
C      Set number of Gauss's points, its coordinates and weights.
       IF(IREDSEL.NE.1) THEN
       NG    =  9
       XG(1) = -DSQRT(3._8)/DSQRT(5._8)
       XG(2) =  XG(1)
       XG(3) =  XG(1)
       XG(4) =  0._8
       XG(5) =  XG(4)
       XG(6) =  XG(4)
       XG(7) = -XG(1)
       XG(8) =  XG(7)
       XG(9) =  XG(7)
C
       YG(1) =  XG(1)
       YG(2) =  XG(4)
       YG(3) =  XG(7)
       YG(4) =  XG(1)
       YG(5) =  XG(4)
       YG(6) =  XG(7)
       YG(7) =  XG(1)
       YG(8) =  XG(4)
       YG(9) =  XG(7)
C
       WG(1) =  25._8/81._8
       WG(2) =  40._8/81._8
       WG(3) =  WG(1)
       WG(4) =  WG(2)
       WG(5) =  64._8/81._8
       WG(6) =  WG(2)
       WG(7) =  WG(1)
       WG(8) =  WG(2)
       WG(9) =  WG(1)
       ELSE
       NG    =  4
       XG(1) = -1._8/DSQRT(3._8)
       XG(2) = -1._8/DSQRT(3._8)
       XG(3) =  1._8/DSQRT(3._8)
       XG(4) =  1._8/DSQRT(3._8)
C
       YG(1) = -1._8/DSQRT(3._8)
       YG(2) =  1._8/DSQRT(3._8)
       YG(3) = -1._8/DSQRT(3._8)
       YG(4) =  1._8/DSQRT(3._8)
C
       WG(1) =  1._8
       WG(2) =  1._8
       WG(3) =  1._8
       WG(4) =  1._8
       ENDIF
C     6-node triangular element.
      ELSE
      NN   =  6         !Number of nodes.
```

```
      NDOF =  NDF*NN    !Number of element degrees of freedom.
C      Set number of Gauss's points, its coordinates and weights.
       IF(IREDSEL.NE.1) THEN
       NG    =  7
       XG(1) =  1._8/3._8
       XG(2) =  0.101286507323456_8
       XG(3) =  0.797426985353087_8
       XG(4) =  XG(2)
       XG(5) =  0.470142064105115_8
       XG(6) =  0.059715871789770_8
       XG(7) =  XG(5)
C
       YG(1) =  XG(1)
       YG(2) =  XG(2)
       YG(3) =  XG(2)
       YG(4) =  XG(3)
       YG(5) =  XG(5)
       YG(6) =  XG(5)
       YG(7) =  XG(6)
C
       WG(1) =  0.225_8
       WG(2) =  0.125939180544827_8
       WG(3) =  WG(2)
       WG(4) =  WG(2)
       WG(5) =  0.132394152788506_8
       WG(6) =  WG(5)
       WG(7) =  WG(5)
C
       WG(1) =  WG(1)/2._8
       WG(2) =  WG(2)/2._8
       WG(3) =  WG(3)/2._8
       WG(4) =  WG(4)/2._8
       WG(5) =  WG(5)/2._8
       WG(6) =  WG(6)/2._8
       WG(7) =  WG(7)/2._8
       ELSE
       NG    =  3
       XG(1) =  1._8/6._8
       XG(2) =  2._8/3._8
       XG(3) =  1._8/6._8
C
       YG(1) =  1._8/6._8
       YG(2) =  1._8/6._8
       YG(3) =  2._8/3._8
C
       WG(1) =  1._8/3._8
       WG(2) =  1._8/3._8
       WG(3) =  1._8/3._8
C
       WG(1) =  WG(1)/2._8
       WG(2) =  WG(2)/2._8
       WG(3) =  WG(3)/2._8
       ENDIF
      ENDIF
C     Create elemant nodal displacement and temperature vector.
      DO 100 I = 1, NN
      II          = INTMAT9(IE,I)
      X(I)        = COORD(II,1)
      Y(I)        = COORD(II,2)
      ETEMP(I)    = TEMP(II,IPOS)
      DETEMP(I)   = TEMP(II,INC+1) - TEMP(II,INC)
      EDISP(2*I-1) = PT(2*II-1)
      EDISP(2*I)   = PT(2*II)
  100 CONTINUE
C     Set initial element tangent stiffness matrix to zero.
      IF(IMOD.NE.1) THEN
      DO 300  I  = 1, NDOF
      DO 300  J  = 1, NDOF
      ETSMv(I,J) = 0._8
  300 CONTINUE
      ENDIF
C     Set initial element stiffness matrix to zero.
      IF(IMOD.NE.2) THEN
      DO 301 I  = 1, NDOF
      DO 301 J  = 1, NDOF
      EKMv(I,J) = 0._8
  301 CONTINUE
```

```
      ENDIF
C     Set initial element thermal load vector to zero.
      IF(IMOD.NE.2 .AND. IRTemp.NE.0) THEN
      DO 303  I  = 1, NDOF
      EFIvolT(I) = 0._8
  303 CONTINUE
      ENDIF
C     Set initial element incremental thermal load vector to zero.
      IF(IMOD.EQ.2 .AND. IRTemp.NE.0) THEN
      DO 304   I  = 1, NDOF
      DEFIvolT(I) = 0._8
  304 CONTINUE
      ENDIF
C==============================================================================
C     Loop over each Gauss's point on an element.
C==============================================================================
      DO 336 K = 1, NG
C     Compute strain-displacement matrix.
      A = XG(K)
      B = YG(K)
      CALL BJ9(X,Y,A,B,BMAT,EKBG,AJ,AJI,DETJAC,DNDA,DNDB,
     *         NN,NDF,IPLANE)
C     Compute temperature and temperature increment at a Gauss's point.
      IF(IRTemp.NE.0) THEN
       CALL GVALUE(A,B,ETEMP,T,NN)
       CALL GVALUE(A,B,DETEMP,DT,NN)
      ELSE
       T  = 0._8
       DT = 0._8
      ENDIF
C     Compute radius from the axis of rotation of a Gauss's point.
      IF(IPLANE.EQ.3) THEN
      CALL GVALUE(A,B,X,R,NN)
      ENDIF
C     Set initial strain vector to zero.
      DO 331 I  = 1, NDIM
      STRAIN(I) = 0._8
  331 CONTINUE
C     Compute strain vector.
      DO 332 I  = 1, NDIM
      DO 332 J  = 1, NDOF
      STRAIN(I) = STRAIN(I) + BMAT(I,J)*EDISP(J)
  332 CONTINUE
C     Set coefficient of thermal expansion vector.
      IF(IPLANE.EQ.1) THEN  !Plane stress case.
      THETA(1) = COTHR
      THETA(2) = COTHR
      THETA(3) = 0._8
      ENDIF
      IF(IPLANE.EQ.2) THEN  !Plane strain case.
      THETA(1) = 1.5_8*COTHR
      THETA(2) = 1.5_8*COTHR
      THETA(3) = 0.0_8
      ENDIF
      IF(IPLANE.EQ.3) THEN  !Axisymmetric case.
      THETA(1) = COTHR
      THETA(2) = COTHR
      THETA(3) = 0._8
      THETA(4) = COTHR
      ENDIF
C==============================================================================
C     Find effective stress.
C==============================================================================
      CALL FINDSTRSS(EFSTRSS,STRAIN,YSTRSS,AHARD,ALPHA,PR,
     *               ELAS,AMOD,EFRATIO,IPLANE,T,COTHR,BETA)
C==============================================================================
C     Plane stress case.
C==============================================================================
C     {STRESSvol} = [EMAT]{ STRAIN-STRAIN0 }
C
C                                                [  1.  1.  0. ]
C     [EMAT] = BETA*ELAS/(1.-2.*PR)/(1.+2.*BETA)*[  1.  1.  0. ]
C                                                [  0.  0.  0. ]
C
C     BETA = 2./3.*(1.-2.*PR)*EFRATIO/ELAS
C
C     EFRATIO = EFSTRSS/EFSTRAN
```

```
C===============================================================================
      IF(IPLANE.EQ.1) THEN
      EMAT(1,1) =  BETA*ELAS/(1._8-2._8*PR)/(1._8+2._8*BETA)
      EMAT(1,2) =  EMAT(1,1)
      EMAT(1,3) =  0._8
      EMAT(2,1) =  EMAT(1,1)
      EMAT(2,2) =  EMAT(1,1)
      EMAT(2,3) =  0._8
      EMAT(3,1) =  0._8
      EMAT(3,2) =  0._8
      EMAT(3,3) =  0._8
C
      DMAT(1,1) =  0.5_8
      DMAT(1,2) =  DMAT(1,1)
      DMAT(1,3) =  0._8
      DMAT(2,1) =  DMAT(1,1)
      DMAT(2,2) =  DMAT(1,1)
      DMAT(2,3) =  0._8
      DMAT(3,1) =  0._8
      DMAT(3,2) =  0._8
      DMAT(3,3) =  0._8
C
      CVEC(1) = ( 1.00_8+BETA+BETA*BETA )*STRAIN(1) +
     *          ( 0.50_8-BETA-BETA*BETA )*STRAIN(2) - 1.5_8*COTHR*T
      CVEC(2) = ( 0.50_8-BETA-BETA*BETA )*STRAIN(1) +
     *          ( 1.00_8+BETA+BETA*BETA )*STRAIN(2) - 1.5_8*COTHR*T
      CVEC(3) = ( 0.25_8+BETA+BETA*BETA )*STRAIN(3)
C
      AVEC(1) = ( STRAIN(1)+STRAIN(2) )*0.5_8 - COTHR*T
      AVEC(2) = AVEC(1)
      AVEC(3) = 0._8
      ENDIF
C===============================================================================
C     Plane strain case.
C===============================================================================
C     {STRESSvol} = [EMAT]{ STRAIN-STRAIN0 }
C
C                                  [  1.  1.  0. ]
C     [EMAT] = ELAS/(3.*(1.-2.*PR))*[  1.  1.  0. ]
C                                  [  0.  0.  0. ]
C===============================================================================
      IF(IPLANE.EQ.2) THEN
      EMAT(1,1) = ELAS/( 3._8*(1._8-2._8*PR) )
      EMAT(1,2) = EMAT(1,1)
      EMAT(1,3) = 0._8
      EMAT(2,1) = EMAT(1,1)
      EMAT(2,2) = EMAT(1,1)
      EMAT(2,3) = 0._8
      EMAT(3,1) = 0._8
      EMAT(3,2) = 0._8
      EMAT(3,3) = 0._8
C
      CVEC(1) = 2.0_8*STRAIN(1) - STRAIN(2)
      CVEC(2) = 2.0_8*STRAIN(2) - STRAIN(1)
      CVEC(3) = 1.5_8*STRAIN(3)
      ENDIF
C===============================================================================
C     Axisymmetric case.
C===============================================================================
C     {STRESSvol} = [EMAT]{ STRAIN-STRAIN0 }
C
C                                  [  1.  1.  0.  1. ]
C     [EMAT] = ELAS/(3.*(1.-2.*PR))*[  1.  1.  0.  1. ]
C                                  [  0.  0.  0.  0. ]
C                                  [  1.  1.  0.  1. ]
C===============================================================================
      IF(IPLANE.EQ.3) THEN
      EMAT(1,1) = ELAS/( 3._8*(1._8-2._8*PR) )
      EMAT(1,2) = EMAT(1,1)
      EMAT(1,3) = 0._8
      EMAT(1,4) = EMAT(1,1)
      EMAT(2,1) = EMAT(1,1)
      EMAT(2,2) = EMAT(1,1)
      EMAT(2,3) = 0._8
      EMAT(2,4) = EMAT(1,1)
      EMAT(3,1) = 0._8
      EMAT(3,2) = 0._8
```

```
      EMAT(3,3) = 0._8
      EMAT(3,4) = 0._8
      EMAT(4,1) = EMAT(1,1)
      EMAT(4,2) = EMAT(1,1)
      EMAT(4,3) = 0._8
      EMAT(4,4) = EMAT(1,1)
C
      CVEC(1) = 2.0_8*STRAIN(1) - STRAIN(2) - STRAIN(4)
      CVEC(2) = 2.0_8*STRAIN(2) - STRAIN(4) - STRAIN(1)
      CVEC(3) = 1.5_8*STRAIN(3)
      CVEC(4) = 2.0_8*STRAIN(4) - STRAIN(1) - STRAIN(2)
      ENDIF
C================================================================================
C     Compute element tangent stiffness matrix.
C================================================================================
      IF(IMOD.NE.1) THEN
      DO 330 I = 1, NDOF
      DO 330 J = 1, NDOF
      DO 330 L = 1, NDIM
      DO 330 M = 1, NDIM
C     Plane stress case.
      IF(IPLANE.EQ.1) THEN
      BELOW1 = STRAIN(1)*STRAIN(1) + 2._8*STRAIN(1)*STRAIN(2) +
     *         STRAIN(2)*STRAIN(2) -
     *         4._8*COTHR*T*( STRAIN(1)+STRAIN(2)-COTHR*T )
      BELOW2 = 1._8 + 27._8*AMOD/ELAS*(1._8-2._8*PR)*BELOW1/
     *                (1._8+2._8*BETA)**3._8
      COTANG = 36._8*AMOD/(1._8+2._8*BETA)**4._8/BELOW2
      ETSMv(I,J) = ETSMv(I,J) + WG(K)*BMAT(L,I)*( EMAT(L,M)+
     *             COTANG*AVEC(L)*CVEC(M) )*BMAT(M,J)*DETJAC*THICK
      ENDIF
C     Plane strain case.
      IF(IPLANE.EQ.2) THEN
      ETSMv(I,J) = ETSMv(I,J) +
     *             WG(K)*BMAT(L,I)*EMAT(L,M)*BMAT(M,J)*DETJAC*THICK
      ENDIF
C     Axisymmetric case.
      IF(IPLANE.EQ.3) THEN
      ETSMv(I,J) = ETSMv(I,J) +
     *             WG(K)*BMAT(L,I)*EMAT(L,M)*BMAT(M,J)*DETJAC*2._8*PI*R
      ENDIF
  330 CONTINUE
      ENDIF
C================================================================================
C     Compute element stiffness matrix.
C================================================================================
      IF(IMOD.NE.2) THEN
      DO 360 I = 1, NDOF
      DO 360 J = 1, NDOF
      DO 360 L = 1, NDIM
      DO 360 M = 1, NDIM
      IF(IPLANE.EQ.3) THEN
      EKMv(I,J) = EKMv(I,J) +
     *            WG(K)*BMAT(L,I)*EMAT(L,M)*BMAT(M,J)*DETJAC*2._8*PI*R
      ELSE
      EKMv(I,J) = EKMv(I,J) +
     *            WG(K)*BMAT(L,I)*EMAT(L,M)*BMAT(M,J)*DETJAC*THICK
      ENDIF
  360 CONTINUE
      ENDIF
C================================================================================
C     Compute element thermal force vector.
C================================================================================
      IF(IRTemp.NE.0) THEN
      IF(IMOD.NE.2) THEN
      DO 335 I = 1, NDOF
      DO 335 L = 1, NDIM
      DO 335 M = 1, NDIM
      IF(IPLANE.EQ.3) THEN
      EFIvolT(I) = EFIvolT(I) + WG(K)*BMAT(L,I)*EMAT(L,M)*THETA(M)*T*
     *                          DETJAC*2._8*PI*R
      ELSE
      EFIvolT(I) = EFIvolT(I) + WG(K)*BMAT(L,I)*EMAT(L,M)*THETA(M)*T*
     *                          DETJAC*THICK
      ENDIF
  335 CONTINUE
C================================================================================
```

```
C     Compute element incremental thermal force vector.
C================================================================================
      ELSE
      DO 365 I = 1, NDOF
      DO 365 L = 1, NDIM
      DO 365 M = 1, NDIM
C     Plane stress case.
      IF(IPLANE.EQ.1) THEN
      BELOW1 = STRAIN(1)*STRAIN(1) + 2._8*STRAIN(1)*STRAIN(2) +
     *         STRAIN(2)*STRAIN(2) -
     *         4._8*COTHR*T*( STRAIN(1)+STRAIN(2)-COTHR*T )
      BELOW2 = 1._8 + 27._8*AMOD/ELAS*(1._8-2._8*PR)*BELOW1/
     *                (1._8+2._8*BETA)**3._8
      COTHER = 54._8*AMOD*( STRAIN(1)+STRAIN(2)-2._8*COTHR*T )*COTHR/
     *         (1._8+2._8*BETA)**4._8/BELOW2
      DEFIvolT(I) = DEFIvolT(I) +
     *              WG(K)*BMAT(L,I)*( EMAT(L,M)*THETA(M)+
     *              DMAT(L,M)*(STRAIN(M)-THETA(M)*T)*COTHER )*DT*
     *              DETJAC*THICK
      ENDIF
C     Plane strain case.
      IF(IPLANE.EQ.2) THEN
      DEFIvolT(I) = DEFIvolT(I) +
     *              WG(K)*BMAT(L,I)*EMAT(L,M)*THETA(M)*DT*DETJAC*THICK
      ENDIF
C     Axisymmetric case.
      IF(IPLANE.EQ.3) THEN
      DEFIvolT(I) = DEFIvolT(I) + WG(K)*BMAT(L,I)*EMAT(L,M)*THETA(M)*
     *                            DT*DETJAC*2._8*PI*R
      ENDIF
  365 CONTINUE
      ENDIF
      ENDIF
C     End each Gauss's point.
  336 CONTINUE
C================================================================================
C     Assemble element matrices and vectors to the system matrices and vectors.
C================================================================================
C     Form system tangent stiffness matrix.
      IF(IMOD.NE.1) THEN
      CALL ASSMBLE(IE,INTMAT9,ETSMv,STSMv,MXPOI,MXELE,NN,NDF)
      ENDIF
C     Form system stiffness matrix.
      IF(IMOD.NE.2) THEN
      CALL ASSMBLE(IE,INTMAT9,EKMv,SKMv,MXPOI,MXELE,NN,NDF)
      ENDIF
C     Form system (incremental) thermal force vector.
      IF(IRTemp.NE.0)THEN
      DO 351 I  = 1, NN
      DO 351 ID = 1, NDF
      INODE = INTMAT9(IE,I)
      IEQ   = (INODE-1)*NDF + ID
      IEE   = (I-1)*NDF +ID
      IF(IMOD.NE.2) THEN   !Thermal force vector.
      FIvolT(IEQ)  = FIvolT(IEQ)  + EFIvolT(IEE)
      ELSE                 !Incremental thermal force vector.
      DFIvolT(IEQ) = DFIvolT(IEQ) + DEFIvolT(IEE)
      ENDIF
  351 CONTINUE
      ENDIF
C     End each element.
 5000 CONTINUE
C
      RETURN
      END
C================================================================================


      SUBROUTINE DEVIATORIC(NPOIN,MXPOI,NELEM,MXELE,INC,MXSTATE,IPLANE,
     *                      NDF,PI,IMOD,INTMAT9,IETIP,COORD,PT,TEMP,
     *                      BDf,STSMd,DFIdevT,DFIBODY,FIdevT,FIBODY,
     *                      PROP,IEMAT,MXMAT,IRTemp,IRBody,SKMd)
C================================================================================
C     THIS SUBROUTINE COMPUTES ALL MATRICES AND VECTORS CORRESPONDING TO
C     DEVIATORIC STRESSES AND BODY FORCEES BY USING HIGHER NUMBER OF GAUSS'S
C     POINTS.
C================================================================================
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COORD(MXPOI,2)
      DIMENSION X(9),Y(9),XG(9),YG(9),WG(9)
      DIMENSION BMAT(4,9*2),DNDA(9),DNDB(9),AJ(2,2),AJI(2,2)
      DIMENSION PT(MXPOI*2),EDISP(9*2)
      DIMENSION GMAT(4,4),PMAT(4,4),HMAT(4,4),DMAT(4,4),CVEC(4),AVEC(4)
      DIMENSION STSMd(MXPOI*2,MXPOI*2),ETSMd(9*2,9*2)
      DIMENSION SKMd(MXPOI*2,MXPOI*2),EKMd(9*2,9*2)
      DIMENSION SKB(MXPOI*2,MXPOI*2),EKB(9*2,9*2),EKBG(9*2,9*2)
      DIMENSION STRAIN(4),THETA(4)
      DIMENSION TEMP(MXPOI,MXSTATE),ETEMP(9),DETEMP(9)
      DIMENSION BDf(MXPOI*2,MXSTATE)
      DIMENSION FIdevT(MXPOI*2),DFIdevT(MXPOI*2)
      DIMENSION FIBODY(MXPOI*2),DFIBODY(MXPOI*2)
      DIMENSION EFIdevT(9*2),DEFIdevT(9*2)
      DIMENSION PROP(MXMAT,7)
      INTEGER INTMAT9(MXELE,9),IETIP(MXELE),IEMAT(MXELE)

C     Number of equations.
      NEQ = NPOIN*NDF
C     Set element matrix dimensions for analyzed problem.
      IF(IPLANE.EQ.3) THEN
      NDIM = 4  !Axisymmetric case.
      ELSE
      NDIM = 3  !Plane stress or plane strain case.
      ENDIF
C     Set consistent analysis state.
C     IMOD = 1 ,compute load vectors.
C          = 2 ,compute tangent stiffness matrix and incremental load vector.
C          = 3 ,compute both load vectors and tangent stiffness matrix.
      IF(IMOD.EQ.2) IPOS = INC      !Incremental part.
      IF(IMOD.NE.2) IPOS = INC + 1  !Iterative part.
C     Set initial system body force stiffness matrix to zero (This matrix used
C     in both incremental and iterative solution part.
      IF(IRBody.NE.0) THEN
      DO 297 I = 1, NEQ
      DO 297 J = 1, NEQ
      SKB(I,J) = 0._8
  297 CONTINUE
      ENDIF
C     Set initial system load vectors to zeros.
      IF(IMOD.EQ.2) THEN
      DO 302  I  = 1, NEQ
      DFIBODY(I) = 0._8   !Incremental body force vector.
  302 CONTINUE
      ELSE
      DO 307 I  = 1, NEQ
      FIBODY(I) = 0._8 !Body force vector.
  307 CONTINUE
      ENDIF
C===============================================================================
C     Loop over all elements.
C===============================================================================
      DO 5000 IE = 1, NELEM
C     Read material properties of each element.
      ELAS   = PROP(IEMAT(IE),1)
      PR     = PROP(IEMAT(IE),2)
      YSTRSS = PROP(IEMAT(IE),3)
      AHARD  = PROP(IEMAT(IE),4)
      ALPHA  = PROP(IEMAT(IE),5)
      COTHR  = PROP(IEMAT(IE),6)
      IF(IPLANE.NE.3) THICK = PROP(IEMAT(IE),7)
C     9-node rectangular crack tip element.
      IF(IETIP(IE).EQ.1) THEN
      NN    =  9         !Number of nodes.
      NG    =  9         !Number of Gauss's points.
      NDOF  =  NDF*NN    !Number of element degrees of freedom.
C     X coordinate at each Gauss's point for this element.
      XG(1) = -DSQRT(3._8)/DSQRT(5._8)
      XG(2) =  XG(1)
      XG(3) =  XG(1)
      XG(4) =  0._8
      XG(5) =  XG(4)
      XG(6) =  XG(4)
      XG(7) = -XG(1)
      XG(8) =  XG(7)
      XG(9) =  XG(7)
```

```
C     Y coordinate at each Gauss's point.
      YG(1) =  XG(1)
      YG(2) =  XG(4)
      YG(3) =  XG(7)
      YG(4) =  XG(1)
      YG(5) =  XG(4)
      YG(6) =  XG(7)
      YG(7) =  XG(1)
      YG(8) =  XG(4)
      YG(9) =  XG(7)
C     Weight at each Gauss's point.
      WG(1) =  25._8/81._8
      WG(2) =  40._8/81._8
      WG(3) =  WG(1)
      WG(4) =  WG(2)
      WG(5) =  64._8/81._8
      WG(6) =  WG(2)
      WG(7) =  WG(1)
      WG(8) =  WG(2)
      WG(9) =  WG(1)
      ELSE
C     6-node triangular element.
      NN    =  6         !Number of nodes.
      NG    =  7         !Number of Gauss's points.
      NDOF  =  NDF*NN    !Number of element degrees of freedom.
C     X coordinate at each Gauss's point for this element.
      XG(1) =  1._8/3._8
      XG(2) =  0.101286507323456_8
      XG(3) =  0.797426985353087_8
      XG(4) =  XG(2)
      XG(5) =  0.470142064105115_8
      XG(6) =  0.059715871789770_8
      XG(7) =  XG(5)
C     Y coordinate at each Gauss's point for this element.
      YG(1) =  XG(1)
      YG(2) =  XG(2)
      YG(3) =  XG(2)
      YG(4) =  XG(3)
      YG(5) =  XG(5)
      YG(6) =  XG(5)
      YG(7) =  XG(6)
C     Weight at each Gauss's point.
      WG(1) =  0.225_8
      WG(2) =  0.125939180544827_8
      WG(3) =  WG(2)
      WG(4) =  WG(2)
      WG(5) =  0.132394152788506_8
      WG(6) =  WG(5)
      WG(7) =  WG(5)
C     Area of triangle must be devided by two.
      WG(1) =  WG(1)/2._8
      WG(2) =  WG(2)/2._8
      WG(3) =  WG(3)/2._8
      WG(4) =  WG(4)/2._8
      WG(5) =  WG(5)/2._8
      WG(6) =  WG(6)/2._8
      WG(7) =  WG(7)/2._8
      ENDIF
C     Create elemant nodal displacement and temperature vectors.
      DO 100 I = 1, NN
      II          = INTMAT9(IE,I)
      X(I)        = COORD(II,1)
      Y(I)        = COORD(II,2)
      ETEMP(I)    = TEMP(II,IPOS)
      DETEMP(I)   = TEMP(II,INC+1) - TEMP(II,INC)
      EDISP(2*I-1) = PT(2*II-1)
      EDISP(2*I)   = PT(2*II)
  100 CONTINUE
C     Set initial element tangent stiffness matrix to zero.
      IF(IMOD.NE.1) THEN
      DO 300  I  = 1, NDOF
      DO 300  J  = 1, NDOF
      ETSMd(I,J) = 0._8
  300 CONTINUE
      ENDIF
C     Set initial element stiffness matrix to zero.
      IF(IMOD.NE.2) THEN
```

```
      DO 301 I  = 1, NDOF
      DO 301 J  = 1, NDOF
      EKMd(I,J) = 0._8
  301 CONTINUE
      ENDIF
C     Set initial element body force stiffness matrix to zero.
      IF(IRBody.NE.0) THEN
      DO 305 I = 1, NDOF
      DO 305 J = 1, NDOF
      EKB(I,J) = 0._8
  305 CONTINUE
      ENDIF
C     Set initial element thermal load vector to zero.
      IF(IMOD.NE.2 .AND. IRTemp.NE.0) THEN
      DO 303 I = 1, NDOF
      EFIdevT(I)  = 0._8
  303 CONTINUE
      ENDIF
C     Set initial element incremental thermal load vector to zero.
      IF(IMOD.EQ.2 .AND. IRTemp.NE.0) THEN
      DO 304 I = 1, NDOF
      DEFIdevT(I) = 0._8
  304 CONTINUE
      ENDIF
C================================================================================
C     Loop over each Gauss's point on an element.
C================================================================================
      DO 336 K = 1, NG
C     Compute strain-displacement matrix.
      A = XG(K)
      B = YG(K)
      CALL BJ9(X,Y,A,B,BMAT,EKBG,AJ,AJI,DETJAC,DNDA,DNDB,
     *         NN,NDF,IPLANE)
C     Compute temperature and temperature increment at a Gauss's point.
      IF(IRTemp.NE.0) THEN
       CALL GVALUE(A,B,ETEMP,T,NN)
       CALL GVALUE(A,B,DETEMP,DT,NN)
      ELSE
       T  = 0._8
       DT = 0._8
      ENDIF
C     Compute radius from the axis of rotation of a Gauss's point.
      IF(IPLANE.EQ.3) THEN
      CALL GVALUE(A,B,X,R,NN)
      ENDIF
C     Set initial strain vector to zero.
      DO 331 I  = 1, NDIM
      STRAIN(I) = 0._8
  331 CONTINUE
C     Compute strain vector.
      DO 332 I  = 1, NDIM
      DO 332 J  = 1, NDOF
      STRAIN(I) = STRAIN(I) + BMAT(I,J)*EDISP(J)
  332 CONTINUE
C     Set coefficient of thermal expansion vector.
      IF(IPLANE.EQ.1) THEN  !Plane stress case.
      THETA(1) = COTHR
      THETA(2) = COTHR
      THETA(3) = 0._8
      ENDIF
C================================================================================
C     Find effective stress.
C================================================================================
      CALL FINDSTRSS(EFSTRSS,STRAIN,YSTRSS,AHARD,ALPHA,PR,
     *               ELAS,AMOD,EFRATIO,IPLANE,T,COTHR,BETA)
C================================================================================
C     Plane stress case.
C================================================================================
C     {STRESSdev} = [GMAT]{STRAIN} - [HMAT]{STRAIN0}
C
C                                                [ 1.+BETA    -BETA  0.       ]
C     [GMAT] = BETA*ELAS/(1.-2.*PR)/(1.+2.*BETA)*[   -BETA  1.+BETA  0.       ]
C                                                [ 0.       0.       0.5+BETA ]
C
C                                                [ 1.       0.       0.       ]
C     [HMAT] = BETA*ELAS/(1.-2.*PR)/(1.+2.*BETA)*[ 0.       1.       0.       ]
C                                                [ 0.       0.       0.       ]
```

```
C
C     BETA = 2./3.*(1.-2.*PR)*EFRATIO/ELAS
C
C     EFRATIO = EFSTRSS/EFSTRAN
C=========================================================================
      IF(IPLANE.EQ.1) THEN
      COND = BETA*ELAS/(1._8-2._8*PR)/(1._8+2._8*BETA)
      GMAT(1,1) =  COND*(1._8+BETA)
      GMAT(1,2) = -COND*BETA
      GMAT(1,3) =  0._8
      GMAT(2,1) =  GMAT(1,2)
      GMAT(2,2) =  GMAT(1,1)
      GMAT(2,3) =  0._8
      GMAT(3,1) =  0._8
      GMAT(3,2) =  0._8
      GMAT(3,3) =  COND*(0.5_8+BETA)
C
      PMAT(1,1) =  0.50_8+BETA+BETA*BETA
      PMAT(1,2) =        -BETA-BETA*BETA
      PMAT(1,3) =  0._8
      PMAT(2,1) =  PMAT(1,2)
      PMAT(2,2) =  PMAT(1,1)
      PMAT(2,3) =  0._8
      PMAT(3,1) =  0._8
      PMAT(3,2) =  0._8
      PMAT(3,3) =  0.25_8+BETA+BETA*BETA
C
      HMAT(1,1) =  COND
      HMAT(1,2) =  0._8
      HMAT(1,3) =  0._8
      HMAT(2,1) =  0._8
      HMAT(2,2) =  HMAT(1,1)
      HMAT(2,3) =  0._8
      HMAT(3,1) =  0._8
      HMAT(3,2) =  0._8
      HMAT(3,3) =  0._8
C
      DMAT(1,1) =  0.5_8
      DMAT(1,2) =  0.0_8
      DMAT(1,3) =  0.0_8
      DMAT(2,1) =  0.0_8
      DMAT(2,2) =  DMAT(1,1)
      DMAT(2,3) =  0.0_8
      DMAT(3,1) =  0.0_8
      DMAT(3,2) =  0.0_8
      DMAT(3,3) =  0.0_8
C
      CVEC(1) = ( 1.00_8+BETA+BETA*BETA )*STRAIN(1) +
     *          ( 0.50_8-BETA-BETA*BETA )*STRAIN(2) - 1.5_8*COTHR*T
      CVEC(2) = ( 0.50_8-BETA-BETA*BETA )*STRAIN(1) +
     *          ( 1.00_8+BETA+BETA*BETA )*STRAIN(2) - 1.5_8*COTHR*T
      CVEC(3) = ( 0.25_8+BETA+BETA*BETA )*STRAIN(3)
C
      AVEC(1) = ( 0.50_8+BETA+BETA*BETA )*STRAIN(1) +
     *          (      -BETA-BETA*BETA )*STRAIN(2)
      AVEC(2) = (      -BETA-BETA*BETA )*STRAIN(1) +
     *          ( 0.50_8+BETA+BETA*BETA )*STRAIN(2)
      AVEC(3) = ( 0.25_8+BETA+BETA*BETA )*STRAIN(3)
      ENDIF
C=========================================================================
C     Plane strain case.
C=========================================================================
C     {STRESSdev} = [GMAT]{STRAIN}
C
C                           [  2.  -1.   0. ]
C     [GMAT] = 2./9.*EFRATIO*[ -1.   2.   0. ]
C                           [  0.   0.   1.5]
C=========================================================================
      IF(IPLANE.EQ.2) THEN
      GMAT(1,1) =  2._8/9._8*EFRATIO*2._8
      GMAT(1,2) = -2._8/9._8*EFRATIO
      GMAT(1,3) =  0._8
      GMAT(2,1) =  GMAT(1,2)
      GMAT(2,2) =  GMAT(1,1)
      GMAT(2,3) =  0._8
      GMAT(3,1) =  0._8
      GMAT(3,2) =  0._8
```

```
      GMAT(3,3) =  2._8/9._8*EFRATIO*1.5_8
C
      CVEC(1) = 2.0_8*STRAIN(1)-STRAIN(2)
      CVEC(2) = 2.0_8*STRAIN(2)-STRAIN(1)
      CVEC(3) = 1.5_8*STRAIN(3)
      ENDIF
C================================================================================
C     Axisymmetric case.
C================================================================================
C     {STRESSdev} = [GMAT]{STRAIN}
C
C                             [  2. -1.  0.  -1. ]
C     [GMAT] = 2./9.*EFRATIO*[ -1.  2.  0.  -1. ]
C                             [  0.  0.  1.5  0. ]
C                             [ -1. -1.  0.   2. ]
C================================================================================
      IF(IPLANE.EQ.3) THEN
      GMAT(1,1) =  2._8/9._8*EFRATIO*2._8
      GMAT(1,2) = -2._8/9._8*EFRATIO
      GMAT(1,3) =  0._8
      GMAT(1,4) =  GMAT(1,2)
      GMAT(2,1) =  GMAT(1,2)
      GMAT(2,2) =  GMAT(1,1)
      GMAT(2,3) =  0._8
      GMAT(2,4) =  GMAT(1,2)
      GMAT(3,1) =  GMAT(1,3)
      GMAT(3,2) =  GMAT(2,3)
      GMAT(3,3) =  2._8/9._8*EFRATIO*1.5_8
      GMAT(3,4) =  0._8
      GMAT(4,1) =  GMAT(1,4)
      GMAT(4,2) =  GMAT(2,4)
      GMAT(4,3) =  GMAT(3,4)
      GMAT(4,4) =  GMAT(1,1)
C
      CVEC(1) = 2.0_8*STRAIN(1)-STRAIN(2)-STRAIN(4)
      CVEC(2) = 2.0_8*STRAIN(2)-STRAIN(4)-STRAIN(1)
      CVEC(3) = 1.5_8*STRAIN(3)
      CVEC(4) = 2.0_8*STRAIN(4)-STRAIN(1)-STRAIN(2)
      ENDIF
C================================================================================
C     Compute element body force stiffness matrix.
C================================================================================
      IF(IRBody.NE.0) THEN
      DO 330 I = 1, NDOF
      DO 330 J = 1, NDOF
      IF(IPLANE.EQ.3) THEN
      EKB(I,J) = EKB(I,J) + WG(K)*EKBG(I,J)*DETJAC*2._8*PI*R
      ELSE
      EKB(I,J) = EKB(I,J) + WG(K)*EKBG(I,J)*DETJAC*THICK
      ENDIF
  330 CONTINUE
      ENDIF
C================================================================================
C     Compute element tangent stiffness matrix.
C================================================================================
      IF(IMOD.NE.1) THEN
      DO 350 I = 1, NDOF
      DO 350 J = 1, NDOF
      DO 350 L = 1, NDIM
      DO 350 M = 1, NDIM
C     Plane stress case.
      IF(IPLANE.EQ.1) THEN
      BELOW1 = STRAIN(1)*STRAIN(1) + 2._8*STRAIN(1)*STRAIN(2) +
     *         STRAIN(2)*STRAIN(2) -
     *         4._8*COTHR*T*( STRAIN(1)+STRAIN(2)-COTHR*T )
      BELOW2 = 1._8 + 27._8*AMOD/ELAS*(1._8-2._8*PR)*BELOW1/
     *                (1._8+2._8*BETA)**3._8
      COTANG = 36._8*AMOD/(1._8+2._8*BETA)**4._8/BELOW2
      ETSMd(I,J) = ETSMd(I,J) + WG(K)*BMAT(L,I)*( GMAT(L,M)+
     *             COTANG*(AVEC(L)-THETA(L)*T/2._8)*CVEC(M) )*
     *             BMAT(M,J)*DETJAC*THICK
      ENDIF
C     Plane strain case.
      IF(IPLANE.EQ.2) THEN
      ETSMd(I,J) = ETSMd(I,J) + WG(K)*BMAT(L,I)*( GMAT(L,M)+AMOD*
     *             CVEC(L)*CVEC(M) )*BMAT(M,J)*DETJAC*THICK
      ENDIF
```

```
C     Axisymmetric case.
      IF(IPLANE.EQ.3) THEN
      ETSMd(I,J) = ETSMd(I,J) + WG(K)*BMAT(L,I)*( GMAT(L,M)+AMOD*
     *             CVEC(L)*CVEC(M) )*BMAT(M,J)*DETJAC*2._8*PI*R
      ENDIF
  350 CONTINUE
      ENDIF
C================================================================================
C     Compute element stiffness matrix.
C================================================================================
      IF(IMOD.NE.2) THEN
      DO 355 I = 1, NDOF
      DO 355 J = 1, NDOF
      DO 355 L = 1, NDIM
      DO 355 M = 1, NDIM
      IF(IPLANE.EQ.3)THEN
      EKMd(I,J) = EKMd(I,J) +
     *            WG(K)*BMAT(L,I)*GMAT(L,M)*BMAT(M,J)*DETJAC*2._8*PI*R
      ELSE
      EKMd(I,J) = EKMd(I,J) +
     *            WG(K)*BMAT(L,I)*GMAT(L,M)*BMAT(M,J)*DETJAC*THICK
      ENDIF
  355 CONTINUE
      ENDIF
C================================================================================
C     Compute element thermal force vector.
C================================================================================
      IF(IMOD.NE.2 .AND. IPLANE.EQ.1 .AND. IRTemp.NE.0) THEN
      DO 335 I = 1, NDOF
      DO 335 L = 1, NDIM
      DO 335 M = 1, NDIM
      EFIdevT(I) = EFIdevT(I) + WG(K)*BMAT(L,I)*HMAT(L,M)*THETA(M)*T*
     *                          DETJAC*THICK
  335 CONTINUE
      ENDIF
C================================================================================
C     Compute element incremental thermal force vector.
C================================================================================
      IF(IMOD.EQ.2 .AND. IPLANE.EQ.1 .AND. IRTemp.NE.0) THEN
      DO 340 I = 1, NDOF
      DO 340 L = 1, NDIM
      DO 340 M = 1, NDIM
      BELOW1 = STRAIN(1)*STRAIN(1) + 2._8*STRAIN(1)*STRAIN(2) +
     *         STRAIN(2)*STRAIN(2) -
     *         4._8*COTHR*T*( STRAIN(1)+STRAIN(2)-COTHR*T )
      BELOW2 = 1._8 + 27._8*AMOD/ELAS*(1._8-2._8*PR)*BELOW1/
     *                (1._8+2._8*BETA)**3._8
      COTHER = 54._8*AMOD*( STRAIN(1)+STRAIN(2)-2._8*COTHR*T )*COTHR/
     *         (1._8+2._8*BETA)**4._8/BELOW2
      DEFIdevT(I) = DEFIdevT(I) +
     *              WG(K)*BMAT(L,I)*( HMAT(L,M)*THETA(M)+
     *              (PMAT(L,M)*STRAIN(M)-DMAT(L,M)*THETA(M)*T)*COTHER )*
     *              DT*DETJAC*THICK
  340 CONTINUE
      ENDIF
C     End each Gauss's point.
  336 CONTINUE
C================================================================================
C     Assemble element matrices and vectors to the system matrices and vectors.
C================================================================================
C     Form system tangent stiffness matrix.
      IF(IMOD.NE.1) THEN
      CALL ASSMBLE(IE,INTMAT9,ETSMd,STSMd,MXPOI,MXELE,NN,NDF)
      ENDIF
C     Form system stiffness matrix.
      IF(IMOD.NE.2) THEN
      CALL ASSMBLE(IE,INTMAT9,EKMd,SKMd,MXPOI,MXELE,NN,NDF)
      ENDIF
C     Form system body force stiffness matrix.
      IF(IRBody.NE.0) THEN
      CALL ASSMBLE(IE,INTMAT9,EKB,SKB,MXPOI,MXELE,NN,NDF)
      ENDIF
C     Form system (incremental) thermal force vector.
      IF(IPLANE.EQ.1 .AND. IRTemp.NE.0) THEN
      DO 351 IN = 1, NN
      DO 351 ID = 1, NDF
      INODE = INTMAT9(IE,IN)
```

```
      IEQ   = (INODE-1)*NDF + ID
      IEE   = (IN-1)*NDF +ID
      IF(IMOD.NE.2) THEN   !Thermal force vector.
      FIdevT(IEQ)  = FIdevT(IEQ)  + EFIdevT(IEE)
      ELSE                 !Incremental thermal force vector.
      DFIdevT(IEQ) = DFIdevT(IEQ) + DEFIdevT(IEE)
      ENDIF
  351 CONTINUE
      ENDIF
C     End each element.
 5000 CONTINUE
C================================================================================
C     Compute system incremental body force vector and body force vector.
C================================================================================
      IF(IMOD.EQ.2 .AND. IRBody.NE.0) THEN
       DO 380  I  = 1, NEQ
       DO 380  J  = 1, NEQ
       DFIBODY(I) = DFIBODY(I) + SKB(I,J)*(BDf(J,INC+1)-BDf(J,INC))
  380  CONTINUE
      ENDIF
C
      IF(IMOD.NE.2 .AND. IRBody.NE.0) THEN
       DO 361 I  = 1, NEQ
       DO 361 J  = 1, NEQ
       FIBODY(I) = FIBODY(I) + SKB(I,J)*BDf(J,INC+1)
  361  CONTINUE
      ENDIF
C
      RETURN
      END
C================================================================================


      SUBROUTINE BJ9(X,Y,A,B,BMAT,EKBG,AJ,AJI,DETJAC,DNDA,DNDB,
     *               NN,NDF,IPLANE)
C================================================================================
C     THIS SUBROUTINE COMPUTES STRAIN-DISPLACEMENT,GAUSS'S POINT ELEMENT BODY
C     FORCE STIFFNESS, JACOBIAN, INVERSE JACOBIAN'S MATRIX AND ITS DETERMINANT
C     ,AND DERIVATIVES OF SHAPE FUNCTIONS.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BMAT(4,9*2),X(9),Y(9)
      DIMENSION DNDA(9),DNDB(9),AJ(2,2),AJI(2,2)
      DIMENSION AN(9),ANBAR(2,9*2),EKBG(9*2,9*2)

      NDOF = NN*NDF
C     Set matrix dimensions for this problem.
      IF(IPLANE.EQ.3) THEN  !Axisymmetric case.
      NDIM = 4
      ELSE                  !Plane stress or plane strain case.
      NDIM = 3
      ENDIF
C     Find shape function values at a Gauss's point.
      SELECT CASE(NN)
C     6-node triangular element.
      CASE(6)
C     Shape functions at a Gauss's point.
      AN(1)   =  (1._8-(A+B))*(1._8-2._8*(A+B))
      AN(2)   =  A*(2._8*A-1._8)
      AN(3)   =  B*(2._8*B-1._8)
      AN(4)   =  4._8*A*(1._8-(A+B))
      AN(5)   =  4._8*A*B
      AN(6)   =  4._8*B*(1._8-(A+B))
C     Derivative of shape functions w.r.t. an elemental coordinate.
      DNDA(1) =  4._8*A - 3._8   + 4._8*B
      DNDA(2) =  4._8*A - 1._8
      DNDA(3) =  0._8
      DNDA(4) =  4._8   - 8._8*A - 4._8*B
      DNDA(5) =  4._8*B
      DNDA(6) = -4._8*B
C     Derivative of shape functions w.r.t. an elemental coordinate.
      DNDB(1) =  4._8*B - 3._8   + 4._8*A
      DNDB(2) =  0._8
      DNDB(3) =  4._8*B - 1._8
      DNDB(4) = -4._8*A
      DNDB(5) =  4._8*A
      DNDB(6) =  4._8   - 4._8*A - 8._8*B
```

```
C     9-node quadrilateral crack tip element.
      CASE(9)
C     Shape functions at a Gauss's point.
      AN(1)   =  A*(1._8-A)*B*(1._8-B)/4._8
      AN(2)   = -A*(1._8+A)*B*(1._8-B)/4._8
      AN(3)   =  A*(1._8+A)*B*(1._8+B)/4._8
      AN(4)   = -A*(1._8-A)*B*(1._8+B)/4._8
      AN(5)   = -(1._8-A*A)*B*(1._8-B)/2._8
      AN(6)   =  A*(1._8+A)*(1._8-B*B)/2._8
      AN(7)   =  (1._8-A*A)*B*(1._8+B)/2._8
      AN(8)   = -A*(1._8-A)*(1._8-B*B)/2._8
      AN(9)   =  (1._8-A*A)*(1._8-B*B)
C     Derivative of shape functions w.r.t. an elemental coordinate.
      DNDA(1) =  B*(1._8-B)*(1._8-2._8*A)/4._8
      DNDA(2) = -B*(1._8-B)*(1._8+2._8*A)/4._8
      DNDA(3) =  B*(1._8+B)*(1._8+2._8*A)/4._8
      DNDA(4) = -B*(1._8+B)*(1._8-2._8*A)/4._8
      DNDA(5) =  B*(1._8-B)*A
      DNDA(6) =  (1._8-B*B)*(1._8+2._8*A)/2._8
      DNDA(7) = -B*(1._8+B)*A
      DNDA(8) = -(1._8-B*B)*(1._8-2._8*A)/2._8
      DNDA(9) = -2._8*A*(1._8-B*B)
C     Derivative of shape functions w.r.t. an elemental coordinate.
      DNDB(1) =  A*(1._8-A)*(1._8-2._8*B)/4._8
      DNDB(2) = -A*(1._8+A)*(1._8-2._8*B)/4._8
      DNDB(3) =  A*(1._8+A)*(1._8+2._8*B)/4._8
      DNDB(4) = -A*(1._8-A)*(1._8+2._8*B)/4._8
      DNDB(5) = -(1._8-A*A)*(1._8-2._8*B)/2._8
      DNDB(6) = -A*(1._8+A)*B
      DNDB(7) =  (1._8-A*A)*(1._8+2._8*B)/2._8
      DNDB(8) =  A*(1._8-A)*B
      DNDB(9) = -2._8*B*(1._8-A*A)
C     Wrong number of nodes in an element.
      CASE DEFAULT
      WRITE(*,*) 'WRONG NUMBER OF NODES IN THE ELEMENT'
      STOP
      END SELECT
C     Compute Gauss's point radius from axis of rotation.
      IF(IPLANE.EQ.3) THEN
      R = 0._8
      DO 80 I = 1, NN
      R = R + AN(I)*X(I)
   80 CONTINUE
      ENDIF
C     Set shape function matrix to zero.
      DO 90 I = 1, NDF
      DO 90 J = 1, NDOF
      ANBAR(I,J) = 0._8
   90 CONTINUE
C     Compute [ANBAR].
      DO 100 I = 1, NN
      ANBAR(1,2*I-1) = AN(I)
      ANBAR(2,2*I)   = AN(I)
  100 CONTINUE
C     Set Gauss's point element body force stiffness matrix to zero.
      DO 110 I  = 1, NDOF
      DO 110 J  = 1, NDOF
      EKBG(I,J) = 0._8
  110 CONTINUE
C     [EKBG] = tran[ANBAR][ANBAR].
      DO 120 I  = 1, NDOF
      DO 120 J  = 1, NDOF
      DO 120 L  = 1, NDF
      EKBG(I,J) = EKBG(I,J) + ANBAR(L,I)*ANBAR(L,J)
  120 CONTINUE
C     Set Jacobian's matrix to zero.
      AJ(1,1) = 0._8
      AJ(1,2) = 0._8
      AJ(2,1) = 0._8
      AJ(2,2) = 0._8
C     Compute Jacobian's matrix.
      DO 5 IN = 1, NN
      AJ(1,1) = AJ(1,1) + DNDA(IN)*X(IN)
      AJ(1,2) = AJ(1,2) + DNDA(IN)*Y(IN)
      AJ(2,1) = AJ(2,1) + DNDB(IN)*X(IN)
      AJ(2,2) = AJ(2,2) + DNDB(IN)*Y(IN)
    5 CONTINUE
```

```
C     Compute determinant of Jacobian's matrix.
      DETJAC  = AJ(1,1)*AJ(2,2) - AJ(2,1)*AJ(1,2)
C     Compute inverse Jacobian's matrix.
      AJI(1,1) =  AJ(2,2)/DETJAC
      AJI(1,2) = -AJ(1,2)/DETJAC
      AJI(2,1) = -AJ(2,1)/DETJAC
      AJI(2,2) =  AJ(1,1)/DETJAC
C     Set strain-displacement matrix to zero.
      DO 10 I = 1, NDIM
      DO 10 J = 1, NDOF
      BMAT(I,J) = 0._8
   10 CONTINUE
C     Compute strain-displacement matrix.
      DO 50 J = 1, NN
      BMAT(1,2*J-1) = AJI(1,1)*DNDA(J) + AJI(1,2)*DNDB(J)
      BMAT(2,2*J  ) = AJI(2,1)*DNDA(J) + AJI(2,2)*DNDB(J)
      BMAT(3,2*J-1) = BMAT(2,2*J  )
      BMAT(3,2*J  ) = BMAT(1,2*J-1)
   50 CONTINUE
      IF(IPLANE.EQ.3) THEN  !For axisymmetric case.
      DO 60 J = 1, NN
      BMAT(4,2*J-1) = AN(J)/R
   60 CONTINUE
      ENDIF
C
      RETURN
      END
C================================================================================


      SUBROUTINE GVALUE(A,B,ANVAL,GVAL,NN)
C================================================================================
C     THIS SUBROUTINE COMPUTES QUANTITY AT ANY POINT IN THE ELEMENT.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AN(9),ANVAL(9)
C     Choose which element to interpolate.
      SELECT CASE(NN)
C     6-node triangular element.
      CASE(6)
      AN(1) = (1._8-(A+B))*(1._8-2._8*(A+B))
      AN(2) = A*(2._8*A-1._8)
      AN(3) = B*(2._8*B-1._8)
      AN(4) = 4._8*A*(1._8-(A+B))
      AN(5) = 4._8*A*B
      AN(6) = 4._8*B*(1._8-(A+B))
C     9-node rectangular crack tip element.
      CASE(9)
      AN(1) =  A*(1._8-A)*B*(1._8-B)/4._8
      AN(2) = -A*(1._8+A)*B*(1._8-B)/4._8
      AN(3) =  A*(1._8+A)*B*(1._8+B)/4._8
      AN(4) = -A*(1._8-A)*B*(1._8+B)/4._8
      AN(5) = -(1._8-A*A)*B*(1._8-B)/2._8
      AN(6) =  A*(1._8+A)*(1._8-B*B)/2._8
      AN(7) =  (1._8-A*A)*B*(1._8+B)/2._8
      AN(8) = -A*(1._8-A)*(1._8-B*B)/2._8
      AN(9) =  (1._8-A*A)*(1._8-B*B)
C     Wrong number of nodes in an element.
      CASE DEFAULT
      WRITE(*,*) 'WRONG NUMBER OF NODES IN THE ELEMENT'
      STOP
      END SELECT
C     Compute Gauss's point value.
      GVAL = 0._8
      DO 10 IN = 1, NN
      GVAL = GVAL + AN(IN)*ANVAL(IN)
   10 CONTINUE
C
      RETURN
      END
C================================================================================


      SUBROUTINE FINDSTRSS(EFSTRSS,STRAIN,YSTRSS,AHARD,ALPHA,PR,
     *                     ELAS,AMOD,EFRATIO,IPLANE,T,COTHR,BETA)
C================================================================================
C     THIS SUBROUTINE COMPUTES EFFECTIVE STRESS, STRAIN AND THEIR RATIO AT A
```

```
C     GAUESS'S POINT.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION STRAIN(4)

C     Set tolerance number and maximum number of iterations for computing
C     effective stress.
      TOR = 1.E-12_8
      MXITER = 150
C     Compute the effective strain.
      IF(IPLANE.EQ.2) THEN  !Plane strain case.
      BEsqrt  = ( 2._8*STRAIN(1)-STRAIN(2)-   0._8   )**2._8 +
     *          ( 2._8*STRAIN(2)-   0._8  -STRAIN(1) )**2._8 +
     *          (        0._8   -STRAIN(1)-STRAIN(2) )**2._8
      BEsqrt  =  BEsqrt/9._8 + STRAIN(3)*STRAIN(3)/2._8
      BEsqrt  =  BEsqrt*2._8/3._8
      EFSTRAN =  DSQRT(BEsqrt)
      ENDIF
      IF(IPLANE.EQ.3) THEN  !Axisymmetric case.
      BEsqrt  = ( 2._8*STRAIN(1)-STRAIN(2)-STRAIN(4) )**2._8 +
     *          ( 2._8*STRAIN(2)-STRAIN(4)-STRAIN(1) )**2._8 +
     *          ( 2._8*STRAIN(4)-STRAIN(1)-STRAIN(2) )**2._8
      BEsqrt  =  BEsqrt/9._8 + STRAIN(3)*STRAIN(3)/2._8
      BEsqrt  =  BEsqrt*2._8/3._8
      EFSTRAN =  DSQRT(BEsqrt)
      ENDIF
C     Compute effective strain per yield strain.
      EFSTRAN0 = EFSTRAN*ELAS/YSTRSS
C================================================================================
C     Use Newton-Raphson iteration scheme to find the effective stress.
C================================================================================
      IF(IPLANE.NE.1) THEN  !Plane strain or axisymmetric case.
       IF(EFSTRAN0.GT.(ALPHA+2._8/3._8*(1._8+PR))) THEN    !Plastic state.
       EFSTRSS0 = (EFSTRAN0/ALPHA)**(1._8/AHARD)
       ELSE    !Elastic state.
       EFSTRSS0 = 3._8/(2._8*(1._8+PR))*EFSTRAN0
       ENDIF
      ELSE                  !Plane stress case.
       EFSTRSS0 = 0._8
      ENDIF
C================================================================================
C     Begin iterative loop.
C================================================================================
      DO 150 I = 1, MXITER
      IF(IPLANE.EQ.1) THEN  !Plane stress case.
      B = ALPHA*EFSTRSS0**(AHARD-1._8)+2._8/3._8*(1._8+PR)
      EFRATIO = ELAS/B
      BETA = 2._8/3._8*(1._8-2._8*PR)/ELAS*EFRATIO
      EPS33 = (3._8*COTHR*T+(BETA-1._8)*(STRAIN(1)+STRAIN(2)))/
     *        (1._8+2._8*BETA)
      BEsqrt  = ( 2._8*STRAIN(1)-STRAIN(2)-EPS33     )**2._8 +
     *          ( 2._8*STRAIN(2)-EPS33    -STRAIN(1) )**2._8 +
     *          ( 2._8*EPS33    -STRAIN(1)-STRAIN(2) )**2._8
      BEsqrt  =  BEsqrt/9._8 + STRAIN(3)*STRAIN(3)/2._8
      BEsqrt  =  BEsqrt*2._8/3._8
      EFSTRAN =  DSQRT(BEsqrt)
C     Compute effective strain per yield strain.
      EFSTRAN0 = EFSTRAN/YSTRSS*ELAS
      ENDIF
C     Compute increment of effective stress per yield stress.
      DEFSTRSS0 = -(ALPHA*EFSTRSS0**AHARD+2._8/3._8*(1._8+PR)*EFSTRSS0
     *               -EFSTRAN0)/(ALPHA*AHARD*EFSTRSS0**(AHARD-1._8)+
     *                            2._8/3._8*(1._8+PR))
C     Compute new effective stress per yield stress predictor.
      EFSTRSS0  = EFSTRSS0 + DEFSTRSS0
C     Check solution convergence.
      IF(DABS(DEFSTRSS0).LE.TOR) GOTO 160
C     End each iteration.
  150 CONTINUE
      WRITE(*,155)
  155 FORMAT(/,' !!! CANNOT FIND EFFECTIVE STRESS WITHIN A SPECIFIED',
     *       /,'     TOLERANCE LIMIT AND NUMBER OF ITERATIONS')
C     Compute effective stress.
  160 EFSTRSS = EFSTRSS0*YSTRSS
C     Check negative effective stress.
      IF(EFSTRSS.LT.0._8) WRITE(*,170)
  170 FORMAT(/,' *** THE EFFECTIVE STRESS IS NEGATIVE *** ')
```

```
C     Compute necessary variables for LST subroutine.
      B = ALPHA*EFSTRSS0**(AHARD-1._8)+2._8/3._8*(1._8+PR)
      EFRATIO = ELAS/B
      BETA = 2._8/3._8*(1._8-2._8*PR)/ELAS*EFRATIO
      IF(EFSTRSS0.EQ.0._8 .AND. AHARD.LT.3._8) THEN
      AMOD = 0._8
      ELSE
      AMOD = -4._8/81._8*ALPHA*(AHARD-1._8)/(YSTRSS**2._8)*ELAS**3._8
     *        *EFSTRSS0**(AHARD-3._8)/(B**3._8*(AHARD*B-2._8/3._8
     *        *(1._8+PR)*(AHARD-1._8)))
      ENDIF
C
      RETURN
      END
C===============================================================================


      SUBROUTINE ASSMBLE(IE,INTMAT9,ELEMAT,SYSMAT,MXPOI,MXELE,NN,NDF)
C===============================================================================
C     THIS SUBROUTINE ASSEMBLES ELEMENT MATRICES INTO A SYSTEM MATRIX.
C===============================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SYSMAT(MXPOI*2,MXPOI*2),ELEMAT(9*2,9*2)
      INTEGER INTMAT9(MXELE,9)
C
      DO 100 INR  = 1, NN
      INODER = INTMAT9(IE,INR)
      DO 100 IDFR = 1, NDF
C     ISR = row position in a system matrix.
C     IER = row position in an element matrix.
      ISR = (INODER-1)*NDF + IDFR
      IER = (INR   -1)*NDF + IDFR
      DO 200 INC  = 1, NN
      INODEC = INTMAT9(IE,INC)
      DO 200 IDFC = 1, NDF
C     ISC = collumn position in a system matrix.
C     IEC = collumn position in an element matrix.
      ISC = (INODEC-1)*NDF + IDFC
      IEC = (INC   -1)*NDF + IDFC
      SYSMAT(ISR,ISC) = SYSMAT(ISR,ISC) + ELEMAT(IER,IEC)
  200 CONTINUE
  100 CONTINUE
C
      RETURN
      END
C===============================================================================


      SUBROUTINE APPLYBC(NEQ,IBC,STSM,QFINC,MXPOI)
C===============================================================================
C     APPLY DISPLACEMENT BOUNDARY CONDITIONS WITH CONDITION CODES OF
C        0 = free to move (force is known).
C       +1 = fixed (displacement is zero).
C       -1 = prescribed displacement to be increment
C            (displacement is non-zero).
C===============================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION STSM(MXPOI*2,MXPOI*2),QFINC(MXPOI*2)
      INTEGER IBC(MXPOI*2)
C
      DO 10 I = 1, NEQ
      II = IBC(I)
      IF(II.NE.0) STSM(I,I) = 1._8
      IF(I.EQ.NEQ) GOTO 10
      DO 20 J = I+1, NEQ
      JJ = IBC(J)
C     Both free.
      IF(II.EQ.0.AND.JJ.EQ.0) GOTO 20
C     Both restraint.
      IF(II.NE.0.AND.JJ.NE.0) GOTO 25
C     I restraint or prescribed.
      IF(II.NE.0) THEN
      QFINC(J) = QFINC(J) - STSM(I,J)*QFINC(I)
C     J restraint or prescribed.
      ELSE
      QFINC(I) = QFINC(I) - STSM(I,J)*QFINC(J)
      ENDIF
```

```
   25 STSM(I,J)= 0._8
   20 CONTINUE
   10 CONTINUE
C
      RETURN
      END
C===============================================================================


      SUBROUTINE CROUT(STSM,PIVOT,NEQ,MXPOI)
C===============================================================================
C     THIS SUBROUTINE APPLIES CROUT'S FACTORISATION TO CHANGE SYSTEM TANGENT
C     STIFFNESS MATRIX INTO THE MULTIPLICATION OF L, T AND TRANSVERSED L
C     MATRICES.
C     *The input matrix is the system tangent stiffness matrix, STSM(NEQ,NEQ).
C      The output matrices are the upper triangle matrix, STSM(NEQ,NEQ) and
C      diagonal pivots, PIVOT(NEQ).
C===============================================================================
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION STSM(MXPOI*2,MXPOI*2),PIVOT(MXPOI*2)
C
      PIVOT(1) = STSM(1,1)
      DO 1 J = 2, NEQ
      DO 2 I = 1, J-1
      A = STSM(I,J)
      IF(I.EQ.1) GOTO 2
      DO 3 L = 1, I-1
    3 A = A - STSM(L,J)*STSM(L,I)
    2 STSM(I,J) = A
      DO 4 I = 1, J-1
    4 STSM(I,J) = STSM(I,J)/STSM(I,I)
      DO 5 L = 1, J-1
    5 STSM(J,J) = STSM(J,J) - STSM(L,L)*STSM(L,J)*STSM(L,J)
    1 PIVOT(J) = STSM(J,J)
C     Check negative pivots.
      NEG = 0
      DO 30 IEQ = 1, NEQ
      IF(PIVOT(IEQ).LT.0._8) NEG = NEG + 1
   30 CONTINUE
      IF(NEG.GT.0) WRITE(*,100) NEG
  100 FORMAT(/,' *** WARNING NO. OF NEGATIVE PIVOTS = ',I5)
C
      RETURN
      END
C===============================================================================


      SUBROUTINE SOLVE(NEQ,STSM,PIVOT,QFINC,MXPOI)
C===============================================================================
C     THIS SUBROUTINE SOLVES A SET OF SIMULTANEOUS EQUATIONS.
C===============================================================================
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION STSM(MXPOI*2,MXPOI*2),QFINC(MXPOI*2),PIVOT(MXPOI*2)

C     Apply forward and backward Crout's substitutions on the QFINC vector.
      DO 1 J = 2, NEQ  !Forward substitutions.
      DO 2 L = 1, J-1
    2 QFINC(J) = QFINC(J) - STSM(L,J)*QFINC(L)
    1 CONTINUE
      DO 3 I = 1, NEQ  !Backward substitutions.
    3 QFINC(I) = QFINC(I)/PIVOT(I)
      DO 4 JJ = 2, NEQ
      J = NEQ + 2 - JJ
      DO 5 L = 1, J-1
    5 QFINC(L) = QFINC(L) - STSM(L,J)*QFINC(J)
    4 CONTINUE
C
      RETURN
      END
C===============================================================================


      SUBROUTINE ITER(PT,BETOK,QEX,IBC,STSM,ITERTY,GM,NPOIN,MXPOI,
     *                MXELE,COORD,INTMAT9,PIVOT,NDF,NELEM,IETIP,
     *                TEMP,MXSTATE,INC,IPLANE,PI,BDf,PROP,FISURF,
     *                BET,ICFLOAD,IREDSEL,IREMESH,BAS,IEMAT,MXMAT,
     *                IRTemp,IRBody)
```

```
C==============================================================================
C     THIS SUBROUTINE ITERATES SOLUTIONS TO EQUILIBRIUM USING NEWTON-RAPHSON AND
C     MODIFIED NEWTON-RAPHSON METHOD.
C     *In Newton-Raphson method, the tangent stiffness matrix must be computed
C      at every iteration loop but in modified Newton-Raphson method only
C      tangent stiffness matrix computed at the incremtal solution part is used
C      for all remaining number of iterations.
C==============================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COORD(MXPOI,2)
      DIMENSION PT(MXPOI*2),QEX(MXPOI*2),GM(MXPOI*2)
      DIMENSION STSM(MXPOI*2,MXPOI*2),PIVOT(MXPOI*2)
      DIMENSION TEMP(MXPOI,MXSTATE),BDf(MXPOI*2,MXSTATE)
      DIMENSION FIMECH(MXPOI*2),FITHER(MXPOI*2)
      DIMENSION FIBODY(MXPOI*2),FISURF(MXPOI*2)
      DIMENSION DFITHER(MXPOI*2),DFIBODY(MXPOI*2),PROP(MXMAT,7)
      INTEGER INTMAT9(MXELE,9),IETIP(MXELE),IEMAT(MXELE),IBC(MXPOI*2)

C     SMALL  is the convergence tolerance limit.
C     NITMAX is maximum number of iterations.
C     IMOD   = 1 compute all load vectors.
C            = 2 compute tangent stiffness matrix and incremtal load vectors.
C            = 3 compute both load vectors and tangent stiffness matrix.
      NEQ    = NPOIN*NDF
      SMALL  = 0.0001_8
      NITMAX = 20
      IMOD   = 1
      IF(ITERTY.EQ.1) THEN
      IMOD = 3
      NITMAX = 10
      ENDIF
C==============================================================================
C     Begin iteration loops.
C==============================================================================
      DO 1000 ITE = 1, NITMAX
      WRITE(*,110) ITE
  110 FORMAT(/,5X,'ITERATIVE LOOP WITH ITE =',I3)
      CALL LST(NPOIN,MXPOI,NELEM,MXELE,INC,MXSTATE,IPLANE,
     *         NDF,PI,IMOD,INTMAT9,IETIP,COORD,PT,TEMP,BDf,
     *         STSM,DFITHER,DFIBODY,FIMECH,FITHER,FIBODY,
     *         PROP,IREDSEL,IEMAT,MXMAT,IRTemp,IRBody)
C     GM(NEQ) is the out-of-balance force vector.
C     FISURF(NEQ) is the nodal force vector due to surface tractions.
      DO 120 IEQ  = 1, NEQ
      IF(IBC(IEQ).EQ.0) THEN
      GM(IEQ)     = QEX(IEQ) + FITHER(IEQ) + FIBODY(IEQ) - FIMECH(IEQ)
      FISURF(IEQ) = QEX(IEQ)
      ELSE
      GM(IEQ)     = 0._8
      FISURF(IEQ) = FIMECH(IEQ) - FITHER(IEQ) - FIBODY(IEQ)
      ENDIF
      IF(ICFLOAD.NE.1) FISURF(IEQ) = 0._8
  120 CONTINUE
C==============================================================================
C     Check solution's convergence.
C==============================================================================
      GNORM = 0._8   !GNORM is the out-of-balance force vector norm.
      DO 130 IEQ = 1, NEQ
      GNORM = GNORM + GM(IEQ)*GM(IEQ)
  130 CONTINUE
      GNORM = DSQRT(GNORM)
C     Set the norm from incremental part as the based norm.
      IF(ITE.EQ.1 .AND. IREMESH.EQ.0)  BAS = MAX(GNORM,SMALL)
      BET = GNORM/BAS
C     Show iteration number and its corresponding convergence factor. Note that
C     the CON. FAC. at ITERATION NO. = 0 is the convergence factor computed from
C     the out-of-balance force vector in the incremental solution part.
      WRITE(*,140) ITE-1, BET
  140 FORMAT(5X,'ITERATION NO. =',I3,' CONV. FAC. =',G13.5)
C     Use convergence tolerance factor as iterative stopping criteria.
      IF(BET.LE.BETOK) GOTO 2000
C     Full Newton-Raphson method.
      IF(ITERTY.EQ.1) THEN
      CALL APPLYBC(NEQ,IBC,STSM,GM,MXPOI)
      CALL CROUT(STSM,PIVOT,NEQ,MXPOI)
      ENDIF
C     Modified Newton-Raphson method.
```

```
      CALL SOLVE(NEQ,STSM,PIVOT,GM,MXPOI)
C     Update displacements.
      DO 150 IEQ = 1, NEQ
      IF(IBC(IEQ).EQ.0) THEN
      PT(IEQ) = PT(IEQ) + GM(IEQ)
      ELSE
      PT(IEQ) = QEX(IEQ)
      ENDIF
  150 CONTINUE
C     End each iteration.
 1000 CONTINUE
C     Solutions failed to converge over this number of iterations
C     and tolerance limit.
      WRITE(*,1010)
 1010 FORMAT(/,' *** FAILED TO CONVERGE ***',
     *       /,' WHAT DO YOU WANT TO DO NEXT ?',
     *       /,' 1 = PROCEED',
     *       /,' 0 = STOP')
      READ(*,*) IDECISION
      IF(IDECISION.EQ.0) STOP
C     Solutions has converged.
 2000 CONTINUE
C
      RETURN
      END
C===============================================================================


      SUBROUTINE GAUSSNODE(IPLANE,NDF,NPOIN,MXPOI,INC,MXSTATE,NELEM,
     *                     MXELE,IETIP,INTMAT9,COORD,PT,TEMP,SIGXX,
     *                     SIGYY,SIGXY,SIGVM,PROP,MXCTE,ICTN,NCTN,
     *                     SIGXXMAX,SIGYYMAX,SIGXYMAX,SIGVMMAX,
     *                     IREDSEL,IEMAT,MXMAT)
C===============================================================================
C     THIS SUBROUTINE COMPUTES NODAL STRESSES FROM GAUSS'S POINT VALUES AND
C     NODAL CHARACTERISTIC ELEMENT SIZES.
C===============================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COORD(MXPOI,2),PT(MXPOI*2)
      DIMENSION TEMP(MXPOI,MXSTATE),PROP(MXMAT,7)
      DIMENSION SIGXX(MXPOI),SIGYY(MXPOI),SIGXY(MXPOI)
      DIMENSION SIGZZ(MXPOI),SIGVM(MXPOI)
      DIMENSION SXX(MXPOI),SYY(MXPOI),SXY(MXPOI)
      DIMENSION SZZ(MXPOI),SMM(MXPOI)
      INTEGER INTMAT9(MXELE,9),IETIP(MXELE),IEMAT(MXELE),ICTN(MXCTE*2+1)

C     Number of equations.
      NEQ = NPOIN*NDF
C===============================================================================
C     Calculate nodal volumetric stresses.
C===============================================================================
      CALL VOLSTRESS(NPOIN,MXPOI,NELEM,MXELE,INC,MXSTATE,IPLANE,
     *               NDF,INTMAT9,IETIP,COORD,PT,TEMP,PROP,SMM,
     *               IREDSEL,IEMAT,MXMAT)
C===============================================================================
C     Calculate nodal deviatoric stresses.
C===============================================================================
      CALL DEVSTRESS(NPOIN,MXPOI,NELEM,MXELE,INC,MXSTATE,IPLANE,
     *               NDF,INTMAT9,IETIP,COORD,PT,TEMP,PROP,SXX,
     *               SYY,SXY,SZZ,IEMAT,MXMAT)
C===============================================================================
C     Calculate nodal total stresses.
C===============================================================================
      DO 1100 I = 1, NPOIN
      SIGXX(I) = SMM(I) + SXX(I)
      SIGYY(I) = SMM(I) + SYY(I)
      SIGXY(I) = SXY(I)
      SIGZZ(I) = SMM(I) + SZZ(I)
      SIGVM(I) = DSQRT( (2._8*SIGXX(I)-SIGYY(I)-SIGZZ(I))**2._8/6._8 +
     *                  (2._8*SIGYY(I)-SIGZZ(I)-SIGXX(I))**2._8/6._8 +
     *                  (2._8*SIGZZ(I)-SIGXX(I)-SIGYY(I))**2._8/6._8 +
     *                   3._8*SIGXY(I)*SIGXY(I) )
 1100 CONTINUE
C===============================================================================
C     Find the maximum von Mises stress of all crack tip nodes in 9-node
C     degenerated crack tip elements.
C===============================================================================
```

```
      SIGXXMAX = 0._8
      SIGYYMAX = 0._8
      SIGXYMAX = 0._8
      SIGVMMAX = 0._8
      DO 1105 I = 1, NCTN
      IF( DABS(SIGXXMAX) .LT. DABS(SIGXX(ICTN(I))) )
     *    SIGXXMAX = SIGXX(ICTN(I))
      IF( DABS(SIGYYMAX) .LT. DABS(SIGYY(ICTN(I))) )
     *    SIGYYMAX = SIGYY(ICTN(I))
      IF( DABS(SIGXYMAX) .LT. DABS(SIGXY(ICTN(I))) )
     *    SIGXYMAX = SIGXY(ICTN(I))
      IF( DABS(SIGVMMAX) .LT. DABS(SIGVM(ICTN(I))) )
     *    SIGVMMAX = SIGVM(ICTN(I))
 1105 CONTINUE
C
      RETURN
      END
C================================================================================


      SUBROUTINE VOLSTRESS(NPOIN,MXPOI,NELEM,MXELE,INC,MXSTATE,IPLANE,
     *                     NDF,INTMAT9,IETIP,COORD,PT,TEMP,PROP,SMM,
     *                     IREDSEL,IEMAT,MXMAT)
C================================================================================
C     THIS SUBROUTINE COMPUTES NODAL VOLUMETRIC STRESSES BY TRANSFORMING THEIR
C     GAUSS'S POINT VALUES TO NODES.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COORD(MXPOI,2)
      DIMENSION X(9),Y(9),XG(9),YG(9),SGMM(9)
      DIMENSION BMAT(4,9*2),DNDA(9),DNDB(9),AJ(2,2),AJI(2,2)
      DIMENSION PT(MXPOI*2),EDISP(9*2),EKBG(9*2,9*2)
      DIMENSION STRAIN(4),PROP(MXMAT,7),TR(9,9)
      DIMENSION TEMP(MXPOI,MXSTATE),ETEMP(9)
      DIMENSION SMM(MXPOI),ONE(MXPOI)
      INTEGER INTMAT9(MXELE,9),IETIP(MXELE),IEMAT(MXELE)

C     Number of equations.
      NEQ = NPOIN*NDF
C     Set element matrix dimensions for analyzed problem.
      IF(IPLANE.EQ.3) THEN
      NDIM = 4  !Axisymmetric case.
      ELSE
      NDIM = 3  !Plane stress or plane strain case.
      ENDIF
C     Set initial nodal quantities to zero.
      DO 10 I = 1, NPOIN
      SMM(I)  = 0._8
      ONE(I)  = 0._8
   10 CONTINUE
C================================================================================
C     Loop over all elements.
C================================================================================
      DO 5000 IE = 1, NELEM
C     Read material properties of each element.
      ELAS   = PROP(IEMAT(IE),1)
      PR     = PROP(IEMAT(IE),2)
      YSTRSS = PROP(IEMAT(IE),3)
      AHARD  = PROP(IEMAT(IE),4)
      ALPHA  = PROP(IEMAT(IE),5)
      COTHR  = PROP(IEMAT(IE),6)
C     9-node rectangular crack tip element.
      IF(IETIP(IE).EQ.1) THEN
      NN   =  9         !Number of nodes.
      NDOF =  NDF*NN    !Number of element degrees of freedom.
C      Set number of Gauss's points, its coordinates and weights.
       IF(IREDSEL.NE.1) THEN
       NG    =  9
       XG(1) = -DSQRT(3._8)/DSQRT(5._8)
       XG(2) =  XG(1)
       XG(3) =  XG(1)
       XG(4) =  0._8
       XG(5) =  XG(4)
       XG(6) =  XG(4)
       XG(7) = -XG(1)
       XG(8) =  XG(7)
       XG(9) =  XG(7)
```

```
C
        YG(1) =  XG(1)
        YG(2) =  XG(4)
        YG(3) =  XG(7)
        YG(4) =  XG(1)
        YG(5) =  XG(4)
        YG(6) =  XG(7)
        YG(7) =  XG(1)
        YG(8) =  XG(4)
        YG(9) =  XG(7)
        ELSE
        NG    =  4
        XG(1) = -1._8/DSQRT(3._8)
        XG(2) = -1._8/DSQRT(3._8)
        XG(3) =  1._8/DSQRT(3._8)
        XG(4) =  1._8/DSQRT(3._8)
C
        YG(1) = -1._8/DSQRT(3._8)
        YG(2) =  1._8/DSQRT(3._8)
        YG(3) = -1._8/DSQRT(3._8)
        YG(4) =  1._8/DSQRT(3._8)
        ENDIF
C      6-node triangular element.
       ELSE
       NN   =  6         !Number of nodes.
       NDOF =  NDF*NN    !Number of element degrees of freedom.
C       Set number of Gauss's points, its coordinates and weights.
        IF(IREDSEL.NE.1) THEN
        NG    =  7
        XG(1) =  1._8/3._8
        XG(2) =  0.101286507323456_8
        XG(3) =  0.797426985353087_8
        XG(4) =  XG(2)
        XG(5) =  0.470142064105115_8
        XG(6) =  0.059715871789770_8
        XG(7) =  XG(5)
C
        YG(1) =  XG(1)
        YG(2) =  XG(2)
        YG(3) =  XG(2)
        YG(4) =  XG(3)
        YG(5) =  XG(5)
        YG(6) =  XG(5)
        YG(7) =  XG(6)
        ELSE
        NG    =  3
        XG(1) =  1._8/6._8
        XG(2) =  2._8/3._8
        XG(3) =  1._8/6._8
C
        YG(1) =  1._8/6._8
        YG(2) =  1._8/6._8
        YG(3) =  2._8/3._8
        ENDIF
       ENDIF
C      Set initial Gauss's point quantities to zeros in each element.
       DO 5 I  = 1, NG
       SGMM(I) = 0._8
     5 CONTINUE
C      Create elemant nodal displacement and temperature vector.
       DO 100 I = 1, NN
       II           = INTMAT9(IE,I)
       X(I)         = COORD(II,1)
       Y(I)         = COORD(II,2)
       ETEMP(I)     = TEMP(II,INC+1)
       EDISP(2*I-1) = PT(2*II-1)
       EDISP(2*I)   = PT(2*II)
   100 CONTINUE
C================================================================================
C      Loop over each Gauss's point on an element.
C================================================================================
       DO 336 K = 1, NG
C      Compute strain-displacement matrix.
       A = XG(K)
       B = YG(K)
       CALL BJ9(X,Y,A,B,BMAT,EKBG,AJ,AJI,DETJAC,DNDA,DNDB,
      *         NN,NDF,IPLANE)
```

```
C     Compute temperature at a Gauss's point.
      CALL GVALUE(A,B,ETEMP,T,NN)
C     Set initial strain vector to zero.
      DO 331 I  = 1, NDIM
      STRAIN(I) = 0._8
  331 CONTINUE
C     Compute strain vector.
      DO 332 I  = 1, NDIM
      DO 332 J  = 1, NDOF
      STRAIN(I) = STRAIN(I) + BMAT(I,J)*EDISP(J)
  332 CONTINUE
C     Find effective stress.
      CALL FINDSTRSS(EFSTRSS,STRAIN,YSTRSS,AHARD,ALPHA,PR,
     *               ELAS,AMOD,EFRATIO,IPLANE,T,COTHR,BETA)
C     Plane stress case.
      IF(IPLANE.EQ.1) THEN
      COND = BETA*ELAS/(1._8-2._8*PR)/(1._8+2._8*BETA)
      SGMM(K) = COND*( STRAIN(1)+STRAIN(2)-2._8*COTHR*T )
      ENDIF
C     Plane strain case.
      IF(IPLANE.EQ.2) THEN
      SGMM(K) = ELAS/(3._8*(1._8-2._8*PR))*
     *          ( STRAIN(1)+STRAIN(2)-3._8*COTHR*T )
      ENDIF
C     Axisymmetric case.
      IF(IPLANE.EQ.3)THEN
      SGMM(K) = ELAS/(3._8*(1._8-2._8*PR))*
     *          ( STRAIN(1)+STRAIN(2)+STRAIN(4)-3._8*COTHR*T )
      ENDIF
C     End each Gauss's point.
  336 CONTINUE
C================================================================================
C     Transform all Gauss's point stress to nodes.
C================================================================================
      CALL TRMAT(NN,NG,TR)
C     Sum all nodal stress in the entire model before nodal averaging.
      DO 655 I = 1, NN
      II = INTMAT9(IE,I)
      DO 650 J = 1, NG
      SMM(II)  = SMM(II) + TR(I,J)*SGMM(J)
  650 CONTINUE
      ONE(II)  = ONE(II) + 1._8
  655 CONTINUE
C     End each element.
 5000 CONTINUE
C================================================================================
C     Calculate nodal stresses.
C================================================================================
      DO 1100 I = 1, NPOIN
      IF(ONE(I).LE.0._8) WRITE(*,1200) I
 1200 FORMAT(' *** WARNING ***  NO CONTRIBUTION AT NODE', I5)
      IF(ONE(I).LE.0._8) STOP
      SMM(I) = SMM(I)/ONE(I)
 1100 CONTINUE
C
      RETURN
      END
C================================================================================


      SUBROUTINE DEVSTRESS(NPOIN,MXPOI,NELEM,MXELE,INC,MXSTATE,IPLANE,
     *                     NDF,INTMAT9,IETIP,COORD,PT,TEMP,PROP,SXX,
     *                     SYY,SXY,SZZ,IEMAT,MXMAT)
C================================================================================
C     THIS SUBROUTINE COMPUTES NODAL DEVIATORIC STRESSES BY TRANSFORMING THEIR
C     GAUSS'S POINT VALUES TO NODES.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COORD(MXPOI,2)
      DIMENSION X(9),Y(9),XG(9),YG(9)
      DIMENSION ONE(MXPOI),SXX(MXPOI),SYY(MXPOI),SXY(MXPOI),SZZ(MXPOI)
      DIMENSION SGXX(9),SGYY(9),SGXY(9),SGZZ(9)
      DIMENSION BMAT(4,9*2),DNDA(9),DNDB(9),AJ(2,2),AJI(2,2)
      DIMENSION PT(MXPOI*2),EDISP(9*2),EKBG(9*2,9*2)
      DIMENSION GMAT(4,4),HMAT(4,4),STRAIN(4),STRAIN0(4)
      DIMENSION TEMP(MXPOI,MXSTATE),ETEMP(9)
      DIMENSION PROP(MXMAT,7),TR(9,9)
```

```
      INTEGER INTMAT9(MXELE,9),IETIP(MXELE),IEMAT(MXELE)

C     Number of equations.
      NEQ = NPOIN*NDF
C     Set element matrix dimensions for analyzed problem.
      IF(IPLANE.EQ.3) THEN
      NDIM = 4  !Axisymmetric case.
      ELSE
      NDIM = 3  !Plane stress or plane strain case.
      ENDIF
C     Set initial nodal quantities to zero.
      DO 10 I = 1, NPOIN
      SXX(I) = 0._8
      SYY(I) = 0._8
      SXY(I) = 0._8
      SZZ(I) = 0._8
      ONE(I) = 0._8
   10 CONTINUE
C===============================================================================
C     Loop over all elements.
C===============================================================================
      DO 5000 IE = 1, NELEM
C     Read material properties of each element.
      ELAS   = PROP(IEMAT(IE),1)
      PR     = PROP(IEMAT(IE),2)
      YSTRSS = PROP(IEMAT(IE),3)
      AHARD  = PROP(IEMAT(IE),4)
      ALPHA  = PROP(IEMAT(IE),5)
      COTHR  = PROP(IEMAT(IE),6)
C     9-node rectangular crack tip element.
      IF(IETIP(IE).EQ.1) THEN
      NN    =  9         !Number of nodes.
      NG    =  9         !Number of Gauss's points.
      NDOF  =  NDF*NN    !Number of element degrees of freedom.
C     X coordinate at each Gauss's point for this element.
      XG(1) = -DSQRT(3._8)/DSQRT(5._8)
      XG(2) =  XG(1)
      XG(3) =  XG(1)
      XG(4) =  0._8
      XG(5) =  XG(4)
      XG(6) =  XG(4)
      XG(7) = -XG(1)
      XG(8) =  XG(7)
      XG(9) =  XG(7)
C     Y coordinate at each Gauss's point.
      YG(1) =  XG(1)
      YG(2) =  XG(4)
      YG(3) =  XG(7)
      YG(4) =  XG(1)
      YG(5) =  XG(4)
      YG(6) =  XG(7)
      YG(7) =  XG(1)
      YG(8) =  XG(4)
      YG(9) =  XG(7)
      ELSE
C     6-node triangular element.
      NN    =  6         !Number of nodes.
      NG    =  7         !Number of Gauss's points.
      NDOF  =  NDF*NN    !Number of element degrees of freedom.
C     X coordinate at each Gauss's point for this element.
      XG(1) =  1._8/3._8
      XG(2) =  0.101286507323456_8
      XG(3) =  0.797426985353087_8
      XG(4) =  XG(2)
      XG(5) =  0.470142064105115_8
      XG(6) =  0.059715871789770_8
      XG(7) =  XG(5)
C     Y coordinate at each Gauss's point for this element.
      YG(1) =  XG(1)
      YG(2) =  XG(2)
      YG(3) =  XG(2)
      YG(4) =  XG(3)
      YG(5) =  XG(5)
      YG(6) =  XG(5)
      YG(7) =  XG(6)
      ENDIF
C     Set initial Gauss's point quantities to zeros in each element.
```

```
      DO 5 I  = 1, NG
      SGXX(I) = 0._8
      SGYY(I) = 0._8
      SGXY(I) = 0._8
      SGZZ(I) = 0._8
    5 CONTINUE
C     Create elemant nodal displacement and temperature vector.
      DO 100 I = 1, NN
      II           = INTMAT9(IE,I)
      X(I)         = COORD(II,1)
      Y(I)         = COORD(II,2)
      ETEMP(I)     = TEMP(II,INC+1)
      EDISP(2*I-1) = PT(2*II-1)
      EDISP(2*I)   = PT(2*II)
  100 CONTINUE
C================================================================================
C     Loop over each Gauss's point on an element.
C================================================================================
      DO 336 K = 1, NG
C     Compute strain-displacement matrix.
      A = XG(K)
      B = YG(K)
      CALL BJ9(X,Y,A,B,BMAT,EKBG,AJ,AJI,DETJAC,DNDA,DNDB,
     *         NN,NDF,IPLANE)
C     Compute temperature at a Gauss's point.
      CALL GVALUE(A,B,ETEMP,T,NN)
C     Set initial strain vector to zero.
      DO 331 I  = 1, NDIM
      STRAIN(I) = 0._8
  331 CONTINUE
C     Compute strain vector.
      DO 332 I  = 1, NDIM
      DO 332 J  = 1, NDOF
      STRAIN(I) = STRAIN(I) + BMAT(I,J)*EDISP(J)
  332 CONTINUE
C     Set coefficient of thermal expansion vector.
      IF(IPLANE.EQ.1) THEN  !Plane stress case.
      STRAIN0(1) = COTHR*T
      STRAIN0(2) = COTHR*T
      STRAIN0(3) = 0._8
      ENDIF
C================================================================================
C     Find effective stress.
C================================================================================
      CALL FINDSTRSS(EFSTRSS,STRAIN,YSTRSS,AHARD,ALPHA,PR,
     *               ELAS,AMOD,EFRATIO,IPLANE,T,COTHR,BETA)
C================================================================================
C     Plane stress case.
C================================================================================
C     {STRESSdev} = [GMAT]{STRAIN} - [HMAT]{STRAIN0}
C================================================================================
      IF(IPLANE.EQ.1) THEN
      COND = BETA*ELAS/(1._8-2._8*PR)/(1._8+2._8*BETA)
      GMAT(1,1) =  COND*(1._8+BETA)
      GMAT(1,2) = -COND*BETA
      GMAT(1,3) =  0._8
      GMAT(2,1) =  GMAT(1,2)
      GMAT(2,2) =  GMAT(1,1)
      GMAT(2,3) =  0._8
      GMAT(3,1) =  0._8
      GMAT(3,2) =  0._8
      GMAT(3,3) =  COND*(0.5_8+BETA)
C
      HMAT(1,1) =  COND
      HMAT(1,2) =  0._8
      HMAT(1,3) =  0._8
      HMAT(2,1) =  0._8
      HMAT(2,2) =  HMAT(1,1)
      HMAT(2,3) =  0._8
      HMAT(3,1) =  0._8
      HMAT(3,2) =  0._8
      HMAT(3,3) =  0._8
      ENDIF
C================================================================================
C     Plane strain case.
C================================================================================
C     {STRESSdev} = [GMAT]{STRAIN}
```

```
C===============================================================================
      IF(IPLANE.EQ.2) THEN
      GMAT(1,1) =  2._8/9._8*EFRATIO*2._8
      GMAT(1,2) = -2._8/9._8*EFRATIO
      GMAT(1,3) =  0._8
      GMAT(2,1) =  GMAT(1,2)
      GMAT(2,2) =  GMAT(1,1)
      GMAT(2,3) =  0._8
      GMAT(3,1) =  0._8
      GMAT(3,2) =  0._8
      GMAT(3,3) =  2._8/9._8*EFRATIO*1.5_8
      ENDIF
C===============================================================================
C     Axisymmetric case.
C===============================================================================
C     {STRESSdev} = [GMAT]{STRAIN}
C===============================================================================
      IF(IPLANE.EQ.3)THEN
      GMAT(1,1) =  2._8/9._8*EFRATIO*2._8
      GMAT(1,2) = -2._8/9._8*EFRATIO
      GMAT(1,3) =  0._8
      GMAT(1,4) =  GMAT(1,2)
      GMAT(2,1) =  GMAT(1,2)
      GMAT(2,2) =  GMAT(1,1)
      GMAT(2,3) =  0._8
      GMAT(2,4) =  GMAT(1,2)
      GMAT(3,1) =  GMAT(1,3)
      GMAT(3,2) =  GMAT(2,3)
      GMAT(3,3) =  2._8/9._8*EFRATIO*1.5_8
      GMAT(3,4) =  0._8
      GMAT(4,1) =  GMAT(1,4)
      GMAT(4,2) =  GMAT(2,4)
      GMAT(4,3) =  GMAT(3,4)
      GMAT(4,4) =  GMAT(1,1)
      ENDIF
C===============================================================================
C     Compute stresses at a Gauss's point.
C===============================================================================
      IF(IPLANE.EQ.1) THEN   !Plane stress case.
      DO 500 I = 1, NDIM
      SGXX(K)  = SGXX(K) + GMAT(1,I)*STRAIN(I) - HMAT(1,I)*STRAIN0(I)
      SGYY(K)  = SGYY(K) + GMAT(2,I)*STRAIN(I) - HMAT(2,I)*STRAIN0(I)
      SGXY(K)  = SGXY(K) + GMAT(3,I)*STRAIN(I) - HMAT(3,I)*STRAIN0(I)
  500 CONTINUE
      ELSE   !Plane strain or axisymmetric case.
      DO 550 I = 1, NDIM
      SGXX(K)  = SGXX(K) + GMAT(1,I)*STRAIN(I)
      SGYY(K)  = SGYY(K) + GMAT(2,I)*STRAIN(I)
      SGXY(K)  = SGXY(K) + GMAT(3,I)*STRAIN(I)
  550 CONTINUE
      ENDIF
C     Compute deviatoric stress in z direction.
      IF(IPLANE.EQ.1) THEN  !Plane stress case.
      EPS33   = ( 3._8*COTHR*T+(BETA-1._8)*(STRAIN(1)+STRAIN(2)) )/
     *          (1._8+2._8*BETA)
      SGZZ(K) = 2._8/9._8*EFRATIO*( 2._8*EPS33-STRAIN(1)-STRAIN(2) )
      ENDIF
      IF(IPLANE.EQ.2) THEN  !Plane strain case.
      SGZZ(K) = 2._8/9._8*EFRATIO*( -STRAIN(1)-STRAIN(2) )
      ENDIF
      IF(IPLANE.EQ.3) THEN  !Axisymmetric case.
      SGZZ(K) = 2._8/9._8*EFRATIO*( 2._8*STRAIN(4)-STRAIN(1)-STRAIN(2) )
      ENDIF
C     End each Gauss's point.
  336 CONTINUE
C===============================================================================
C     Transform all Gauss's point stress to nodes.
C===============================================================================
      CALL TRMAT(NN,NG,TR)
C     Sum all nodal stress in the entire model before nodal averaging.
      DO 655 I = 1, NN
      II = INTMAT9(IE,I)
      DO 650 J = 1, NG
      SXX(II)  = SXX(II) + TR(I,J)*SGXX(J)
      SYY(II)  = SYY(II) + TR(I,J)*SGYY(J)
      SXY(II)  = SXY(II) + TR(I,J)*SGXY(J)
      SZZ(II)  = SZZ(II) + TR(I,J)*SGZZ(J)
```

```
  650 CONTINUE
      ONE(II)  = ONE(II) + 1._8
  655 CONTINUE
C     End each element.
 5000 CONTINUE
C================================================================================
C     Calculate nodal stresses.
C================================================================================
      DO 1100 I = 1, NPOIN
      IF(ONE(I).LE.0._8) WRITE(*,1200) I
 1200 FORMAT(' *** WARNING ***  NO CONTRIBUTION AT NODE', I5)
      IF(ONE(I).LE.0._8) STOP
      SXX(I) = SXX(I)/ONE(I)
      SYY(I) = SYY(I)/ONE(I)
      SXY(I) = SXY(I)/ONE(I)
      SZZ(I) = SZZ(I)/ONE(I)
 1100 CONTINUE
C
      RETURN
      END
C================================================================================


      SUBROUTINE TRMAT(NN,NG,TR)
C================================================================================
C     THIS SUBROUTINE RETURNS ANY GAUSS'S POINT-TO-NODE TRANSFORMATION MATRIX.
C================================================================================

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TR(9,9)

C================================================================================
C     9-node rectangular element with 9 Gauss's points.
C================================================================================
      IF(NN.EQ.9 .AND. NG.EQ.9) THEN
C
      TR(1,1) =  .21869398183909500E+01_8
      TR(1,2) = -.98588703846749160E+00_8
      TR(1,3) =  .27777777777777800E+00_8
      TR(1,4) = -.98588703846749160E+00_8
      TR(1,5) =  .44444444444444510E+00_8
      TR(1,6) = -.12522407264362050E+00_8
      TR(1,7) =  .27777777777777800E+00_8
      TR(1,8) = -.12522407264362050E+00_8
      TR(1,9) =  .35282403831273010E-01_8
C
      TR(2,1) =  .27777777777777840E+00_8
      TR(2,2) = -.12522407264362070E+00_8
      TR(2,3) =  .35282403831273120E-01_8
      TR(2,4) = -.98588703846749160E+00_8
      TR(2,5) =  .44444444444444510E+00_8
      TR(2,6) = -.12522407264362050E+00_8
      TR(2,7) =  .21869398183909500E+01_8
      TR(2,8) = -.98588703846749120E+00_8
      TR(2,9) =  .27777777777777790E+00_8
C
      TR(3,1) =  .35282403831273350E-01_8
      TR(3,2) = -.12522407264362070E+00_8
      TR(3,3) =  .27777777777777800E+00_8
      TR(3,4) = -.12522407264362050E+00_8
      TR(3,5) =  .44444444444444510E+00_8
      TR(3,6) = -.98588703846749120E+00_8
      TR(3,7) =  .27777777777777800E+00_8
      TR(3,8) = -.98588703846749120E+00_8
      TR(3,9) =  .21869398183909490E+01_8
C
      TR(4,1) =  .27777777777777800E+00_8
      TR(4,2) = -.98588703846749160E+00_8
      TR(4,3) =  .21869398183909500E+01_8
      TR(4,4) = -.12522407264362050E+00_8
      TR(4,5) =  .44444444444444510E+00_8
      TR(4,6) = -.98588703846749120E+00_8
      TR(4,7) =  .35282403831273120E-01_8
      TR(4,8) = -.12522407264362050E+00_8
      TR(4,9) =  .27777777777777790E+00_8
C
      TR(5,1) =  .13877787807814460E-15_8
```

```
      TR(5,2) = -.11102230246251570E-15_8
      TR(5,3) =  .55511151231257830E-16_8
      TR(5,4) =  .14788305577012360E+01_8
      TR(5,5) = -.66666666666666690E+00_8
      TR(5,6) =  .18783610896543060E+00_8
      TR(5,7) =  .55511151231257830E-16_8
      TR(5,8) = -.55511151231257830E-16_8
      TR(5,9) =  .27755575615628910E-16_8
C
      TR(6,1) =  .83266726846886740E-16_8
      TR(6,2) =  .18783610896543080E+00_8
      TR(6,3) = -.11102230246251570E-15_8
      TR(6,4) = -.55511151231257830E-16_8
      TR(6,5) = -.66666666666666690E+00_8
      TR(6,6) = -.55511151231257830E-16_8
      TR(6,7) =  .16653345369377350E-15_8
      TR(6,8) =  .14788305577012360E+01_8
      TR(6,9) =  .13877787807814460E-15_8
C
      TR(7,1) = -.83266726846886740E-16_8
      TR(7,2) = -.11102230246251570E-15_8
      TR(7,3) =  .16653345369377350E-15_8
      TR(7,4) =  .18783610896543080E+00_8
      TR(7,5) = -.66666666666666690E+00_8
      TR(7,6) =  .14788305577012360E+01_8
      TR(7,7) = -.55511151231257830E-16_8
      TR(7,8) = -.55511151231257830E-16_8
      TR(7,9) =  .13877787807814460E-15_8
C
      TR(8,1) = -.27755575615628910E-16_8
      TR(8,2) =  .14788305577012360E+01_8
      TR(8,3) =  .00000000000000000E+00_8
      TR(8,4) = -.55511151231257830E-16_8
      TR(8,5) = -.66666666666666690E+00_8
      TR(8,6) = -.55511151231257830E-16_8
      TR(8,7) =  .55511151231257830E-16_8
      TR(8,8) =  .18783610896543060E+00_8
      TR(8,9) =  .27755575615628910E-16_8
C
      TR(9,1) =  .27755575615628910E-16_8
      TR(9,2) =  .11102230246251570E-15_8
      TR(9,3) = -.55511151231257830E-16_8
      TR(9,4) =  .55511151231257830E-16_8
      TR(9,5) =  .99999999999999990E+00_8
      TR(9,6) =  .55511151231257830E-16_8
      TR(9,7) =  .00000000000000000E+00_8
      TR(9,8) =  .55511151231257830E-16_8
      TR(9,9) = -.27755575615628910E-16_8
      GOTO 100
      ENDIF
C================================================================================
C     9-node rectangular element with 4 Gauss's points.
C================================================================================
      IF(NN.EQ.9 .AND. NG.EQ.4) THEN
C
      TR(1,1) =  .18660254037844380E+01_8
      TR(1,2) = -.49999999999999980E+00_8
      TR(1,3) = -.49999999999999980E+00_8
      TR(1,4) =  .13397459621556130E+00_8
C
      TR(2,1) = -.49999999999999980E+00_8
      TR(2,2) =  .13397459621556130E+00_8
      TR(2,3) =  .18660254037844380E+01_8
      TR(2,4) = -.49999999999999980E+00_8
C
      TR(3,1) =  .13397459621556130E+00_8
      TR(3,2) = -.49999999999999980E+00_8
      TR(3,3) = -.49999999999999980E+00_8
      TR(3,4) =  .18660254037844380E+01_8
C
      TR(4,1) = -.49999999999999980E+00_8
      TR(4,2) =  .18660254037844380E+01_8
      TR(4,3) =  .13397459621556130E+00_8
      TR(4,4) = -.49999999999999980E+00_8
C
      TR(5,1) =  .68301270189221920E+00_8
      TR(5,2) = -.18301270189221920E+00_8
```

```
      TR(5,3) =  .68301270189221920E+00_8
      TR(5,4) = -.18301270189221920E+00_8
C
      TR(6,1) = -.18301270189221920E+00_8
      TR(6,2) = -.18301270189221920E+00_8
      TR(6,3) =  .68301270189221920E+00_8
      TR(6,4) =  .68301270189221920E+00_8
C
      TR(7,1) = -.18301270189221920E+00_8
      TR(7,2) =  .68301270189221920E+00_8
      TR(7,3) = -.18301270189221920E+00_8
      TR(7,4) =  .68301270189221920E+00_8
C
      TR(8,1) =  .68301270189221920E+00_8
      TR(8,2) =  .68301270189221920E+00_8
      TR(8,3) = -.18301270189221920E+00_8
      TR(8,4) = -.18301270189221920E+00_8
C
      TR(9,1) =  .25000000000000000E+00_8
      TR(9,2) =  .25000000000000000E+00_8
      TR(9,3) =  .25000000000000000E+00_8
      TR(9,4) =  .25000000000000000E+00_8
      GOTO 100
      ENDIF
C===============================================================================
C     6-node triangular element with 7 Gauss's points.
C===============================================================================
      IF(NN.EQ.6 .AND. NG.EQ.7) THEN
C
      TR(1,1) = -.69230769230764320E+00_8
      TR(1,2) =  .19743924601202800E+01_8
      TR(1,3) =  .14344495818924320E+00_8
      TR(1,4) =  .14344495818922900E+00_8
      TR(1,5) =  .25637677064884160E+00_8
      TR(1,6) = -.41267572741996480E+00_8
      TR(1,7) = -.41267572741999300E+00_8
C
      TR(2,1) = -.69230769230768990E+00_8
      TR(2,2) =  .14344495818923650E+00_8
      TR(2,3) =  .19743924601203660E+01_8
      TR(2,4) =  .14344495818923960E+00_8
      TR(2,5) = -.41267572742002570E+00_8
      TR(2,6) =  .25637677064887310E+00_8
      TR(2,7) = -.41267572742003100E+00_8
C
      TR(3,1) = -.69230769230768410E+00_8
      TR(3,2) =  .14344495818920590E+00_8
      TR(3,3) =  .14344495818923220E+00_8
      TR(3,4) =  .19743924601203570E+01_8
      TR(3,5) = -.41267572742002700E+00_8
      TR(3,6) = -.41267572742000700E+00_8
      TR(3,7) =  .25637677064886730E+00_8
C
      TR(4,1) =  .17751479289940540E+00_8
      TR(4,2) =  .98125809310777970E-01_8
      TR(4,3) =  .98125809310778190E-01_8
      TR(4,4) =  .15626601555207210E+00_8
      TR(4,5) = -.28229740694391450E+00_8
      TR(4,6) = -.28229740694392030E+00_8
      TR(4,7) =  .10345623868147880E+01_8
C
      TR(5,1) =  .17751479289941500E+00_8
      TR(5,2) =  .15626601555204820E+00_8
      TR(5,3) =  .98125809310774580E-01_8
      TR(5,4) =  .98125809310770860E-01_8
      TR(5,5) =  .10345623868147820E+01_8
      TR(5,6) = -.28229740694390500E+00_8
      TR(5,7) = -.28229740694391580E+00_8
C
      TR(6,1) =  .17751479289939320E+00_8
      TR(6,2) =  .98125809310794290E-01_8
      TR(6,3) =  .15626601555207180E+00_8
      TR(6,4) =  .98125809310782410E-01_8
      TR(6,5) = -.28229740694390900E+00_8
      TR(6,6) =  .10345623868147800E+01_8
      TR(6,7) = -.28229740694392950E+00_8
      GOTO 100
```

```
      ENDIF
C===============================================================================
C     6-node triangular element with 3 Gauss's points.
C===============================================================================
      IF(NN.EQ.6 .AND. NG.EQ.3) THEN
C
      TR(1,1) =  .16666666666666665E+01_8
      TR(1,2) = -.33333333333333331E+00_8
      TR(1,3) = -.33333333333333331E+00_8
C
      TR(2,1) = -.33333333333333348E+00_8
      TR(2,2) =  .16666666666666667E+01_8
      TR(2,3) = -.33333333333333331E+00_8
C
      TR(3,1) = -.33333333333333348E+00_8
      TR(3,2) = -.33333333333333331E+00_8
      TR(3,3) =  .16666666666666667E+01_8
C
      TR(4,1) =  .66666666666666652E+00_8
      TR(4,2) =  .66666666666666674E+00_8
      TR(4,3) = -.33333333333333331E+00_8
C
      TR(5,1) = -.33333333333333348E+00_8
      TR(5,2) =  .66666666666666674E+00_8
      TR(5,3) =  .66666666666666674E+00_8
C
      TR(6,1) =  .66666666666666652E+00_8
      TR(6,2) = -.33333333333333331E+00_8
      TR(6,3) =  .66666666666666674E+00_8
      GOTO 100
      ENDIF
C     Incorrect number of nodes or Gauss's points.
      WRITE(*,200) NN, NG
  200 FORMAT(/,' *** NO TRANSFORMATION MATRIX FOR THIS ELEMENT WITH',
     *       /,' NN = ',I2,' NG = ',I2,' ***')
C
  100 CONTINUE
C
      RETURN
      END
C===============================================================================


      SUBROUTINE CJINT(IPLANE,NDF,NDOM,MXDOM,MXEDOM,NEIND,IEIND,
     *                 MXCFNODE,IFACEN,IFACEE,NFACEN,NFACE,IFACE,
     *                 NPOIN,MXPOI,INC,MXSTATE,MXELE,IETIP,INTMAT9,
     *                 COORD,PT,TEMP,BDf,NODEK1new,CANGLE,ROR,PI,
     *                 PROP,AJdom,AJface,AJint,FISURF,CTQEX,FDQEX,
     *                 ICFLOAD,IDOMTY,IREDSEL,IEMAT,MXMAT)
C===============================================================================
C     THIS SUBROUTINE CALCULATES PARAMETER J-INTEGRAL USING DOMAIN INTEGRAL
C     METHOD BASED ON DELORENZI'S PAPER, 'ENERGY RELEASE RATE CALCULATIONS BY
C     FINITE ELEMENT METHOD' AND SHIH'S PAPER, 'ENERGY RELEASE RATE ALONG THREE-
C     DIMENTIONAL CRACK FRONT IN A THERMALLY STRESSED BODY'.
C===============================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COORD(MXPOI,2),PT(MXPOI*2),ROR(MXDOM,4)
      DIMENSION TEMP(MXPOI,MXSTATE),BDf(MXPOI*2,MXSTATE)
      DIMENSION PROP(MXMAT,7),FISURF(MXPOI*2)
      DIMENSION CTQEX(4),FDQEX(4)
      DIMENSION AJvol(MXDOM),AJdev(MXDOM),AJdom(MXDOM)
      DIMENSION AJface(MXDOM),AJint(MXDOM)
      INTEGER INTMAT9(MXELE,9),IETIP(MXELE),IEMAT(MXELE)
      INTEGER NEIND(MXDOM),IEIND(MXEDOM,MXDOM)
      INTEGER IFACEN(2,MXCFNODE,MXDOM),NFACEN(2,MXDOM)
      INTEGER IFACEE(2,(MXCFNODE-1)/2,MXDOM)

C===============================================================================
C     Compute J-integral parameter from volumetric stress expression.
C===============================================================================
      CALL CJVOL(IPLANE,NDF,NDOM,MXDOM,MXEDOM,NEIND,IEIND,
     *           NPOIN,MXPOI,INC,MXSTATE,MXELE,IETIP,INTMAT9,
     *           COORD,PT,TEMP,NODEK1new,CANGLE,ROR,PROP,AJvol,
     *           IDOMTY,IREDSEL,IEMAT,MXMAT)
C===============================================================================
C     Compute J-integral parameter from deviatoric stress expression.
C===============================================================================
```

```
      CALL CJDEV(IPLANE,NDF,NDOM,MXDOM,MXEDOM,NEIND,IEIND,
     *           NPOIN,MXPOI,INC,MXSTATE,MXELE,IETIP,INTMAT9,
     *           COORD,PT,TEMP,BDf,NODEK1new,CANGLE,ROR,PROP,
     *           AJdev,IDOMTY,IEMAT,MXMAT)
C===============================================================================
C     Compute J-integral parameter from crack face expression.
C===============================================================================
      IF(ICFLOAD.EQ.1) THEN
       CALL CJFACE(IPLANE,NDF,NDOM,MXDOM,MXCFNODE,IFACEN,
     *             IFACEE,NFACEN,NFACE,IFACE,NPOIN,MXPOI,
     *             MXELE,IETIP,INTMAT9,COORD,PT,NODEK1new,
     *             CANGLE,ROR,PI,PROP,AJface,FISURF,CTQEX,
     *             FDQEX,IDOMTY,IEMAT,MXMAT)
      ELSE
       DO 10 IDOM = 1, NDOM
        AJface(IDOM) = 0._8
   10  CONTINUE
      ENDIF
C===============================================================================
C     Compute total J-integral parameter of each integrated domain.
C===============================================================================
      DO 3000 IDOM = 1, NDOM
      AJdom(IDOM)  = AJvol(IDOM) + AJdev(IDOM)
      AJint(IDOM)  = AJdom(IDOM) - AJface(IDOM)
      WRITE(*,3001) IDOM
 3001 FORMAT(/,' DOMAIN [',I2,']')
      WRITE(*,*) ' J-DOMAIN     =', AJdom(IDOM)
      WRITE(*,*) ' J-CRACK FACE =', AJface(IDOM)
      WRITE(*,*) ' J-INTEGRAL   =', AJint(IDOM)
 3000 CONTINUE
C
      RETURN
      END
C===============================================================================


      SUBROUTINE CJVOL(IPLANE,NDF,NDOM,MXDOM,MXEDOM,NEIND,IEIND,
     *                 NPOIN,MXPOI,INC,MXSTATE,MXELE,IETIP,INTMAT9,
     *                 COORD,PT,TEMP,NODEK1new,CANGLE,ROR,PROP,AJvol,
     *                 IDOMTY,IREDSEL,IEMAT,MXMAT)
C===============================================================================
C     THIS SUBROUTINE CALCULATES J-INTEGRAL PARAMETER ACCORDING TO VOLUMETRIC
C     STRESS EXPRESSION USING DOMAIN INTEGRAL METHOD.
C===============================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COORD(MXPOI,2),PT(MXPOI*2),ROR(MXDOM,4)
      DIMENSION TEMP(MXPOI,MXSTATE),PROP(MXMAT,7)
      DIMENSION EDISP(9*2),ETEMP(9),STRAIN(4)
      DIMENSION BMAT(4,9*2),EKBG(9*2,9*2)
      DIMENSION AJ(2,2),AJI(2,2),AJID(2,2)
      DIMENSION EUr(9),EQ1(9),EQ2(9)
      DIMENSION DNDA(9),DNDB(9),P(9,2),PJID(9,2)
      DIMENSION dUdXMat(2,2),dTdXMat(1,2)
      DIMENSION QMat(2,1),dQdXMat(2,2)
      DIMENSION X(9),Y(9),XG(9),YG(9),WG(9),Q(9)
      DIMENSION AJvol(MXDOM)
      INTEGER INTMAT9(MXELE,9),IETIP(MXELE),IEMAT(MXELE)
      INTEGER NEIND(MXDOM),IEIND(MXEDOM,MXDOM)

C     Number of equations.
      NEQ = NPOIN*NDF
C     Set matrix dimensions according to the analyzed problem.
      IF(IPLANE.EQ.3) THEN !Axisymmetric case.
      NDIM = 4
      Rtip = COORD(NODEK1new,1) !Compute crack tip radius from axis of rotation.
      ELSE !Plane stress or plane strain case.
      NDIM = 3
      ENDIF
C===============================================================================
C     Compute J-integral from volumetric stress expression.
C===============================================================================
C     Loop over each integrated domain.
      DO 100 IDOM = 1, NDOM
      AJvol(IDOM) = 0._8
C     Loop over each element in this integrated domain.
      DO 120 IED  = 1, NEIND(IDOM)
C     Read material properties of each element.
```

```
      ELAS   = PROP(IEMAT(IEIND(IED,IDOM)),1)
      PR     = PROP(IEMAT(IEIND(IED,IDOM)),2)
      YSTRSS = PROP(IEMAT(IEIND(IED,IDOM)),3)
      AHARD  = PROP(IEMAT(IEIND(IED,IDOM)),4)
      ALPHA  = PROP(IEMAT(IEIND(IED,IDOM)),5)
      COTHR  = PROP(IEMAT(IEIND(IED,IDOM)),6)
C     9-node rectangular crack tip element.
      IF(IETIP(IEIND(IED,IDOM)).EQ.1) THEN
      NN   =  9         !Number of nodes.
      NMN  =  4         !Number of main (corner) nodes.
      NDOF =  NDF*NN    !Number of element degrees of freedom.
C      Set number of Gauss's points, its coordinates and weights.
       IF(IREDSEL.NE.1) THEN
       NG    =  9
       XG(1) = -DSQRT(3._8)/DSQRT(5._8)
       XG(2) =  XG(1)
       XG(3) =  XG(1)
       XG(4) =  0._8
       XG(5) =  XG(4)
       XG(6) =  XG(4)
       XG(7) = -XG(1)
       XG(8) =  XG(7)
       XG(9) =  XG(7)
C
       YG(1) =  XG(1)
       YG(2) =  XG(4)
       YG(3) =  XG(7)
       YG(4) =  XG(1)
       YG(5) =  XG(4)
       YG(6) =  XG(7)
       YG(7) =  XG(1)
       YG(8) =  XG(4)
       YG(9) =  XG(7)
C
       WG(1) =  25._8/81._8
       WG(2) =  40._8/81._8
       WG(3) =  WG(1)
       WG(4) =  WG(2)
       WG(5) =  64._8/81._8
       WG(6) =  WG(2)
       WG(7) =  WG(1)
       WG(8) =  WG(2)
       WG(9) =  WG(1)
       ELSE
       NG    =  4
       XG(1) = -1._8/DSQRT(3._8)
       XG(2) = -1._8/DSQRT(3._8)
       XG(3) =  1._8/DSQRT(3._8)
       XG(4) =  1._8/DSQRT(3._8)
C
       YG(1) = -1._8/DSQRT(3._8)
       YG(2) =  1._8/DSQRT(3._8)
       YG(3) = -1._8/DSQRT(3._8)
       YG(4) =  1._8/DSQRT(3._8)
C
       WG(1) =  1._8
       WG(2) =  1._8
       WG(3) =  1._8
       WG(4) =  1._8
       ENDIF
C     6-node triangular element.
      ELSE
      NN   =  6         !Number of nodes.
      NMN  =  3         !Number of main (corner) nodes.
      NDOF =  NDF*NN    !Number of element degrees of freedom.
C      Set number of Gauss's points, its coordinates and weights.
       IF(IREDSEL.NE.1) THEN
       NG    =  7
       XG(1) =  1._8/3._8
       XG(2) =  0.101286507323456_8
       XG(3) =  0.797426985353087_8
       XG(4) =  XG(2)
       XG(5) =  0.470142064105115_8
       XG(6) =  0.059715871789770_8
       XG(7) =  XG(5)
C
       YG(1) =  XG(1)
```

```
        YG(2) =  XG(2)
        YG(3) =  XG(2)
        YG(4) =  XG(3)
        YG(5) =  XG(5)
        YG(6) =  XG(5)
        YG(7) =  XG(6)
C
        WG(1) =  0.225_8
        WG(2) =  0.125939180544827_8
        WG(3) =  WG(2)
        WG(4) =  WG(2)
        WG(5) =  0.132394152788506_8
        WG(6) =  WG(5)
        WG(7) =  WG(5)
C
        WG(1) =  WG(1)/2._8
        WG(2) =  WG(2)/2._8
        WG(3) =  WG(3)/2._8
        WG(4) =  WG(4)/2._8
        WG(5) =  WG(5)/2._8
        WG(6) =  WG(6)/2._8
        WG(7) =  WG(7)/2._8
        ELSE
        NG    =  3
        XG(1) =  1._8/6._8
        XG(2) =  2._8/3._8
        XG(3) =  1._8/6._8
C
        YG(1) =  1._8/6._8
        YG(2) =  1._8/6._8
        YG(3) =  2._8/3._8
C
        WG(1) =  1._8/3._8
        WG(2) =  1._8/3._8
        WG(3) =  1._8/3._8
C
        WG(1) =  WG(1)/2._8
        WG(2) =  WG(2)/2._8
        WG(3) =  WG(3)/2._8
        ENDIF
       ENDIF
C      Compute the magnitude of shift function.
       DO 135 I = 1, NN
       II = INTMAT9(IEIND(IED,IDOM),I)
       DX = COORD(II,1)-COORD(NODEK1new,1)
       DY = COORD(II,2)-COORD(NODEK1new,2)
       IF(IDOMTY.EQ.1) THEN  !Square domain.
        AR = MAX(DABS(DX),DABS(DY))/ROR(IDOM,1)
       ENDIF
C
       IF(IDOMTY.EQ.2) THEN  !Circular domain.
        AR = DSQRT(DX*DX+DY*DY)/ROR(IDOM,1)
       ENDIF
C
       IF(IDOMTY.EQ.3) THEN  !Rectangular domain.
        IF(DX.LE.0._8) THEN
         IF(ROR(IDOM,1).GT.0._8) THEN
          AX = -DX/ROR(IDOM,1)
         ELSE
          AX = -1._8
         ENDIF
        ELSE
         IF(ROR(IDOM,3).GT.0._8) THEN
          AX =  DX/ROR(IDOM,3)
         ELSE
          AX = -1._8
         ENDIF
        ENDIF
C
        IF(DY.LE.0._8) THEN
         IF(ROR(IDOM,4).GT.0._8) THEN
          AY = -DY/ROR(IDOM,4)
         ELSE
          AY = -1._8
         ENDIF
        ELSE
         IF(ROR(IDOM,2).GT.0._8) THEN
```

```
          AY =  DY/ROR(IDOM,2)
         ELSE
          AY = -1._8
         ENDIF
        ENDIF
        AR = MAX(AX,AY)
       ENDIF
C
       Q(I) = 1._8 - AR
       IF(Q(I).LE.1.0E-12_8) Q(I) = 0._8
   135 CONTINUE
C      Make the magnitude of shift function zero at circular domain boundary.
       IF(IDOMTY.EQ.2) THEN
       DO 145 I = 1, NMN
       I2ND = I + 1
       IF(I.EQ.NMN) I2ND = 1
       IF(Q(I).EQ.0._8 .AND. Q(I2ND).EQ.0._8) Q(I+NMN) = 0._8
   145 CONTINUE
       ENDIF
C      Create element nodal quantities.
       DO 130 I = 1, NN
       II           = INTMAT9(IEIND(IED,IDOM),I)
       X(I)         = COORD(II,1)
       Y(I)         = COORD(II,2)
       EQ1(I)       = Q(I)*DCOSD(CANGLE)
       EQ2(I)       = Q(I)*DSIND(CANGLE)
       EUr(I)       = PT(2*II-1)
       ETEMP(I)     = TEMP(II,INC+1)
       EDISP(2*I-1) = PT(2*II-1)
       EDISP(2*I)   = PT(2*II)
   130 CONTINUE
C================================================================================
C      Loop over each Gauss's point on this element.
C================================================================================
       DO 1000 K = 1, NG
       A = XG(K)
       B = YG(K)
       CALL BJ9(X,Y,A,B,BMAT,EKBG,AJ,AJI,DETJAC,DNDA,DNDB,
      *        NN,NDF,IPLANE)
C      Compute all Gauss's point quantities.
       CALL GVALUE(A,B,EQ1,Q1G,NN)
       CALL GVALUE(A,B,EQ2,Q2G,NN)
       CALL GVALUE(A,B,ETEMP,T,NN)
C      For axisymmetric case.
       IF(IPLANE.EQ.3) THEN
       CALL GVALUE(A,B,X,RG,NN)
       CALL GVALUE(A,B,EUr,UrG,NN)
       ENDIF
C      Set initial strain vector to zero.
       DO 200 I  = 1, NDIM
       STRAIN(I) = 0._8
   200 CONTINUE
C      Compute strain vector.
       DO 210 I  = 1, NDIM
       DO 210 J  = 1, NDOF
       STRAIN(I) = STRAIN(I) + BMAT(I,J)*EDISP(J)
   210 CONTINUE
C      Find effective stress.
       CALL FINDSTRSS(EFSTRSS,STRAIN,YSTRSS,AHARD,ALPHA,PR,
      *              ELAS,AMOD,EFRATIO,IPLANE,T,COTHR,BETA)
C      Plane stress case.
       IF(IPLANE.EQ.1) THEN
       COND = BETA*ELAS/(1._8-2._8*PR)/(1._8+2._8*BETA)
       SGMM = COND*(STRAIN(1)+STRAIN(2)-2._8*COTHR*T)
       ENDIF
C      Plane strain case.
       IF(IPLANE.EQ.2) THEN
       SGMM = ELAS/(3._8*(1._8-2._8*PR))*
      *       (STRAIN(1)+STRAIN(2)-3._8*COTHR*T)
       ENDIF
C      Axisymmetric case.
       IF(IPLANE.EQ.3)THEN
       SGMM = ELAS/(3._8*(1._8-2._8*PR))*
      *       (STRAIN(1)+STRAIN(2)+STRAIN(4)-3._8*COTHR*T)
       ENDIF
C      Compute strain energy density at each Gauss's point.
       Wvol = (1.5_8-3._8*PR)/ELAS*SGMM*SGMM
```

```
C     Form shape function derivative matrix.
      DO 140 I = 1, NN
      P(I,1) = DNDA(I)
      P(I,2) = DNDB(I)
  140 CONTINUE
C     Form inverse Jacobian's matrix in deLorenzi's form.
      AJID(1,1) = AJI(1,1)
      AJID(1,2) = AJI(2,1)
      AJID(2,1) = AJI(1,2)
      AJID(2,2) = AJI(2,2)
C     [PJID] = [P][AJID].
      DO 150 I  = 1, NN
      DO 150 J  = 1, NDF
      PJID(I,J) = P(I,1)*AJID(1,J) + P(I,2)*AJID(2,J)
  150 CONTINUE
C     Set initial matrices to zeros.
      DO 160 I = 1, NDF
      DO 170 J = 1, NDF
      dUdXMat(I,J) = 0._8
      dQdXMat(I,J) = 0._8
  170 CONTINUE
      dTdXMat(1,I) = 0._8
  160 CONTINUE
C     Compute all derivative matrices at each Gauss's point.
      DO 180 I = 1, NN
C     Form displacement derivative matrix.
      dUdXMat(1,1) = dUdXMat(1,1) + EDISP(2*I-1)*PJID(I,1)
      dUdXMat(1,2) = dUdXMat(1,2) + EDISP(2*I-1)*PJID(I,2)
      dUdXMat(2,1) = dUdXMat(2,1) + EDISP(2*I  )*PJID(I,1)
      dUdXMat(2,2) = dUdXMat(2,2) + EDISP(2*I  )*PJID(I,2)
C     Form shift function derivative matrix.
      dQdXMat(1,1) = dQdXMat(1,1) + EQ1(I)*PJID(I,1)
      dQdXMat(1,2) = dQdXMat(1,2) + EQ1(I)*PJID(I,2)
      dQdXMat(2,1) = dQdXMat(2,1) + EQ2(I)*PJID(I,1)
      dQdXMat(2,2) = dQdXMat(2,2) + EQ2(I)*PJID(I,2)
C     Form temperature derivative matrix.
      dTdXMat(1,1) = dTdXMat(1,1) + ETEMP(I)*PJID(I,1)
      dTdXMat(1,2) = dTdXMat(1,2) + ETEMP(I)*PJID(I,2)
  180 CONTINUE
C     Form shift function matrix.
      QMat(1,1) = Q1G
      QMat(2,1) = Q2G
C     Compute J-integral in volumetric stress expression.
      AJvolSUM = 0._8
      DO 190 I = 1, NDF
      AJvolSUM = AJvolSUM - Wvol*dQdXMat(I,I)
     *                    + 3._8*COTHR*SGMM*dTdXMat(1,I)*QMat(I,1)
      DO 190 J = 1, NDF
      AJvolSUM = AJvolSUM + SGMM*dUdXMat(I,J)*dQdXMat(J,I)
  190 CONTINUE
C     Check problem case.
      IF(IPLANE.EQ.3) THEN  !Axisymmetric case.
      AJvol(IDOM) = AJvol(IDOM) + WG(K)*AJvolSUM*RG*DETJAC +
     *              WG(K)*(SGMM*UrG/RG-Wvol)*QMat(1,1)*DETJAC
      ELSE                  !Plane stress or plane strain case.
      AJvol(IDOM) = AJvol(IDOM) + WG(K)*AJvolSUM*DETJAC
      ENDIF
C     End each Gauss's point.
 1000 CONTINUE
C     End each element in the integrated domain.
  120 CONTINUE
      IF(IPLANE.EQ.3)THEN
      AJvol(IDOM) = AJvol(IDOM)/Rtip
      ENDIF
C     End each integrated domain.
  100 CONTINUE
C
      RETURN
      END
C================================================================================


      SUBROUTINE CJDEV(IPLANE,NDF,NDOM,MXDOM,MXEDOM,NEIND,IEIND,
     *                 NPOIN,MXPOI,INC,MXSTATE,MXELE,IETIP,INTMAT9,
     *                 COORD,PT,TEMP,BDf,NODEK1new,CANGLE,ROR,PROP,
     *                 AJdev,IDOMTY,IEMAT,MXMAT)
C================================================================================
```

```
C     THIS SUBROUTINE CALCULATES J-INTEGRAL PARAMETER ACCORDING TO DEVIATORIC
C     STRESS EXPRESSION USING DOMAIN INTEGRAL METHOD.
C===============================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COORD(MXPOI,2),PT(MXPOI*2),ROR(MXDOM,4)
      DIMENSION TEMP(MXPOI,MXSTATE),BDf(MXPOI*2,MXSTATE)
      DIMENSION PROP(MXMAT,7)
      DIMENSION EDISP(9*2),ETEMP(9),STRAIN(4),STRAIN0(4)
      DIMENSION BMAT(4,9*2),EKBG(9*2,9*2)
      DIMENSION AJ(2,2),AJI(2,2),AJID(2,2)
      DIMENSION EUr(9),EBDf1(9),EBDf2(9),EQ1(9),EQ2(9)
      DIMENSION DNDA(9),DNDB(9),P(9,2),PJID(9,2)
      DIMENSION GMAT(4,4),HMAT(4,4)
      DIMENSION SMat(2,2),dUdXMat(2,2)
      DIMENSION QMat(2,1),dQdXMat(2,2),fMat(2,1)
      DIMENSION X(9),Y(9),XG(9),YG(9),WG(9),Q(9)
      DIMENSION AJdev(MXDOM)
      INTEGER INTMAT9(MXELE,9),IETIP(MXELE),IEMAT(MXELE)
      INTEGER NEIND(MXDOM),IEIND(MXEDOM,MXDOM)

C     Number of equations.
      NEQ = NPOIN*NDF
C     Set matrix dimensions according to the analyzed problem.
      IF(IPLANE.EQ.3) THEN !Axisymmetric case.
      NDIM = 4
      Rtip = COORD(NODEK1new,1) !Compute crack tip radius from axis of rotation.
      ELSE !Plane stress or plane strain case.
      NDIM = 3
      ENDIF
C===============================================================================
C     Compute J-integral from deviatoric stress expression.
C===============================================================================
C     Loop over each integrated domain.
      DO 100 IDOM = 1, NDOM
      AJdev(IDOM) = 0._8
C     Loop each element in integrated domain.
      DO 120 IED  = 1, NEIND(IDOM)
C     Read material properties of each element.
      ELAS   = PROP(IEMAT(IEIND(IED,IDOM)),1)
      PR     = PROP(IEMAT(IEIND(IED,IDOM)),2)
      YSTRSS = PROP(IEMAT(IEIND(IED,IDOM)),3)
      AHARD  = PROP(IEMAT(IEIND(IED,IDOM)),4)
      ALPHA  = PROP(IEMAT(IEIND(IED,IDOM)),5)
      COTHR  = PROP(IEMAT(IEIND(IED,IDOM)),6)
C     9-node rectangular crack tip element.
      IF(IETIP(IEIND(IED,IDOM)).EQ.1) THEN
      NN    =  9        !Number of nodes.
      NG    =  9        !Number of Gauss's points.
      NMN   =  4        !Number of main (corner) nodes.
      NDOF  =  NDF*NN   !Number of element degrees of freedom.
C     X coordinate at each Gauss's point for this element.
      XG(1) = -DSQRT(3._8)/DSQRT(5._8)
      XG(2) =  XG(1)
      XG(3) =  XG(1)
      XG(4) =  0._8
      XG(5) =  XG(4)
      XG(6) =  XG(4)
      XG(7) = -XG(1)
      XG(8) =  XG(7)
      XG(9) =  XG(7)
C     Y coordinate at each Gauss's point for this element.
      YG(1) =  XG(1)
      YG(2) =  XG(4)
      YG(3) =  XG(7)
      YG(4) =  XG(1)
      YG(5) =  XG(4)
      YG(6) =  XG(7)
      YG(7) =  XG(1)
      YG(8) =  XG(4)
      YG(9) =  XG(7)
C     Gauss's point weights.
      WG(1) =  25._8/81._8
      WG(2) =  40._8/81._8
      WG(3) =  WG(1)
      WG(4) =  WG(2)
      WG(5) =  64._8/81._8
      WG(6) =  WG(2)
```

```
      WG(7) =  WG(1)
      WG(8) =  WG(2)
      WG(9) =  WG(1)
C     6-node triangular element.
      ELSE
      NN    =  6        !Number of nodes.
      NG    =  7        !Number of Gauss's points.
      NMN   =  3        !Number of main (corner) nodes.
      NDOF  =  NDF*NN    !Number of element degrees of freedom.
C     X coordinate at each Gauss's point for this element.
      XG(1) =  1._8/3._8
      XG(2) =  0.101286507323456_8
      XG(3) =  0.797426985353087_8
      XG(4) =  XG(2)
      XG(5) =  0.470142064105115_8
      XG(6) =  0.059715871789770_8
      XG(7) =  XG(5)
C     Y coordinate at each Gauss's point for this element.
      YG(1) =  XG(1)
      YG(2) =  XG(2)
      YG(3) =  XG(2)
      YG(4) =  XG(3)
      YG(5) =  XG(5)
      YG(6) =  XG(5)
      YG(7) =  XG(6)
C     Gauss's point weights.
      WG(1) =  0.225_8
      WG(2) =  0.125939180544827_8
      WG(3) =  WG(2)
      WG(4) =  WG(2)
      WG(5) =  0.132394152788506_8
      WG(6) =  WG(5)
      WG(7) =  WG(5)
C     Area of a triangle must be devided by two.
      WG(1) =  WG(1)/2._8
      WG(2) =  WG(2)/2._8
      WG(3) =  WG(3)/2._8
      WG(4) =  WG(4)/2._8
      WG(5) =  WG(5)/2._8
      WG(6) =  WG(6)/2._8
      WG(7) =  WG(7)/2._8
      ENDIF
C     Compute the magnitude of shift function.
      DO 135 I = 1, NN
      II = INTMAT9(IEIND(IED,IDOM),I)
      DX = COORD(II,1)-COORD(NODEK1new,1)
      DY = COORD(II,2)-COORD(NODEK1new,2)
      IF(IDOMTY.EQ.1) THEN  !Square domain.
       AR = MAX(DABS(DX),DABS(DY))/ROR(IDOM,1)
      ENDIF
C
      IF(IDOMTY.EQ.2) THEN  !Circular domain.
       AR = DSQRT(DX*DX+DY*DY)/ROR(IDOM,1)
      ENDIF
C
      IF(IDOMTY.EQ.3) THEN  !Rectangular domain.
       IF(DX.LE.0._8) THEN
        IF(ROR(IDOM,1).GT.0._8) THEN
         AX = -DX/ROR(IDOM,1)
        ELSE
         AX = -1._8
        ENDIF
       ELSE
        IF(ROR(IDOM,3).GT.0._8) THEN
         AX =  DX/ROR(IDOM,3)
        ELSE
         AX = -1._8
        ENDIF
       ENDIF
C
       IF(DY.LE.0._8) THEN
        IF(ROR(IDOM,4).GT.0._8) THEN
         AY = -DY/ROR(IDOM,4)
        ELSE
         AY = -1._8
        ENDIF
       ELSE
```

```
        IF(ROR(IDOM,2).GT.0._8) THEN
         AY =  DY/ROR(IDOM,2)
        ELSE
         AY = -1._8
        ENDIF
       ENDIF
       AR = MAX(AX,AY)
      ENDIF
C
      Q(I) = 1._8 - AR
      IF(Q(I).LE.1.0E-12_8) Q(I) = 0._8
  135 CONTINUE
C     Make the magnitude of shift function zero at circular domain boundary.
      IF(IDOMTY.EQ.2) THEN
      DO 145 I = 1, NMN
      I2ND = I + 1
      IF(I.EQ.NMN) I2ND = 1
      IF(Q(I).EQ.0._8 .AND. Q(I2ND).EQ.0._8) Q(I+NMN) = 0._8
  145 CONTINUE
      ENDIF
C     Create element nodal quantities.
      DO 130 I = 1, NN
      II           = INTMAT9(IEIND(IED,IDOM),I)
      X(I)         = COORD(II,1)
      Y(I)         = COORD(II,2)
      EQ1(I)       = Q(I)*DCOSD(CANGLE)
      EQ2(I)       = Q(I)*DSIND(CANGLE)
      EUr(I)       = PT(2*II-1)
      ETEMP(I)     = TEMP(II,INC+1)
      EBDf1(I)     = BDf(2*II-1,INC+1)
      EBDf2(I)     = BDf(2*II  ,INC+1)
      EDISP(2*I-1) = PT(2*II-1)
      EDISP(2*I)   = PT(2*II)
  130 CONTINUE
C================================================================================
C     Loop over each Gauss's point on this element.
C================================================================================
      DO 1000 K = 1, NG
      A = XG(K)
      B = YG(K)
      CALL BJ9(X,Y,A,B,BMAT,EKBG,AJ,AJI,DETJAC,DNDA,DNDB,
     *         NN,NDF,IPLANE)
C     Compute all Gauss's point quantities.
      CALL GVALUE(A,B,EQ1,Q1G,NN)
      CALL GVALUE(A,B,EQ2,Q2G,NN)
      CALL GVALUE(A,B,ETEMP,T,NN)
      CALL GVALUE(A,B,EBDf1,BDf1G,NN)
      CALL GVALUE(A,B,EBDf2,BDf2G,NN)
C     For axisymmetric case.
      IF(IPLANE.EQ.3) THEN
      CALL GVALUE(A,B,X,RG,NN)
      CALL GVALUE(A,B,EUr,UrG,NN)
      ENDIF
C     Set initial strain vector to zero.
      DO 200 I  = 1, NDIM
      STRAIN(I) = 0._8
  200 CONTINUE
C     Compute strain vector.
      DO 210 I  = 1, NDIM
      DO 210 J  = 1, NDOF
      STRAIN(I) = STRAIN(I) + BMAT(I,J)*EDISP(J)
  210 CONTINUE
C     Compute thermal strain vector.
      IF(IPLANE.EQ.1) THEN  !Plane stress case.
      STRAIN0(1) = COTHR*T
      STRAIN0(2) = COTHR*T
      STRAIN0(3) = 0._8
      ENDIF
C================================================================================
C     Find effective stress.
C================================================================================
      CALL FINDSTRSS(EFSTRSS,STRAIN,YSTRSS,AHARD,ALPHA,PR,
     *               ELAS,AMOD,EFRATIO,IPLANE,T,COTHR,BETA)
C================================================================================
C     Plane stress case.
C================================================================================
C     {STRESSdev} = [GMAT]{STRAIN} - [HMAT]{STRAIN0}
```

```
C===============================================================================
      IF(IPLANE.EQ.1) THEN
      COND = BETA*ELAS/(1._8-2._8*PR)/(1._8+2._8*BETA)
      GMAT(1,1) =  COND*(1._8+BETA)
      GMAT(1,2) = -COND*BETA
      GMAT(1,3) =  0._8
      GMAT(2,1) =  GMAT(1,2)
      GMAT(2,2) =  GMAT(1,1)
      GMAT(2,3) =  0._8
      GMAT(3,1) =  0._8
      GMAT(3,2) =  0._8
      GMAT(3,3) =  COND*(0.5_8+BETA)
C
      HMAT(1,1) =  COND
      HMAT(1,2) =  0._8
      HMAT(1,3) =  0._8
      HMAT(2,1) =  0._8
      HMAT(2,2) =  HMAT(1,1)
      HMAT(2,3) =  0._8
      HMAT(3,1) =  0._8
      HMAT(3,2) =  0._8
      HMAT(3,3) =  0._8
      ENDIF
C===============================================================================
C     Plane strain case.
C===============================================================================
C     {STRESSdev} = [GMAT]{STRAIN}
C===============================================================================
      IF(IPLANE.EQ.2) THEN
      GMAT(1,1) =  2._8/9._8*EFRATIO*2._8
      GMAT(1,2) = -2._8/9._8*EFRATIO
      GMAT(1,3) =  0._8
      GMAT(2,1) =  GMAT(1,2)
      GMAT(2,2) =  GMAT(1,1)
      GMAT(2,3) =  0._8
      GMAT(3,1) =  0._8
      GMAT(3,2) =  0._8
      GMAT(3,3) =  2._8/9._8*EFRATIO*1.5_8
      ENDIF
C===============================================================================
C     Axisymmetric case.
C===============================================================================
C     {STRESSdev} = [GMAT]{STRAIN}
C===============================================================================
      IF(IPLANE.EQ.3)THEN
      GMAT(1,1) =  2._8/9._8*EFRATIO*2._8
      GMAT(1,2) = -2._8/9._8*EFRATIO
      GMAT(1,3) =  0._8
      GMAT(1,4) =  GMAT(1,2)
      GMAT(2,1) =  GMAT(1,2)
      GMAT(2,2) =  GMAT(1,1)
      GMAT(2,3) =  0._8
      GMAT(2,4) =  GMAT(1,2)
      GMAT(3,1) =  GMAT(1,3)
      GMAT(3,2) =  GMAT(2,3)
      GMAT(3,3) =  2._8/9._8*EFRATIO*1.5_8
      GMAT(3,4) =  0._8
      GMAT(4,1) =  GMAT(1,4)
      GMAT(4,2) =  GMAT(2,4)
      GMAT(4,3) =  GMAT(3,4)
      GMAT(4,4) =  GMAT(1,1)
      ENDIF
C     Set stresses to be zeros before summing.
      SGXX = 0._8
      SGYY = 0._8
      SGXY = 0._8
C     Compute deviatoric stresses.
      IF(IPLANE.EQ.1) THEN   !Plane stress case.
      DO 220 I = 1, NDIM
      SGXX = SGXX + GMAT(1,I)*STRAIN(I) - HMAT(1,I)*STRAIN0(I)
      SGYY = SGYY + GMAT(2,I)*STRAIN(I) - HMAT(2,I)*STRAIN0(I)
      SGXY = SGXY + GMAT(3,I)*STRAIN(I) - HMAT(3,I)*STRAIN0(I)
  220 CONTINUE
      ELSE   !Plane strain or axisymmetric case.
      DO 230 I = 1, NDIM
      SGXX = SGXX + GMAT(1,I)*STRAIN(I)
      SGYY = SGYY + GMAT(2,I)*STRAIN(I)
```

```
      SGXY = SGXY + GMAT(3,I)*STRAIN(I)
  230 CONTINUE
      ENDIF
C     Compute deviatoric stress in z direction.
      IF(IPLANE.EQ.3) THEN  !Axisymmetric case.
      SGZZ = 2._8/9._8*EFRATIO*( 2._8*STRAIN(4)-STRAIN(1)-STRAIN(2) )
      ENDIF
C     Compute strain energy density at each Gauss's point.
      Wdev = (1._8+PR)/ELAS/3._8*EFSTRSS*EFSTRSS +
     *       AHARD/(AHARD+1._8)*ALPHA*YSTRSS*YSTRSS/ELAS*
     *       (EFSTRSS/YSTRSS)**(AHARD+1._8)
C     Form shape function derivative matrix.
      DO 140 I = 1, NN
      P(I,1) = DNDA(I)
      P(I,2) = DNDB(I)
  140 CONTINUE
C     Form inverse Jacobian's matrix in deLorenzi's form.
      AJID(1,1) = AJI(1,1)
      AJID(1,2) = AJI(2,1)
      AJID(2,1) = AJI(1,2)
      AJID(2,2) = AJI(2,2)
C     [PJID] = [P][AJID].
      DO 150 I  = 1, NN
      DO 150 J  = 1, NDF
      PJID(I,J) = P(I,1)*AJID(1,J) + P(I,2)*AJID(2,J)
  150 CONTINUE
C     Set initial matrices to zeros.
      DO 160 I = 1, NDF
      DO 160 J = 1, NDF
      dUdXMat(I,J) = 0._8
      dQdXMat(I,J) = 0._8
  160 CONTINUE
C     Compute all derivative matrices at each Gauss's point.
      DO 180 I = 1, NN
C     Form displacement derivative matrix.
      dUdXMat(1,1) = dUdXMat(1,1) + EDISP(2*I-1)*PJID(I,1)
      dUdXMat(1,2) = dUdXMat(1,2) + EDISP(2*I-1)*PJID(I,2)
      dUdXMat(2,1) = dUdXMat(2,1) + EDISP(2*I  )*PJID(I,1)
      dUdXMat(2,2) = dUdXMat(2,2) + EDISP(2*I  )*PJID(I,2)
C     Form shift function derivative matrix.
      dQdXMat(1,1) = dQdXMat(1,1) + EQ1(I)*PJID(I,1)
      dQdXMat(1,2) = dQdXMat(1,2) + EQ1(I)*PJID(I,2)
      dQdXMat(2,1) = dQdXMat(2,1) + EQ2(I)*PJID(I,1)
      dQdXMat(2,2) = dQdXMat(2,2) + EQ2(I)*PJID(I,2)
  180 CONTINUE
C     Form stress matrix.
      SMat(1,1) = SGXX
      SMat(1,2) = SGXY
      SMat(2,1) = SMat(1,2)
      SMat(2,2) = SGYY
C     Form shift function matrix.
      QMat(1,1) = Q1G
      QMat(2,1) = Q2G
C     Form body force matrix.
      fMat(1,1) = BDf1G
      fMat(2,1) = BDf2G
C     Compute J-integral in deviatoric stress expression.
      AJdevSUM = 0._8
      DO 190 I = 1, NDF
      AJdevSUM = AJdevSUM - Wdev*dQdXMat(I,I)
      DO 190 J = 1, NDF
      AJdevSUM = AJdevSUM - fMat(I,1)*dUdXMat(I,J)*QMat(J,1)
      DO 190 L = 1, NDF
      AJdevSUM = AJdevSUM + SMat(I,J)*dUdXMat(J,L)*dQdXMat(L,I)
  190 CONTINUE
C     Check problem case.
      IF(IPLANE.EQ.3) THEN  !Axisymmetric case.
      AJdev(IDOM) = AJdev(IDOM) + WG(K)*AJdevSUM*RG*DETJAC +
     *              WG(K)*(SGZZ*UrG/RG-Wdev)*QMat(1,1)*DETJAC
      ELSE                  !Plane stress or plane strain case.
      AJdev(IDOM) = AJdev(IDOM) + WG(K)*AJdevSUM*DETJAC
      ENDIF
C     End each Gauss's point.
 1000 CONTINUE
C     End each element in the integrated domain.
  120 CONTINUE
      IF(IPLANE.EQ.3)THEN
```

```
      AJdev(IDOM) = AJdev(IDOM)/Rtip
      ENDIF
C     End each integrated domain.
  100 CONTINUE
C
      RETURN
      END
C================================================================================


      SUBROUTINE CJFACE(IPLANE,NDF,NDOM,MXDOM,MXCFNODE,IFACEN,
     *                  IFACEE,NFACEN,NFACE,IFACE,NPOIN,MXPOI,
     *                  MXELE,IETIP,INTMAT9,COORD,PT,NODEK1new,
     *                  CANGLE,ROR,PI,PROP,AJface,FISURF,CTQEX,
     *                  FDQEX,IDOMTY,IEMAT,MXMAT)
C================================================================================
C     THIS SUBROUTINE CALCULATES J-INTEGRAL PARAMETER ACCORDING TO CRACK FACE
C     EXPRESSION.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COORD(MXPOI,2),PT(MXPOI*2)
      DIMENSION PROP(MXMAT,7),ROR(MXDOM,4)
      DIMENSION X(9),Y(9),Q(9)
      DIMENSION BMAT(4,9*2),EKBG(9*2,9*2)
      DIMENSION AJ(2,2),AJI(2,2),AJID(2,2)
      DIMENSION EQ1(9),EQ2(9),EDISP(9*2)
      DIMENSION DNDA(9),DNDB(9),P(9,2),PJID(9,2)
      DIMENSION SNODE(3),ANF(3),DNDAF(3),Tx(3),Ty(3)
      DIMENSION CTQEX(4),FDQEX(4)
      DIMENSION dUdXMatF(2,2),QMatF(2,1),TMatF(2,1)
      DIMENSION XGF(5),WGF(5)
      DIMENSION ETFMat(3,3),STFMat(MXPOI*2,MXPOI*2),PIVOTF(MXPOI*2)
      DIMENSION CFLVec(MXPOI*2),TractN(2,MXPOI*2,2),FISURF(MXPOI*2)
      DIMENSION AJface(MXDOM)
      INTEGER INTMAT9(MXELE,9),IETIP(MXELE),IEMAT(MXELE)
      INTEGER IFACEN(2,MXCFNODE,MXDOM),NFACEN(2,MXDOM)
      INTEGER IFACEE(2,(MXCFNODE-1)/2,MXDOM)
      INTEGER INODE(3)

C     Number of equations.
      NEQ = NPOIN*NDF
C     Compute crack tip radius from axis of rotation.
      IF(IPLANE.EQ.3) THEN
      Rtip = COORD(NODEK1new,1)
      ENDIF
C================================================================================
C     Transform crack face nodal loads to nodal crack face tractions.
C     *In this subroutine, nodal crack face tractions are computed from nodal
C      loads along crack face elements within the last integrated domain which
C      is the biggest one. The subroutine also requires you to input the nodal
C      load contributions of the farthest crack face element nodes in the last
C      integrated domain for accurate crack face nodal load-to-traction
C      transformation.
C================================================================================
C     Number of Gauss's points for integration over a crack face.
      NGF    =  5
C     Gauss's point coordinates for one dimensional integration.
      XGF(1) =  0.0000000000_8
      XGF(2) =  0.5384693101_8
      XGF(3) = -0.5384693101_8
      XGF(4) =  0.9061798459_8
      XGF(5) = -0.9061798459_8
C     Its corresponding weights.
      WGF(1) =  0.5688888889_8
      WGF(2) =  0.4786286705_8
      WGF(3) =  0.4786286705_8
      WGF(4) =  0.2369268850_8
      WGF(5) =  0.2369268850_8
C     Loop over each crack face in the last integrated domain.
      DO 2100 ICFACE = 1, NFACE
      IF(NFACE.EQ.2) IFACE = ICFACE
C     Number of nodes on each crack face within the biggest domain.
      NFNODE = NFACEN(IFACE,NDOM)
C     Set system crack face traction-to-load transformation matrix to be zero.
      DO 2105  I  = 1, NFNODE
      DO 2105  J  = 1, NFNODE
      STFMat(I,J) = 0._8
```

```
 2105 CONTINUE
C     Loop over each crack face element.
      DO 2110 IEF = 1, (NFNODE-3)/2+1
C     Set element thickness.
      IF(IPLANE.NE.3) THICK = PROP(IEMAT(IFACEE(IFACE,IEF,NDOM)),7)
C     Find nodal numbers for this element.
      INODE(1) = IFACEN(IFACE,1+2*(IEF-1),NDOM)
      INODE(2) = IFACEN(IFACE,3+2*(IEF-1),NDOM)
      INODE(3) = IFACEN(IFACE,2+2*(IEF-1),NDOM)
C     Find crack face local coordinates for each node.
      SNODE(1) = 0._8
      SNODE(2) = DSQRT( (COORD(INODE(2),1)-COORD(INODE(1),1))**2._8 +
     *                  (COORD(INODE(2),2)-COORD(INODE(1),2))**2._8 )
      SNODE(3) = DSQRT( (COORD(INODE(3),1)-COORD(INODE(1),1))**2._8 +
     *                  (COORD(INODE(3),2)-COORD(INODE(1),2))**2._8 )
C     Set element crack face traction-to-load transformation matrix to zero.
      DO 2120  I  = 1, 3
      DO 2120  J  = 1, 3
      ETFMat(I,J) = 0._8
 2120 CONTINUE
C     Loop over each Gauss's point on a crack face element.
      DO 2130 IG = 1 ,NGF
      AF = XGF(IG)
      ANF(1)   =  0.5_8*AF*(AF-1._8)
      ANF(2)   =  0.5_8*AF*(AF+1._8)
      ANF(3)   =  1.0_8 - AF*AF
      DNDAF(1) =  AF - 0.5_8
      DNDAF(2) =  AF + 0.5_8
      DNDAF(3) = -2.0_8*AF
      DETJACF  =  DNDAF(1)*SNODE(1) +
     *            DNDAF(2)*SNODE(2) + DNDAF(3)*SNODE(3)
      IF(IPLANE.EQ.3) THEN  !Axisymmetric case.
      RGF = ANF(1)*COORD(INODE(1),1) + ANF(2)*COORD(INODE(2),1) +
     *      ANF(3)*COORD(INODE(3),1)
      ENDIF
C     Compute element crack face traction-to-load transformation matrix.
      DO 2140 I = 1, 3
      DO 2140 J = 1, 3
      IF(IPLANE.EQ.3) THEN
      ETFMat(I,J) = ETFMat(I,J) +
     *              WGF(IG)*ANF(I)*ANF(J)*2._8*PI*RGF*DETJACF
      ELSE
      ETFMat(I,J) = ETFMat(I,J) +
     *              WGF(IG)*ANF(I)*ANF(J)*THICK*DETJACF
      ENDIF
 2140 CONTINUE
C     End each crack face Gauss's point.
 2130 CONTINUE
C     Assemble element matrices into a system matrix.
      DO 2150 IER = 1, 3
      IF(IER.EQ.1) IR = 1
      IF(IER.EQ.2) IR = 3
      IF(IER.EQ.3) IR = 2
      ISR = 2*(IEF-1) + IR
      DO 2150 IEC = 1, 3
      IF(IEC.EQ.1) IC = 1
      IF(IEC.EQ.2) IC = 3
      IF(IEC.EQ.3) IC = 2
      ISC = 2*(IEF-1) + IC
      STFMat(ISR,ISC) = STFMat(ISR,ISC) + ETFMat(IER,IEC)
 2150 CONTINUE
C     End each crack face element.
 2110 CONTINUE
C     Apply Crout's factorization.
      CALL CROUT(STFMat,PIVOTF,NFNODE,MXPOI)
C     Loop over nodal crack face tractions in X and Y direction.
      DO 2160 I = 1, NDF
C     Form crack face nodal load vector in a direction.
      DO 2170 INF = 1, NFNODE
      CFLVec(INF) = FISURF(2*IFACEN(IFACE,INF,NDOM)+I-2)
C      The farthest node on C.W. crack face in the last integrated domain.
       IF(IFACE.EQ.1 .AND. INF.EQ.1) THEN
        CFLVec(INF) = FDQEX(2*(IFACE-1)+I)
       ENDIF
C      The crack tip node on C.W. crack face in the last integrated domain.
       IF(IFACE.EQ.1 .AND. INF.EQ.NFNODE) THEN
        CFLVec(INF) = CTQEX(2*(IFACE-1)+I)
```

```fortran
       ENDIF
C      The farthest node on C.C.W. crack face in the last integrated domain.
       IF(IFACE.EQ.2 .AND. INF.EQ.NFNODE) THEN
        CFLVec(INF) = FDQEX(2*(IFACE-1)+I)
       ENDIF
C      The crack tip node on C.C.W. crack face in the last integrated domain.
       IF(IFACE.EQ.2 .AND. INF.EQ.1) THEN
        CFLVec(INF) = CTQEX(2*(IFACE-1)+I)
       ENDIF
 2170 CONTINUE
C     Solve a set of simultaneous equations.
      CALL SOLVE(NFNODE,STFMat,PIVOTF,CFLVec,MXPOI)
      DO 2180 INF = 1, NFNODE
      TractN(IFACE,INF,I) = CFLVec(INF)
 2180 CONTINUE
C     End each direction.
 2160 CONTINUE
C     End each crack face in the last integrated domain.
 2100 CONTINUE
C================================================================================
C     Print nodal tractions on screen.
C================================================================================
      DO 2200 ICFACE = 1, NFACE
      IF(NFACE.EQ.2) IFACE = ICFACE
      WRITE(*,*)
      IF(IFACE.EQ.1) WRITE(*,*) '[CLOCKWISE FACE]'
      IF(IFACE.EQ.2) WRITE(*,*) '[COUNTER CLOCKWISE FACE]'
      WRITE(*,2211)
 2211 FORMAT(4X,'NO.',4X,'NODE',14X,'Tx',24X,'Ty')
      DO 2210 INF = 1, NFNODE
      WRITE(*,2212) INF, IFACEN(IFACE,INF,NDOM),
     *              TractN(IFACE,INF,1), TractN(IFACE,INF,2)
 2212 FORMAT(2X,I4,4X,I5,4X,E22.16,4X,E22.16)
 2210 CONTINUE
 2200 CONTINUE
C================================================================================
C     Compute J-integral from crack face expression.
C================================================================================
C     Loop over each integrated domain.
      DO 1100 IDOM = 1, NDOM
      AJface(IDOM) = 0._8
C     Loop over each crack face.
      DO 1200 ICFACE = 1, NFACE
      IF(NFACE.EQ.2) IFACE = ICFACE
C     Loop over each crack face element.
      DO 1300 IEF = 1, (NFACEN(IFACE,IDOM)-3)/2+1
C     Set nodal number along three crack face nodes of this element.
      INODE(1) = IFACEN(IFACE,2*(IEF-1)+1,IDOM)
      INODE(2) = IFACEN(IFACE,2*(IEF-1)+3,IDOM)
      INODE(3) = IFACEN(IFACE,2*(IEF-1)+2,IDOM)
C     Search crack face tractions at nodes on this crack face element from nodes
C     on the largest integrated domain.
      DO 1405 I = 1, 3
      DO 1405 J = 1, NFACEN(IFACE,NDOM)
      IF(INODE(I).EQ.IFACEN(IFACE,J,NDOM)) THEN
      Tx(I) = TractN(IFACE,J,1)
      Ty(I) = TractN(IFACE,J,2)
      ENDIF
 1405 CONTINUE
C     Find crack face element coordinate at each crack face node.
      SNODE(1) = 0._8
      SNODE(2) = DSQRT( (COORD(INODE(2),1)-COORD(INODE(1),1))**2._8 +
     *                  (COORD(INODE(2),2)-COORD(INODE(1),2))**2._8 )
      SNODE(3) = DSQRT( (COORD(INODE(3),1)-COORD(INODE(1),1))**2._8 +
     *                  (COORD(INODE(3),2)-COORD(INODE(1),2))**2._8 )
C     Set number of nodes and degrees of freedom for this crack face element.
      IF(IETIP(IFACEE(IFACE,IEF,IDOM)).EQ.1) THEN !9-node rectangular element.
      NN   = 9        !Number of nodes.
      NMN  = 4        !Number of main (corner) nodes.
      NDOF = NDF*NN   !Number of element degrees of freedom.
      ELSE            !6-node triangular element.
      NN   = 6        !Number of nodes.
      NMN  = 3        !Number of main (corner) nodes.
      NDOF = NDF*NN   !Number of element degrees of freedom.
      ENDIF
C     Compute the magnitude of shift function.
      DO 135 I = 1, NN
```

```
      II = INTMAT9(IFACEE(IFACE,IEF,IDOM),I)
      DX = COORD(II,1)-COORD(NODEK1new,1)
      DY = COORD(II,2)-COORD(NODEK1new,2)
      IF(IDOMTY.EQ.1) THEN  !Square domain.
       AR = MAX(DABS(DX),DABS(DY))/ROR(IDOM,1)
      ENDIF
C
      IF(IDOMTY.EQ.2) THEN  !Circular domain.
       AR = DSQRT(DX*DX+DY*DY)/ROR(IDOM,1)
      ENDIF
C
      IF(IDOMTY.EQ.3) THEN  !Rectangular domain.
       IF(DX.LE.0._8) THEN
        IF(ROR(IDOM,1).GT.0._8) THEN
         AX = -DX/ROR(IDOM,1)
        ELSE
         AX = -1._8
        ENDIF
       ELSE
        IF(ROR(IDOM,3).GT.0._8) THEN
         AX =  DX/ROR(IDOM,3)
        ELSE
         AX = -1._8
        ENDIF
       ENDIF
C
       IF(DY.LE.0._8) THEN
        IF(ROR(IDOM,4).GT.0._8) THEN
         AY = -DY/ROR(IDOM,4)
        ELSE
         AY = -1._8
        ENDIF
       ELSE
        IF(ROR(IDOM,2).GT.0._8) THEN
         AY =  DY/ROR(IDOM,2)
        ELSE
         AY = -1._8
        ENDIF
       ENDIF
       AR = MAX(AX,AY)
      ENDIF
C
      Q(I) = 1._8 - AR
      IF(Q(I).LE.1.0E-12_8) Q(I) = 0._8
  135 CONTINUE
C     Make the magnitude of shift function zero at circular domain boundary.
      IF(IDOMTY.EQ.2) THEN
      DO 145 I = 1, NMN
      I2ND = I + 1
      IF(I.EQ.NMN) I2ND = 1
      IF(Q(I).EQ.0._8 .AND. Q(I2ND).EQ.0._8) Q(I+NMN) = 0._8
  145 CONTINUE
      ENDIF
C     Create element nodal temperatures, coordinates, displacements and shift
C     functions.
      DO 1315 I = 1, NN
      II            = INTMAT9(IFACEE(IFACE,IEF,IDOM),I)
      X(I)          = COORD(II,1)
      Y(I)          = COORD(II,2)
      EQ1(I)        = Q(I)*DCOSD(CANGLE)
      EQ2(I)        = Q(I)*DSIND(CANGLE)
      EDISP(2*I-1) = PT(2*II-1)
      EDISP(2*I)   = PT(2*II)
 1315 CONTINUE
C     Loop over each Gauss's point along a crack face element.
      DO 2000 IG = 1, NGF
C     Shape functions and their derivatives of crack face local coordinates.
      AF = XGF(IG)
      ANF(1)   =  0.5_8*AF*(AF-1._8)
      ANF(2)   =  0.5_8*AF*(AF+1._8)
      ANF(3)   =  1.0_8 - AF*AF
      DNDAF(1) =  AF - 0.5_8
      DNDAF(2) =  AF + 0.5_8
      DNDAF(3) = -2.0_8*AF
      DETJACF  =  DNDAF(1)*SNODE(1) +
     *            DNDAF(2)*SNODE(2) + DNDAF(3)*SNODE(3)
      IF(IPLANE.EQ.3) THEN  !Axisymmetric case.
```

```
      RGF = ANF(1)*COORD(INODE(1),1) + ANF(2)*COORD(INODE(2),1) +
     *      ANF(3)*COORD(INODE(3),1)
      ENDIF
C     Compute crack face traction matrix at a Gauss's point.
      TMatF(1,1) = ANF(1)*Tx(1) + ANF(2)*Tx(2) + ANF(3)*Tx(3)
      TMatF(2,1) = ANF(1)*Ty(1) + ANF(2)*Ty(2) + ANF(3)*Ty(3)
C================================================================================
C     Find element coordinates in 2D from a Gauss's point on a crack face
C     coordinate in 1D and calculate all quantities according to this element
C     coordinates.
C================================================================================
      IF(IETIP(IFACEE(IFACE,IEF,IDOM)).EQ.1) THEN  !9-node rectangular element.
      NMN = 4     !Number of main (corner) nodes.
      DO 1410 I = 1, NMN
      IF(INTMAT9(IFACEE(IFACE,IEF,IDOM),I).EQ.INODE(1)) THEN
C     Crack face is the lower edge in a mapped rectangular element.
      IF(I.EQ.1) THEN
      A =  XGF(IG)
      B = -1._8
      ENDIF
C     Crack face is the right edge in a mapped rectangular element.
      IF(I.EQ.2) THEN
      A =  1._8
      B =  XGF(IG)
      ENDIF
C     Crack face is the upper edge in a mapped rectangular element.
      IF(I.EQ.3) THEN
      A = -XGF(IG)
      B =  1._8
      ENDIF
C     Crack face is the left edge in a mapped rectangular element.
      IF(I.EQ.4) THEN
      A = -1._8
      B = -XGF(IG)
      ENDIF
      GOTO 1450
      ENDIF
 1410 CONTINUE
      ELSE        !6-node triangular element.
      NMN = 3     !Number of main (corner) nodes.
      DO 1420 I = 1, NMN
      IF(INTMAT9(IFACEE(IFACE,IEF,IDOM),I).EQ.INODE(1)) THEN
C     Crack face is the lower edge in a mapped triangular element.
      IF(I.EQ.1) THEN
      A = 0.5_8*(XGF(IG)+1._8)
      B = 0._8
      ENDIF
C     Crack face is the right edge in a mapped triangular element.
      IF(I.EQ.2) THEN
      A = 0.5_8*(1._8-XGF(IG))
      B = 0.5_8*(XGF(IG)+1._8)
      ENDIF
C     Crack face is the left edge in a mapped triangular element.
      IF(I.EQ.3) THEN
      A = 0._8
      B = 0.5_8*(1._8-XGF(IG))
      ENDIF
      GOTO 1450
      ENDIF
 1420 CONTINUE
      ENDIF
C     Exit searching coordinate loop.
 1450 CONTINUE
C     Compute Gauss's point radius.
      IF(IPLANE.EQ.3) THEN
      CALL GVALUE(A,B,X,RGF,NN)
      ENDIF
C     Compute strain-displacement matrix.
      CALL BJ9(X,Y,A,B,BMAT,EKBG,AJ,AJI,DETJAC,DNDA,DNDB,
     *         NN,NDF,IPLANE)
C     Form shape function derivative matrix.
      DO 1500 I = 1, NN
      P(I,1) = DNDA(I)
      P(I,2) = DNDB(I)
 1500 CONTINUE
C     Form inverse Jacobian matrix according to deLorenzi's paper.
      AJID(1,1) = AJI(1,1)
```

```
      AJID(1,2) = AJI(2,1)
      AJID(2,1) = AJI(1,2)
      AJID(2,2) = AJI(2,2)
C     [PJID] = [P][AJID].
      DO 1510 I = 1, NN
      DO 1510 J = 1, NDF
      PJID(I,J) = P(I,1)*AJID(1,J) + P(I,2)*AJID(2,J)
 1510 CONTINUE
C     Set displacement derivative matrix to zero.
      DO 1520 I = 1, NDF
      DO 1520 J = 1, NDF
      dUdXMatF(I,J) = 0._8
 1520 CONTINUE
C     Form displacement derivative matrix.
      DO 1530 I = 1, NN
      dUdXMatF(1,1) = dUdXMatF(1,1) + EDISP(2*I-1)*PJID(I,1)
      dUdXMatF(1,2) = dUdXMatF(1,2) + EDISP(2*I-1)*PJID(I,2)
      dUdXMatF(2,1) = dUdXMatF(2,1) + EDISP(2*I  )*PJID(I,1)
      dUdXMatF(2,2) = dUdXMatF(2,2) + EDISP(2*I  )*PJID(I,2)
 1530 CONTINUE
C     Form shift function matrix.
      CALL GVALUE(A,B,EQ1,GQ1,NN)
      CALL GVALUE(A,B,EQ2,GQ2,NN)
      QMatF(1,1) = GQ1
      QMatF(2,1) = GQ2
C     Compute J-integral from crack face expression.
      AJfaceSUM = 0._8
      DO 1400 I = 1, NDF
      DO 1400 J = 1, NDF
      AJfaceSUM = AJfaceSUM + TMatF(I,1)*dUdXMatF(I,J)*QMatF(J,1)
 1400 CONTINUE
C     Check problem case.
      IF(IPLANE.EQ.3) THEN  !Axisymmetric case.
      AJface(IDOM) = AJface(IDOM) + WGF(IG)*AJfaceSUM*RGF*DETJACF
      ELSE                  !Plane stress or plane strain case.
      AJface(IDOM) = AJface(IDOM) + WGF(IG)*AJfaceSUM*DETJACF
      ENDIF
C     End each crack face Gauss's point.
 2000 CONTINUE
C     End each crack face element.
 1300 CONTINUE
C     End each crack face.
 1200 CONTINUE
      IF(IPLANE.EQ.3) THEN
      AJface(IDOM) = AJface(IDOM)/Rtip
      ENDIF
C     End each integrated domain.
 1100 CONTINUE
C
      RETURN
      END
C================================================================================


      SUBROUTINE GETSTRING(NUMint,NUMstr,MXFIG)
C================================================================================
C     THIS SUBROUTINE CHANGES INTEGER NUMBERS TO INTEGER STRINGS.
C================================================================================
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER NUMint,NUMfig(MXFIG)
      CHARACTER(1) STRfig(MXFIG)
      CHARACTER(MXFIG) NUMstr

C     Check whether number of figures exceed its maximum.
      IF(NUMint.GE.10**MXFIG) WRITE(*,1000)
 1000 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXFIG IN'
     *        ,' GETSTRING SUBROUTINE')
      IF(NUMint.GE.10**MXFIG) STOP
C     Check number of figures.
      DO 5 IFIG = MXFIG, 1, -1
      IF(NUMint.LT.10**IFIG) THEN
      NFIG = IFIG
      ELSE
      GOTO 15
      ENDIF
    5 CONTINUE
C     Find string of each figure.
```

```
   15 NUMBER = NUMint
      DO 10 IFIG = NFIG, 1, -1
      NUMfig(IFIG) = INT(NUMBER/10**(IFIG-1))
      NUMBER = NUMBER - NUMfig(IFIG)*10**(IFIG-1)
      SELECT CASE(NUMfig(IFIG))
      CASE(0)
      STRfig(IFIG) = '0'
      CASE(1)
      STRfig(IFIG) = '1'
      CASE(2)
      STRfig(IFIG) = '2'
      CASE(3)
      STRfig(IFIG) = '3'
      CASE(4)
      STRfig(IFIG) = '4'
      CASE(5)
      STRfig(IFIG) = '5'
      CASE(6)
      STRfig(IFIG) = '6'
      CASE(7)
      STRfig(IFIG) = '7'
      CASE(8)
      STRfig(IFIG) = '8'
      CASE(9)
      STRfig(IFIG) = '9'
      END SELECT
   10 CONTINUE
C     Combine all strings.
      NUMstr = ''
      NUMstr(1:1) = STRfig(NFIG)
      DO 20 IFIG  = NFIG-1, 1, -1
      LEN = LEN_TRIM(NUMstr)
      NUMstr(1:LEN+1) = NUMstr(1:LEN)//STRfig(IFIG)
   20 CONTINUE
C
      RETURN
      END
C===============================================================================


      SUBROUTINE MAPPING(MXPOI,MXELE,MXCTE,NDF,IBC,QEX,NPOIN,NELEMp,
     *                   PT,PTp,COORD,COORDp,INTMAT9p,IETIPp,ICTN,
     *                   ICTNp,NCTN,RTHETA,RTHETAp,SXX,SXXp,SXY,SXYp,
     *                   SYY,SYYp,SVM,SVMp,IOCTEp,NCTELEp,ICTETRAN,
     *                   THETA0p,ICTEBNp)
C===============================================================================
C     THIS SUBROUTINE MAPS ALL NODAL DISPLACEMETS FROM THE OLD MESH INTO THE NEW
C     REFINED ONE.
C===============================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION QEX(MXPOI*2),AL(3),ALmin(3),ALRec(4)
      DIMENSION COORD(MXPOI,2),COORDp(MXPOI,2)
      DIMENSION PT(MXPOI*2),PTp(MXPOI*2)
      DIMENSION SXX(MXPOI),SXXp(MXPOI)
      DIMENSION SXY(MXPOI),SXYp(MXPOI)
      DIMENSION SYY(MXPOI),SYYp(MXPOI)
      DIMENSION SVM(MXPOI),SVMp(MXPOI)
      DIMENSION RTHETA(MXCTE*2+1),RTHETAp(MXCTE*2+1)
      DIMENSION Xp(9),Yp(9),EUp(9),EVp(9)
      DIMENSION ESXXp(9),ESXYp(9),ESYYp(9),ESVMp(9)
      INTEGER IBC(MXPOI*2),INTMAT9p(MXELE,9)
      INTEGER IETIPp(MXELE),IOCTEp(MXCTE)
      INTEGER ICTN(MXCTE*2+1),ICTNp(MXCTE*2+1),ICTEBNp(MXCTE*2+1)

C===============================================================================
C     Loop over all nodes in new refined mesh model except its crack tip nodes
C     to find what element in the old mesh model this node is on and map
C     displacement solutions to this new mesh node.
C===============================================================================
      DO 100 IP = 1, NPOIN
C      Check that this node is not one of 9-node element crack tip nodes.
       IF(ICTETRAN.EQ.1) THEN
       DO 110 I = 1, NCTN
        IF(IP.EQ.ICTN(I)) GOTO 100
  110  CONTINUE
       ENDIF
C      Find its coordinates.
```

```
      X = COORD(IP,1)
      Y = COORD(IP,2)
C     Search all old mesh elements.
      DO 120 IEp = 1, NELEMp
C      9-node rectangular crack tip element.
       IF(IETIPp(IEp).EQ.1) THEN
        NMNp = 4
        DO 130 I = 1, NMNp
         Xp(I) = COORDp(INTMAT9p(IEp,I),1)
         Yp(I) = COORDp(INTMAT9p(IEp,I),2)
  130   CONTINUE
        Atot1 = 0.5_8*( Xp(2)*Yp(3)-Xp(3)*Yp(2)+
     *               (Yp(2)-Yp(3))*Xp(1)+(Xp(3)-Xp(2))*Yp(1) )
        Atot2 = 0.5_8*( Xp(3)*Yp(4)-Xp(4)*Yp(3)+
     *               (Yp(3)-Yp(4))*Xp(1)+(Xp(4)-Xp(3))*Yp(1) )
        ARec1 = 0.5_8*( Xp(2)*Yp(3)-Xp(3)*Yp(2)+
     *               (Yp(2)-Yp(3))*X+(Xp(3)-Xp(2))*Y )
        ARec2 = 0.5_8*( Xp(3)*Yp(4)-Xp(4)*Yp(3)+
     *               (Yp(3)-Yp(4))*X+(Xp(4)-Xp(3))*Y )
        ARec3 = 0.5_8*( Xp(1)*Yp(2)-Xp(2)*Yp(1)+
     *               (Yp(1)-Yp(2))*X+(Xp(2)-Xp(1))*Y )
        ARec4 = 0.5_8*( Xp(4)*Yp(1)-Xp(1)*Yp(4)+
     *               (Yp(4)-Yp(1))*X+(Xp(1)-Xp(4))*Y )
        Atot  = Atot1 + Atot2
        ALRec(1) = ARec1/Atot
        ALRec(2) = ARec2/Atot
        ALRec(3) = ARec3/Atot
        ALRec(4) = ARec4/Atot
C       Compute the relative difference of the two area to check whether this
C       new mesh node is in the old mesh element.
        DA = 0._8
        DO 147 I = 1, 4
         IF(ALRec(I).LT.0._8) DA = DA + DABS(ALRec(I))
  147   CONTINUE
C      6-node triangular element.
       ELSE
        NMNp = 3
        DO 140 I = 1, NMNp
         Xp(I) = COORDp(INTMAT9p(IEp,I),1)
         Yp(I) = COORDp(INTMAT9p(IEp,I),2)
  140   CONTINUE
        Atot = 0.5_8*( Xp(2)*Yp(3)-Xp(3)*Yp(2)+
     *              (Yp(2)-Yp(3))*Xp(1)+(Xp(3)-Xp(2))*Yp(1) )
        A1   = 0.5_8*( Xp(2)*Yp(3)-Xp(3)*Yp(2)+
     *              (Yp(2)-Yp(3))*X+(Xp(3)-Xp(2))*Y )
        A2   = 0.5_8*( Xp(3)*Yp(1)-Xp(1)*Yp(3)+
     *              (Yp(3)-Yp(1))*X+(Xp(1)-Xp(3))*Y )
        A3   = 0.5_8*( Xp(1)*Yp(2)-Xp(2)*Yp(1)+
     *              (Yp(1)-Yp(2))*X+(Xp(2)-Xp(1))*Y )
C       Calculate local area coordinates for triangle.
        AL(1) = A1/Atot
        AL(2) = A2/Atot
        AL(3) = A3/Atot
C       Compute the relative difference of the two area to check whether this
C       new mesh node is in the old mesh element.
        DA = 0._8
        DO 144 I = 1, 3
         IF(AL(I).LT.0._8) DA = DA + DABS(AL(I))
  144   CONTINUE
       ENDIF
C     In case the new mesh node is the node on its model boundary, it may not
C     be in the old element mesh so we have to search for the nearest element
C     of this new mesh node in the old mesh element and project displacement
C     solutions to this node.
      IF(IEp.EQ.1) THEN
       DAmin = DA  !Set 1st element as the nearest element.
       IEmin = 1
      ENDIF
C     Find the nearest element.
      IF(DA.LE.DAmin) THEN
       DAmin = DA
       IEmin = IEp
       IF(IETIPp(IEp).NE.1) THEN
        ALmin(1) = AL(1)
        ALmin(2) = AL(2)
        ALmin(3) = AL(3)
       ENDIF
```

```
      ENDIF
C      End searching in each element in old model.
  120  CONTINUE
C================================================================================
C      Calculate natural coordinates of this node in this nearest element of the
C      old mesh model and transfer displacement solutions to this new mesh node.
C================================================================================
      IF(IETIPp(IEmin).EQ.1) THEN
      NNp = 9
      DX = X - COORDp(ICTNp(1),1)
      DY = Y - COORDp(ICTNp(1),2)
      AR = DSQRT(DX*DX + DY*DY)
      CALL XYLOCAL(DX,DY,DXL,DYL,THETA0p)
      CALL FTHETA(DXL,DYL,THETA)
      DO 300 ICEp = 1, NCTELEp
       IF(IEmin.EQ.IOCTEp(ICEp)) THEN
        DS = RTHETAp(2*ICEp+1) - RTHETAp(2*ICEp-1)
        DP = THETA - RTHETAp(2*ICEp-1)
        DXH = COORDp(ICTEBNp(2*ICEp),1) - COORDp(ICTNp(1),1)
        DYH = COORDp(ICTEBNp(2*ICEp),2) - COORDp(ICTNp(1),2)
        AH = DSQRT(DXH*DXH + DYH*DYH)
        ABAR = AR/( 0.5_8*AH*DSQRT(1._8 +
     *              DTAND(DS/2._8-DP)*DTAND(DS/2._8-DP)) ) - 1._8
        BBAR = DTAND(DS/2._8-DP)/DTAND(DS/2._8)
        DO 310 IN = 1, 4
         IF(ICTNp(2*ICEp-1).EQ.INTMAT9p(IOCTEp(ICEp),IN)) THEN
          IF(IN.EQ.1) THEN
           Ap = -BBAR
           Bp =  ABAR
          ENDIF
          IF(IN.EQ.2) THEN
           Ap = -ABAR
           Bp = -BBAR
          ENDIF
          IF(IN.EQ.3) THEN
           Ap =  BBAR
           Bp = -ABAR
          ENDIF
          IF(IN.EQ.4) THEN
           Ap =  ABAR
           Bp =  BBAR
          ENDIF
          GOTO 320
         ENDIF
  310   CONTINUE
       ENDIF
  300  CONTINUE
      ELSE
C       Calculate shape functions according to this point.
       NNp = 6
       AN1 = 2._8*ALmin(1)*ALmin(1) - ALmin(1)
       AN2 = 2._8*ALmin(2)*ALmin(2) - ALmin(2)
       AN3 = 2._8*ALmin(3)*ALmin(3) - ALmin(3)
       AN4 = 4._8*ALmin(1)*ALmin(2)
       AN5 = 4._8*ALmin(2)*ALmin(3)
       AN6 = 4._8*ALmin(3)*ALmin(1)
      ENDIF
  320  CONTINUE
C      Get nodal soluions of this nearest element.
      DO 190 I = 1, NNp
       II = INTMAT9p(IEmin,I)
       EUp(I)   = PTp(2*II-1)
       EVp(I)   = PTp(2*II)
       ESXXp(I) = SXXp(II)
       ESXYp(I) = SXYp(II)
       ESYYp(I) = SYYp(II)
       ESVMp(I) = SVMp(II)
  190  CONTINUE
C      Map soluions to this new node according to an element type.
      IF(IETIPp(IEmin).EQ.1) THEN
       CALL GVALUE(Ap,Bp,EUp  ,GEUp  ,NNp)
       CALL GVALUE(Ap,Bp,EVp  ,GEVp  ,NNp)
       CALL GVALUE(Ap,Bp,ESXXp,GESXXp,NNp)
       CALL GVALUE(Ap,Bp,ESXYp,GESXYp,NNp)
       CALL GVALUE(Ap,Bp,ESYYp,GESYYp,NNp)
       CALL GVALUE(Ap,Bp,ESVMp,GESVMp,NNp)
      ELSE
```

```
        GEUp   = AN1*EUp(1) + AN2*EUp(2) + AN3*EUp(3) +
     *           AN4*EUp(4) + AN5*EUp(5) + AN6*EUp(6)
        GEVp   = AN1*EVp(1) + AN2*EVp(2) + AN3*EVp(3) +
     *           AN4*EVp(4) + AN5*EVp(5) + AN6*EVp(6)
        GESXXp = AN1*ESXXp(1) + AN2*ESXXp(2) + AN3*ESXXp(3) +
     *           AN4*ESXXp(4) + AN5*ESXXp(5) + AN6*ESXXp(6)
        GESXYp = AN1*ESXYp(1) + AN2*ESXYp(2) + AN3*ESXYp(3) +
     *           AN4*ESXYp(4) + AN5*ESXYp(5) + AN6*ESXYp(6)
        GESYYp = AN1*ESYYp(1) + AN2*ESYYp(2) + AN3*ESYYp(3) +
     *           AN4*ESYYp(4) + AN5*ESYYp(5) + AN6*ESYYp(6)
        GESVMp = AN1*ESVMp(1) + AN2*ESVMp(2) + AN3*ESVMp(3) +
     *           AN4*ESVMp(4) + AN5*ESVMp(5) + AN6*ESVMp(6)
       ENDIF
       PT(2*IP-1) = GEUp
       PT(2*IP)   = GEVp
        SXX(IP)   = GESXXp
        SXY(IP)   = GESXYp
        SYY(IP)   = GESYYp
        SVM(IP)   = GESVMp
C     End each node in new model except crack tip nodes.
  100 CONTINUE
C================================================================================
C     Map all crack tip nodal displacements from the old mesh into the new one.
C================================================================================
      IF(ICTETRAN.EQ.1) THEN
C     Transfer the first crack tip node solutions.
      PT(2*ICTN(1)-1) = PTp(2*ICTNp(1)-1)
      PT(2*ICTN(1))   = PTp(2*ICTNp(1))
       SXX(ICTN(1))   =  SXXp(ICTNp(1))
       SXY(ICTN(1))   =  SXYp(ICTNp(1))
       SYY(ICTN(1))   =  SYYp(ICTNp(1))
       SVM(ICTN(1))   =  SVMp(ICTNp(1))
C     Transfer the last crack tip node solutions.
      PT(2*ICTN(NCTN)-1) = PTp(2*ICTNp(2*NCTELEp+1)-1)
      PT(2*ICTN(NCTN))   = PTp(2*ICTNp(2*NCTELEp+1))
       SXX(ICTN(NCTN))   =  SXXp(ICTNp(2*NCTELEp+1))
       SXY(ICTN(NCTN))   =  SXYp(ICTNp(2*NCTELEp+1))
       SYY(ICTN(NCTN))   =  SYYp(ICTNp(2*NCTELEp+1))
       SVM(ICTN(NCTN))   =  SVMp(ICTNp(2*NCTELEp+1))
C     Map all remaining crack tip nodes.
      DO 200 IC = 2, NCTN-1
C      Search in old mesh crack tip elements.
       DO 210 ICEp = 1, NCTELEp
        IF(RTHETAp(2*ICEp-1).LE.RTHETA(IC) .AND.
     *                RTHETA(IC).LE.RTHETAp(2*ICEp+1)) THEN
         DP = RTHETA(IC) - RTHETAp(2*ICEp-1)
         DS = RTHETAp(2*ICEp+1) - RTHETAp(2*ICEp-1)
         DO 220 IN = 1, 4
          IF(ICTNp(2*ICEp-1).EQ.INTMAT9p(IOCTEp(ICEp),IN)) THEN
           IF(IN.EQ.1) THEN
            Ap =  DTAND(DP-DS/2._8)/DTAND(DS/2._8)
            Bp = -1._8
           ENDIF
           IF(IN.EQ.2) THEN
            Ap =  1._8
            Bp =  DTAND(DP-DS/2._8)/DTAND(DS/2._8)
           ENDIF
           IF(IN.EQ.3) THEN
            Ap = -DTAND(DP-DS/2._8)/DTAND(DS/2._8)
            Bp =  1._8
           ENDIF
           IF(IN.EQ.4) THEN
            Ap = -1._8
            Bp = -DTAND(DP-DS/2._8)/DTAND(DS/2._8)
           ENDIF
           GOTO 230
          ENDIF
C        End searching each corner position of the old mesh crack tip element.
  220    CONTINUE
        ENDIF
C      End searching in each old mesh crack tip elements.
  210  CONTINUE
       WRITE(*,225) IC
  225  FORMAT(/,' CRACK TIP NODE NO.',I3,' HAS NOT BEEN MAPPED')
  230  CONTINUE   !Natural coordinates have been found.
C      Transfer crack tip nodal solutions.
       NNp = 9
```

```
       DO 240 I = 1, NNp
        II = INTMAT9p(IOCTEp(ICEp),I)
        EUp(I)   = PTp(2*II-1)
        EVp(I)   = PTp(2*II  )
        ESXXp(I) = SXXp(II)
        ESXYp(I) = SXYp(II)
        ESYYp(I) = SYYp(II)
        ESVMp(I) = SVMp(II)
  240  CONTINUE
C
       CALL GVALUE(Ap,Bp,EUp  ,GEUp  ,NNp)
       CALL GVALUE(Ap,Bp,EVp  ,GEVp  ,NNp)
       CALL GVALUE(Ap,Bp,ESXXp,GESXXp,NNp)
       CALL GVALUE(Ap,Bp,ESXYp,GESXYp,NNp)
       CALL GVALUE(Ap,Bp,ESYYp,GESYYp,NNp)
       CALL GVALUE(Ap,Bp,ESVMp,GESVMp,NNp)
       PT(2*ICTN(IC)-1) = GEUp
       PT(2*ICTN(IC))   = GEVp
        SXX(ICTN(IC))   = GESXXp
        SXY(ICTN(IC))   = GESXYp
        SYY(ICTN(IC))   = GESYYp
        SVM(ICTN(IC))   = GESVMp
C     End each new model crack tip node
  200 CONTINUE
      ENDIF
C================================================================================
C     Change nodal displacements to the prescribed ones.
C================================================================================
      NEQ = NPOIN*NDF
      DO 250 IEQ = 1, NEQ
      IF(IBC(IEQ).NE.0) THEN
      PT(IEQ) = QEX(IEQ)
      ENDIF
  250 CONTINUE
C
      RETURN
      END
C================================================================================


      SUBROUTINE TemBDfFUNC(ICASE,X,Y,TEMP,BDfFX,BDfFY)
C================================================================================
C     THIS SUBROUTINE COMPUTES FIXED NODAL TEMPERATURE AND BODY FORCES ACCORDING
C     TO NODAL COORDINATES.
C================================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
C
      SELECT CASE(ICASE)
      CASE(1) !Single edge crack panel, SECP.
      TEMP = 125._8 + 400._8*X - 100._8*X*X
      TEMP = TEMP*2._8
      CASE(2) !Axially crack cylinder, ACC.
      T0 = 0.25_8
      Ri = 20._8
      r  = DSQRT(X*X+Y*Y)
      TEMP = 250._8 + 800._8*(r-Ri) - 200._8*(r-Ri)*(r-Ri)
      TEMP = TEMP*T0
      CASE(3) !Circumferentially crack cylinder, CCC.
      Ri = 80._8
      r  = X
      TEMP = 125._8 + 100._8*(r-Ri) - 6.25_8*(r-Ri)*(r-Ri)
      CASE(4) !Center cracked panel, CCP.
      TEMP = 100._8*X*X
      CASE(5) !Centrifugal force.
      OMEGA = 0.25_8
      BDfFX = 10._8*OMEGA*OMEGA*X
      BDfFY = 10._8*OMEGA*OMEGA*Y
      CASE DEFAULT !No temperature gradient and body forces.
      TEMP  = 0._8
      BDfFX = 0._8
      BDfFY = 0._8
      END SELECT
C
      RETURN
      END
C================================================================================
```

ภาคผนวก ฉ

Proceeding of the 18th ME-NETT Conference
18-20 October 2004, Khon Kean

Stress Intensity Factor Calculation by the Domain Integral Method and Adaptive FEM Remeshing Technique

Kobsak POTJANANAPASIRI, Sutthisak PHONGTHANAPANICH and Pramote DECHAUMPHAI
Department of Mechanical Engineering, Faculty of Engineering,
Chulalongkorn University, 10330, Thailand
Tel: 0-2218-6621 Fax: 0-2218-6621 E-mail: fmepdc@eng.chula.ac.th

Abstract

This paper presents a finite element method for analyzing two-dimensional linear elastic fracture mechanics problems with cracks presented in material bodies. Stress intensity factor is used as the parameter to characterize the severity of the stresses near the crack tip. The domain integral method, for which all relevant quantities are integrated over any arbitrary element area around the crack tip, is utilized as the stress intensity factor solution scheme. The six-node triangular elements are placed around the crack tip. An adaptive remeshing technique is implemented for automatically generating small elements in the regions with high stress gradients to improve solution accuracy. Many benchmark problems are analyzed to demonstrate the efficiency of the numerical solution scheme.

1. Introduction

In linear elastic material behavior, Stress Intensity Factor, SIF, is the most widely used parameter characterizing the intensity of stresses near a crack tip. Many numerical procedures have been developed to estimate the SIF such as stress and displacement matching, contour integration and virtual crack extension [1], etc. One efficient method that has many advantages is the energy domain integral. Originally formulated by Shih, et. al. [2], this approach is remarkably versatile because it can be applied to both quasistatic and dynamic problems with elastic, plastic, or viscoplastic material responses, as well as thermal loading. Moreover, it can numerically be employed to efficiently calculate the other two important elastoplastic crack tip parameters; J and T^*-integral which based respectively on the deformation and incremental theory of plasticity [3].

In this paper, the domain integral method is used to calculate the energy release when a crack grows and convert it to the SIF by relations between stresses and energy. Adaptive remeshing technique and crack tip element in which mid-side nodes near the tip displaced from its nominal positions to quarter points [4] are also implemented to enhance the solution accuracy. Several problems have been analyzed to demonstrate the algorithm.

2. The energy domain integral

For stable crack growth in a two-dimensional body having a line crack along the x_1 axis, the energy release per unit crack advance is,

$$J = \lim_{\Gamma \to 0} \int_{\Gamma} \left(W\delta_{1i} - \sigma_{ij} u_{j,1} \right) n_i \, dC \tag{1}$$

where W is the stress work density, σ_{ij} and u_i are components of the stress and displacement along the x_i axis, n_i is the unit vector normal to Γ contour and dC is the infinitetesimal arc length as depicted in Fig. 1.

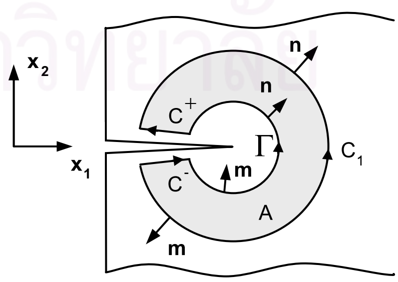


Fig. 1. Closed contour $C = C_1 - \Gamma + C^+ + C^-$ enclosing a simply connected region A

In the absence of thermal strain, body force and crack face traction, Eq. (1) can be rewritten in the form,

$$J=\int_{C}\left[\sigma_{ij}u_{j1}-W\delta_{1i}\right]m_{i}q_{1}dC \qquad (2)$$

where $C=C_1+C^{+}+C^{-}-\Gamma$ is the closed curve, q_1 is a sufficiently smooth function in the area enclosed by C which is unity on Γ and zero on C_1, and m_j is the components of outward normal unit vector as shown in Fig. 1. By applying the divergence theorem to (2),

$$J=\int_{A}\left[\left(\sigma_{ij}u_{j,1}-W\delta_{1t}\right)q_{1}\right]_{,1}dA \qquad (3)$$

where A is the area enclosed by C. Invoking the equilibrium equation, the domain expression for the energy release rate is,

$$J=\int_{A}\left[\sigma_{ij}u_{j,1}-W\delta_{1i}\right]q_{1,i}dA \qquad (4)$$

The function q_1 can be interpreted as a unit translation on Γ in the x_1 direction while keeping the material points on C_1 fixed. According to the vanishing of Γ around the tip, this can be viewed as the growing of the crack.

3. Stress intensity factor

In linear elastic material response, the stress intensity factor in opening mode can be computed from the energy release rate by the expression [1],

$$K_I=\sqrt{JE'} \qquad (5)$$

where $E'=E,\dfrac{E}{1-\nu}$ for plane strain and plane stress case respectively, E is the modulus of elasticity, and ν is the Poisson's ratio.

4. Finite element formulation for the domain integral method

For the six-node isoparametric element, the coordinates, displacements, and a smooth function are,

$$x_i=\sum_{K=1}^{6}N_K X_{iK} \qquad (6)$$

$$u_i=\sum_{K=1}^{6}N_K U_{iK} \qquad (7)$$

$$q_1=\sum_{K=1}^{6}N_K Q_{1K} \qquad (8)$$

where N_K are the shape functions, X_{iK} are the nodal coordinates, U_{iK} are the nodal displacements and Q_{1I} are the nodal values of the smooth function varying between 1 and 0.

Using Eq. (6) and (8) and the chain rule, the spatial gradient of q_1 is,

$$\frac{\partial q_1}{\partial x_j}=\sum_{I=1}^{6}\sum_{k=1}^{2}\frac{\partial N_I}{\partial \eta_k}\frac{\partial \eta_k}{\partial x_j}Q_{1I} \qquad (9)$$

where $\dfrac{\partial \eta_k}{\partial x_j}$ is the inverse Jacobian matrix.

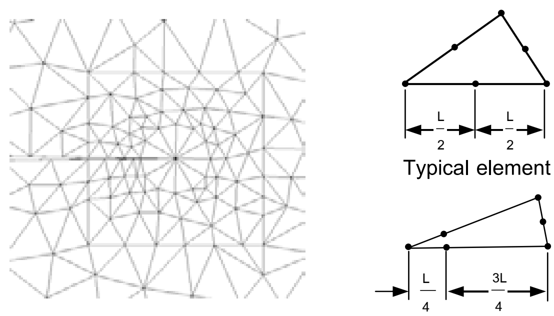
For 2×2 Gaussian integration, the energy release rate expression in Eq. (4) is,

$$J=\sum_{\substack{\text{all}\\ \text{elements}\\ \text{in A}}}\sum_{p=1}^{4}w_p\left\{\left[\sigma_{ij}\frac{\partial u_j}{\partial x_1}-W\delta_{1i}\right]\frac{\partial q_1}{\partial x_1}\det\left(\frac{\partial x_k}{\partial \eta_k}\right)\right\}_p t \qquad (10)$$

where all quantities are calculated at the 4 Gauss points with w_p as their respective weights and t is the specimen thickness.

5. Crack tip elements and the smooth function

Fig. 2 shows elements and finite element mesh on the domain used in this scheme. In this paper, the six-node rosette elements which the mid-side nodes near a tip located on the one-fourth of their sides from the tip are placed around the crack tip. These element can improve the solution because they have the same $1/\sqrt{r}$ singularity of displacement solutions as the exact solution does at the tip. The other elements out of this rosette are standard six-node isoparametric triangular elements.



Rosette element around crack tip

Fig. 2. Finite element mesh and the elements used on integrated domain

According to Shih, et. al. [2], the simple pyramid function as depicted in Fig. 3 is utilized as the smooth function which is unity at the crack tip and varies to zero on the edges of the domain. The base of this pyramid smooth fuction which coincides with the square mesh surrounding the tip is also shown in the figure.

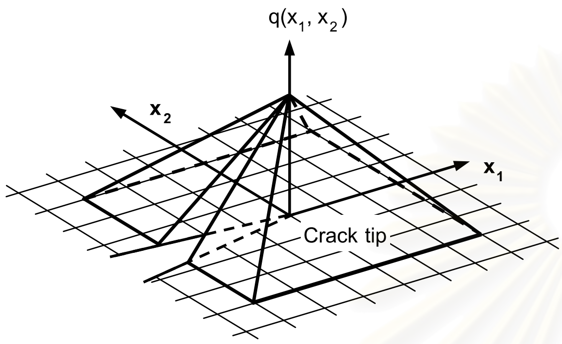


Fig. 3. A smooth function on integrated domain

6. Adaptive remeshing technique

The adaptive remeshing technique generates an entirely new mesh based on the solution obtained from the previous mesh. The technique generates small elements in the regions with large change in the stress gradients to increase the analysis solution accuracy. At the same time, larger elements are generated in the other regions where the stress is nearly uniform to reduce the computational time and the computer memory. The adaptive remeshing procedure thus consists of two main steps: the computation of proper element sizes and the generation of a new mesh for the entire domain.

6.1 Element sizes

To determine proper element sizes at different locations in the domain, the solid mechanics concept for determining the principal stresses from a given state of stresses at a point is employed. Because small elements must be placed in the region where large changes in the stress gradients, such as the von Mises stress σ, occur. Thus the second derivatives of the von Mises stress at a point with respect to global coordinates x_1 and x_2 are needed to compute. Then the principal quantities in the principal directions X_1 and X_2 where the cross derivatives vanish are determined,

$$\begin{bmatrix} \dfrac{\partial^2\sigma}{\partial x_1^2} & \dfrac{\partial^2\sigma}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2\sigma}{\partial x_1 \partial x_2} & \dfrac{\partial^2\sigma}{\partial x_2^2} \end{bmatrix} \Rightarrow \begin{bmatrix} \dfrac{\partial^2\sigma}{\partial X_1^2} & 0 \\ 0 & \dfrac{\partial^2\sigma}{\partial X_2^2} \end{bmatrix} \tag{11}$$

The maximum principal quantities are then used to compute the proper element size, h_i, by requiring that the error should be uniform for all elements,

$$h_i^2\lambda_i = h_{min}^2\lambda_{max} = \text{constant} \tag{12}$$

where $$\lambda_i = \max\left(\left|\frac{\partial^2\sigma}{\partial X_1^2}\right|, \left|\frac{\partial^2\sigma}{\partial X_2^2}\right|\right)$$

λ_{max} is the maximum principal quantity for all elements and h_{min} is the minimum element size specified by users.

6.2 Mesh regeneration

The mesh regeneration with adaptive remeshing technique is implemented based on the Delaunay triangulation and mesh refinement [5]. The main idea is to construct a new mesh over the background mesh (mesh from the previous step). Therefore, the new mesh consists of small elements in the regions with large change in solution gradients and large elements in the other regions where the change in solution gradients in small. The capability of such adaptive remeshing technique will be demonstrated by benchmark examples.

7. Algorithm evaluation

Several examples have been used to demonstrate the efficiency of the combined domain integral, the finite element method, and the adaptive remeshing technique. The examples of a single edge cracked plate, a compact tension specimen and a center cracked plate are used to determine the stress intensity factor in the opening mode under the plane strain condition.

7.1 The single edge cracked plate

The geometry of the single edge cracked plate and its final adaptive mesh are shown in Fig. 4. The stress intensity factor can be calculated from [6],

$$K_I = F\sigma\sqrt{\pi a} \tag{13}$$

where $F = 1.12 - 0.23\alpha + 10.55\alpha^2 - 21.72\alpha^3 + 30.39\alpha^4$

and $\alpha = a/b$

The final adaptive mesh consists of 444 triangles and 931 nodes. The computed stress intensity factor from this adaptive mesh is 2.366 comparing to 2.363 from Eq. (13) with the difference of 0.127%

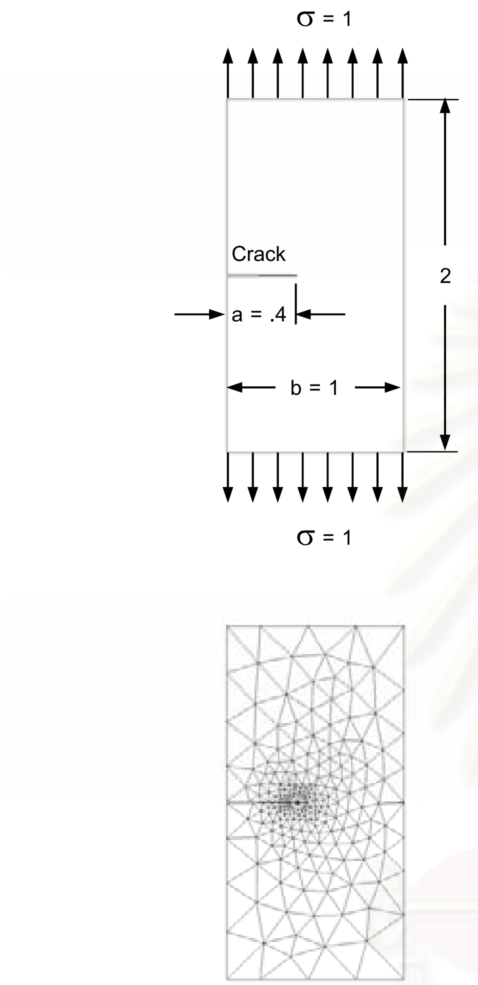


Fig. 4. Problem statement and the final mesh of the single edge cracked plate.

7.2 The compact tension specimen

The geometry of the compact tension specimen and its final adaptive mesh are shown in Fig. 5. The final adaptive mesh consists of 1,396 triangles and 2,939 nodes. The stress intensity factor can be calculated from [7],

$$K_I = P\left(2 + a/w\right)\left[0.886 + 4.64\left(a/b\right) - 13.32\left(a/b\right)^2 + 14.72\left(a/b\right)^3 - 5.6\left(a/b\right)^4\right] / t\sqrt{w}\left(1 - a/w\right)^{3/2} \quad (14)$$

where the thickness $t = 25.4$ mm. The computed stress intensity factor from the adaptive mesh is 28.599 comparing to 27.804 from Eq. (14) with the difference of 2.859%

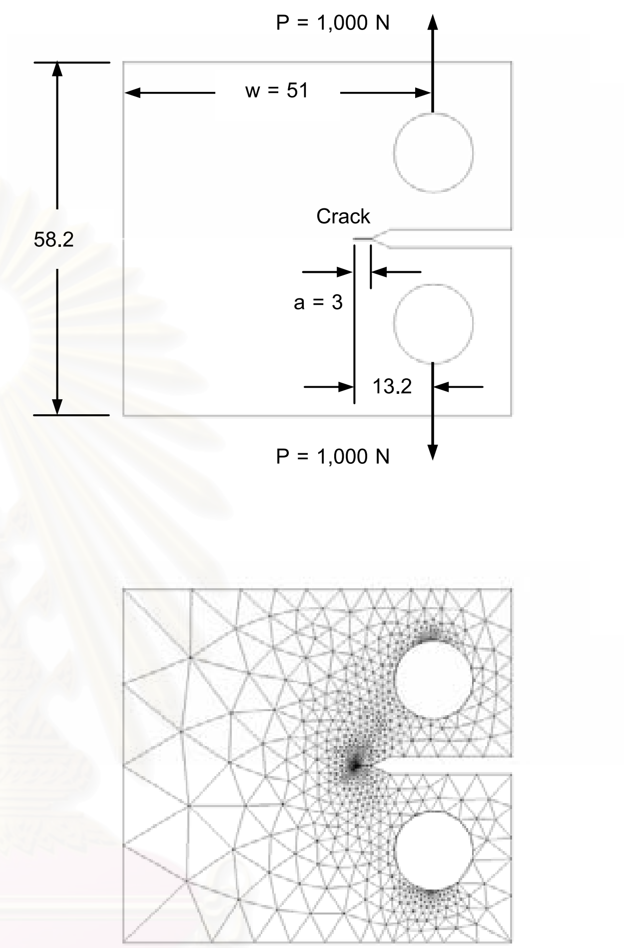


Fig. 5. Problem statement and the final mesh of the compact tension specimen.

7.3 The center cracked plate

The geometry of the center cracked plate and its final adaptive mesh are shown in Fig. 6. The plate has an initial crack length $2a = 100$ units, and the thickness $t = 1$ unit. The stress intensity factor for this problem was derived [8] in closed-form as,

$$K_I = 1.334\sigma\sqrt{\pi a} \quad (15)$$

The final adaptive mesh consists of 1,254 triangles and 2,580 nodes. The computed stress intensity factor from this adaptive mesh is 16.7133 comparing to 16.7192 from Eq. (15) with the difference of 0.04%

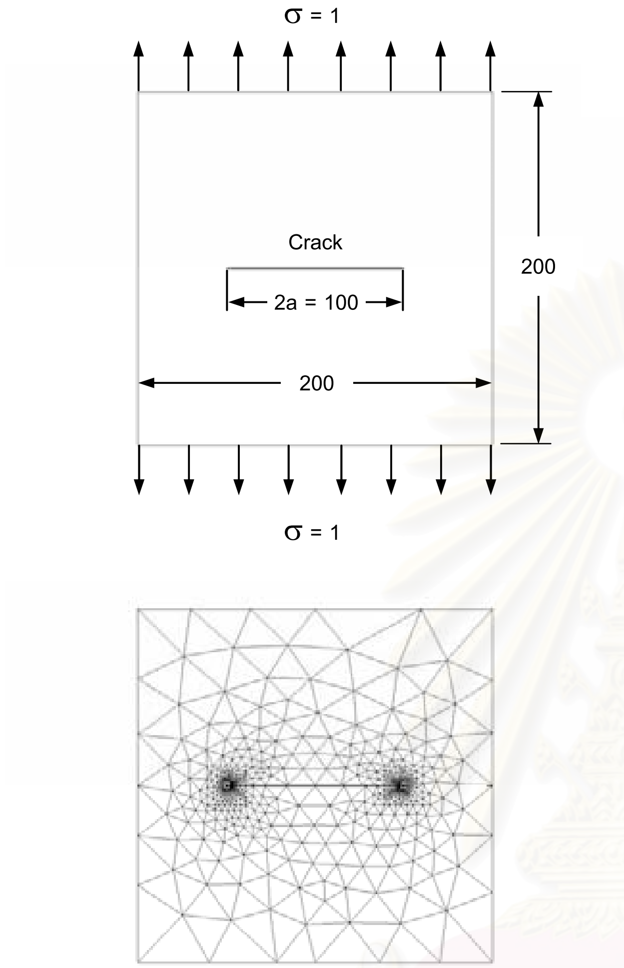


Fig. 6. Problem statement and the final mesh of the center cracked plate.

7.4 Conclusions

Domain integral was combined with the finite element method and the adaptive remeshing technique for analysis of linear elastic fracture mechanics problems. The concept of the domain integral and its smooth function for two-dimensional geometry were explained. The finite element method using the six-node triangular elements was described. These triangular elements with mid-side nodes displaced from their nominal position to a quarter point of the crack tip were employed to form up a circular zone surrounding the crack tip for providing accurate solution. The solution accuracy was further enhanced by incorporating an adaptive remeshing technique. The technique places small elements around the crack tips and in the regions with large change of stress gradients for solution accuracy. At the same, larger elements are generated in the other regions to minimize the total number of unknowns and the computational time.

The efficiency of the combined procedure was demonstrated by examples for determining the stress intensity factor. These examples demonstrate the capability of the combined adaptive remeshing technique with domain integral method for analysis of fracture mechanics problems effectively.

References

[1] T.L. Anderson "Fracture mechanics: fundamentals and applications", CRC Press, 1995.

[2] C.F. Shih, B. Moran and T. Nakamura " Energy Release Rate along a Three-dimensional Crack Front in a Thermally Stressed Body", International Journal of Fracture, 1986, Vol. 30, pp. 79-102.

[3] Y. Omori, A. S. Kobayashi, H. Okada, S. N. Atluri, P. W. Tan " T_{ε}^{*} integral as a crack growth criterion", Mechanics of Materials, 1998, Vol. 28, pp. 147-154.

[4] R.S. Barsoum "On the Use of Isoparametric Finite Elements in Linear Fracture Mechanics", International Journal for Numerical Methods in Engineering, 1976, Vol. 10, pp. 25-37.

[5] P. Dechaumphai, S. Phongthanapanich and P. Bhandhubanyong "Adaptive Delaunay Meshing Technique for Fracture Mechanics Problems", Key Engineering Materials, 2003, Vol. 233-236, pp. 157-162.

[6] Murakami Y. (Editor) Stress intensity Factors Handbook, Pergamon Press, Oxford, 1987.

[7] ASTM 1996 Annual Book of ASTM Standards, American Society for Testing and Materials, Philadelphia, 1996.

[8] M. Isida "Effect of Width and Length on Stress Intensity Factors of Internally Cracked Plates Under Various Boundary Conditions", International Journal of Fracture, 1971, Vol. 7, pp. 301-316.

ประวัติผู้เขียนวิทยานิพนธ์

นายกอบศักดิ์ พจนานภาศิริ เกิดเมื่อวันที่ 10 เดือนพฤศจิกายน พุทธศักราช 2521 จังหวัดสมุทรสงคราม สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิตจากภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ เมื่อปีการศึกษา 2543 เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2545