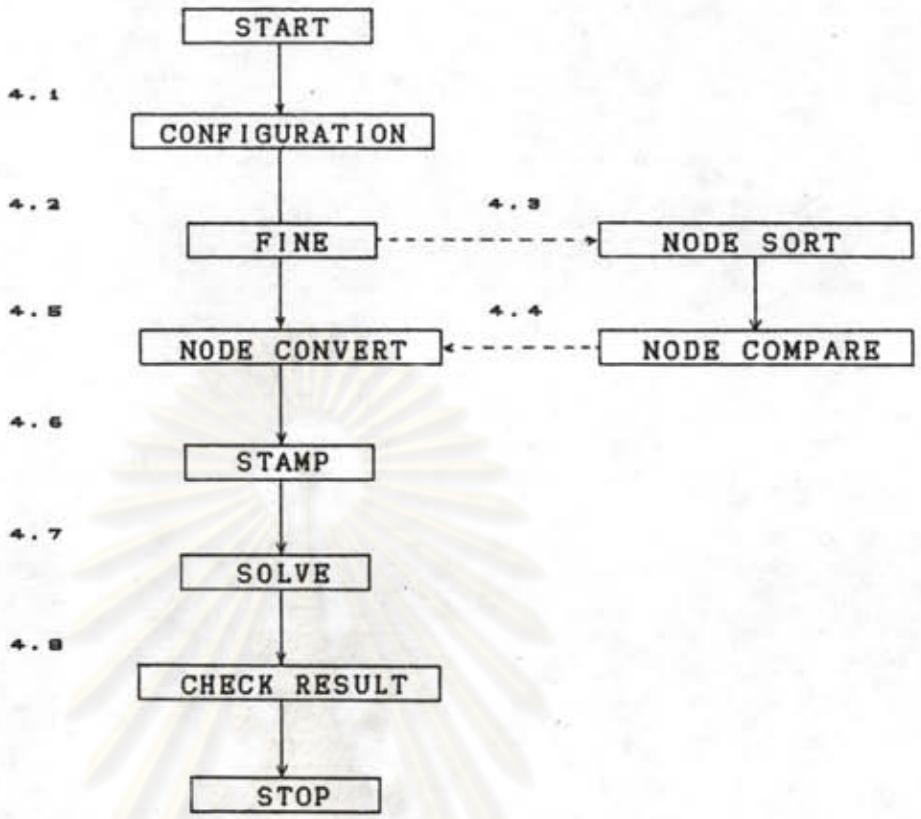
บทที่ 4

โครงสร้างของโปรแกรม

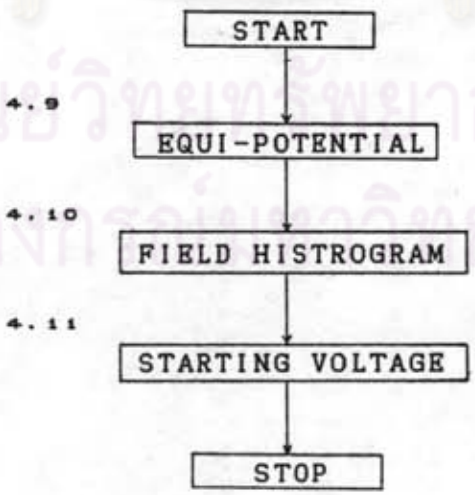
โปรแกรมที่จะกล่าวถึงต่อไปนี้สามารถแก้ไขปัญหาสนามไฟฟ้าสถิตสามมิติที่ไม่มีประจุอิสระอยู่ภายใน และสนามไฟฟ้านั้นต้องเป็นสนามไฟฟ้าที่มีสภาพสมมาตรตามระยะทางหรือมีสภาพสมมาตรตามแกนหมุนเพื่อที่จะได้สามารถแทนให้อยู่ในรูปของสนามไฟฟ้าสองมิติได้ เท่านั้น

ผังของโปรแกรมแสดงในรูปที่ 4.1 และ 4.2 ซึ่งจะอธิบายโดยละเอียดในภายหลัง ภาษาที่ใช้เขียนคือ Applesoft BASIC และแปลเป็นภาษาเครื่องเพื่อเพิ่มความเร็วในการทำงานด้วยโปรแกรมชื่อ TASC BASIC Compiler โปรแกรนี้เขียนขึ้นเพื่อใช้งานกับเครื่องคอมพิวเตอร์ที่เทียบเท่ากับเครื่อง APPLE II+ ที่มีหน่วยความจำที่ใช้งานได้อย่างน้อย 48K และต้องย้าย Disk Operating System ไปอยู่ในหน่วยความจำสำรอง นอกจากนั้นยังต้องมีเครื่องขับจานข้อมูล (Disk Drive) อย่างน้อยอีกหนึ่งเครื่อง การแสดงผลจะแสดงบนจอมอนิเตอร์เสมอ และถ้าหากมีเครื่องพิมพ์ต่ออยู่ก็จะสามารถสั่งให้แสดงผลทางเครื่องพิมพ์ได้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.1 ผังโปรแกรการคำนวณศักย์ในสนามไฟฟ้าด้วยวิธีไฟไนต์อิเล็มเมนต์



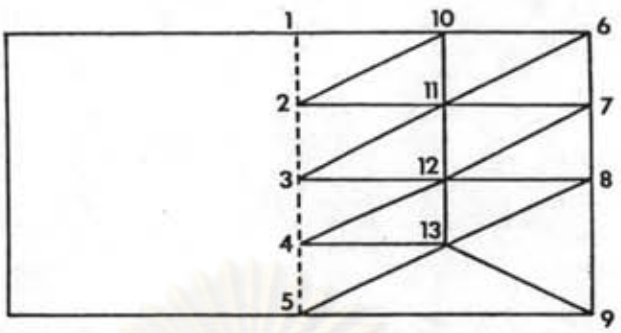
รูปที่ 4.2 ผังโปรแกรมการคำนวณแรงดันเริ่มต้นในอากาศและก๊าซ SF_6

4.1 โปรแกรมย่อย CONFIGURATION

โปรแกรมย่อยนี้จะมีอยู่ด้วยกันสามโปรแกรม คือ โปรแกรมสำหรับปัญหาเงื่อนไขขอบเขตแบบภาคตัดขวางสี่เหลี่ยมผืนผ้า, สำหรับทรงกระบอกซ้อนแกนร่วม, และสำหรับทรงกลมซ้อนศูนย์กลางร่วม ตัวอย่างโปรแกรมย่อยสำหรับทรงกระบอกซ้อนแกนร่วมมี pseudo code ดังแสดงในภาคผนวก จ

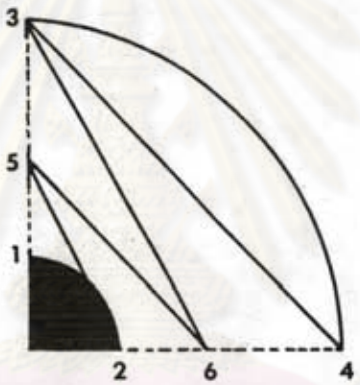
โปรแกรมจะถามผู้ใช้ถึงข้อมูลที่จำเป็นสำหรับแต่ละปัญหาที่ผู้ใช้เลือก เช่น รัศมีของทรงกระบอกใน, รัศมีของทรงกระบอกนอก, หรือความยาวของด้านในปัญหาเงื่อนไขขอบเขตแบบภาคตัดขวางสี่เหลี่ยมผืนผ้า เป็นต้น จากนั้นก็จะแสดงขนาดของปัญหาที่เลือกบนจอมอนิเตอร์ด้วยภาพ, คำนวณพิกัดของจุดต่างๆในสนามไฟฟ้าซึ่งกำหนดได้ไว้ล่วงหน้าว่าอยู่ที่ตำแหน่งใดเมื่อเทียบกับขนาดของปัญหาที่ผู้ใช้ป้อนค่าพิกัดที่ได้จะเก็บไว้ในตัวแปร X(I) และ Y(I) ซึ่ง I คือหมายเลขของจุด, แบ่งสนามไฟฟ้าออกเป็นสามเหลี่ยมย่อยแล้วกำหนดว่าจะมีจุดใดบ้างที่ประกอบขึ้นเป็นสามเหลี่ยมแต่ละรูป ด้วยตัวแปร RE(J,K) ซึ่งมีความหมายว่าจุดหมายเลขนั้นอยู่ที่จุดยอดที่ K ของสามเหลี่ยมหมายเลข J, กำหนดค่าแรงดันที่เงื่อนไขขอบเขตให้กับจุดที่อยู่บนเงื่อนไขขอบเขตด้วยตัวแปร POT(I) ซึ่ง I คือหมายเลขของจุด, และต้องมีตัวชี้เพื่อบอกว่าจุดใดในสนามไฟฟ้าที่อยู่บนเงื่อนไขขอบเขต ตัวชี้นี้จะเก็บที่ตัวแปร CX(I) ซึ่ง I คือหมายเลขจุด

นอกเหนือจากที่กล่าวมาแล้ว โปรแกรมจะต้องส่งข้อมูลของสนามไฟฟ้าที่ถูกแบ่งออกเป็นสามเหลี่ยมย่อยเบื้องต้นดังแสดงในรูปที่ 4.3 ไปให้โปรแกรมย่อย FINE คือ จำนวนของจุดในสนามไฟฟ้า (NP), จำนวนสามเหลี่ยมย่อยที่ประกอบขึ้นเป็นสนามไฟฟ้า (NE), และจำนวนด้านของสามเหลี่ยมที่มีอยู่ในสนามไฟฟ้า (IS)



NP = 13 : NE = 14 : IS = 26

ก. ปัญหาเงื่อนไขขอบเขตแบบภาคตัดขวางสี่เหลี่ยมผืนผ้าที่มีสมมาตรตามระยะทาง



NP = 6 : NE = 4 : IS = 9

ข. ทรงกระบอกซ้อนแกนร่วมที่มีสมมาตรตามระยะทาง

NP = 6 : NE = 4 : IS = 9

ค. ทรงกลมซ้อนศูนย์กลางร่วม

รูปที่ 4.3 การแบ่งสนามไฟฟ้าออกเป็นสามเหลี่ยมย่อยเบื้องต้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.2 โปรแกรมย่อย FINE

โปรแกรมนี้มีหน้าที่แบ่งสนามไฟฟ้าออกเป็นสามเหลี่ยมย่อย ๆ ตามที่ต้องการ เพราะการแบ่งสนามไฟฟ้าออกเป็นสามเหลี่ยมย่อย ๆ ด้วยการให้ผู้ใช้ป้อนข้อมูลเป็นงานที่น่าเบื่อ ใช้เวลานาน และที่สำคัญที่สุดคือมีโอกาสผิดพลาดมาก

โปรแกรมย่อย FINE มี pseudo code ดังแสดงในภาคผนวก จ โดยโปรแกรมจะเริ่มจากการคำนวณล่วงหน้าว่าการแบ่งสนามไฟฟ้าครั้งต่อไปจะทำให้มีจำนวนของจุดในสนามไฟฟ้า, จำนวนของสามเหลี่ยมที่ประกอบขึ้นเป็นสนามไฟฟ้า, และจำนวนด้านของสามเหลี่ยมในสนามไฟฟ้าเป็นเท่าไรโดยอาศัยสมการดังต่อไปนี้ [10]

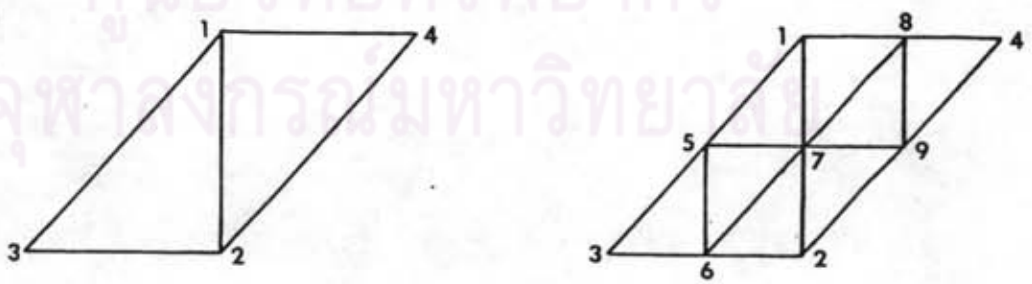
$$(NE)_{n+1} = 4 \cdot (NE)_n \qquad (4.2.1)$$

$$(NP)_{n+1} = (NP)_n + (IS)_n \qquad (4.2.2)$$

$$(IS)_{n+1} = (IS)_n \cdot 2 + (NE)_n \cdot 3 \qquad (4.2.3)$$

ยกตัวอย่างเช่นการแบ่งสามเหลี่ยมสองรูปที่ติดกันดังแสดงในรูปที่ 4.4 ก เป็นสามเหลี่ยม 8 รูปในรูปที่ 4.4 ข

$NE_0 = 2 : NP_0 = 4 : IS_0 = 5$ $NE_1 = 8 : NP_1 = 9 : IS_1 = 16$



ก. ก่อนการแบ่ง ข. หลังการแบ่งครั้งแรก

รูปที่ 4.4 การแบ่งสนามไฟฟ้าออกเป็นสามเหลี่ยมย่อย

นอกจากนั้นผลของการคำนวณล่วงหน้ายังใช้เป็นตัวเปรียบเทียบด้วยว่าการแบ่งนั้นถูกต้องหรือไม่ การแบ่งจะกระทำซ้ำกี่ครั้งก็ได้ขึ้นอยู่กับจำนวนจุดในสนามไฟฟ้าที่เราต้องการและขึ้นกับหน่วยความจำของเครื่องคอมพิวเตอร์ โปรแกรมที่สร้างขึ้นได้กำหนดให้มีจุดในสนามไฟฟ้าได้ไม่เกิน 780 จุด และมีจำนวนสามเหลี่ยมย่อยไม่เกิน 1410 รูป

ในกรณีที่ด้านของสามเหลี่ยมอยู่บนส่วนโค้ง โปรแกรมจะแบ่งด้านนั้นด้วยการใช้รัศมีของส่วนโค้งและมุมของจุดปลายของด้านเพื่อให้จุดที่เกิดจากการแบ่งอยู่บนส่วนโค้งด้วย เช่นในกรณีของทรงกระบอกซ้อนแกนร่วมและทรงกลมซ้อนศูนย์กลางร่วม เป็นต้น

เมื่อได้จุดใหม่ที่เกิดจากการแบ่งด้านของสามเหลี่ยมแล้ว โปรแกรมต้องทดสอบดูว่าจุดใหม่ที่ได้มีพิกัดซ้ำกับจุดที่มีอยู่แล้วในสนามไฟฟ้าหรือไม่ ถ้าซ้ำก็ไม่ต้องให้หมายเลขใหม่กับจุดนั้น และโปรแกรมยังต้องทดสอบดูว่าจุดนั้นอยู่บนเงื่อนไขขอบเขตหรือไม่ ถ้าอยู่ต้องกำหนดให้ตัวชี้บอกว่าจุดนั้นอยู่บนเงื่อนไขขอบเขต

หลังจากแบ่งสนามไฟฟ้าจนได้จุดในสนามไฟฟ้าตามที่ต้องการแล้ว โปรแกรมจะเข้าสู่โปรแกรมย่อยถัดไปซึ่งมีอยู่สองโปรแกรม คือ

ก. โปรแกรมย่อย NODE SORT จะเข้าสู่โปรแกรมนี้ในกรณีที่เริ่มหาศักย์ไฟฟ้าของสนามไฟฟ้าแบบใดแบบหนึ่งเป็นครั้งแรกเท่านั้น และต้องเก็บชุดข้อมูลของพิกัด (X-Y file) ไว้ในหน่วยความจำสำรองเพื่อใช้ในโปรแกรมย่อย NODE COMPARE

ข. โปรแกรมย่อย NODE CONVERT จะเข้าสู่โปรแกรมนี้ทุกครั้งเมื่อต้องการหาศักย์ไฟฟ้าของสนามไฟฟ้าแบบเดิม แต่ขนาดของสนามไฟฟ้าเปลี่ยนไป

4.3 โปรแกรมย่อย NODE SORT

โปรแกรมนี้ไม่ได้มีจุดประสงค์ที่จะเรียงหมายเลขจุดที่ต้องการทราบค่าศักย์ไฟฟ้าขึ้นก่อนจุดที่เป็นเงื่อนไขขอบเขตดังที่กล่าวในหัวข้อที่ 2.1 แต่มีจุดประสงค์ที่จะลดจำนวนหน่วยความจำที่ต้องใช้เก็บสมาชิกของเมตริกซ์ Dirichlet

เมตริกซ์ Dirichlet ที่จะสร้างขึ้นจากพิกัดของจุดต่างๆในสนามไฟฟ้าเป็นเมตริกซ์ที่สมมาตรและมีสมาชิกที่เป็นศูนย์อยู่เป็นจำนวนมาก (Sparse Matrix) การเก็บสมาชิกของเมตริกซ์ในหน่วยความจำจึงสามารถเก็บเพียงส่วนล่างดังแสดงในรูปที่ 4.5 ก เท่านั้นก็พอ จำนวนของหน่วยความจำที่ใช้เก็บสามารถคำนวณได้จาก

$$\text{จำนวนหน่วยความจำ} = NP \cdot (NP+1) / 2 \qquad (4.3.1)$$

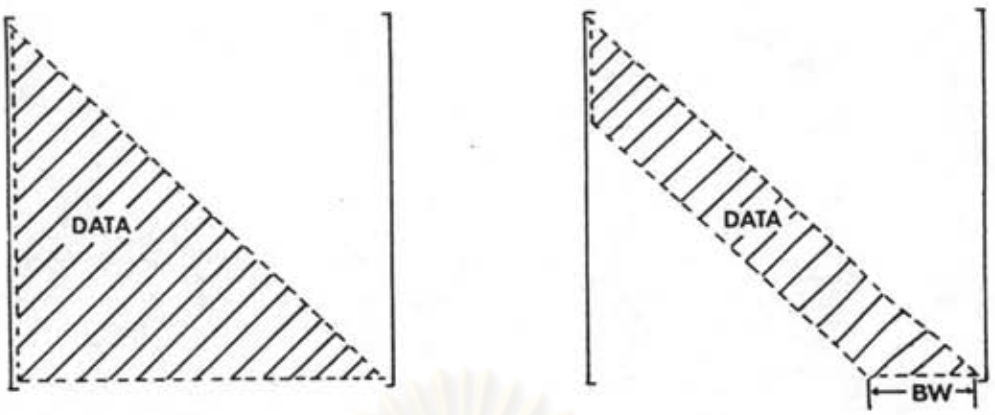
เมื่อ NP = จำนวนจุดในสนามไฟฟ้า = ขนาดของเมตริกซ์

โปรแกรม NODE SORT จะทำหน้าที่เรียงหมายเลขจุดในสนามไฟฟ้าโดยให้จุดที่ใกล้กันมีหมายเลขต่างกันน้อยที่สุดเท่าที่จะทำได้ด้วยวิธีที่คล้ายกับ One-Way Dissection Algorithm [11] ผลที่ได้รับก็คือสมาชิกในเมตริกซ์จะเกาะกลุ่มกันเป็นแถบ (Band) ดังแสดงในรูปที่ 4.5 ข ความกว้างของแถบเราเรียกว่าแบนด์วิดท์ หน่วยความจำที่ต้องใช้เก็บสมาชิกของเมตริกซ์ในลักษณะนี้จะคำนวณได้จากสมการ

$$\text{จำนวนหน่วยความจำ} = \left(\sum_{i=1}^{BW-1} i\right) + (NP+1-BW).BW \qquad (4.3.2)$$

เมื่อ BW = แบนด์วิดท์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ก. การเก็บข้อมูลเฉพาะส่วนล่าง ข. การเก็บข้อมูลเฉพาะที่อยู่ในแถบ

รูปที่ 4.5 การเก็บข้อมูลของเมตริกซ์ Dirichlet

เมื่อเปรียบเทียบสมการที่ (4.3.1) และ (4.3.2) จะเห็นว่ายิ่งมีจำนวนจุดมากขึ้นเท่าไรก็จะทำให้ประหยัดหน่วยความจำมากขึ้นเท่านั้น ตารางที่ 4.1 ได้แสดงจำนวนหน่วยความจำที่ประหยัดได้เป็นจำนวนเท่าของปัญหาทรงกระบอกซ้อนแกนร่วม เมื่อมีจำนวนจุดในสนามไฟฟ้าค่าต่างๆ ซึ่งคำนวณจากสมการที่ (4.2.2) และแบ่งสนามไฟฟ้าเบื้องต้นดังแสดงในรูปที่ 4.3 ข

ตารางที่ 4.1 ตารางเปรียบเทียบหน่วยความจำที่ต้องใช้ในการเก็บข้อมูลของเมตริกซ์ Dirichlet ของปัญหาทรงกระบอกซ้อนแกนร่วม

แบ่งครั้งที่	จำนวนจุด	เก็บเฉพาะส่วนล่าง	เก็บเป็นแถบ	การประหยัด
0	6	21	15	1.40
1	15	120	54	2.22
2	45	1,035	255	4.06
3	153	11,781	1,485	7.93
4	561	157,641	9,945	15.85
5	2,145	2,301,585	72,369	31.80
6	8,385	35,158,305	551,265	63.78

จากตารางที่ 4.1 ถ้าเราแบ่งสนามไฟฟ้าของทรงกระบอกซ้อนแกนร่วมออกเป็น 561 จุดซึ่งเป็นจำนวนจุดในสนามไฟฟ้าที่สูงสุดที่โปรแกรมย่อย FINE แบ่งได้ (การแบ่งครั้งที่ 5 จะมีจำนวนจุดเป็น 2,145 จุด แต่โปรแกรมย่อย FINE แบ่งได้มากที่สุดไม่เกิน 780 จุด) และเนื่องจากเครื่องคอมพิวเตอร์ที่เทียบเท่ากับเครื่อง APPLE II+ ใช้หน่วยความจำ 5 หน่วยในการเก็บตัวแปรจริง การเก็บข้อมูลแบบแถบจะใช้หน่วยความจำน้อยกว่าการเก็บข้อมูลเฉพาะส่วนล่างถึง 738K

โปรแกรมย่อย NODE SORT มี pseudo code ดังแสดงในภาคผนวก จ โดยโปรแกรมจะเริ่มเรียงหมายเลขใหม่ตั้งแต่หมายเลข 2 โดยขั้นแรกจะหาว่าจุดหมายเลข 1 อยู่ที่จุดยอดที่เท่าไรของสามเหลี่ยมหมายเลขใด เมื่อพบแล้วก็จะเปลี่ยนหมายเลขจุดที่จุดยอดที่เหลือของสามเหลี่ยมรูปนั้นเช่น m และ n ให้เป็นหมายเลข 2 และ 3 จากนั้นก็ต้องมีการสลับที่กันระหว่างข้อมูลของจุดหมายเลข 2 และ 3 กับข้อมูลของจุด m และ n เดิม คือ สลับที่ข้อมูล X, Y, POT, CX, และต้องสลับหมายเลขจุดที่จุดยอดของสามเหลี่ยมรูปอื่นด้วย

ขั้นต่อมาก็จะหาจุดที่ใกล้กับจุดหมายเลข 2 และ 3 มากที่สุด สมมุติว่าเป็นจุดหมายเลข a, b, และ c เราจะเรียงลำดับของจุดเหล่านี้ด้วยพิกัดทางแกน Y (ข้อมูล Y) จากค่าพิกัดมากไปหาค่าพิกัดน้อย สมมุติว่าเรียงแล้วจุดเหล่านี้มีลำดับเป็น c, a, และ b เราจะเปลี่ยนหมายเลข c, a, และ b ให้เป็นหมายเลข 4, 5, และ 6 ตามลำดับ สลับที่ข้อมูล X, Y, POT, CX, และ หมายเลขจุดที่จุดยอดของสามเหลี่ยมรูปอื่น เช่นเดียวกับขั้นตอนแรก

นอกเหนือจากที่กล่าวมาแล้ว ระหว่างแต่ละขั้นตอน โปรแกรมจะสลับหมายเลขของสามเหลี่ยมที่ได้เปลี่ยนแปลงหมายเลขจุดที่จุดยอดครบทั้งสามจุดยอดแล้วให้มีหมายเลขน้อยที่สุดเท่าที่จะทำได้ เพื่อลดเวลาที่จะต้องใช้ในการหาจุดที่ใกล้กับจุดที่ได้รับการเปลี่ยนแปลงหมายเลขล่าสุด

เมื่อทำซ้ำตามขั้นตอนดังที่กล่าวมาแล้วจนครบทุกจุดในสนามไฟฟ้า โปรแกรมจะเรียงลำดับจุดยอดของสามเหลี่ยมแต่ละรูป โดยให้จุดยอดที่มีหมายเลขจุดต่ำที่สุดเป็นจุดยอดที่ 1 ของสามเหลี่ยมรูปนั้นจากนั้นโปรแกรมจะเรียงลำดับหมายเลข

ของสามเหลี่ยม โดยให้สามเหลี่ยมซึ่งมีหมายเลขจุดต่ำสุดที่จุดยอดที่ 1 มีหมายเลขของสามเหลี่ยมขึ้นต้น การเรียงลำดับในขั้นตอนนี้ใช้วิธี Shell sort [12] เนื่องจากข้อมูลที่จะเรียงลำดับมีจำนวนมากและวิธีนี้มีความเร็วในการเรียงลำดับมากกว่าวิธี Bubble sort สำหรับเหตุผลที่ต้องเรียงลำดับหมายเลขของสามเหลี่ยมจะอธิบายในหัวข้อที่ 4.6

ผลของโปรแกรมนี้คือชุดข้อมูลมาตรฐาน ซึ่งประกอบด้วย ชุดข้อมูลมาตรฐานของสามเหลี่ยมที่ประกอบขึ้นเป็นสนามไฟฟ้า (STD. RE file), ชุดข้อมูลมาตรฐานของแรงดันที่เงื่อนไขขอบเขตและตัวชี้จุดที่เงื่อนไขขอบเขต (STD. POT-CX file), และชุดข้อมูลมาตรฐานของพิกัด (STD. X-Y file)

4.4 โปรแกรมย่อย NODE COMPARE

โปรแกรมนี้ทำหน้าที่หาความสัมพันธ์ของหมายเลขจุดก่อนการเรียงหมายเลขและภายหลังการเรียงหมายเลข โดยการเปรียบเทียบพิกัดของหมายเลขจุดระหว่างชุดข้อมูลมาตรฐานของพิกัด (STD. X-Y file) กับชุดข้อมูลของพิกัด (X-Y file) ความสัมพันธ์ที่ได้จะเก็บไว้ในชุดข้อมูลการเปลี่ยนหมายเลขจุด (NODE CHANGE file) pseudo code ของโปรแกรมได้แสดงไว้ในภาคผนวก จ

โปรแกรมนี้จะใช้สำหรับปัญหาสนามไฟฟ้าแบบใดแบบหนึ่งเพียงครั้งแรกครั้งเดียวเท่านั้น

4.5 โปรแกรมย่อย NODE CONVERT

โปรแกรมนี้ทำหน้าที่เปลี่ยนหมายเลขจุดของพิกัดที่ได้จากโปรแกรมย่อย FINE ให้อยู่ในรูปมาตรฐาน ด้วยการใช้ชุดข้อมูลการเปลี่ยนหมายเลขจุด (NODE CHANGE file) นอกจากนั้นยังเรียกชุดข้อมูลมาตรฐานของสามเหลี่ยมที่ประกอบขึ้นเป็นสนามไฟฟ้า (STD. RE file) และชุดข้อมูลมาตรฐาน

ของแรงดันที่เงื่อนไขขอบเขตและตัวชี้จุดที่เงื่อนไขขอบเขต (STD. POT-CX file) เข้ามาแทนที่ข้อมูลเดิมอีกด้วย pseudo code ของโปรแกรมได้แสดงไว้ในภาคผนวก จ

4.6 โปรแกรมย่อย STAMP

จากหัวข้อที่ 2.7 สมการที่ใช้หาศักย์ในสนามไฟฟ้าจะอยู่ในรูปของ

$$SU = 0 \qquad (4.6.1)$$

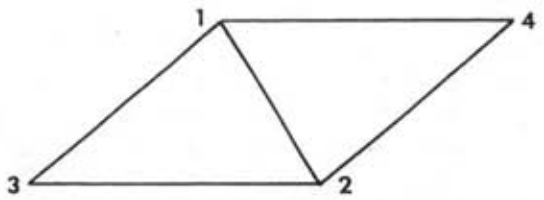
ซึ่งสมการนี้จะเป็นจริง [13] เมื่อตัวกำหนดของเมตริกซ์ Dirichlet มีค่าเป็นศูนย์ หรือเมตริกซ์ U เป็นศูนย์ แต่เนื่องจากเมตริกซ์ U ไม่เป็นศูนย์ ดังนั้นตัวกำหนดของเมตริกซ์ Dirichlet จึงต้องเท่ากับศูนย์ ซึ่งหมายความว่ามีอย่างน้อยหนึ่งแถวในเมตริกซ์ Dirichlet ที่เป็น Linearly dependent กับแถวอื่น จำนวนสมการจึงมีน้อยกว่าจำนวนตัวแปร ทำให้ไม่สามารถหาค่าในเมตริกซ์ U ได้ครบในทางปฏิบัติ ดังนั้นเราจำเป็นที่จะต้องเปลี่ยนสมการที่ (4.6.1) ให้อยู่ในรูปของสมการต่อไปนี้ คือ

$$S'U = R \qquad (4.6.2)$$

โดยที่เมตริกซ์ R จะต้องมีสมาชิกที่ไม่ศูนย์อยู่อย่างน้อยหนึ่งตัว และเราจะเรียกเมตริกซ์นี้ว่าเมตริกซ์เงื่อนไขขอบเขต

พิจารณารูปที่ 4.6 และสมการที่ (2.1.13) ในหัวข้อที่ 2.1 คือ

$$S_{ff}U_f = -S_{fp}U_p \qquad (4.6.3)$$



รูปที่ 4.6 สามเหลี่ยมสองรูปที่ติดกันในสนามไฟฟ้า

สมมุติว่าจุดที่ 1 และ 2 ในรูปที่ 4.6 เป็นจุดที่ต้องการทราบค่าศักย์ และเราทราบค่าศักย์ที่จุด 3 และ 4 เนื่องจากเป็นจุดที่อยู่บนเงื่อนไขขอบเขต สมการที่ (4.6.3) จะสามารถเขียนได้เป็น

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = - \begin{bmatrix} S_{13} & S_{14} \\ S_{23} & S_{24} \end{bmatrix} \begin{bmatrix} U_3 \\ U_4 \end{bmatrix} \qquad (4.6.4)$$

ถ้ากำหนดให้ D เป็น Diagonal Matrix จากสมการที่ (4.6.4) เมื่อกำหนดให้เมตริกซ์ U ทางซ้ายมือของสมการประกอบด้วยศักย์ทุกจุดในรูปที่ 4.6 จะเขียนใหม่ได้เป็น

$$\begin{bmatrix} S_{11} & S_{12} & & \\ S_{21} & S_{22} & & \\ & & D_{33} & \\ & & & D_{44} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} -S_{13} & -S_{14} \\ -S_{23} & -S_{24} \\ D_{33} & \\ & D_{44} \end{bmatrix} \begin{bmatrix} U_3 \\ U_4 \end{bmatrix} \qquad (4.6.5)$$

หรือหากกำหนดให้จุดที่ 2 และ 4 เป็นจุดที่ต้องการทราบค่าศักย์ และจุดที่ 1 และ 3 เป็นจุดที่เงื่อนไขขอบเขต สมการที่ (4.6.4) ก็จะเขียนใหม่ได้เป็น

$$\begin{bmatrix} D_{11} & & & \\ & S_{22} & & S_{24} \\ & & D_{33} & \\ & S_{42} & & S_{44} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} D_{11} & \\ -S_{21} & -S_{23} \\ & D_{33} \\ -S_{41} & -S_{43} \end{bmatrix} \begin{bmatrix} U_1 \\ U_3 \end{bmatrix} \qquad (4.6.6)$$

เนื่องจากสมาชิกของเมตริกซ์ Dirichlet เป็นค่าคงที่ดังที่กล่าวแล้วในบทที่ 2 และโดยปรกติเมตริกซ์ D จะเป็นเมตริกซ์เอกลักษณ์เพื่อความสะดวกในการคำนวณ ทางด้านขวามือของสมการที่ (4.6.5) และ (4.6.6) จึงสามารถคูณกันเพื่อให้เป็นเมตริกซ์แนวตั้งได้ ผลก็คือสมการทั้งสองจะอยู่ในรูปของสมการที่ (4.6.2) นั่นเอง ดังนั้นจะเห็นว่าการสร้างเมตริกซ์ DiIrichlet และเมตริกซ์เงื่อนไขขอบเขตสามารถสร้างพร้อมๆกันได้

สำหรับสามเหลี่ยมแต่ละรูป ถ้าจุดยอดของสามเหลี่ยมที่กำลังสนใจมีหมายเลขจุดในสนามไฟฟ้าเป็นหมายเลขที่ต้องการทราบค่าศักย์ สมาชิกของเมตริกซ์ Dirichlet ก็จะหาด้วยวิธีปรกติ แต่ถ้าเป็นจุดที่เงื่อนไขขอบเขตเราจะแทนแถวและสดมภ์ของเมตริกซ์เอกลักษณ์ในเมตริกซ์ Dirichlet แล้วขยายเมตริกซ์เงื่อนไขขอบเขตทางด้านขวามือตาม

pseudo code โปรแกรมย่อย STAMP ได้แสดงไว้ในภาคผนวก จ โปรแกรมนี้จะมีอยู่สองโปรแกรมคือ โปรแกรมสำหรับสนามไฟฟ้าที่มีสภาพสมมาตรตามระยะทางและโปรแกรมสำหรับสนามไฟฟ้าที่มีสภาพสมมาตรตามแกนหมุน ซึ่งแตกต่างกันเฉพาะส่วนที่สร้างเมตริกซ์ Dirichlet และเมตริกซ์เงื่อนไขขอบเขตเท่านั้น

โปรแกรมจะสร้างเมตริกซ์ Dirichlet ทีละ 10 แถว เนื่องจากเมตริกซ์มีขนาดใหญ่ไม่สามารถสร้างทีเดียวทั้งหมดได้ ยกตัวอย่างเช่น การสร้างแถวที่ 11 ถึง 20 ของเมตริกซ์ โปรแกรมจะทดสอบจุดยอดทุกจุดของสามเหลี่ยมว่ามีหมายเลขจุดต่ำกว่า 11 หรือไม่ ถ้าไม่จึงเริ่มสร้างเมตริกซ์ นอกจากนั้นโปรแกรมยังต้องทดสอบจุดยอดที่ 1 ของสามเหลี่ยมว่ามีหมายเลขจุดสูงกว่า 20 หรือไม่ ถ้าสูงกว่าก็ให้เก็บเมตริกซ์แถวที่ 11 ถึง 20 นั้น

ในหน่วยความจำสำรอง ก่อนการสร้างอีก 10 แถวถัดมา

นี่เป็นเหตุผลที่ว่าทำไมโปรแกรมย่อย NODE SORT จึงต้องมีการเรียงลำดับหมายเลขของสามเหลี่ยม และต้องเรียงให้จุดยอดที่ 1 ของสามเหลี่ยมแต่ละรูปมีหมายเลขจุดต่ำสุด

ผลของโปรแกรมจะได้ข้อมูลของเมตริกซ์ Dirichlet (S file) ในหน่วยความจำสำรอง, ข้อมูลของเมตริกซ์เงื่อนไขขอบเขต (RT), และโปรแกรมจะเก็บข้อมูลพิกัด (X,Y)ของจุดต่างๆในสนามไฟฟ้าไว้ในชุดข้อมูลผลลัพธ์ (DATA file) เพื่อใช้ในโปรแกรมย่อย CHECK RESULT ต่อไป

4.7 โปรแกรมย่อย SOLVE

โปรแกรมนี้จะแก้สมการที่ (4.6.2) โดยทำให้เมตริกซ์ Dirichlet อยู่ในรูปของ

$$S' = LL^T \qquad (4.7.1)$$

โดยที่ L เป็น Lower Triangular Matrix ซึ่งมีศูนย์อยู่เหนือและไปทางด้านขวาของเส้นทแยงมุมหลักของเมตริกซ์ ส่วน L^T ก็คือ Upper Triangular Matrix ซึ่งเป็นทรานสโพสของเมตริกซ์ L นั่นเองและวิธีที่ทำให้เมตริกซ์ S′ อยู่ในรูปของสมการที่ (4.7.1) จะเรียกว่าวิธี Triangular Factorization หรือ Triangular Decomposition

แทนสมการที่ (4.7.1) ลงในสมการที่ (4.6.2) จะได้สมการ

$$LL^TU = R \qquad (4.7.2)$$

หรือ

$$Lz = R \qquad (4.7.3)$$

เมื่อ

$$L^T U = z \qquad (4.7.4)$$

จากการพิจารณาสมการที่ (4.7.3) เนื่องจากเมตริกซ์ L เป็นเมตริกซ์ที่มีสมาชิกที่ไม่เป็นศูนย์เฉพาะสามเหลี่ยมล่าง แถวที่ k ของเมตริกซ์ L จึงมีสมาชิกที่ไม่เป็นศูนย์ระหว่างสดมภ์ที่ 1 ถึง k เท่านั้น ดังนั้นแถวแรกของเมตริกซ์ L จึงมีสมาชิกที่ไม่เป็นศูนย์ในสดมภ์แรกเพียงตัวเดียว ทำให้สามารถคำนวณสมาชิกตัวแรกของเมตริกซ์ z ได้ทันที

สำหรับแถวที่สองของเมตริกซ์ L สมาชิกที่ไม่เป็นศูนย์จะมีเพียงสองตัว คือ ในสดมภ์แรกและสดมภ์ที่สอง ดังนั้นจึงมีสมาชิกของเมตริกซ์ z เพียงสองตัวที่ต้องคำนวณ แต่เนื่องจากเราทราบค่าของสมาชิกตัวแรกในเมตริกซ์ z แล้ว เราจึงสามารถคำนวณค่าของสมาชิกตัวที่สองได้

ทำเช่นนี้จนครบทุกแถวก็จะทำให้ทราบค่าของสมาชิกทุกตัวในเมตริกซ์ z วิธีหาสมาชิกของเมตริกซ์ z แบบนี้มีชื่อเรียกว่า Forward Elimination

เช่นเดียวกับสมการที่ (4.7.4) เนื่องจาก L^T เป็นเมตริกซ์ที่มีสมาชิกที่ไม่เป็นศูนย์เฉพาะสามเหลี่ยมบน การหาค่าของสมาชิกในเมตริกซ์ U จึงทำได้ด้วยวิธีเดียวกันกับการหาสมาชิกของเมตริกซ์ z เพียงแต่ต้องทำย้อนจากแถวสุดท้ายขึ้นมาเท่านั้น และมีชื่อเรียกวิธีนี้อีกอย่างหนึ่งว่า Backward Substitution

ดังที่เห็นแล้วว่าสิ่งที่สำคัญก่อนการหาผลลัพธ์ของสมการที่ (4.6.2) ก็คือการหาเมตริกซ์ L ในสมการที่ (4.7.1) สมาชิกแต่ละตัวของเมตริกซ์ Dirichlet ในรูปของสมการที่ (4.7.1) จะสามารถแสดงได้ด้วยสมการ

$$S_{ik} = \sum_{j=1}^{\min(i,k)} L_{ij} L_{kj} \qquad (4.7.5)$$

ขอบเขตของการรวมยอดในสมการที่ (4.7.5) จะขึ้นอยู่กับค่า i และ k ว่าตัวไหนมีค่าน้อยกว่ากัน

การแก้สมการที่ (4.7.5) จะให้ผลลัพธ์เป็นสมาชิกของเมตริกซ์ L ดังนี้ [2]

ก. สมาชิกของเมตริกซ์ L ที่อยู่บนเส้นทแยงมุมจะคำนวณได้จาก

$$L_{ii} = \left(S_{ii} - \sum_{j=1}^{i-1} L^2_{ij} \right)^{1/2} \qquad (4.7.6)$$

ข. สมาชิกในแถวที่ i และ สดมภ์ที่ k โดยที่ k=1,2,...,i-1 จะคำนวณได้จาก

$$L_{ik} = \left(S_{ik} - \sum_{j=1}^{k-1} L_{ij} L_{kj} \right) / L_{kk} \qquad (4.7.7)$$

โปรแกรมย่อย SOLVE มี pseudo code ดังแสดงในภาคผนวก จ โปรแกรมจะเริ่มด้วยการแก้สมการที่ (4.6.2) ตามวิธีที่กล่าวมาแล้วโดยเริ่มจากการหาสมาชิกของเมตริกซ์ L สมาชิกที่หาได้จะแทนในตำแหน่งเดิมของสมาชิกในเมตริกซ์ Dirichlet (S file) เพื่อประหยัดหน่วยความจำสำรอง พร้อมกันนั้นก็จะหาสมาชิกของเมตริกซ์ z ไปด้วย เพื่อลดเวลาในการเรียกข้อมูลจากหน่วยความจำสำรอง เหตุที่สามารถคำนวณไปพร้อมกันได้ก็เนื่องจากสมาชิกในแถวที่ i ของเมตริกซ์ z จะคำนวณได้จาก L_{ii}, L_{ik}, และ z_k โดยที่ k=1,2,...,i-1 เสมอ สมาชิกของเมตริกซ์ z ที่คำนวณได้จะแทนที่สมาชิกของเมตริกซ์เงื่อนไขขอบเขต (RT) เพื่อประหยัดหน่วยความจำ เมื่อหาสมาชิกของเมตริกซ์ z ครบทุกตัวแล้วก็จะทำย้อนกลับเพื่อหาสมาชิกของเมตริกซ์ U

ผลของโปรแกรมก็คือสมาชิกทุกตัวในเมตริกซ์ U ซึ่งก็คือศักย์ทุกจุดในสนามไฟฟ้านั่นเอง และโปรแกรมจะเก็บผลลัพธ์ของศักย์ในสนามไฟฟ้าที่คำนวณได้ไว้ในชุดข้อมูลผลลัพธ์ (DATA file) เพื่อใช้ในโปรแกรมย่อย CHECK RESULT ต่อไป

4.8 โปรแกรมย่อย CHECK RESULT

โปรแกรย่อยนี้มี pseudo code ดังแสดงในภาคผนวก จ โปรแกรมจะทำหน้าที่เปรียบเทียบผลการคำนวณศักย์ในสนามไฟฟ้าด้วยวิธีไฟไนต์อิเล็มเมนต์ในชุดข้อมูลผลลัพธ์ (DATA file) ที่ได้จากจากโปรแกรม SOLVE กับผลที่ได้จากสมการคณิตศาสตร์วิเคราะห์ (ดูภาคผนวก ฉ) ดังนี้

ก. <u>ปัญหาเงื่อนไขขอบเขตแบบภาคตัดขวางสี่เหลี่ยมผืนผ้า</u> คือ

$$U = 4 \cdot U_o \cdot \left(\sum_{m=1}^{\infty} (\sinh m\pi y/A) \cdot (\sin m\pi x/A) \right.$$

$$\left. /(\sinh m\pi B/A)/m) \right) / \pi$$

เมื่อ A = ความยาวของปัญหาทางแกนพิกัด x

B = ความยาวของปัญหาทางแกนพิกัด y

x,y = พิกัดของจุดที่ต้องการทราบค่าศักย์ไฟฟ้า

โดย $A > x > 0$ และ $B > y > 0$

m = ดรรชนีที่เป็นเลขคี่

U_o = ศักย์ที่พิกัด y เท่ากับ B

ข. <u>ทรงกระบอกซ้อนแกนร่วม</u> คือ

$$U = U_o \cdot \ln(r/R_o) / \ln(R_i/R_o)$$

เมื่อ R_i = รัศมีของทรงกระบอกใน

R_o = รัศมีของทรงกระบอกนอก

r = ระยะทางในแนวรัศมีจากจุดศูนย์กลางของทรงกระบอกถึงพิกัดของจุดที่ต้องการทราบค่าศักย์ไฟฟ้า โดย $R_o > r > R_i$

U_o = ความต่างศักย์ระหว่างทรงกระบอกนอกกับทรงกระบอกใน

ค. <u>ทรงกลมซ้อนศูนย์กลางร่วม</u> คือ

$$U = U_o \cdot R_i \cdot (R_o - r) / (R_o - R_i) / r$$

เมื่อ R_i = รัศมีของทรงกลมใน

R_o = รัศมีของทรงกลมนอก

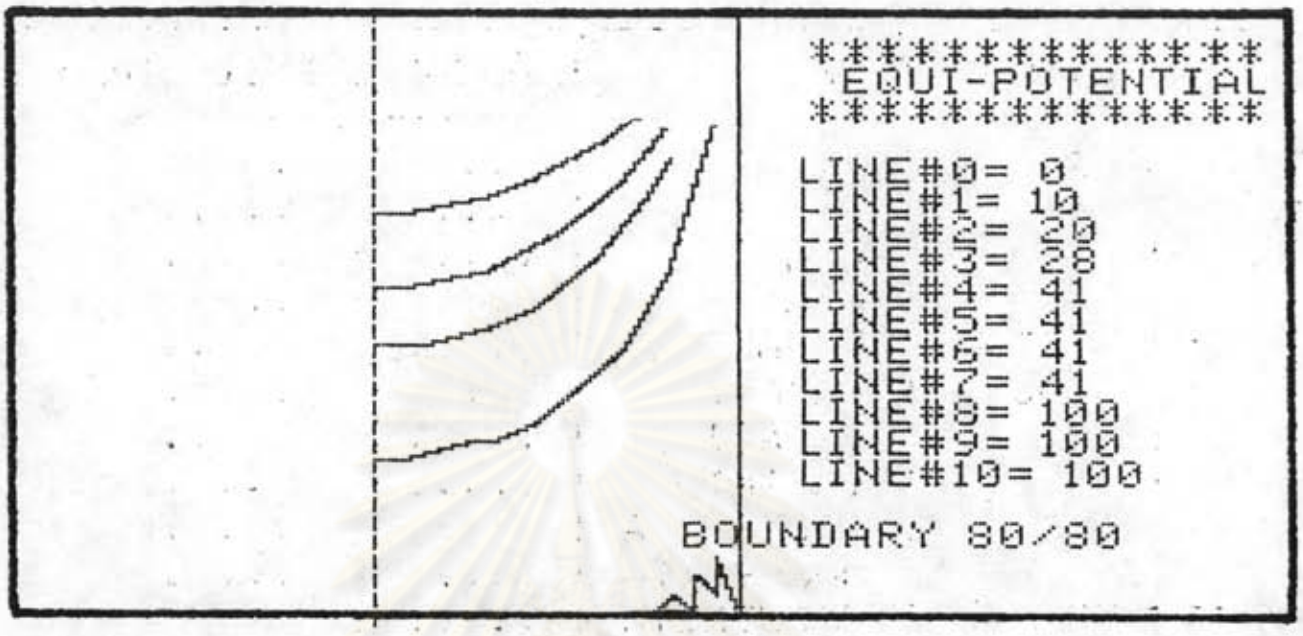
r = ระยะทางในแนวรัศมีจากจุดศูนย์กลางของทรงกลมถึงพิกัดของจุดที่ต้องการทราบค่าศักย์ไฟฟ้า โดย $R_o > r > R_i$

U_o = ความต่างศักย์ระหว่างทรงกลมนอกกับทรงกลมใน

ผลที่ได้จากการเปรียบเทียบจะกล่าวถึงในบทที่ 5

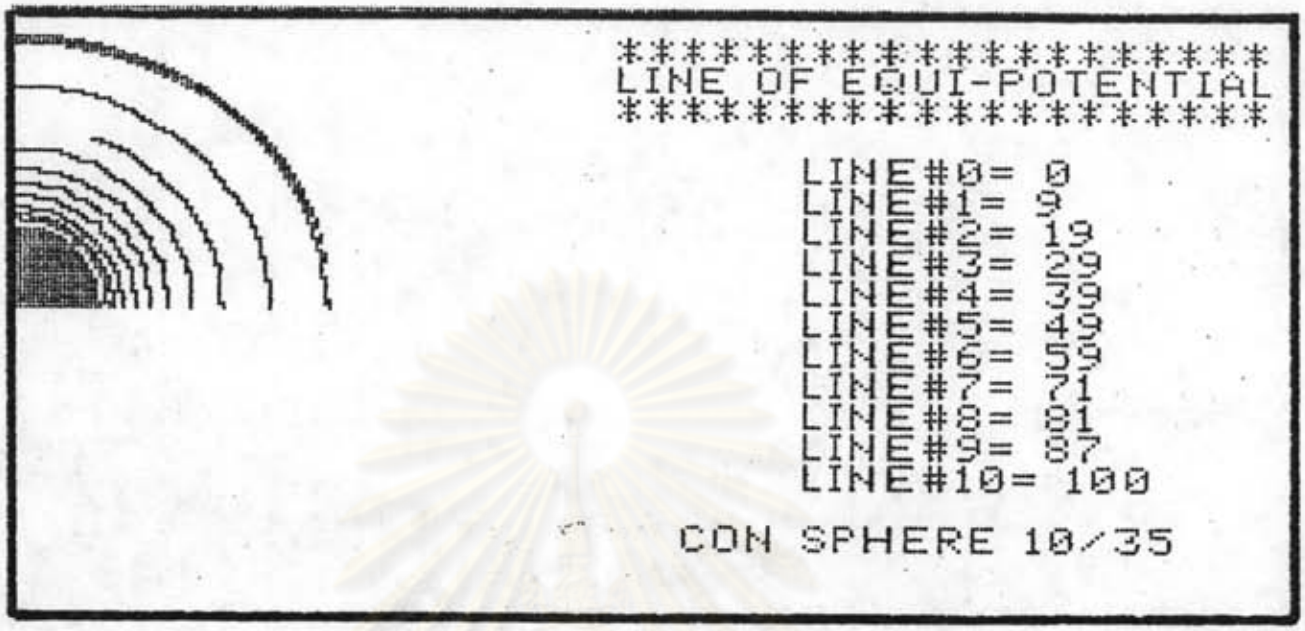
4.9 <u>โปรแกรมย่อย EQUI-POTENTIAL</u>

โปรแกรมนี้มีหน้าที่หาเส้นสมศักย์ในสนามไฟฟ้าทุกเส้นที่ประกอบด้วยจุดที่มีศักย์เท่ากันอย่างน้อย 5 จุด แล้วจะเลือกแสดงบนจอภาพเฉพาะเส้นที่ใกล้กับค่า 10, 20, ... ,90 เปอร์เซนต์ของศักย์ที่เงื่อนไขขอบเขตซึ่งมีค่าสูงสุด นอกจากนั้นโปรแกรมจะส่งข้อมูลของจำนวนเส้นสมศักย์ที่หาได้ (S) และ ค่าศักย์ของแต่ละเส้น (C(I)) ไปให้โปรแกรมย่อย FIELD HISTROGRAM pseudo code ของโปรแกรมได้แสดงไว้ในภาคผนวก จ และตัวอย่างของการแสดงผลแสดงในรูปที่ 4.7



รูปที่ 4.7 ก ตัวอย่างการแสดงผลของโปรแกรมย่อย EQUI-POTENTIAL ของปัญหาเงื่อนไขเขตแบบภาคตัดขวางสี่เหลี่ยมผืนผ้าที่มีสมมาตรตามระยะทาง

รูปที่ 4.7 ข ตัวอย่างการแสดงผลของโปรแกรมย่อย EQUI-POTENTIAL ของทรงกระบอกซ้อนแกนร่วมที่มีสมมาตรตามระยะทาง



รูปที่ 4.7 ค ตัวอย่างการแสดงผลของโปรแกรมย่อย EQUI-POTENTIAL ของทรงกลมซ้อนศูนย์กลางร่วม

4.10 โปรแกรมย่อย FIELD HISTROGRAM

โปรแกรมนี้ทำหน้าที่หาความเครียดสนามไฟฟ้ากระจายในรูปของสมการดังที่กล่าวมาแล้วในหัวข้อที่ 3.1 โปรแกรมจะเริ่มหาคู่ความสัมพันธ์ทีละคู่จากข้อมูลศักย์ของเส้นสมศักย์ (C(i)) และข้อมูลพิกัด (X,Y) โดยที่ความเครียดสนามไฟฟ้าจะคำนวณด้วยสมการที่ (3.1.3) เมื่อได้ครบทุกคู่แล้วก็จะใช้วิธีที่อธิบายในภาคผนวก ค หาค่าคงที่ของสมการ ผลของโปรแกรมคือค่าคงที่ (A และ B) ซึ่งจะนำไปใช้คำนวณแรงดันเริ่มต้นในโปรแกรมย่อย STARTING VOLTAGE

โปรแกรมนี้มี pseudo code ดังแสดงในภาคผนวก จ

4.11 โปรแกรมย่อย STARTING VOLTAGE

โปรแกรมนี้จะหาค่าแรงดันเริ่มต้นของของทรงกระบอกซ้อนแกนร่วม และทรงกลมซ้อนศูนย์กลางร่วม เมื่อมีตัวกลางของสนามไฟฟ้าเป็นอากาศหรือก๊าซ SF_6 ที่ความดันก๊าซ (p) ต่างๆ โดยการอินทิเกรตสมการที่ (3.2.4) หรือ (3.2.5) ผลของโปรแกรมจะทำให้ทราบค่าระยะวิกฤติ (x_c), แรงดันเริ่มต้น (U_i), ความเครียดสนามไฟฟ้าสูงสุดต่อความดันก๊าซ (E_m/p), และแฟกเตอร์สนามไฟฟ้า (n^*) ที่ความดันก๊าซต่างๆ

ในก๊าซ SF_6 จะเปรียบเทียบผลที่คำนวณได้กับผลที่คำนวณจากสมการคณิตศาสตร์วิเคราะห์ [3] ดังนี้

ก. ทรงกระบอกซ้อนแกนร่วม

$$x_c = R_i \cdot ((E_m/p)/(E/p)_c - 1)$$

$$U_i = (E/p)_c \cdot p \cdot d \cdot n^* \cdot (1 + b/(p.R_i)^{1/2})$$

$$E_m/p = (E/p)_c \cdot (1 + b/(p.R_i)^{1/2})$$

$$n^* = R_i \cdot \ln (R_o/R_i) / d$$

เมื่อ

$$d = R_o - R_i$$

$$b = ((1 - K/C/(E/p)_c/p/R_i)/$$

$$(1 - \ln((E_m/p)/(E/p)_c)) - 1) \cdot (p.R_i)^{1/2}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข. <u>ทรงกลมซ้อนศูนย์กลางร่วม</u>

$$x_o = R_i \cdot (((E_m/p)/(E/p)_o)^{1/2}) - 1)$$

$$U_i = (E/p)_o \cdot p \cdot d \cdot n^* \cdot (1 + a/(p.R_i)^{1/2})^2$$

$$E_m/p = (E/p)_o \cdot (1 + a/(p.R_i)^{1/2})^2$$

$$n^* = R_i/R_o$$

เมื่อ

$$d = R_i - R_o$$

$$a = (K / C / (E/p)_o)^{1/2}$$

ในอากาศจะเปรียบเทียบผลการคำนวณที่ได้กับผลจากการอินทิเกรตสมการที่ (3.2.4) โดยตรง

psedo code ของโปรแกรมได้แสดงไว้ในภาคผนวก จ และผลการเปรียบเทียบจะกล่าวถึงในบทที่ 5

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย