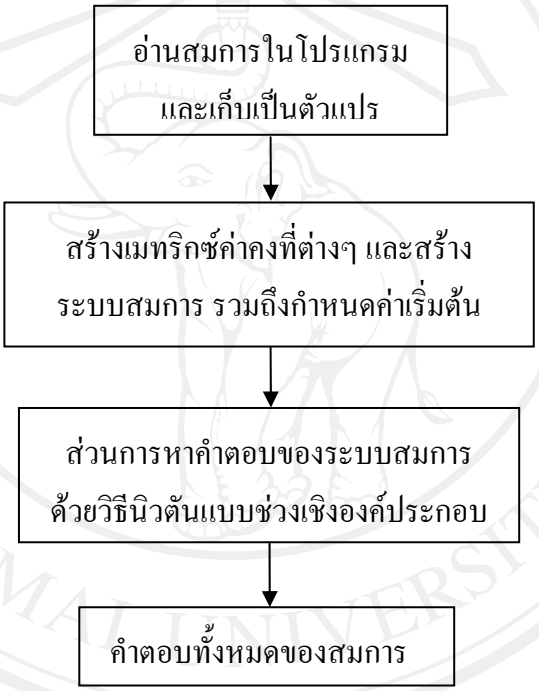
บทที่ 5
การพัฒนาโปรแกรมเพื่อหารากสมการ โดยวิธีนิวตันแบบช่วงทีละองค์ประกอบ

ในบทนี้กล่าวถึงการพัฒนาโปรแกรมเพื่อหารากสมการทั้งหมดของสมการไม่เชิงเส้น โดยใช้วิธีนิวตันแบบช่วงเชิงองค์ประกอบ พัฒนาโปรแกรมโดยใช้ภาษาซีพลัสพลัส และใช้ไลบรารี C-XSC ซึ่งเป็นไลบรารีเสริมสำหรับการพัฒนาโปรแกรมคำนวณคณิตศาสตร์แบบช่วง ซึ่งขั้นตอนการทำงานของโปรแกรมโดยรวมแสดงได้ดังรูปที่ 5.1



รูปที่ 5.1 ขั้นตอนการทำงานของโปรแกรมโดยรวม

จะแบ่งอธิบายโปรแกรมออกเป็นสองส่วนคือ ส่วนการสร้างเมทริกซ์และระบบสมการ และส่วนการหาคำตอบด้วยวิธีนิวตันแบบช่วงเชิงองค์ประกอบทั้งแบบการประมวลผลเดี่ยวและการประมวลผลขนาน

5.1 ส่วนการสร้างเมทริกซ์และระบบสมการ

การระบบสมการจะใช้ฟังก์ชันพื้นฐานของภาษาซีพลัสพลัส โดยใช้รูปแบบ $f(x)=0$ จากนั้นนำเข้าฟังก์ชันเวกเตอร์ที่มีขนาดมิติของเมทริกซ์เท่ากับจำนวนตัวแปรของสมการซึ่งมีการเก็บข้อมูลเป็นแบบโครงสร้าง ฟังก์ชันนี้สามารถหาจาโคเบียนของระบบสมการแบบช่วงได้โดยไม่ต้องหาเมทริกซ์ผกผัน เช่น

ต้องการหารากสมการ

$$\begin{aligned} x_3x_9 + x_5x_9 + x_7x_9 - 1.9791 &= 0 \\ 3x_2x_4 + 2x_2x_6 + x_2x_8 - 4.0616 &= 0 \\ 3x_1x_4 + 2x_1x_6 + x_1x_8 - 1.7172 &= 0 \\ 3x_3 + 2x_5 + x_7 - 3.9701 &= 0 \\ x_1^2 + x_2^2 - 1 &= 0 \\ x_3^2 + x_4^2 - 1 &= 0 \end{aligned}$$

จะได้

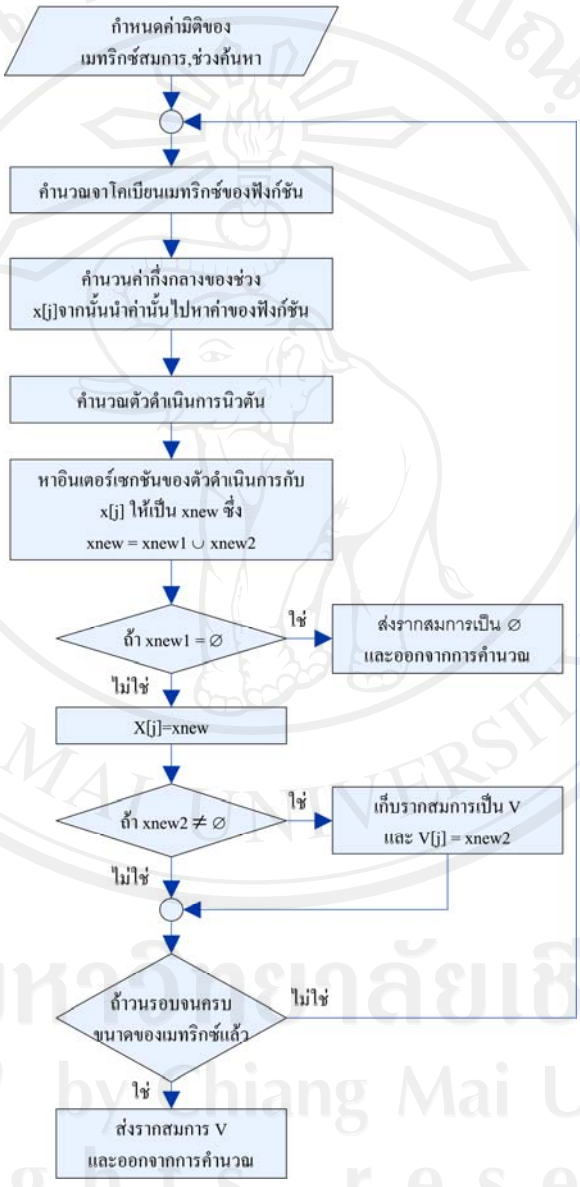
```
GTvector fx(const GTvector& x)
{
        int fDim = x.Dim();
        GTvector Result(fDim);

        Result[1] = x[3]*x[9] + x[5]*x[9] + x[7]*x[9] - 1.9791;
        Result[2] = 3*x[2]*x[4] + 2*x[2]*x[6] + x[2]*x[8] - 4.0616;
        Result[3] = 3*x[1]*x[4] + 2*x[1]*x[6] + x[1]*x[8] - 1.7172;
        Result[4] = 3*x[3] + 2*x[5] + x[7]-3.9701;
        Result[5] = sqr(x[1]) + sqr(x[2]) - 1;
        Result[6] = sqr(x[3]) + sqr(x[4]) - 1;

        return Result;
}
```

โดย Result เป็นฟังก์ชันเวกเตอร์ที่ใช้เก็บสมการในการหารากของสมการซึ่งมีมิติแบบไดนามิกส์ตามจำนวนตัวแปรของสมการของคลาส GTvector

5.2 ส่วนการหาคำตอบด้วยวิธีนิวตันแบบช่วงเชิงองค์ประกอบ

5.2.1 การประมวลผลเดี่ยว

โปรแกรมส่วนนี้เป็นขั้นตอนการหารากของสมการที่สร้างขึ้นในหัวข้อที่ 5.1.2 ด้วยด้วยวิธีนิวตันแบบช่วงเชิงองค์ประกอบโดยการประมวลผลเดี่ยว สามารถสรุปขั้นตอนการทำงานได้ดังรูปที่ 5.2

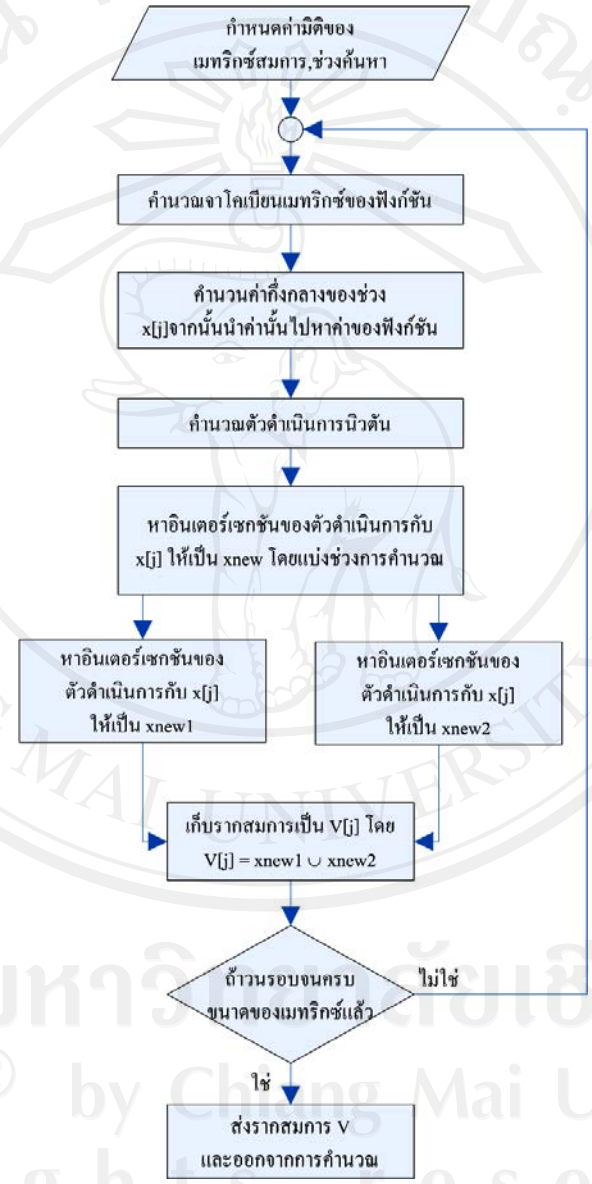


รูปที่ 5.2 ส่วนขั้นตอนการคำนวณด้วยวิธีนิวตันแบบช่วงเชิงองค์ประกอบโดยประมวลผลเดี่ยว

โปรแกรมในส่วนนี้อยู่ในฟังก์ชัน CmpNewtonStep() โดยมีอินพุทเป็นฟังก์ชันที่ได้จากหัวข้อที่ 5.1 และค่าช่วงเริ่มต้นในการค้นหา ได้เอาต์พุทเป็นเมทริกซ์รากของสมการ

5.2.1 การประมวลผลขนาน

โปรแกรมส่วนนี้แตกต่างจากการประมวลผลเดี่ยวเล็กน้อย เมื่อโปรแกรมพบว่ามีรากของสมการมากกว่าหนึ่ง จะทำการแบ่งการคำนวณออกเป็นสองส่วนแยกเป็นสองเทรดอิสระต่อกัน สามารถสรุปขั้นตอนการทำงานได้ดังรูปที่ 5.3



รูปที่ 5.3 ส่วนขั้นตอนการคำนวณด้วยวิธีนิวตันแบบช่วงเชิงองค์ประกอบโดยประมวลผลขนาน

โปรแกรมในส่วนนี้อยู่ในฟังก์ชัน split() โดยมีอินพุทเป็นฟังก์ชันที่ได้จาก XINcmp() จากนั้นรวมรากของสมการเข้าฟังก์ชัน CmpNewtonStep()

5.3 ไลบรารี C-XSC

ไลบรารี C-XSC เป็นเครื่องมือในการแก้ปัญหาทางวิทยาศาสตร์ที่มีประสิทธิภาพสูง ซึ่งมีการตรวจสอบคำตอบแบบอัตโนมัติ โดยพัฒนาจากภาษาซีพลัสพลัสที่เหมาะสมการการทำงานทั้งเครื่องคอมพิวเตอร์ส่วนบุคคล, เครื่องคอมพิวเตอร์ลูกข่ายและเครื่องคอมพิวเตอร์เมนแฟรม

ปกติแล้วภาษาซีพลัสพลัสจะมีการตรวจสอบคำต่างๆ, ตรวจสอบตัวเชื่อมโยงของโปรแกรม โดยไลบรารี C-XSC มีการเพิ่มเติมการตรวจสอบค่าช่วงของดรรชนีสำหรับเวกเตอร์และเมทริกซ์ การตรวจสอบค่าผิดพลาดของตัวเลข เช่น การมีค่ามากเกินขอบเขต (overflow), ค่าน้อยกว่าขอบเขต (underflow), ค่าความคลาดเคลื่อน และความผิดปกติทางตรรกะ เป็นต้น

การดำเนินการเลขคณิตทั้งหมดของไลบรารี C-XSC นั้นจะให้ผลลัพธ์ที่มีความเที่ยงตรงสูงที่สุด หน้าที่ของคณิตศาสตร์พื้นฐานคือให้ผลลัพธ์ที่มีความถูกต้องสูงแต่ไลบรารี C-XSC มีความเป็นไปได้ที่จะได้ผลลัพธ์ที่ถูกต้องที่สุด อย่างไรก็ตามการใช้ก็ต้องอยู่ในสิ่งที่ไลบรารีนี้มีด้วยดังนี้

1. เลขคณิตจำนวนจริงและมาตรฐานจำนวนจริง
2. เลขคณิตแบบช่วงและมาตรฐานจำนวนแบบช่วง
3. ชนิดของข้อมูลที่เกี่ยวกับคณิตศาสตร์ต่างๆ ดังนี้
    3.1จำนวนจริง
    3.2 จำนวนเชิงซ้อน
    3.3 จำนวนแบบช่วง
    3.4 จำนวนแบบช่วงเชิงซ้อน
    3.5 เวคเตอร์จำนวนจริง, เวคเตอร์เชิงซ้อน, เวคเตอร์แบบช่วง, เวคเตอร์แบบช่วงเชิงซ้อน
    3.6 เมทริกซ์จำนวนจริง, เมทริกซ์เชิงซ้อน, เมทริกซ์แบบช่วง, เมทริกซ์แบบช่วงเชิงซ้อน
4. การควบคุมข้อมูลชนิดต่างๆ
5. สมบัติสำหรับการประเมินค่าความถูกต้องของผลลัพธ์

เลขคณิตจำนวนจริงนั้นได้รับการยอมรับในมาตรฐาน IEEE ข้อ 754 สำหรับเลขคณิตทศนิยมดังนั้นจึงมั่นใจได้ว่าเลขคณิตแบบช่วงก็มีความแม่ยำสูงเช่นเดียวกัน ในการค้นคว้าอิสระนี้มีการใช้งานฟังก์ชันของไลบรารีโดยสังเขปดังนี้

5.3.1 AllNLSS(fx,SearchBox,Epsilon,Sol,Uniqe,N,Error)

ฟังก์ชัน AllNLSS() เป็นฟังก์ชันที่ใช้ในการแก้สมการโดยวิธีนิวตันแบบช่วง ภายในฟังก์ชันมีการเรียกใช้ฟังก์ชัน NewtonStep(), XINNewton() และ VerificationStep() เพื่อคำนวณตามระเบียบวิธีของนิวตัน โดยใช้ฟังก์ชันที่บรรจุสมการเป็นแบบ GTvector ,ช่วงค้นหาในการคำนวณหาเมทริกซ์ของรากสมการ และขนาดของบริเวณค้นหา ซึ่งใช้ตัวแปร fx, SearchBox และ Epsilon ตามลำดับ จากนั้นฟังก์ชันจะทำการคืนค่ารากของสมการในรูปแบบเมทริกซ์, การมีคำตอบเดียว, จำนวนรากสมการ และข้อความแสดงข้อผิดพลาด ซึ่งใช้ตัวแปร Sol, Uniqe, N และ Error ตามลำดับ

5.3.2 fJEvalJ(f,x, fx, Jfx)

ฟังก์ชัน fJEvalJ() ใช้ในการหาจาโคเบียนเมทริกซ์ ซึ่งในการค้นคว้าอิสระนี้นำมาช่วยในการแก้สมการ (3.6)โดยแทนฟังก์ชันที่บรรจุสมการเป็นแบบ GTvector และตัวแปรของสมการเป็น f และ x จากนั้นฟังก์ชันจะทำการคืนค่าของสมการในรูปแบบเวคเตอร์และค่าของจาโคเบียนเมทริกซ์ ซึ่งใช้ตัวแปร fx และ Jfx

5.3.3 Resize(L1,cnt-1)

ในการจัดการเมทริกซ์และเวคเตอร์ในการคำนวณโดยไลบรารี C-XCS จะใช้ฟังก์ชัน Resize() เพื่อให้เหมาะสมกับการนำไปใช้ ดังนี้

```
Resize(x,n);            // x[1] … x[n]
Resize(y,-n,-1);        // y[-n] … y[-1]
Resize(A,n,m);          // A[1][1] … A[n][m]
Resize(B,0,n-1,m+4)     // B[0][5] … B[n-1][m+4]
```

5.3.4 accumulate(Accu,A[i][j],y[j])

ฟังก์ชัน accumulate() ใช้ในการเพิ่มผลลัพธ์ที่แม่นยำหรือตัวดำเนินการปริมาณสเกลาร์ยังตัวเพิ่มความแม่ยำของผลคูณแบบสเกลาร์ (dotprecision accumulator) หรือเพิ่มผลลัพธ์ที่แม่นยำแบบสเกลาร์ของสองเวคเตอร์ที่กระทำต่อกันลงในตัวเพิ่มความแม่ยำของผลคูณแบบสเกลาร์ ดังนี้

$$\sum_{j=1}^{n} A[i][j] \square y[j]$$