บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาโปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม ขึ้นมาเป็นชุด โดยแต่ละโปรแกรมย่อยจะมีความสัมพันธ์กันในลักษณะเอื้ออำนวยความสะดวกต่อการเรียกใช้ในโปรแกรมของผู้ใช้คือ

- มีขั้นตอนการใช้ที่แน่นอน ทำให้ผู้ใช้ไม่ต้องสังเลใจในการใช้ (ดังจะปรากฏในตัวอย่างการเรียกใช้โปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม ในภาคผนวก ก.)
- รูปแบบโครงสร้างของข้อมูลที่ผู้ใช้ส่งไปยังโปรแกรมย่อยเป็นรูปแบบที่ง่าย
- ผู้ใช้สามารถที่จะพัฒนาโปรแกมย่อยของผู้ใช้ขึ้นมาเอง โดยไม่จำเป็นต้องเข้าไปยุ่งเกี่ยวกับอุปกรณ์แสดงผลลัพธ์ต่างๆเลย

โปรแกรมย่อยทางกราฟฟิก สำหรับงานวิศวกรรมนี้สามารถแสดงผลลัพธ์ได้สองทางคือ ทางหนึ่ง แสดงผลลัพธ์ทางจอภาพ และอีกทางหนึ่งคือ แสดงผลลัพธ์ทางเครื่องพิมพ์ โดยผู้ใช้ไม่จำเป็นต้องพัฒนาโปรแกรมขึ้นมาสองโปรแกรมสำหรับแสดงผลลัพธ์ในแต่ละอย่าง (จอภาพ หรือ เครื่องพิมพ์) นอกจากนี้ ยังประกอบด้วยโปรแกรมย่อยที่ทำหน้าที่ สำหรับการสร้างกราฟในรูปแบบต่าง ๆ เช่น

- โปรแกรมย่อย SCALE สำหรับคำนวณค่าที่ใช้นำไปใช้ในการพลอตกราฟ เพื่อให้ได้รูปกราฟมีขนาดที่พอเหมาะพอดีกับขนาดความยาวของแกนกราฟ และค่าที่ปรากฏไม่ผิดจากข้อมูลจริงที่ต้องการพลอต
- โปรแกรมย่อย AXIS สำหรับใช้พลอตแกนกราฟ เพื่อใช้อ้างอิงในการอ่านค่าของข้อมูลจากกราฟที่พลอต
- โปรแกรมย่อย LINE สำหรับ พลอตแกนกราฟเส้น
- โปรแกรมย่อย BAR สำหรับ พลอตกราฟแท่ง

- โปรแกรมย่อย BAR2 สำหรับ พลอตกราฟแท่งเปรียบเทียบ
- โปรแกรมย่อย PIE สำหรับ พลอตกราฟวงกลม

และยังมีโปรแกรมย่อยที่ทำหน้าที่วาดรูปทางวิศวกรรม เช่น

- โปรแกรมย่อย CIRCLE สำหรับ พลอตรูปวงกลม
- โปรแกรมย่อย RCTNGL สำหรับ พลอตรูปสี่เหลี่ยม
- โปรแกรมย่อย RCTBLK สำหรับ พลอตรูปสี่เหลี่ยมทึบ
- โปรแกรมย่อย DASHLN สำหรับ สร้างเส้นในลักษณะต่าง ๆ
- โปรแกรมย่อย AROHD สำหรับ สร้างลูกศรในลักษณะต่าง ๆ
- โปรแกรมย่อย SYMBOL และ NUMBER สำหรับใช้เขียนข้อความต่างๆ

นอกจากนี้โปรแกรมย่อยทางกราฟฟิกยังมีโปรแกรมย่อยบางส่วนที่มีลักษณะการเรียกใช้เช่นเดียวกับ โปรแกรมย่อยทางกราฟฟิกที่แสดงผลลัพธ์ทางพลอตเตอร์ ทำให้โปรแกรมที่ถูกพัฒนาสำหรับแสดงผลลัพธ์ทางพลอตเตอร์ สามารถนำมาแสดงผลลัพธ์ทาง จอภาพ และ เครื่องพิมพ์ด้วย ตัวอย่างเช่น โปรแกรม LUSAS ที่ใช้ในการคำนวณ FINITE ELEMENTS สำหรับงานวิศวกรรมโยธา ทำให้ในขั้นตอนที่ยังมีการทดลอบผลลัพธ์อยู่ สามารถนำผลลัพธ์มาแสดงทางเครื่องพิมพ์ก่อนได้ ทำให้ลดการใช้เวลา เนื่องจากเวลาต้องการแสดงผลลัพธ์ทางพลอตเตอร์ ต้องมีเจ้าหน้าที่คอยดูแลเวลาแสดงผลลัพธ์ เมื่อผลลัพธ์ที่ผู้ใช้ตรวจสอบผลลัพธ์ที่แสดงทางเครื่องพิมพ์เรียบร้อยแล้วจึงค่อยแสดงผลลัพธ์ทางเครื่องพลอตเตอร์ต่อไป

การออกแบบโปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรมนี้ จะแบ่งลักษณะของโปรแกรมย่อยออกเป็น 3 ลักษณะ คือ

- ลักษณะแรกจะจัดการเกี่ยวกับอุปกรณ์แสดงผลลัพธ์ หรือควบคุมโปรแกรมย่อยทางกราฟฟิกอื่น โดยควบคุมผ่านทางตัวแปรร่วมเช่น การพลอตจุดบนจอภาพหรือบนแพ้มข้อมูล การลร้างเส้นตรง การสร้างแพ้มข้อมูลที่เก็บเวกเตอร์การพลอต การแปลงแพ้มข้อมูลที่เก็บเวกเตอร์ของการพลอตมาเป็นแพ้มข้อมูลที่เก็บข้อมูลเป็นจุดๆของรูปภาพที่ต้องการพลอต ซึ่งแพ้มข้อมูลนี้จะถูกส่งไปพิมพ์ยังเครื่องพิมพ์ต่อไป

- ลักษณะที่สองทำหน้าที่พื้นฐานต่างๆ เช่น การพลอตเส้น การสร้างวงกลม การพลอตตัวอักษร ฯลฯ

- ลักษณะที่สามทำหน้าที่เฉพาะอย่าง เช่น การสร้างกราฟแท่ง การสร้างกราฟวงกลม การสร้างรูปสี่เหลี่ยม การสร้างลูกศร ฯลฯ

เนื่องจากโปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรมนี้ แบ่งลักษณะหน้าที่อย่างชัดเจน ในกรณีที่ต้องการพัฒนาโปรแกรมย่อยทางกราฟฟิกต่อ บนเครื่องคอมพิวเตอร์หรือ อุปกรณ์แสดงผลลัพธ์ที่เปลี่ยนไป การดัดแปลงแก้ไขจะทำกับโปรแกรมย่อยในลักษณะที่หนึ่งเท่านั้น สำหรับกรณีที่ใช้ภาษาในการพัฒนาโปรแกรมย่อยทางกราฟฟิกเหมือนกัน (ในกรณีนี้ภาษาที่ใช้คือ ภาษาฟอร์แทรน) แต่ถ้ามีความแตกต่างของภาษาที่ใช้เขียนโปรแกรม ลักษณะของโครงสร้างของโปรแกรมย่อยทางกราฟฟิกก็ยังคงไม่ต้องทำการเปลี่ยนแปลง เพียงแต่แปลงภาษาให้มีการทำงานเหมือนเท่านั้น สำหรับสาเหตุที่ใช้ภาษาฟอร์แทรนในการพัฒนาโปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม เพราะว่ามีโปรแกรมสำหรับงานวิศวกรรมที่พัฒนาโดยใช้ภาษาฟอร์แทรนเป็นจำนวนมาก[7] และอีกทั้งผู้ใช้โปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรมส่วนใหญ่คงเป็นวิศวกรซึ่งสามารถเขียนภาษาฟอร์แทรนได้แล้ว โปรแกรมย่อยทางกราฟฟิกนี้จะเหมาะสำหรับงานที่มีลักษณะประมวลผลแบบ BATCH (คือเก็บรวบรวมข้อมูลครบหมดแล้วจึงมีการประมวลผลไม่มีการโต้ตอบในช่วงที่มีการทำงาน)

จากข้อมูลที่กล่าวมาแล้วข้างต้น โปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม จะสามารถทำให้ผู้ใช้ที่มีความต้องการพัฒนาโปรแกรมสำหรับงานวิศวกรรมที่ต้องการแสดงผลลัพธ์แบบกราฟฟิกสามารถเรียกใช้ได้อย่างมีประสิทธิภาพ และ แสดงผลลัพธ์ได้ทั้งจอภาพและเครื่องพิมพ์ โดยไม่จำเป็นต้องเขียนโปรแกรมสำหรับแสดงผลทางจอภาพโปรแกรมหนึ่ง และ โปรแกรมสำหรับแสดงผลทางเครื่องพิมพ์อีกโปรแกรมหนึ่ง โดยเพียงแต่เขียนโปรแกรมเดียวแล้วโปรแกรมย่อยทางกราฟฟิกนี้จะจัดการให้เรียบร้อย

5.2 ข้อเสนอแนะ

การวิจัยนี้ เป็นการแสดงแนวทางในการพัฒนาโปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรมขึ้นมาเป็นชุดที่มีแต่ละโปรแกรมย่อยมีความสัมพันธ์กันในการทำงานทางกราฟฟิก และยังสามารถที่จะนำไปใช้ช่วยในการพัฒนาต่อไปนี้คือ

- การสร้างชุดโปรแกรมย่อยให้สามารถในลักษณะโต้ตอบได้ (interactive)
- การสร้างโปรแกรมย่อยที่สามารถสร้างรูป 3 มิติได้[5] ดังมีตัวอย่างของโปรแกรมย่อย 3 มิติ 2 โปรแกรมย่อย ปรากฏอยู่ข้างล่างต่อไปนี้

```
SUBROUTINE PLOT3D (X,Y,Z,TH,IPEN)
CALL PLOT(Y-X*(COSD(TH),Z-X*(SIND(TH)),IPEN)
STOP
END

SUBROUTINE AXIS3D (TH)
CALL PLOT3D(0.,0.,0.,TH,3)
CALL PLOT3D(4.,0.,0.,TH,2)
CALL PLOT3D(0.,0.,0.,TH,3)
CALL PLOT3D(0.,4.,0.,TH,2)
CALL PLOT3D(0.,0.,0.,TH,3)
CALL PLOT3D(0.,0.,4.,TH,2)
CALL PLOT3D(0.,0.,0.,TH,3)
STOP
END
```

- การสร้างรูปที่มีการซ้อนทับของเส้นแล้วบังกัน(HIDDEN LINE)
- การสร้างรูปวงรี
- การสร้างเส้นโค้งของกราฟที่มีลักษณะเรียบ (SPLINE CURVE)
- ฯลฯ

ซึ่งจะทำให้โปรแกรมย่อยทางกราฟฟิกนี้ สามารถขยายขอบเขตทำงานได้มากขึ้นซึ่งจะเป็นประโยชน์สำหรับงานวิศวกรรมต่อไป

ข้อเสนอแนะเพิ่มเติมเกี่ยวกับค่าผิดพลาด ในการคำนวณค่าจำนวนเต็มในเครื่องคอมพิวเตอร์ เนื่องตำแหน่งที่ใช้ในการพลอตบนอุปกรณ์แสดงผลลัพธ์จะถูกแทนค่าด้วยจำนวนเต็ม เช่น ตำแหน่ง (3,3) ซึ่งถ้าค่านี้ถูกหารด้วย 2 จะได้ค่า คือ (1,1) แล้วมีการคูณด้วย 2 อีกครั้งภายหลังจะได้ค่า คือ (2,2) ซึ่งผิดพลาดจากค่าเดิมไป ฯลฯ เพราะฉะนั้นในส่วนนี้ อาจมีการเพิ่มการรับพารามิเตอร์ที่ใช้กำหนดขนาดการแทนค่าเพื่อป้องกันการผิดพลาด เนื่องจากการปัดเศษในการคำนวณคูณหารกับจำนวนเต็ม เช่น เดิมตำแหน่งที่เก็บไว้คือ (3,3) อาจเก็บด้วยค่า (30,30) แทน ซึ่งเมื่อถูกคูณด้วย 2 หารด้วย 2 ก็ยังได้ค่าเดิม