การพัฒนาโปรแกรมวิเคราะห์โครงสร้างสามมิติสำหรับโครงสร้าง Truss และ Frame
Development of 3-D Structural Analysis Programming for Truss & Frame

ชานน อธิปัญญา[1] ณัฐชยา ศิริมาตย์[2] และ ผศ. ดร.วัฒนชัย สมิทธากร[3]

[1,2,3] ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย จ.กรุงเทพฯ

บทคัดย่อ

บทความนี้นำเสนอทางเลือกในการพัฒนาโปรแกรมวิเคราะห์โครงสร้างด้วยภาษาไพทอน มีจุดประสงค์เพื่อเป็นต้นแบบสำหรับการวิจัยหรือพัฒนาต่อไป ใช้วิธีสติฟเนสโดยตรงประกอบกับแนวคิดการเขียนโปรแกรมเชิงวัตถุเป็นแนวทางในการพัฒนา เพื่อลดปัญหาการเข้าถึงชุดคำสั่งของโปรแกรม ทำให้การพัฒนาโปรแกรมเป็นไปโดยสะดวกมากยิ่งขึ้น นอกจากนี้ ด้วยคุณสมบัติของภาษาไพทอนที่ถูกพัฒนาให้มีชุดคำสั่งโปรแกรมที่ง่ายต่อการทำความเข้าใจ มีโครงสร้างภาษาที่ยืดหยุ่นและกระชับ สามารถทำงานบนระบบปฏิบัติการได้หลากหลาย การจัดการหน่วยความจำอัตโนมัติทำให้การประมวลผลเป็นไปอย่างมีประสิทธิภาพ มีไลบรารี่สนับสนุนจำนวนมากที่มีประสิทธิภาพสูงและใช้งานได้ง่าย เป็น Open-source สนับสนุนการพัฒนาโปรแกรมจากหลากหลายแหล่ง อีกทั้งยังมีประสิทธิภาพในการคำนวณงานทางด้านวิศวกรรมศาสตร์ จากคุณสมบัติของไพทอนที่ได้กล่าวมาข้างต้น ทำให้การพัฒนาโปรแกรมให้มีประสิทธิภาพ มีความสามารถต่างๆ เพิ่มเติมจากเดิม (เช่น การขึ้นโมเดลสามมิติ) และวิเคราะห์ปัญหาโครงสร้างที่มีความซับซ้อนมากขึ้นได้ง่ายกว่าเดิมมาก เมื่อเทียบกับภาษาคอมพิวเตอร์ภาษาอื่น โดยในบทความนี้จะยกตัวอย่างการวิเคราะห์โครงสร้างสามมิติสำหรับโครงสร้าง Truss และ Frame ด้วยโปรแกรมที่ถูกพัฒนาขึ้นในโครงงานไว้

คำสำคัญ: ภาษาไพทอน, การเขียนโปรแกรมเชิงวัตถุ, การวิเคราะห์โครงสร้าง, วิธีสติฟเนสโดยตรง

Abstract

This project provides an alternative to analyze structure with computer program which developed by Python language. Using Direct stiffness method together with Object Oriented Programming (OOP) reduce program's code accessing problems and make developing program more comfortable. Furthermore, Python was developed with many comprehensible commands in addition to it is more compact and more flexible while comparing with other languages. Python also has many high-efficient libraries which can be used together with the program to improve the result, for example, displaying 3D model. Besides, Python is an open-source which support engineer calculation work and compatible with many platforms. From previously mentioned, developing structure analysis program with Python language is uncomplicated but efficient method and can be the prototype for anyone who interested in further development. This project is mainly focus on analyzing 3D Truss and Frame which will be demonstrated in this paper.

Keywords: Python language, Object Oriented Programming, Structure analysis, Direct stiffness method

1. บทนำ

วิศวกรโครงสร้างพยายามสร้างสิ่งอำนวยความสะดวกต่างๆ เพื่อให้สามารถใช้งานได้อย่างปลอดภัย และตอบสนองความต้องการของลูกค้า จึงได้นำเทคโนโลยีเข้ามาช่วยในการทำงาน ซึ่งการนำโปรแกรมวิเคราะห์และออกแบบโครงสร้างเข้ามาใช้ในงานด้านวิศวกรรมโครงสร้างก็เป็นทางเลือกหนึ่ง โดยตั้งแต่สมัยคอมพิวเตอร์ยุคแรกๆ ก็ได้มีการนำภาษา Fortran มาพัฒนาโปรแกรมวิเคราะห์และออกแบบโครงสร้างเป็นจำนวนมาก เนื่องจากเป็นภาษาที่มีประสิทธิภาพในการคำนวณด้านวิศวกรรม แต่ก็ยังมีข้อจำกัดที่เป็นอุปสรรคต่อการพัฒนาโปรแกรม เช่น ต้องแก้ไขชุดคำสั่งทุกครั้งเมื่อมีการเปลี่ยนเครื่องประมวลผล และการเขียนชุดคำสั่งรวมอยู่ในไฟล์เดียวทำให้การเข้าถึงชุดคำสั่งมีความยุ่งยาก เป็นต้น นอกจากที่กล่าวมาข้างต้นแล้ว โครงสร้างของภาษายังทำความเข้าใจได้ยาก และขาดความยืดหยุ่น ทำให้การพัฒนาโปรแกรมให้มีความซับซ้อนนั้นมีความยุ่งยาก ส่งผลให้ได้รับความนิยมน้อยลงแม้ว่าจะมีความรวดเร็วในการประมวลผลก็ตาม ต่อมาแม้จะมีการพัฒนาภาษาคอมพิวเตอร์ต่างๆ ขึ้นเพื่อสนับสนุนการพัฒนา

โปรแกรมแล้ว แต่ก็ยังมีปัญหาเรื่องความยุ่งยากในการพัฒนาให้โปรแกรมมีความซับซ้อนอยู่ แต่ภาษาไพทอนที่ผู้วิจัยเลือกนำมาพัฒนาในโครงงานนี้นั้นสามารถแก้ปัญหาดังกล่าวได้ เนื่องจากไพทอนเป็นภาษาที่ง่ายต่อการทำความเข้าใจ มีโครงสร้างภาษาที่ยืดหยุ่นและกระชับ และมีไลบรารี่ที่มีประสิทธิภาพสนับสนุนเป็นจำนวนมาก นอกจากนี้แล้วไพทอนยังสนับสนุนแนวคิดการเขียนโปรแกรมเชิงวัตถุ ทำให้สามารถเข้าถึงชุดคำสั่งโปรแกรมได้ง่าย มีการจัดการหน่วยความจำอัตโนมัติ ทำให้การประมวลผลของโปรแกรมเป็นไปอย่างมีประสิทธิภาพ สามารถทำงานบนระบบปฏิบัติการได้หลากหลาย ซึ่งแก้ปัญหาการพัฒนาโปรแกรมที่มีมาตั้งแต่ในอดีตได้ อีกทั้งยังมีประสิทธิภาพด้านการคำนวณด้านวิศวกรรม และเป็น Open-source สนับสนุนการนำโปรแกรมไปพัฒนาต่อไปอย่างมาก ซึ่งตรงตามจุดประสงค์ของโครงงานที่ต้องการพัฒนาต้นแบบของโปรแกรมวิเคราะห์โครงสร้างสามมิติสำหรับโครงสร้าง Truss และ Frame เพื่อการนำไปวิจัยและพัฒนาต่อโดยผู้สนใจ โดยขอบเขตของโปรแกรมในโครงงานนี้สามารถแก้โจทย์ปัญหาการวิเคราะห์โครงสร้างแบบ Linear-elastic, Static analysis, Space Truss และ Space Frame เท่านั้น (ไม่สามารถวิเคราะห์โครงสร้างแบบผสมระหว่าง Truss และ Frame ได้)

2. งานวิจัยในอดีตและทฤษฎีที่เกี่ยวข้อง

2.1 งานวิจัยในอดีต

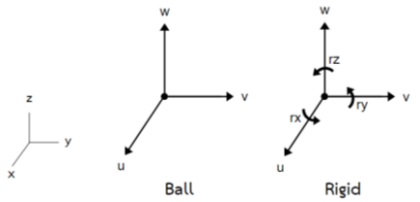
ตั้งแต่ยุคเริ่มต้นของคอมพิวเตอร์ ก็มีการพัฒนาโปรแกรมวิเคราะห์และออกแบบโครงสร้าง เพื่อนำมาช่วยในการทำงานด้านวิศวกรรม โดยได้นำภาษาคอมพิวเตอร์ในแต่ละยุคมาประกอบกับทฤษฎีการวิเคราะห์โครงสร้างเพื่อพัฒนาโปรแกรม สำหรับภาษาคอมพิวเตอร์นั้น ถูกนำมาใช้งานต่างกันไปในแต่ละยุค โดยภาษาที่มีการนำมาพัฒนาโปรแกรมด้านโครงสร้างเป็นครั้งแรกคือภาษา Fortran ซึ่งมีประสิทธิภาพในการคำนวณด้านวิศวกรรม แต่ยังมีข้อจำกัดทางคอมพิวเตอร์ในเรื่องหน่วยความจำ จึงเป็นอุปสรรคในการวิเคราะห์โครงสร้างที่มีขนาดใหญ่ [1] ต่อมาได้มีการนำภาษา Pascal เข้ามาช่วยในการพัฒนาโปรแกรมเพื่อจัดการปัญหาเรื่องหน่วยความจำแรมที่มีจำกัด โดยใช้หลักการ Arrays of pointers [2] ในการดึงข้อมูลจากหน่วยความจำช่วยมาแทนที่ข้อมูลที่ไม่ได้ใช้งานขณะประมวลผลบนหน่วยความจำแรม ส่งผลให้การจัดการหน่วยความจำเป็นไปอย่างมีประสิทธิภาพมากขึ้น นอกจากนี้ยังได้มีการนำเทคโนโลยีด้านกราฟฟิคเข้ามาพัฒนาโปรแกรมเพื่อให้ผู้ใช้งานสามารถเข้าใจและใช้งานได้สะดวกขึ้น ตั้งแต่ยุคภาษา Fortran [3] ยกตัวอย่างโปรแกรมวิเคราะห์โครงสร้างอื่นๆ ที่มีการพัฒนาจากภาษาคอมพิวเตอร์ที่แตกต่างกัน เช่น ANSTRUK [4] พัฒนาด้วยภาษาเบสิก, ชุดคำสั่งการวิเคราะห์โครงสร้างจักรยาน (Frame) ซึ่งพัฒนาจากภาษา C [5], PSA [6] และ JSM [7] พัฒนาด้วยภาษาจาวา เป็นต้น โดยในปัจจุบันก็ได้เริ่มมีการพัฒนาโปรแกรมวิเคราะห์โครงสร้างจากภาษาไพทอนอย่าง anaStruct [8] ที่สามารถวิเคราะห์ได้เฉพาะโครงสร้างสองมิติ ในส่วนของทฤษฎีการวิเคราะห์โครงสร้างที่มักถูกนำไปใช้ในการพัฒนาโปรแกรมนั้นโดยส่วนใหญ่แล้วเป็น Direct Stiffness Method เนื่องจากเป็นวิธีที่มีความเหมาะสมที่สุดในการนำมาประมวลผลบนคอมพิวเตอร์ เพราะเป็นการวิเคราะห์โดยใช้เมตริกซ์เป็นหลักนั่นเอง

2.2 ทฤษฎีที่เกี่ยวข้อง

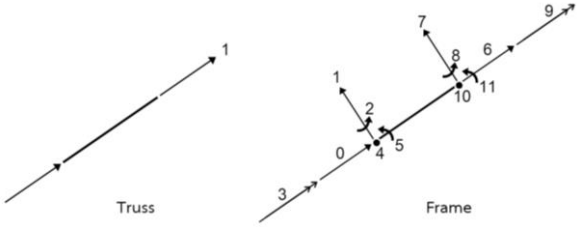
2.2.1 วิธีสติฟเนสโดยตรง [9]

การวิเคราะห์โครงสร้างด้วยวิธี Direct Stiffness Method เป็นการวิเคราะห์โครงสร้างโดยใช้เมตริกซ์ เพื่อหาแรงภายในและระยะการเคลื่อนที่ของโครงสร้าง โดยตัวแปรที่ต้องการทราบค่าเป็นลำดับแรกคือ ระยะการเคลื่อนที่ของจุดต่อ (Joint Displacement) ซึ่งสามารถหาค่าได้โดยการแก้สมการเมตริกซ์ของโครงสร้าง จากนั้นจึงนำระยะการเคลื่อนที่ของจุดต่อไปหาแรงภายในของชิ้นส่วนจากความสัมพันธ์ระหว่างแรงกับระยะการเคลื่อนที่ (Compatibility) ของโครงสร้าง โดยจะอธิบายขั้นตอนการวิเคราะห์โครงสร้าง Truss และ Frame ในโปรแกรม ดังนี้

1) แยกโครงสร้างออกเป็นส่วนย่อย คือ จุดต่อ (nodes) และชิ้นส่วน (elements) โดยทำการกำหนดจำนวน Degree of freedom ตามประเภทของจุดต่อ



รูปที่ 1 ประเภทของจุดต่อ (ตามแกนหลักของโครงสร้าง)



รูปที่ 2 ประเภทของชิ้นส่วน (ตามแกนของชิ้นส่วน)

2) คำนวณค่าสติฟเนสเมตริกซ์ของแต่ละชิ้นส่วน k_L ตามแนวแกนของชิ้นส่วน (Local coordinate)

$$k_L = \frac{EA}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

รูปที่ 3 สติฟเนสเมตริกซ์ของชิ้นส่วน Truss 3 มิติ

$$\mathbf{k} = \frac{E}{L^3}\begin{bmatrix} AL^2 & 0 & 0 & 0 & 0 & 0 & -AL^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12I_z & 0 & 0 & 0 & 6LI_z & 0 & -12I_z & 0 & 0 & 0 & 6LI_z \\ 0 & 0 & 12I_y & 0 & -6LI_y & 0 & 0 & 0 & -12I_y & 0 & -6LI_y & 0 \\ 0 & 0 & 0 & \frac{GJL^2}{E} & 0 & 0 & 0 & 0 & 0 & -\frac{GJL^2}{E} & 0 & 0 \\ 0 & 0 & -6LI_y & 0 & 4L^2I_y & 0 & 0 & 0 & 6LI_y & 0 & 2L^2I_y & 0 \\ 0 & 6LI_z & 0 & 0 & 0 & 4L^2I_z & 0 & -6LI_z & 0 & 0 & 0 & 2L^2I_z \\ -AL^2 & 0 & 0 & 0 & 0 & 0 & AL^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -12I_z & 0 & 0 & 0 & -6LI_z & 0 & 12I_z & 0 & 0 & 0 & -6LI_z \\ 0 & 0 & -12I_y & 0 & 6LI_y & 0 & 0 & 0 & 12I_y & 0 & 6LI_y & 0 \\ 0 & 0 & 0 & -\frac{GJL^2}{E} & 0 & 0 & 0 & 0 & 0 & \frac{GJL^2}{E} & 0 & 0 \\ 0 & 0 & -6LI_y & 0 & 2L^2I_y & 0 & 0 & 0 & 6LI_y & 0 & 4L^2I_y & 0 \\ 0 & 6LI_z & 0 & 0 & 0 & 2L^2I_z & 0 & -6LI_z & 0 & 0 & 0 & 4L^2I_z \end{bmatrix}$$

รูปที่ 4 สติฟเนสเมตริกซ์ของชิ้นส่วน Frame 3 มิติ

3) เปลี่ยนแนวแกนสติฟเนสเมตริกซ์ของชิ้นส่วน จากแกน Local มาเป็นแกน Global ด้วยการใช้ Transformation matrix T ดังแสดงในสมการที่ (1)

$$k_G = T^T k_L T \quad (1)$$

$$T = \begin{bmatrix} cos\theta x & cos\theta y & cos\theta z & 0 & 0 & 0 \\ 0 & 0 & 0 & cos\theta x & cos\theta y & cos\theta z \end{bmatrix}$$

รูปที่ 5 Transformation matrix T ของ Truss 3 มิติ

$$\mathbf{T} = \begin{bmatrix} r_{xX} & r_{xY} & r_{xZ} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r_{yX} & r_{yY} & r_{yZ} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r_{zX} & r_{zY} & r_{zZ} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_{xX} & r_{xY} & r_{xZ} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_{yX} & r_{yY} & r_{yZ} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_{zX} & r_{zY} & r_{zZ} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_{xX} & r_{xY} & r_{xZ} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_{yX} & r_{yY} & r_{yZ} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_{zX} & r_{zY} & r_{zZ} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{xX} & r_{xY} & r_{xZ} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{yX} & r_{yY} & r_{yZ} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{zX} & r_{zY} & r_{zZ} \end{bmatrix}$$

รูปที่ 6 Transformation matrix T ของ Frame 3 มิติ

4) นำค่าสติฟเนสเมตริกซ์ของแต่ละชิ้นส่วนตามแกนโครงสร้างหลัก k_G มาประกอบรวมกันไว้ในสติฟเนสเมตริกซ์ของโครงสร้าง K ตามตำแหน่งของ Degree of freedom ที่ตรงกันระหว่างชิ้นส่วนกับโครงสร้าง

5) กำหนดเงื่อนไขของโครงสร้างให้สอดคล้องกับเงื่อนไขของ Support และกำหนดแรงภายนอกตามแนวแกน Global

6) ทำการแก้สมการเมตริกซ์ดังแสดงในสมการที่ (2) โดยใช้ระเบียบวิธี Gauss elimination เพื่อหาค่าการเคลื่อนที่ที่จุดต่อของโครงสร้าง

$$F = Kv \quad (2)$$

เมื่อ F คือเมตริกซ์ของแรงภายนอก, K คือสติฟเนสเมตริกซ์ของโครงสร้าง และ v คือเมตริกซ์ของการเคลื่อนที่ที่จุดต่อของโครงสร้าง

7) เปลี่ยนแนวแกนของระยะการเคลื่อนที่ที่จุดต่อ จากแกน Global มาเป็นแกน Local ดังแสดงในสมการที่ (3) โดยใช้ Transformation matrix T เช่นเดียวกับการแปลงสติฟเนสเมตริกซ์

$$u = Tv \quad (3)$$

เมื่อ u คือระยะการเคลื่อนตัวของจุดต่อในแนวแกน Local และ v คือระยะการเคลื่อนตัวของจุดต่อในแนวแกน Global

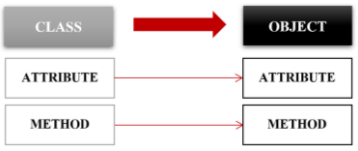
8) คำนวณค่าแรงภายในของแต่ละชิ้นส่วนดังแสดงในสมการที่ (4)

$$Q = k_L u \quad (4)$$

เมื่อ Q คือแรงภายในชิ้นส่วน, k_L คือสติฟเนสเมตริกซ์ตามแนวแกน Local และ u คือระยะการเคลื่อนตัวของจุดต่อในแนวแกน Local

2.2.2 การเขียนโปรแกรมเชิงวัตถุ [10]

โปรแกรมในโครงงานนี้พัฒนาโดยใช้แนวคิดการเขียนโปรแกรมเชิงวัตถุ คือการเขียนชุดคำสั่งของโปรแกรมโดยมองว่าระบบมีคลาสอันเป็นต้นแบบสำหรับการสร้างวัตถุขึ้น ซึ่งวัตถุนั้นจะมีคุณสมบัติ (Attributes) และการทำงาน (Method) พื้นฐานตามคลาสของวัตถุนั้นๆ

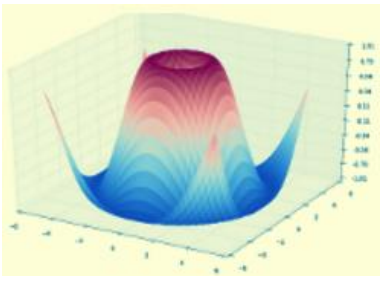


รูปที่ 7 การสร้างวัตถุจากคลาส

สำหรับหลักการการเขียนโปรแกรมเชิงวัตถุที่นำมาใช้พัฒนาโปรแกรมประกอบไปด้วย Encapsulation, Inheritance และ Polymorphism โดยแบ่งองค์ประกอบของโครงสร้างออกเป็นคลาสต่างๆ ซึ่งมีการใช้คุณสมบัติที่กล่าวมาข้างต้นดังจะอธิบายด้วยตัวอย่างของคลาส Element, Truss และ Frame เนื่องจากคลาส Element เป็นคลาสแม่ (Super class) ของทั้งคลาส Truss และคลาส Frame ทำให้คลาส Truss และคลาส Frame ได้รับสืบทอดคุณสมบัติและการทำงานมาจากคลาส Element ทั้งหมด โดยสามารถเพิ่มเติมคุณสมบัติและการทำงานเฉพาะตัวเข้าไปได้ ซึ่งเป็นหลักการของ Inheritance และแม้ว่าคลาส Truss และคลาส Frame จะเป็นคลาสลูก (Sub class) ของคลาส Element เหมือนกัน แต่ไม่จำเป็นต้องมีผลลัพธ์การทำงานและลักษณะเฉพาะตัวที่เหมือนกัน เนื่องจากการเพิ่มเติม Attributes หรือ Method เฉพาะตัวของคลาสลูกเอง อันเป็นหลักการ Polymorphism นอกจากนี้ วัตถุใดๆ ที่ถูกสร้างจากคลาส Truss จะมีคุณสมบัติและเรียกใช้คำสั่งการทำงานได้เฉพาะภายในคลาสของตัวเองและคลาสแม่อย่างคลาส Element เท่านั้น ไม่สามารถเลือกใช้คำสั่งของคลาส Frame ได้ นั่นคือหลักการที่สำคัญอีกอย่างของการเขียนโปรแกรมเชิงวัตถุอย่าง Encapsulation นั่นเอง

2.2.3 ภาษาไพทอน [11]

ภาษาไพทอนเป็นภาษาคอมพิวเตอร์ชั้นสูงที่ถูกพัฒนาโดยมีจุดประสงค์สำคัญคือการทำให้เป็นภาษาที่สามารถทำความเข้าใจได้ง่าย โครงสร้างภาษาสั้น กระชับ และสนับสนุนการเขียนชุดคำสั่งให้เข้าใจได้ง่ายทั้งในโปรแกรมขนาดเล็กไปจนถึงโปรแกรมขนาดใหญ่ นอกจากนั้น ไพทอนยังเป็นภาษาที่มีข้อยกเว้นของโครงสร้างภาษาน้อยกว่าภาษาคอมพิวเตอร์อื่นทำให้การพัฒนาโปรแกรมมีความยืดหยุ่นมากกว่า มีการจัดการหน่วยความจำอย่างอัตโนมัติ สามารถจัดการพื้นที่หน่วยความจำที่ไม่ต่อเนื่องได้อย่างมีประสิทธิภาพ ส่งผลให้การประมวลผลมีประสิทธิภาพ สนับสนุนการเขียนโปรแกรมเชิงวัตถุ โปรแกรมจึงมีชุดคำสั่งที่ไม่ยาวมากนัก จึงช่วยให้การเข้าถึงชุดคำสั่งต่างๆ ในโปรแกรมเป็นไปอย่างสะดวก สามารถทำงานบนระบบปฏิบัติการที่ต่างกันได้โดยไม่ต้องทำการแก้ไขชุดคำสั่งทุกครั้ง เป็น Open-source เปิดโอกาสให้บุคคลจากหลายแหล่งสามารถเข้าถึงและนำโปรแกรมไปพัฒนาต่อไป มีประสิทธิภาพในการประมวลด้านวิศวกรรม และมีไลบรารี่จำนวนมากที่มีคุณภาพ เช่น Matplotlib ช่วยในการวิเคราะห์ข้อมูลและสร้างโมเดล, Numpy ฟังก์ชันทางคณิตศาสตร์ชั้นสูง ช่วยในการคำนวณด้านวิทยาศาสตร์อย่างมีประสิทธิภาพ, Math ฟังก์ชันคณิตศาสตร์พื้นฐาน และค่าคงที่ทางวิทยาศาสตร์ เป็นต้น ด้วยข้อดีของไพทอนที่กล่าวมาข้างต้นจึงเป็นประโยชน์ต่อการพัฒนาโปรแกรมต่อไป

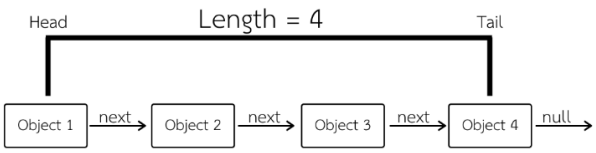


รูปที่ 8 ตัวอย่างการขึ้นโมเดลโดยใช้ Matplotlib [12]

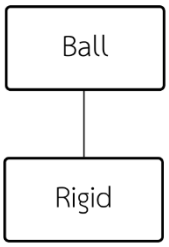
3. องค์ประกอบของโปรแกรม

การเขียนคำสั่งการทำงานของโปรแกรมสามารถแบ่งออกเป็น Class ซึ่งแต่ละ Class จะทำหน้าที่แตกต่างกันโดยจะประกอบด้วย Class ทั้งหมดดังนี้: Ball, Rigid, Truss, Frame, Point, Material, Section, Matrix, Structure, StructureAssemble, UniformLoad และ matplot

Class Ball และ Class Rigid จะทำหน้าที่ในการสร้างจุดต่อ หรือ Node ให้กับชิ้นส่วน Truss และ Frame ตามลำดับ โดยที่จะเก็บข้อมูลตำแหน่งพิกัดและจำนวน Degree of Freedom ซึ่ง Class Rigid จะเป็นคลาสลูก หรือ Subclass ของ Class Ball แต่จะเพิ่มข้อมูลการหมุนรอบแกน x, y และ z ของ Node โดยที่การสร้าง Object จาก Class Ball และ Class Rigid นั้น จะมีลักษณะการสร้างเป็นแบบ Linked List นั่นคือทุก ๆ Object จะมีลักษณะต่อกันเป็นสาย สามารถเรียกใช้ Object อย่างต่อเนื่องได้

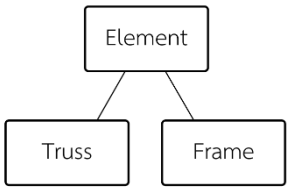


รูปที่ 9 Object ที่มีลักษณะเป็นแบบ Linked List



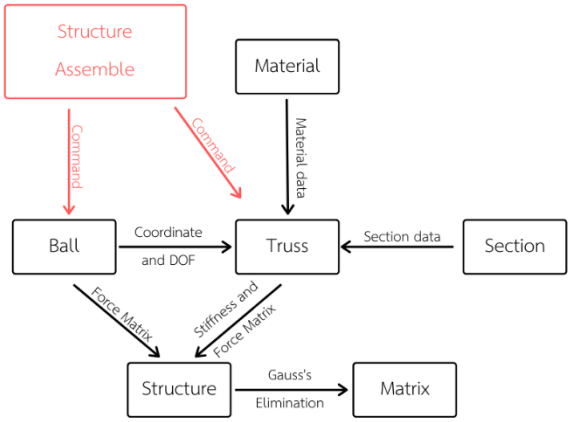
รูปที่ 10 ความสัมพันธ์ระหว่าง Class Ball และ Class Rigid

หลังจากกำหนดจุด Node ของชิ้นส่วนแล้ว การสร้างชิ้นส่วน member จะทำการสร้างผ่าน Class Truss และ Class Frame ซึ่งทั้ง 2 Class จะเป็น subclass ของ Class Element ทำให้ member ที่สร้างขึ้นสามารถใช้คำสั่งของ Class Element ได้ นั่นก็คือคำสั่งที่ใช้ในการประกอบ Matrix ของโครงสร้างจากชิ้นส่วน โดยที่ Class Truss และ Class Frame จะทำการคำนวณ stiffness matrix ของชิ้นส่วนจากจุดปลายของชิ้นส่วน, ข้อมูลวัสดุ และ ข้อมูลหน้าตัด ซึ่งจะกำหนดโดยใช้ Class Material และ Class Section ตามลำดับ

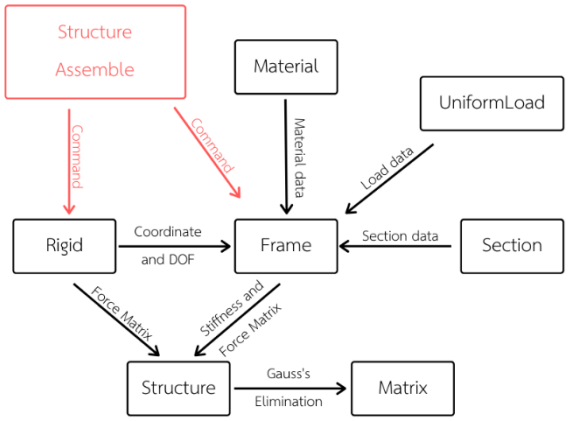


รูปที่ 11 ความสัมพันธ์ระหว่าง Class Element, Class Truss และ Class Frame

การกำหนด Load หรือแรงภายนอกจะสามารถทำได้ 2 วิธี วิธีที่ 1 คือการสร้าง Load ที่จุดต่อหรือ node โดยจะสร้างผ่านคำสั่งที่เขียนไว้ภายใน Class Ball และ Class Rigid วิธีที่ 2 คือการสร้าง uniform load หรือแรงกระจายคงที่ บนชิ้นส่วน Frame โดยที่จะสร้างจาก Class UniformLoad



รูปที่ 12 ขั้นตอนการวิเคราะห์โครงสร้าง Truss ของโปรแกรม



รูปที่ 13 ขั้นตอนการวิเคราะห์โครงสร้าง Frame ของโปรแกรม

การประกอบ stiffness matrix ของโครงสร้างจาก stiffness matrix ของชิ้นส่วนจะใช้คำสั่งของ Class StructureAssemble ซึ่งทำหน้าที่ในการประกอบ node และ member ที่สร้างขึ้นเข้าเป็นโครงสร้าง และเก็บข้อมูลของ stiffness matrix และ force matrix ใน Class Structure เพื่อป้องกันการเรียกใช้งานที่ซ้ำซ้อนของโปรแกรม แล้วจึงทำการวิเคราะห์โครงสร้างด้วยการแก้สมการ stiffness matrix ซึ่งจะใช้วิธี Gauss's Elimination ในการแก้สมการ หลังจากนั้นจึงแสดงผลการวิเคราะห์โดยสามารถแบ่งออกเป็น 2 ส่วน ส่วนที่ 1 แสดงผลเป็นไฟล์ประเภท CSV ที่สามารถเปิดด้วยโปรแกรม Microsoft Excel ซึ่งคำสั่งที่ใช้สำหรับแสดงผลการวิเคราะห์จะเขียนอยู่ใน Class StructureAssemble และส่วนที่ 2 แสดงรูปโครงสร้าง 3 มิติซึ่งคำสั่งที่ใช้จะเขียนอยู่ใน Class matplot

ตัวอย่าง code คำสั่งของ Class matplot

```
class matplot():
    def show():
        fig = plt.figure()
        ax = fig.add_subplot(111,projection='3d')
        truss = Truss.start
        while truss != None:
            X=[ ];Y=[ ];Z=[ ]
            truss.plot(X,Y,Z); ax.plot(X,Y,Z)
            truss = truss.next
        frame = Frame.start
        while frame != None:
            X=[ ];Y=[ ];Z=[ ]
            frame.plot(X,Y,Z)
            ax.plot(X,Y,Z)
            frame = frame.next
        plt.show()
```

4. ผลการดำเนินงาน

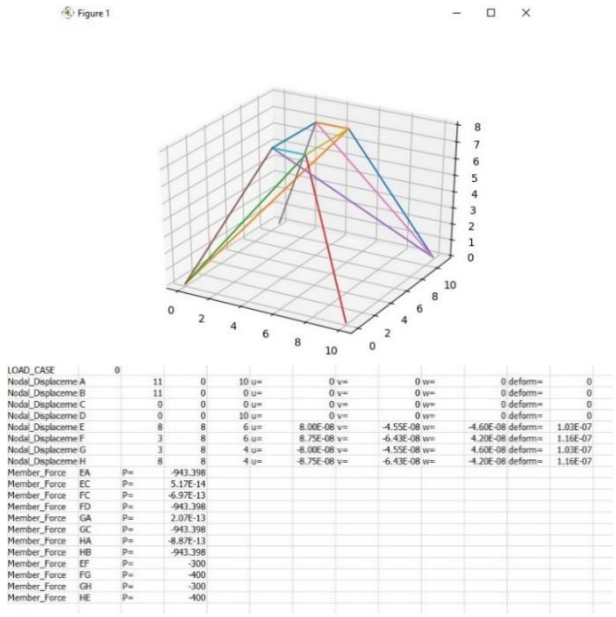
ในการวิเคราะห์โครงสร้างด้วยโปรแกรมภาษา Python ที่ได้เขียนขึ้นนั้น จะต้องมีการใส่ข้อมูล (input) เพื่อเป็นการกำหนดลักษณะของโครงสร้างดังนี้ การกำหนดพิกัดจุดต่อหรือ Node, การสร้างชิ้นส่วน member ที่เชื่อมระหว่างจุดต่อ, การกำหนดประเภทวัสดุหรือคุณสมบัติของวัสดุ, การกำหนดประเภทหน้าตัดหรือคุณสมบัติของหน้าตัด,การใส่ load ที่โครงสร้างหรือจุดต่อ และ การกำหนดประเภทของ support แล้วจึงทำการวิเคราะห์โครงสร้าง ซึ่งจะได้ผลการวิเคราะห์ (output) จากโปรแกรมคือ ระยะการเคลื่อนที่ (displacement) ที่จุดต่อ, แรงภายในของชิ้นส่วน (member force) และ รูปโครงสร้าง 3 มิติ

4.1 ตัวอย่างโครงสร้าง Truss 3 มิติ (Space Truss)[13]

การใส่ข้อมูล (input) ในลักษณะของ Python code

```
mat = Material()
mat.Steel()
sect= Section(1.0)
A = Ball(10.0,11.0,0.0,'A'); B = Ball(0.0,11.0,0.0,'B')
C = Ball(0.0,0.0,0.0,'C'); D = Ball(10.0,0.0,0.0,'D')
E = Ball(6.0,8.0,8.0,'E'); F = Ball(6.0,3.0,8.0,'F')
G = Ball(4.0,3.0,8.0,'G'); H = Ball(4.0,8.0,8.0,'H')
EA = Truss(E,A,mat,sect,'EA'); EC = Truss(E,C,mat,sect,'EC')
FC = Truss(F,C,mat,sect,'FC'); FD = Truss(F,D,mat,sect,'FD')
GA = Truss(G,A,mat,sect,'GA'); GC = Truss(G,C,mat,sect,'GC')
HA = Truss(H,A,mat,sect,'HA'); HB = Truss(H,B,mat,sect,'HB')
EF = Truss(E,F,mat,sect,'EF'); FG = Truss(F,G,mat,sect,'FG')
GH = Truss(G,H,mat,sect,'GH'); HE = Truss(H,E,mat,sect,'HE')
E.load(0,0.0,-800.0,0.0); F.load(0,0.0,-800.0,0.0)
G.load(0,0.0,-800.0,0.0); H.load(0,0.0,-800.0,0.0)
A.socket(); B.socket(); C.socket(); D.socket()
Structure.init()
t5 = StructureAssemble()
t5.solve()
t5.PrintCSV('T5')
matplot.show()
```

ผลการวิเคราะห์โครงสร้าง (output)



รูปที่ 14 ผลการวิเคราะห์โครงสร้าง Truss ด้วยโปรแกรมภาษา Python

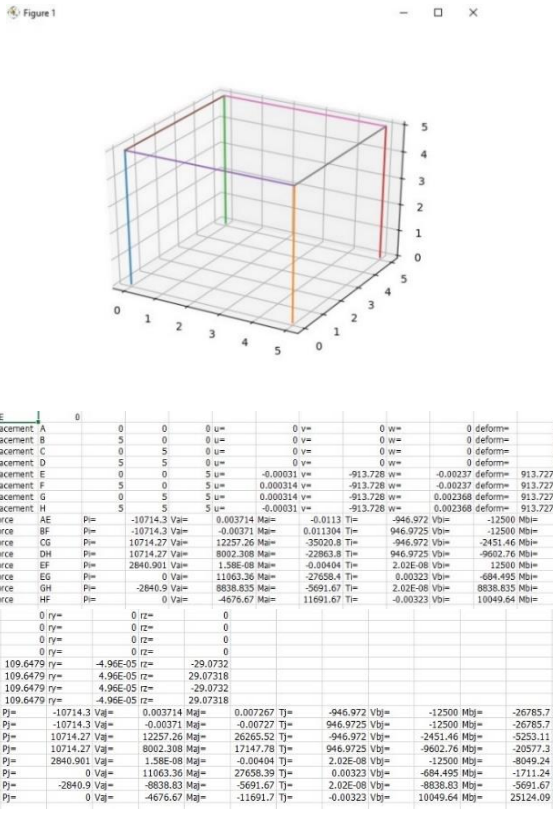
4.2 ตัวอย่างโครงสร้าง Frame 3 มิติ

การใส่ข้อมูล (input) ในลักษณะของ Python code

```
mat = Material()
mat.Steel()
sect = RoundBar(12)
w = UniformLoad(0,-5.0e3,0)
A = Rigid(0,0,0,"A"); B = Rigid(5,0,0,"B")
C = Rigid(0,5,0,"C"); D = Rigid(5,5,0,"D")
E = Rigid(0,0,5,"E"); F = Rigid(5,0,5,"F")
G = Rigid(0,5,5,"G"); H = Rigid(5,5,5,"H")
P = Point(-1,0,0,"P")
AE = Frame(A,E,P,mat,sect,"AE")
BF = Frame(B,F,P,mat,sect,"BF")
CG = Frame(C,G,P,mat,sect,"CG")
DH = Frame(D,H,P,mat,sect,"DH")
EF = Frame(E,F,P,mat,sect,"EF")
EG = Frame(E,G,P,mat,sect,"EG")
GH = Frame(G,H,P,mat,sect,"GH")
HF = Frame(H,F,P,mat,sect,"HF")
EF.load(w); EG.load(w); GH.load(w); HF.load(w)
```

```
A.fixed(); B.fixed(); C.fixed(); D.fixed()
Structure.init()
s=StructureAssemble()
s.solve()
s.PrintCSV("F1")
matplot.show()
```

ผลการวิเคราะห์โครงสร้าง (output)



LOAD CASE		0												
Nodal_Displacement	A		0	0	0 u=		0 v=		0 w=		0 deform=			
Nodal_Displacement	B		5	0	0 u=		0 v=		0 w=		0 deform=			
Nodal_Displacement	C		0	5	0 u=		0 v=		0 w=		0 deform=			
Nodal_Displacement	D		5	5	0 u=		0 v=		0 w=		0 deform=			
Nodal_Displacement	E		0	0	5 u=	-0.00031	v=	-913.728	w=	-0.00237	deform=	913.727		
Nodal_Displacement	F		5	0	5 u=	0.000314	v=	-913.728	w=	-0.00237	deform=	913.727		
Nodal_Displacement	G		0	5	5 u=	0.000314	v=	-913.728	w=	0.002368	deform=	913.727		
Nodal_Displacement	H		5	5	5 u=	-0.00031	v=	-913.728	w=	0.002368	deform=	913.727		
Member_Force	AE	Pi=	-10714.3	Vai=	0.003714	Mai=	-0.0113	Ti=	-946.972	Vbi=	-12500	Mbi=		
Member_Force	BF	Pi=	-10714.3	Vai=	-0.00371	Mai=	0.011304	Ti=	946.9725	Vbi=	-12500	Mbi=		
Member_Force	CG	Pi=	10714.27	Vai=	12257.26	Mai=	-35020.8	Ti=	-946.972	Vbi=	-2451.46	Mbi=		
Member_Force	DH	Pi=	10714.27	Vai=	8002.308	Mai=	-22863.8	Ti=	946.9725	Vbi=	-9602.76	Mbi=		
Member_Force	EF	Pi=	2840.901	Vai=	1.58E-08	Mai=	-0.00404	Ti=	2.02E-08	Vbi=	12500	Mbi=		
Member_Force	EG	Pi=	0	Vai=	11063.36	Mai=	-27658.4	Ti=	0.00323	Vbi=	-684.495	Mbi=		
Member_Force	GH	Pi=	-2840.9	Vai=	8838.835	Mai=	-5691.67	Ti=	2.02E-08	Vbi=	8838.835	Mbi=		
Member_Force	HF	Pi=	0	Vai=	-4676.67	Mai=	11691.67	Ti=	-0.00323	Vbi=	10049.64	Mbi=		
nx=	0	ry=	0	rz=	0									
nx=	0	ry=	0	rz=	0									
nx=	0	ry=	0	rz=	0									
nx=	0	ry=	0	rz=	0									
nx=	109.6479	ry=	-4.96E-05	rz=	-29.0732									
nx=	109.6479	ry=	4.96E-05	rz=	29.07318									
nx=	109.6479	ry=	4.96E-05	rz=	-29.0732									
nx=	109.6479	ry=	-4.96E-05	rz=	29.07318									
35714.32	Pj=	-10714.3	Vaj=	0.003714	Maj=	0.007267	Tj=	-946.972	Vbj=	-12500	Mbj=	-26785.7		
35714.32	Pj=	-10714.3	Vaj=	-0.00371	Maj=	-0.00727	Tj=	946.9725	Vbj=	-12500	Mbj=	-26785.7		
7004.165	Pj=	10714.27	Vaj=	12257.26	Maj=	26265.52	Tj=	-946.972	Vbj=	-2451.46	Mbj=	-5253.11		
27436.49	Pj=	10714.27	Vaj=	8002.308	Maj=	17147.78	Tj=	946.9725	Vbj=	-9602.76	Mbj=	-20577.3		
-8049.24	Pj=	2840.901	Vaj=	1.58E-08	Maj=	-0.00404	Tj=	2.02E-08	Vbj=	-12500	Mbj=	-8049.24		
1711.238	Pj=	0	Vaj=	11063.36	Maj=	27658.39	Tj=	0.00323	Vbj=	-684.495	Mbj=	-1711.24		
-5691.67	Pj=	-2840.9	Vaj=	-8838.83	Maj=	-5691.67	Tj=	2.02E-08	Vbj=	-8838.83	Mbj=	-5691.67		
-25124.1	Pj=	0	Vaj=	-4676.67	Maj=	-11691.7	Tj=	-0.00323	Vbj=	10049.64	Mbj=	25124.09		

รูปที่ 15 ผลการวิเคราะห์โครงสร้าง Frame ด้วยโปรแกรมภาษา Python

จากการเปรียบเทียบผลการวิเคราะห์โครงสร้าง Truss 3 มิติ ด้วยโปรแกรมภาษา Python และโครงสร้าง Frame 3 มิติ ที่วิเคราะห์ด้วยโปรแกรม JSM และโปรแกรมภาษา Python จะพบว่าผลการวิเคราะห์มีค่าตรงกัน ดังนั้นจึงสามารถนำโปรแกรมวิเคราะห์โครงสร้างด้วยภาษา Python ไปวิเคราะห์โครงสร้างเพื่อหาระยะการเคลื่อนที่ของจุดต่อ และแรงภายในของโครงสร้างได้ ตามที่ได้ตั้งวัตถุประสงค์ไว้

5. บทสรุป

การพัฒนาโปรแกรมวิเคราะห์โครงสร้างสามมิติสำหรับโครงสร้าง Truss และ Frame ด้วยภาษาไพทอนในโครงงานนี้ สามารถนำไปพัฒนาต่อยอดสำหรับผู้ที่สนใจได้ต่อไป โดยช่วยลดระยะเวลาในการพัฒนาลง เนื่องจากประสิทธิภาพของภาษาไพทอนนั้นช่วยสนับสนุนการพัฒนาความสามารถ (Features) ของโปรแกรมที่มีความซับซ้อน และสนับสนุนการพัฒนาโปรแกรมให้สามารถวิเคราะห์โครงสร้างที่ซับซ้อนมากกว่า linear-elastic และ static problem ได้ง่ายขึ้น เช่น การแก้ปัญหาโครงสร้างแบบ non-linear และการทำโครงสร้างย่อย sub-structuring เป็นต้น นอกจากนั้นแล้วยังช่วยประหยัดเวลาในการเรียนรู้ภาษาคอมพิวเตอร์ในอดีตที่มีความยากต่อการเข้าใจ เนื่องจากภาษาคอมพิวเตอร์สมัยก่อนมีความใกล้เคียงกับภาษาเครื่องมาก ทำให้มีประสิทธิภาพด้านการคำนวณสูง เพราะคอมพิวเตอร์ไม่ต้องทำการแปลหลายชั้น ถึงแม้ว่าจะสามารถทำการคำนวณได้อย่างรวดเร็ว แต่ก็มีข้อจำกัดในหลายด้านที่เป็นอุปสรรคต่อการพัฒนาโปรแกรม อีกทั้งยังได้รับความนิยมน้อยลงในปัจจุบัน หรืออาจไม่ใช้งานเลย ดังนั้น เมื่อผู้พัฒนาไม่ต้องเสียเวลาในการเรียนรู้ภาษาคอมพิวเตอร์สมัยเก่าแล้วมุ่งศึกษาภาษาไพทอนอย่างจริงจังจะสามารถดึงความสามารถของภาษาไพทอนมาใช้ในการพัฒนาโปรแกรมได้อย่างเต็มประสิทธิภาพ โดยอาจสามารถพัฒนาชุดคำสั่งของไพทอนให้มีประสิทธิภาพมากจนมีความเร็วในการประมวลผลทัดเทียมกับภาษาคอมพิวเตอร์ยุคเก่าได้

สำหรับการพัฒนาโปรแกรมในลำดับถัดไปนั้น สามารถศึกษาหลักการทำงานของโปรแกรม และจัดรูปแบบการทำงานใหม่ให้มีประสิทธิภาพและกระชับมากยิ่งขึ้น เช่น ควรจัดให้คลาส Point เป็นคลาสแม่ต้นสายลำดับแรกของคลาสประเภท node (Ball & Rigid) เพื่อจัดระเบียบของคลาสเป็นต้น ต่อจากนั้นจึงพัฒนาโปรแกรมให้สามารถทำการวิเคราะห์โครงสร้าง 3 มิติแบบผสมระหว่างโครงสร้าง Truss และ Frame อีกทั้งยังสามารถศึกษาไลบรารี่ของไพทอนต่างๆ เพื่อนำมาพัฒนา Graphical user interface (GUI) ในการใส่ข้อมูล (input) เพื่อการกำหนดลักษณะของโครงสร้าง ในรูปแบบอื่น เนื่องจากการใส่ข้อมูลแบบใช้ Python code อาจทำให้ผู้ใช้ที่ไม่มีความรู้ด้านการเขียนโปรแกรม Python มาก่อน รู้สึกยากในการใช้งานตัวโปรแกรม และยังสามารถพัฒนาการแสดงผลในรูปแบบโมเดล 3 มิติแบบเป็นรูปทรงของชิ้นส่วน เช่น แสดงหน้าตัดของเหล็กหรือคอนกรีตเสริมเหล็ก ตามรูปแบบของหน้าตัดที่กำหนดไว้ ร่วมกับการแสดงผลการวิเคราะห์ในรูปแบบของ Graphic เช่น Shear Force Diagram, Bending Moment Diagram และ รูปการเคลื่อนตัวของโครงสร้าง นอกจากนี้ ยังสามารถพัฒนาโปรแกรมเพื่อนำผลการวิเคราะห์ไปออกแบบโครงสร้างเพื่อเลือกหน้าตัดที่เหมาะสมในขั้นตอนต่อไป

เอกสารอ้างอิง

[1] Harrison, H. (1965). *Computer analysis of plane frames*. (Civil Engineering, Lehigh University).

[2] Soriano, H. and Souza, S. (1991). *Structured types in pascal for structural analysis*. (Civil Engineering, Federal University of Rio de Janeiro).

[3] Rodgers, D. (1982). *Structural analysis aided by interactive computer graphics*. (Master's Thesis, Civil Engineering, Iowa State University).

[4] Rivani, A. and Nirmalawati. (n.d.). *Structural analysis program of plane frame with visual basic language.* (Civil Engineering, Tadulako University).

[5] Gupta, A. (n.d.). Finite element method structural analysis of bicycle (frame element) using c programming. (IIITD&M Kancheepuram).

[6] Patil, I. and Annigeri, S. (2016). Introduction to PSA as a free structural analysis software. (n.p.).

[7] Watanachai Smittakorn. (2008). *JSM as a toolbox for structural analysis and design applications.* (Civil Engineering, Chulalongkorn University).

[8] Ritchie46. (n.d.). *anaStruct.* (n.p.).

[9] Kassimali, A. (2012). *Matrix analysis of structures, second edition.* (United States of America.).

[10] *การเขียนโปรแกรมเชิงวัตถุ (Object oriented programming, OOP).* 2556. แหล่งที่มา: https://arit.rmutsv.ac.th/ (11 พฤษภาคม 2563)

[11] *ข้อดีข้อเสียของ Python.* (ออนไลน์). 2562. แหล่งที่มา: https://www.mindphp.com/ (11 พฤษภาคม 2563)

[12] *Python libraries.* (ออนไลน์). 2562. แหล่งที่มา: https://data-flair.training/ (19 พฤษภาคม 2563)

[13] Southern Illinois University Edwardsville (n.d.). *Space Truss*. (n.p.)