การพัฒนาโปรแกรมสำหรับวิเคราะห์โครงสร้าง 2 มิติ ด้วยภาษาไพธอน

A Program for Structural Analysis in 2-D using Python Language

นายชนาธิป พยอมแย้ม นายชลายุ ตั้งชีวินศิริกุล ผศ. ดร. วัฒนชัย สมิทธากร

ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย กรุงเทพฯ

บทคัดย่อ

โครงงานนี้ได้รับแรงบันดาลใจมาจากโปรแกรม JSM (Java Structural Method) ซึ่งเป็นโปรแกรมวิเคราะห์โครงสร้างเพื่อใช้ในการออกแบบและวิเคราะห์โครงสร้างด้วยภาษาจาวา ทางคณะผู้จัดทำเล็งเห็นว่าโปรแกรมที่กล่าวไปข้างต้นมีประโยชน์อยู่มาก และ จะเป็นประโยชน์ต่อผู้อื่น หรือ แม้กระทั่งนิสิตคนอื่นๆอีกมาก ทางคณะผู้จัดทำจึงได้นำโปรแกรมภาษาจาวามาทำการแปลเป็นภาษาไพธอน เนื่องจากเป็นภาษาที่เรียบง่ายและมีความยืดหยุ่น จึงทำให้ง่ายต่อการเรียนรู้อีกด้วย โครงงานนี้มีจุดมุ่งหมายในการทำให้โปรแกรมภาษาไพธอนที่ได้ทำการแปลมานี้เป็นพื้นฐานที่ดีในการนำไปพัฒนาต่อด้วยภาษาไพธอนต่อไปอีกด้วย

คำสำคัญ: ไพธอน, วิธีสติฟเนสตรง, การเขียนโปรแกรมเชิงวัตถุ

Abstract

This project got an inspiration from program JSM (Java Structural Method) Which is structural analysis and design program using Java. The program that mentioned before has many potentials and will be useful for other people or student. So, we have translated it to Python language because of Its flexibility and simplicity which makes it easy to learn. This project's purpose is to make this python program (PSM) is a good start for further development.

Keywords: Python, Direct Stiffness method, Object Oriented Programming

1. บทนำ

ในการวิเคราะห์โครงสร้าง และออกแบบโครงสร้าง วิศวกรโยธาและวิศวกรคอมพิวเตอร์ได้พยายามพัฒนาโปรแกรม หรือแอพพลิเคชั่นในการคำนวณตั้งแต่อดีตจนถึงปัจจุบันโดยภาษาคอมพิวเตอร์ที่นิยมยกตัวอย่าง เช่น Basic, Fortran ซึ่งช่วยยกระดับประสิทธิภาพในการคำนวณขึ้นไปในแต่ละยุคแต่ละสมัย โครงงานนี้ได้รับแรงบันดาลใจมาจากโปรแกรม Structural Analysis และ Optimization ชื่อว่า Java Structural Mechanics (JSM) ของผศ. ดร. วัฒนชัย สมิทธากร JSM ถูกเขียนขึ้นด้วยภาษา Java รองรับการทำงานแบบ Object Oriented Programming (OOP) การคำนวณทางโครงสร้างใช้หลักการ Finite Element Method (FEM) ซึ่งเข้ากันได้ดีกับวิธีการเขียนโปรแกรมเชิงวัตถุ OOP เป็นอย่างมาก เมื่อเวลาผ่านไปเทคโนโลยีก็เจริญก้าวหน้าขึ้น ภาษา Python เริ่มได้รับความนิยมมากขึ้นจนกลายเป็นภาษาที่ได้รับความนิยมอย่างสูงที่สุดในปัจจุบัน เพราะเป็นภาษาที่เข้าใจได้ง่ายมี syntax ของภาษาตรงตัว มี Library ของภาษาหลากหลาย สามารถเขียนโปรแกรมเชิงวัตถุได้เช่นเดียวกับภาษาจาวา มีส่วนเสริมสำเร็จรูปที่เรียกว่า Package หลายรูปแบบให้ได้เลือกใช้งาน และที่สำคัญคือไพธอนเป็นภาษาแบบ open-source จึงเป็นที่ดึงดูดนักพัฒนาโปรแกรมในสาขาต่างๆให้หันมาใช้ หันมาพัฒนาภาษานี้มากขึ้นเรื่อยๆ ในโครงงาน PSM (Python Structural Method) นี้จะเน้นไปที่การทำ Structural analysis truss และ frame 2 มิติ ซึ่งเป็นขั้นต้นของการออกแบบโครงสร้าง เพื่อหาแรงภายในชิ้นส่วนและการกระจัดจากแรงภายนอกที่กระทำต่อโครงสร้างด้วยวิธีสติฟเนสตรง Direct Stiffness Method ซึ่งเป็นส่วนหนึ่งของวิธีการคำนวณแบบ FEM วิธีการรวิเคราะห์โครงสร้างแบบสติฟเนสตรงถูกใช้อย่างแพร่หลายงานวิศวกรรมโยธา วิธีการคำนวณที่ใช้จะเป็นการเมตริกซ์ของแต่ละชิ้นส่วนขึ้นมาเรียกว่า Local Matrices หลังจากนั้นนำเมตริกซ์ของแต่ละชิ้นส่วนมา assemble กันภายใต้เงื่อนไขที่กำหนดไว้ จะได้เป็น Global Matrix ในโปรแกรมนี้จะไม่สามารถวิเคราะห์โครงสร้างประเภท substructure ได้ รวมไปถึงขอบเขตของโครงสร้างที่พิจารณาจะอยู่ในช่วง linear-elastic

เท่านั้น การแสดงผลจะใช้การแสดงผ่านไฟล์ Data csv ซึ่งสามารถเปิดด้วยโปรแกรม Microsoft Excel ได้

2. ทฤษฎีที่เกี่ยวข้อง

วิธีสติฟเนสตรงจะใช้การแบ่งโครงสร้างออกเป็น จุด Node และชิ้นส่วน Element กำหนด Degree of Freedom (DOF) ของแต่ละชิ้นส่วนคำนวณค่าสติฟเนสของแต่ละชิ้นส่วนจากคุณสมบัติทางด้านวัสดุและหน้าตัดของชิ้นส่วน

Local stiffness matrix ของ Link:

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

Local stiffness matrix ของ Beam:

$$k_e = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}$$

กำหนด loads และ structural supports ของโครงสร้างซึ่งเป็นเงื่อนไขเบื่องต้นของโครงสร้าง จากนั้นนำเมตริกซ์ Local มา assemble กันผ่านเมตริกซ์แปลง (Transform Matrix) สำหรับเปลี่ยนทิศทาง DOF ในระดับ Local ให้ตรงกับทิศทางตามแกนพิกัดแบบคาร์ทีเซียนในระดับ Global สมการสำคัญซึ่งใช้สำหรับการคำนวณคือสมการ Stiffness Equation

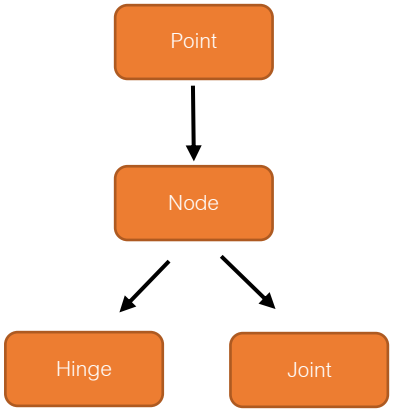
$$[K]\{D\} = \{F\}$$

หลังจากการแก้สมการด้วยวิธี Gaussian's elimination จะได้คำตอบของสมการคือ Displacement $\{D\}$ นำไปคำนวณผ่านเมตริกซ์แปลง และเมตริกซ์ k_eจะได้เป็นแรงภายในของแต่ละชิ้นส่วน

หลักการเขียนคำสั่งในโปรแกรมนี้จะใช้วิธีการเขียนโปรแกรมเชิงวัตถุ OOP ซึ่งใช้งานได้ดีในโปรแกรมที่ต้องใช้คำสั่งรูปแบบเดิมซ้ำหลายๆครั้ง การเขียนโปรแกรมเชิงวัตถุประกอบด้วยคุณลักษณะ 4 อย่างที่สำคัญคือ Encapsulation, Abstraction, Inheritance และ Polymorphism คุณลักษณะเหล่านี้จะถูกกำหนดอยู่ใน Class ข้อมูลที่ถูกสร้างจาก Class จะเรียกว่า object ซึ่งจะมีคุณลักษณะ และความสามารถเช่นเดียวกับ Class ทุกประการ หากต้องการจะเพิ่มเติมหรือแก้ไข module ใดๆ สามารถแก้ไขใน Class ได้เลย ไม่จำเป็นต้องแก้ไขใน objectแต่ละชิ้น

3. ส่วนประกอบของโปรแกรม

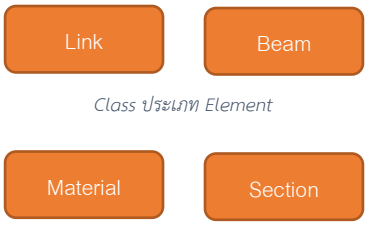
โปรแกรม PSM ใช้การแบ่ง Class ออกเป็น 2 กลุ่มใหญ่ๆคือกลุ่มของ Node และ Element



Class ประเภท Node

กลุ่มของ Node จะทำหน้าที่รับข้อมูลประเภทจุดพิกัดคาร์ทีเซียน (x, y) เข้ามาเก็บไว้ สำหรับนำไปใช้สร้าง Element. โดย Node มี *subclass* ทั้งหมด 2 ประเภท แบ่งตามจำนวน DOF คือ Hinge และ Joint

- Hinge มีจำนวน DOF = *2* คือ u *(แนวแกน x)* และ v *(แนวแกน y)* สามารถรับแรงกระทำแบบจุดได้ 2 แนวตามทิศทางของ *DOF* สามารถกำหนด *support* ประเภท pinned, rollerX, rollerY และ free ได้
- *Node* มีจำนวน DOF = 3 คือ u *(แนวแกน x)*, v *(แนวแกน y)* และ *r (ทิศ CCW)* สามารถรับแรงกระทำแบบจุดได้ 3 แนวตามทิศทางของ *DOF* สามารถกำหนด *support* ประเภท pinned, rollerX, rollerY, fixed, guideX, guideY และ free ได้



Class ประเภท Element

กลุ่มของ *Element* จะรับข้อมูลประเภท *Node* สำหรับใช้เป็นจุดต้นและจุดปลาย โดย Element มีทั้งหมด 2 ประเภทคือ Link และ Beam ซึ่งไม่ได้เป็น subclass

- Link รับข้อมูลประเภทจุดต้นและจุดปลายมาจาก Hinge รับข้อมูลประเภทคุณสมบัติของวัสดุจาก Material และคุณสมบัติประเภทรูปร่างหรือหน้าตัดจาก Section สำหรับนำไปคำนวณ

Local stiffness matrix k_e สามารถกำหนดคุณสมบัติประเภท unstretched length หรือ pretensioned ได้

- Beam รับข้อมูลประเภทจุดต้นและจุดปลายมาจาก Joint รับข้อมูลประเภทคุณสมบัติของวัสดุจาก Material และคุณสมบัติประเภทรูปร่างหรือหน้าตัดจาก Section สำหรับนำไปคำนวณ Local stiffness matrix k_e สามารถกำหนดแรงกระทำได้ 2 แบบคือ point load และ uniform load ไม่สามารถกำหนดแรงประเภท point moment ได้

Structure

Class Structure เป็นหัวใจสำคัญของโปรแกรมนี้ ทำหน้าที่รวบรวมข้อมูลจากเมตริกซ์ *Local* ของแต่ละชิ้นส่วนมา *assemble* เป็นเมตริกซ์ *Global* เมตริกซ์ที่สำคัญคือ Structure.a หรือเรียกว่า Global stiffness matrix $[K]$ ทำหน้าที่เก็บข้อมูลประเภทสติฟเนสของโครงสร้าง ตำแหน่งของข้อมูลจะถูกแบ่งตามลำดับของ DOF เมตริกซ์ที่สำคัญอีกหนึ่งอย่างคือ Structure.b หรือเรียกว่า Global external forces matrix $\{F\}$ ทำหน้าที่เก็บข้อมูลทุกส่วนที่เกิดจากการกระทำจากแรงภายนอกเช่น Nodal load, Support settlement, Pretensioned ใน Link และ point load, uniform load ใน Beam จากสมการ Stiffness Equation เมื่อแก้สมการจะได้คอบตอบคือ Displacement $\{D\}$ การแสดงผลในโปรแกรมนี้จะมีอยู่ทั้งหมด 2 แบบ แบบแรกคือการแสดงผลเฉพาะ load case กรณีเดียวเท่านั้นด้วยคำสั่ง printcase แบบที่สองคือการแสดงผลทุก load factor ที่ป้อนเข้าไปด้วยคำสั่ง printout จะแสดงผลของ member force และ nodal displacement แบบที่ทุก load case คูณ load factor แล้ว

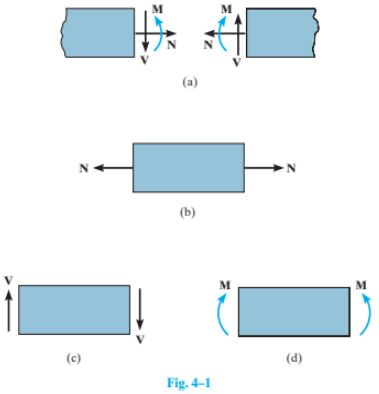
```
from PSM import *


T0 = Structure()
A = Hinge(0,0,"A")
B = Hinge(15,0,"B")
C = Hinge(0,20,"C")
M1 = Material.steel()
SEC1 = Section(1,2,3,4,5,6,"A")
AB = Link(A,B,M1,SEC1)
AC = Link(A,C,M1,SEC1)
CB = Link(C,B,M1,SEC1)

B.load(0,-40)
A.pinned()
C.rollerY()


T0.__init__()
T0.solve()
T0.printout()
```

ตัวอย่างการป้อนข้อมูลโครงสร้างประเภท Truss

การแสดงผลมีทั้ง 2 แบบ

- printcase ใช้สำหรับแสดงผลเฉพาะ load case ที่ทำการ input เข้าไป ยกตัวอย่างเช่น structure.printcase(0) เพื่อแสดงผลของ loadcase ที่ 0 โดยใช้ความสามารถ Linked list เรียกทุก node และ element มาคำนวณ ผลลัพธ์ที่ได้จากการคำนวณประกอบด้วย Nodal Displacement และ Member Force
- printout ใช้สำหรับแสดงผลของ load combination แบบคูณ load factor เพิ่มเข้าไปที่ Nodal displacement และ Local internal forces โดยใช้ความสามารถ Linked list เรียกทุก node และ element มาคำนวณ ในกรณีที่ไม่มี load factor การใช้คำสั่งนี้จะแสดงผลออกมาเฉพาะกรณี *factor0* หรือก็คือ combination แบบที่แทน factor ทุกตัวเท่ากับ 1.0 ผลลัพธ์ที่ได้จากการคำนวณประกอบด้วย Nodal Displacement และ Member Force

ทิศทางของแรง Member force จะถูกกำหนดด้วยเครื่องหมายบวกหรือลบ ภาพด้านล่างจะแสดงทิศทางของแรงตามเครื่องหมายบวก



ทิศทางของแรงเครื่องหมายบวกใน member force

4. ตัวอย่างการการใช้โปรแกรม

โปรแกรมที่พัฒนาขึ้นในโครงงานนี้ ใช้สำหรับวิเคราะห์โครงสร้าง 2 มิติ การใช้งานประกอบด้วยส่วนป้อนข้อมูลเข้า (Input) และส่วนแสดงผล (Output) ดังต่อไปนี้

ส่วน Input ประกอบด้วย Hinge, Joint , Link , Beam , Structure เป็นการจำแนกส่วนต่างๆของโครงสร้างและทำการสร้างโครงสร้างขึ้นมาในโปรแกรมเพื่อคำนวณ

ส่วน Output ประกอบด้วย Node displacement , Member force

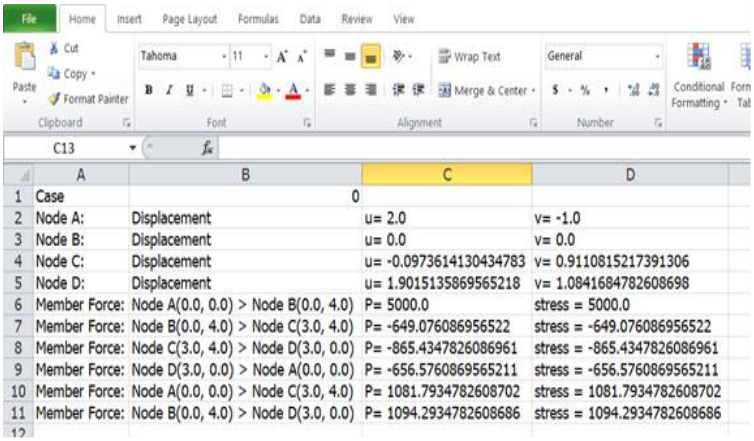
4.1 Truss

T1.py - C:\PSM\T1.py (3.8.2)

File Edit Format Run Options Window Help

```
from PSM import *

T1 = Structure()
mat = Material(2e4)
sect1 = Section(1.0,2,3,4,5,6,"A")
A = Hinge(0.0, 0.0, "A")
B = Hinge(0.0, 4.0, "B")
C = Hinge(3.0, 4.0, "C")
D = Hinge(3.0, 0.0, "D")
AB = Link(A, B, mat, sect1)
BC = Link(B, C, mat, sect1)
CD = Link(C, D, mat, sect1)
DA = Link(D, A, mat, sect1)
AC = Link(A, C, mat, sect1)
BD = Link(B, D, mat, sect1)
D.load(0.0, -10.0)
A.pinned()
B.pinned()
A.settleX(2.0)
A.settleY(-1.0)

T1.__init__()
T1.solve()
T1.printout()
```

Result:



	A	B	C	D
1	Case		0	
2	Node A:	Displacement	u= 2.0	v= -1.0
3	Node B:	Displacement	u= 0.0	v= 0.0
4	Node C:	Displacement	u= -0.0973614130434783	v= 0.9110815217391306
5	Node D:	Displacement	u= 1.9015135869565218	v= 1.0841684782608698
6	Member Force:	Node A(0.0, 0.0) > Node B(0.0, 4.0)	P= 5000.0	stress = 5000.0
7	Member Force:	Node B(0.0, 4.0) > Node C(3.0, 4.0)	P= -649.076086956522	stress = -649.076086956522
8	Member Force:	Node C(3.0, 4.0) > Node D(3.0, 0.0)	P= -865.4347826086961	stress = -865.4347826086961
9	Member Force:	Node D(3.0, 0.0) > Node A(0.0, 0.0)	P= -656.5760869565211	stress = -656.5760869565211
10	Member Force:	Node A(0.0, 0.0) > Node C(3.0, 4.0)	P= 1081.7934782608702	stress = 1081.7934782608702
11	Member Force:	Node B(0.0, 4.0) > Node D(3.0, 0.0)	P= 1094.2934782608686	stress = 1094.2934782608686

4.2 Frame

F5.py - C:\Users\tchal\Desktop\PSM\F5.py (3.8.1)

File Edit Format Run Options Window Help

```
from PSM import *

# Example F5 from JSM

mat = Material.concrete()
beam = RectSection(0.20, 0.40)
column = RectSection(0.30, 0.30)

F5 = Structure()
A = Joint( 0.0, 3.2, "A")
B = Joint( 5.0, 3.2, "B")
C = Joint( 8.0, 3.2, "C")
D = Joint( 0.0, 0.0, "D")
E = Joint( 5.0, 0.0, "E")
F = Joint( 8.0, 0.0, "F")
G = Joint( 0.0, 6.4, "G")
H = Joint( 5.0, 6.4, "H")
I = Joint( 8.0, 6.4, "I")
AB = Beam(A, B, mat, beam)
BC = Beam(B, C, mat, beam)
DA = Beam(D, A, mat, column)
AG = Beam(A, G, mat, column)
EB = Beam(E, B, mat, column)
BH = Beam(B, H, mat, column)
FC = Beam(F, C, mat, column)
CI = Beam(C, I, mat, column)

AB.uniformLoad(0.0, -35890.0)
BC.uniformLoad(0.0, -14784.0)

D.fixed()
E.fixed()
F.fixed()
G.fixed()
H.fixed()
I.fixed()

F5.__init__()
F5.solve()
F5.printcase(0)
```

Result:

1	Loadcase		0	
2	Node A:	Displacement	u= 0.0	v= -7.690265311108678e-05
3	Node B:	Displacement	u= 0.0	v= -0.0001137579321613360
4	Node C:	Displacement	u= 0.0	v= -8.274525838688248e-06
5	Node D:	Displacement	u= 0.0	v= 0.0
6	Node E:	Displacement	u= 0.0	v= 0.0
7	Node F:	Displacement	u= 0.0	v= 0.0
8	Node G:	Displacement	u= 0.0	v= 0.0
9	Node H:	Displacement	u= 0.0	v= 0.0
10	Node I:	Displacement	u= 0.0	v= 0.0
11	Member Force:	Node A(0.0, 3.2) > Node B(5.0, 3.2)	Pi= -0.0	Vi= 86515.48474997263
12	Member Force:	Node B(5.0, 3.2) > Node C(8.0, 3.2)	Pi= -0.0	Vi= 35043.15843147572
13	Member Force:	Node D(0.0, 0.0) > Node A(0.0, 3.2)	Pi= -43257.74237498631	Vi= -12998.823232615585
14	Member Force:	Node A(0.0, 3.2) > Node G(0.0, 6.4)	Pi= 43257.74237498631	Vi= -12998.823232615585
15	Member Force:	Node E(5.0, 0.0) > Node B(5.0, 3.2)	Pi= -63988.83684075154	Vi= 7923.6625616998845
16	Member Force:	Node B(5.0, 3.2) > Node H(5.0, 6.4)	Pi= 63988.83684075154	Vi= 7923.6625616998845
17	Member Force:	Node F(8.0, 0.0) > Node C(8.0, 3.2)	Pi= -4654.42078426214	Vi= -210.9094175898257
18	Member Force:	Node C(8.0, 3.2) > Node I(8.0, 6.4)	Pi= 4654.42078426214	Vi= -210.9094175898257

r= -0.0016433080234812794			
r= 0.001001707464590208			
r= -2.6663116495306368e-05			
r= 0.0			
r= 0.0			
r= 0.0			
r= 0.0			
r= 0.0			
r= 0.0			
Mi= -55461.645792493204	Pj= 0.0	Vj= -92934.51525002737	Mj= -71509.22204263008
Mi= -37701.59511271057	Pj= 0.0	Vj= -9308.84156852428	Mj= 899.8801817165895
Mi= 13865.411448123292	Pj= -43257.74237498631	Vj= -12998.82323261558	Mj= -27730.822896246584
Mi= 27730.822896246584	Pj= 43257.74237498631	Vj= -12998.82323261558	Mj= -13865.411448123292
Mi= -8451.906732479878	Pj= -63988.83684075154	Vj= 7923.6625616998845	Mj= 16903.813464959756
Mi= -16903.813464959756	Pj= 63988.83684075154	Vj= 7923.6625616998845	Mj= 8451.906732479878
Mi= 224.97004542914743	Pj= -4654.42078426214	Vj= -210.9094175898257	Mj= -449.94009085829487
Mi= 449.94009085829487	Pj= 4654.42078426214	Vj= -210.9094175898257	Mj= -224.97004542914743

4.3 Load factor and combinations

```
T4.py - C:\PSM\T4.py (3.8.2)
File  Edit  Format  Run  Options  Window  Help
from PSM import *

factor1 = Factor([1.5, 1.6, 0.5],'factor1')
factor2 = Factor([1.3, 1.3, 1.3],'factor2')
factor3 = Factor([1.7, 1.2, 1.8],'factor3')

T4 = Structure()
A = Hinge(0,0,"A")
B = Hinge(15,0,"B")
C = Hinge(0,20,"C")
M1 = Material.steel()
SEC1 = Section(1,2,3,4,5,6,"A")
AB = Link(A,B,M1,SEC1)
AC = Link(A,C,M1,SEC1)
CB = Link(C,B,M1,SEC1)

B.load(0.0, -40.0)
C.load(0.0, -30.0)
A.load(0.0, -50.0)
A.pinned()
C.rollerY()


T4.__init__()
T4.solve()
T4.printout()
```

Result:

	A	B	C	D
1	Case	0		
2	Node A:	Displacement	u= 0.0	v= 0.0
3	Node B:	Displacement	u= -2.2499999999999999e-09	v= -1.65e-08
4	Node C:	Displacement	u= 0.0	v= -7.000000000000001e-09
5	Member Force:	Node A(0.0, 0.0) > Node B(15.0, 0.0)	P= -29.99999999999999	stress = -29.99999999999999
6	Member Force:	Node A(0.0, 0.0) > Node C(0.0, 20.0)	P= -70.0	stress = -70.0
7	Member Force:	Node C(0.0, 20.0) > Node B(15.0, 0.0)	P= 50.0	stress = 50.0

	A	B	C	D
1	Case	factor1		
2	Node A:	Displacement	u= 0.0	v= 0.0
3	Node B:	Displacement	u= -3.3749999999999988e-09	v= -2.475e-08
4	Node C:	Displacement	u= 0.0	v= -1.0500000000000001e-08
5	Member Force:	Node A(0.0, 0.0) > Node B(15.0, 0.0)	P= -44.999999999999986	stress = -44.999999999999986
6	Member Force:	Node A(0.0, 0.0) > Node C(0.0, 20.0)	P= -105.0	stress = -105.0
7	Member Force:	Node C(0.0, 20.0) > Node B(15.0, 0.0)	P= 75.0	stress = 75.0

	A	B	C	D
1	Case	factor2		
2	Node A:	Displacement	u= 0.0	v= 0.0
3	Node B:	Displacement	u= -2.924999999999999e-09	v= -2.145e-08
4	Node C:	Displacement	u= 0.0	v= -9.1e-09
5	Member Force:	Node A(0.0, 0.0) > Node B(15.0, 0.0)	P= -38.999999999999986	stress = -38.999999999999986
6	Member Force:	Node A(0.0, 0.0) > Node C(0.0, 20.0)	P= -91.0	stress = -91.0
7	Member Force:	Node C(0.0, 20.0) > Node B(15.0, 0.0)	P= 65.0	stress = 65.0

	A	B	C	D
1	Case	factor3		
2	Node A:	Displacement	u= 0.0	v= 0.0
3	Node B:	Displacement	u= -3.8249999999999999e-09	v= -2.8049999999999996e-08
4	Node C:	Displacement	u= 0.0	v= -1.19e-08
5	Member Force:	Node A(0.0, 0.0) > Node B(15.0, 0.0)	P= -50.99999999999998	stress = -50.99999999999998
6	Member Force:	Node A(0.0, 0.0) > Node C(0.0, 20.0)	P= -119.0	stress = -119.0
7	Member Force:	Node C(0.0, 20.0) > Node B(15.0, 0.0)	P= 85.0	stress = 85.0

5. สรุปผล

โปรแกรม PSM ใช้การคำนวณด้วยวิธีคิดแบบ Finite-Element-Method (FEM) การคำนวณใช้วิธีสติฟเนสตรง Direct Stiffness มีสมการที่สำคัญคือ Stiffness Equation *[K]{D} = {F}* การคำนวณจะใช้การแบ่งโครงสร้างที่ศึกษาออกเป็น Node และ Element แต่ละชิ้นส่วนจะมีเมตริกซ์ Local external forces และเมตริกซ์ Local stiffness ของชิ้นส่วนนั้นๆ จากนั้นนำเมตริกซ์ย่อยมา assemble กันตามหลักการของ FEM ได้เป็นเมตริกซ์ Global ของโครงสร้างเพื่อนำไปแทนค่าในสมการ Stiffness Equation และไม่มีส่วนการแสดงผลด้วยกราฟฟิก แต่จะใช้การแสดงผลด้วยไฟล์ Data csv เท่านั้น

ส่วนการป้อนข้อมูลเข้าไปในโปรแกรมและส่วนการแสดงผลของข้อมูล โดยในส่วนการป้อนข้อมูลเพื่อนำไปวิเคราะห์สามารถทำได้กระชับ เข้าใจง่าย การป้อนข้อมูลสามารถป้อนแบบครบทุกแบบ หรือป้อนเฉพาะข้อมูลที่ทราบก็ได้ ในส่วนการแสดงผลสามารถเลือกได้ทั้ง 2 แบบ แบบที่หนึ่งเลือก loadcase คำสั่ง printcase() แบบที่สองแสดงผลทุก load factor คำสั่ง printout() สามารถแสดงค่าที่ถูกต้องทุกเคสเทียบกับโปรแกรม JSM

เอกสารอ้างอิง

[1] Watanachai Smittakorn.JSM as a Toolbox for Structural Analysis and Design Applications.Department of Civil Engineering, Chulalongkorn University, Bangkok 10330, THAILAND

[2] Structural Analysis Eight Edition R. C. Hibbeller

[3] John Hunt.A Beginners Guide to Python 3 Programming.Chippenham, Wiltshire, UK

[4] Klaus-Jürgen Bathe. Finite Element Procedures.second edition. United States of America.2014

[5] G. P. Nikishkov.INTRODUCTION TO THE FINITE ELEMENT METHOD.University of Aizu, Aizu-Wakamatsu 965-8580, Japan.2004

[6] Olivier de Weck& Il Yong Kim.January 12, 2004.Finite Element Method.MIT - Massachusetts Institute of Technology.

[7] อลงกรณ ละมอม.2543.โปรแกรมวิเคราะหโครงสรางเพื่อการเรียนการสอน.มหาวิทยาลัยเทคโนโลยีสุรนาร

กิตติกรรมประกาศ

โครงงานการวิเคราะห์และออกแบบโครงสร้าง 2 มิติ ด้วยภาษาไพธอน นี้เสร็จ สำเร็จลุล่วงได้อย่างสมบูรณ์ด้วยความกรุณาให้ความช่วยเหลือของ ผศ.ดร. วัฒนชัย สมิทธากร ที่ได้สอนการเขียน การอ่านโปรแกรมภาษาทั้งไพธอน และ จาวา พร้อมทั้งให้ข้อเสนอแนะ และ ให้คำแนะนำในการเขียนรายงานวิจัยฉบับนี้อีกด้วย ตลอดเวลาจนกระทั่งรายงานฉบับนี้เสร็จสมบูรณ์ ทางคณะผู้จัดทำขอขอบคุณเป็นอย่างสูง