รายการอ้างอิง

1. Bathe K.J. Finite element Procedures in Engineering Analysis. Englewood Cliff, NJ : Prentice-Hall, 1996.
2. Biggs J.M., Kaynia A.M., and Veneziano D., Seismic Effectiveness of Tuned Mass Dampers, Journal of Structural, ASCE, (August ,1981).
3. Brito R.S., and Ruiz S.E., Influence of Ground Motion Intensity on The Effectiveness of Tuned Mass Dammpers , Earthquake Engineering and Structural Dynamics. 1999.
4. Chang C.C. and Yang H.T.Y., Control of Buildings Using Active Tune Mass Dampers , Journal Engineering Machanics. ASCE,1995.
5. Chopra A.K.. Dynamic of Structures Theory And Applications to Earthquake Engineering. Englewood Cliff, NJ : Prentice-Hall,1995.
6. Clough R.W., and Penzien j.. Dynamic of Structures. New York : McGraw – Hill,1993.
7. Craig J.R.. Structueral Dynamic. New York : John Wiley & Sons, 1981.
8. Cook R.D.,Malkus D.S., and Plesha M.E.. Concepts and Appications of Finite Element Analysis. 3 rd ed, New York : John Wiley & Sons, 1889.
9. Cook R.D.. Finite Element Modeling For Stress Analysis. New York : John Wiley & Sons, 1995.
10. Constantinou M.C., Soong T.T. and dargush.. Passive Energy Dissipation Systems for Structural Design and Retrofit. MCEER, 1998.
11. Inman D.J.. Engineering Vibration. Englewood Cliff, NJ : Prentice-Hall,1996.
12. Jara J.M., and Aguiniga F., Paramitric Study of A Two Degree of Freedom System With Rasonant Masses.11 World Conference on Earthquake Engineering. Elsevier Science Ltd., Paper No1340,1996.
14. Kaynia A.M., Veneziano D. and Biggs J.M., Seismic Effectiveness of Tune mass dampers. Journal of Structural. ASCE, 1981.
13. Muhamad N.S.H. and Arfiadi Y., Optimum Design of Absorber for MDOF Structures Journal of Structural. ASCE, 1998.
15. Kwon Y.W., and Bang H.. The Finite Element Method Using Matlab. CRC Press LLC, 1997.

16. Lin C.C. ,Hu C.M. , Wang J.F. and Hu R.Y., Vibration Control Effectiveness of Passive Tuned Mass Dampers, Journal of Chinese Institute of Engineers,1994.

17. Meirovitch L.. Dynamics and Control of Structures. New York : John Wiley & Sons, 1992.

18. Sadek F., Mohraz B., Taylor A.W., and Chung R.M., A Method of Estimate the Parameters of Tuned Mass Dampers for Seismic Appplications, Earthquake Engineering and Structural Dynamics,1997.

19. Sladek J.R. and Klinger R.E., Effect of Tuned – Mass Dampers on Seismic Response, Journal of Structural. ASCE, 1983.

20. Villaverde R. and Koyoma L.A., Damped Resonant Appendages to Increase Inherent Damping in Building , Earthquake Engineering and Structural Dynamics,1993.

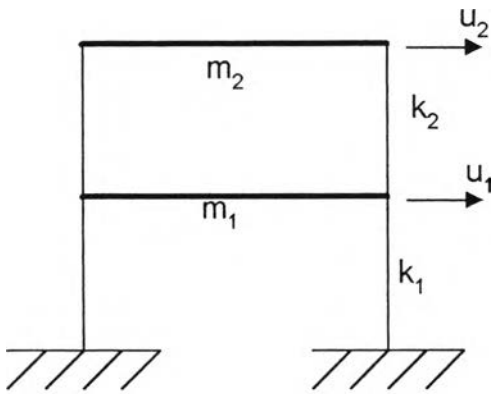
21. Wirsching P.H. and Campbell G.W., Minimal Structural Response Under Random Excitation Using the Vibration Absorber, Earthquake Engineering and Structural Dynamics,1974.

ภาคผนวก

ภาคผนวก ก

ปัญหาการสร้างเมตริกซ์การหน่วงของโปรมแกรมวิเคราะห์เชิงพลวัต

พิจารณาตัวอย่างการสร้างเมตริกซ์การหน่วงของอาคาร 2 ชั้น แสดงดังรูปที่ 1 มีเมตริกซ์ของมวลและเมตริกซ์สติฟเนสดังนี้



รูปที่ 1 โครงสร้างสองชั้น

$$m = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$k = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$

จากมวลและสติฟเนสหาความถี่ธรรมชาติและรูปแบบการสั่นของโครงสร้างได้

$$\omega = \begin{bmatrix} 1.6180 & 0 \\ 0 & 0.6180 \end{bmatrix}$$

$$\phi = \begin{bmatrix} 0.8507 & 0.5257 \\ -.5257 & 0.8507 \end{bmatrix}$$

สร้างเมตริกซ์การหน่วงของโครงสร้าง 2 ชั้น จากวิธีการหน่วงแบบแยกโหมดกำหนดค่าอัตราส่วนการหน่วงทั้งสองโหมดเท่ากับ 0.05 จะได้เมตริกซ์การหน่วง

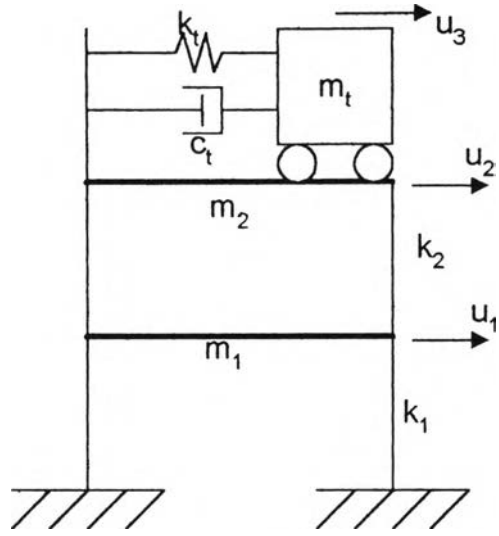
$$C_{str} = \begin{bmatrix} 2\xi_1\omega_1 M_1 & 0 \\ 0 & 2\xi_2\omega_2 M_2 \end{bmatrix}$$

สามารถหาค่าเมตริกซ์การหน่วง c ด้วยการคูณโหมดของการสั่นไหวด้านหน้าและด้านหลังเมตริกซ์การหน่วงสามัญ

$$c_{str} = \left[\phi^{T}\right]^{-1} C\, \phi^{-1}$$

ดังนั้นเมตริกซ์ความหน่วงของโครงสร้าง 2 ชั้นคือ

$$c_{str} = \begin{bmatrix} 0.1342 & -0.0447 \\ -0.0447 & 0.0894 \end{bmatrix}$$



รูปที่ 2 โครงสร้างสองชั้นติดมวลหน่วงปรับค่า

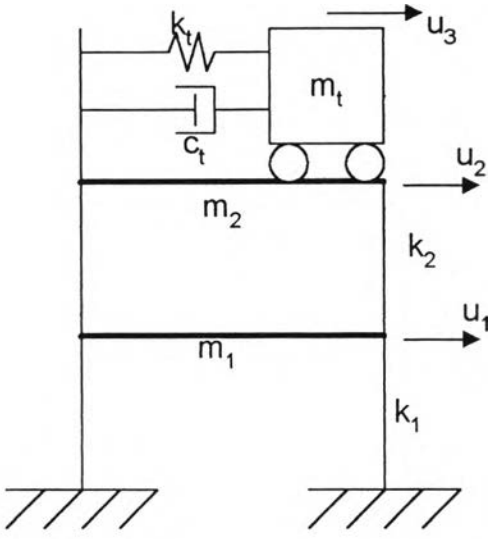
พิจารณากรณีโครงสร้างสองชั้นติดมวลหน่วงปรับค่า กำหนดให้ค่าอัตราหน่วงของมวลหน่วงปรับค่าเท่ากับ ξ_{TMD} ค่าความถี่ของมวลหน่วงปรับค่าเท่ากับ ω_{TMD} และมวลของมวลหน่วงปรับค่า m_{TMD} ดังนั้นเมตริกซ์การหน่วงของโครงสร้าง 2 ชั้นที่ติดมวลหน่วงปรับค่าคือ

$$c_{str+TMD} = \begin{bmatrix} 0.1342 & -0.0447 & 0 \\ -0.0447 & 0.0894 + 2\xi_{TMD}\omega_{TMD}m_{TMD} & -2\xi_{TMD}\omega_{TMD}m_{TMD} \\ 0 & -2\xi_{TMD}\omega_{TMD}m_{TMD} & 2\xi_{TMD}\omega_{TMD}m_{TMD} \end{bmatrix}$$

กำหนดค่าอัตราการหน่วงของมวลหน่วงปรับค่า ξ_{TMD} เท่ากับ 0.104ความถี่ของมวลหน่วงปรับค่าเท่ากับ ω_{TMD} เท่ากับ 0.618 มวลเท่ากับ 0.03 สติฟเนสเท่ากับ 0.0114จาก (Den Hartog (1956)) เมตริกซ์การหน่วงของโครงสร้าง 2 ชั้นที่ติดมวลหน่วงปรับค่า ได้

$$c_{str+TMD} = \begin{bmatrix} 0.1342 & -0.0447 & 0 \\ -0.0447 & 0.09328 & -3.8759E-4 \\ 0 & -3.8759E-4 & 3.8759E-4 \end{bmatrix}$$

พิจารณาตัวอย่างจากรูปที่ 2 การสร้างเมตริกซ์การหน่วงของอาคาร 2 ชั้นและมีการติดมวลหน่วงปรับค่าที่ชั้นบนสุด มีเมตริกซ์ของมวลและเมตริกซ์สติฟเนสดังนี้



รูปที่ 2 โครงสร้างสองชั้นติดมวลหน่วงปรับค่า

$$M = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.03 \end{bmatrix}$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_t & -k_t \\ 0 & -k_t & k_t \end{bmatrix} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 1.0115 & -0.0115 \\ 0 & -0.0115 & 0.0115 \end{bmatrix}$$

จากมวลและสติฟเนสหาความถี่ธรรมชาติและรูปแบบการสั่นของโครงสร้างได้

$$\omega = \begin{bmatrix} 1.6192 & 0 & 0 \\ 0 & 0.6650 & 0 \\ 0 & 0 & 0.5740 \end{bmatrix}$$

$$\phi = \begin{bmatrix} 0.8458 & 0.0995 & 0.0812 \\ -0.5259 & 0.1550 & 0.1357 \\ 0.0897 & -0.9829 & 0.9874. \end{bmatrix}$$

โปรแกรมวิเคราะห์พลวัตสร้างเมตริกซ์การหน่วงด้วยวิธีการหน่วงแบบแยกโหมดโดยให้ใส่ค่าอัตราการหน่วงในแต่ละโหมด จึงมีความเป็นไปได้หรือไม่ที่จะใส่ค่าอัตราการหน่วงที่ให้ค่าเมตริกซ์เท่ากับเมตริกซ์ของโครงสร้างที่ติดตั้งมวลหน่วงปรับค่า โดยสมมติค่าของเมตริกซ์การหน่วงสามัญดังนี้

$$C_{PRO} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}$$

ค่าเมตริกซ์การหน่วง $c = \left[\phi^{T}\right]^{-1} C\, \phi^{-1}$ เขียนอยู่ในรูป

$$c_{PRO} = \begin{bmatrix} D & E & F \\ E & G & H \\ F & H & I \end{bmatrix}$$

แล้วทำการเทียบค่าในเมตริกซ์การหน่วงจากโปรแกรมวิเคราะห์เชิงพลวัตจากการสมมติค่าอัตราการหน่วงเท่ากับค่าเมตริกซ์ของโครงสร้างที่ติดมวลหน่วงปรับค่า สามารถเขียนได้ดังนี้

$$[A][X] = [B]$$

$$\begin{bmatrix} 0.7154 & 0.0099 & 0.0066 \\ -0.4448 & 0.0154 & 0.0110 \\ 0.0758 & -0.0978 & 0.0802 \\ 0.2766 & 0.0240 & 0.0184 \\ -0.0472 & -0.1524 & 0.1340 \\ 0.0080 & 0.9661 & 0.9750 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0.1342 \\ -0.0447 \\ 0 \\ 0.0933 \\ -0.0039 \\ 0.0039 \end{bmatrix}$$

ในการแก้สมการเพื่อจะหาค่า a , b , c ที่ทำให้สมการเป็นจริง โดยสามารถตรวจสอบว่ามีคำตอบหรือไม่จากวิธีการหาแรงค์ โดยจะมีคำตอบเมื่อ $rank[A]=rank[\widetilde{A}]$ ($[\widetilde{A}]$คือเมตริกซ์ [A] ที่รวมกับสดมภ์ B) พบว่า $rank[A]$=3 และ $rank[\widetilde{A}]$ =4 ดังนั้น $rank[A]<rank[\widetilde{A}]$ จึงไม่มีคำตอบ ดังนั้นโปรแกรมวิเคราะห์เชิงพลวัตไม่สามารถประยุกต์วิเคราะห์ระบบโครงสร้างที่ติดมวลหน่วงปรับค่า เนื่องจากไม่สามารถหาค่าอัตราการหน่วงที่ทำให้ค่าเมตริกซ์การหน่วงเท่ากับเมตริกซ์ของโครงสร้างที่ติดมวลหน่วงปรับค่า

ภาคผนวค ข
การใช้โปรแกรม

ตัวอย่างข้อมูลของโครงสร้างที่ใส่ในโปรแกรม

1) General Data (NSTR, NCOLL, NCOLT, NBEAMT, NCOL, NBEAM, NJOINT)

11, 2, 1, 1, 20, 10, 22,

2) mass at joint

0

0

89.5e3

89.5e3

85e3

85e3

80.5e3

80.5e3

76e3

76e3

71.5e3

71.5e3

67e3

67e3

62.5e3

62.5e3

58e3

58e3

53.5e3

53.5e3

49e3

49e3

3) Column Data (CNO, CEI(i), CXI(i), CA(i), CLENG(i))

1, 2.602916667e6, 1, 1, 1,

2, 2.602916667e6, 1, 1, 1,

3, 2.1775e6, 1, 1, 1,

4, 2.1775e6, 1, 1, 1,

5, 2.339166667e6, 1, 1, 1,

6, 2.339166667e6, 1, 1, 1,

7, 2.209166667e6, 1, 1, 1,

8, 2.209166667e6, 1, 1, 1,

9, 2.0795833333e6, 1, 1, 1,

10, 2.0795833333e6, 1, 1, 1,

11, 1.9495833333e6, 1, 1, 1,

12, 1.9495833333e6, 1, 1, 1,

13, 1.8195833333e6, 1, 1, 1,

14, 1.8195833333e6, 1, 1, 1,

15, 1.689583333e6, 1, 1, 1,

16, 1.689583333e6, 1, 1, 1,

17, 1.5595833333e6, 1, 1, 1,

18, 1.5595833333e6, 1, 1, 1,

19, 1.429583333e6, 1, 1, 1,

20, 1.429583333e6, 1, 1, 1,

4) Beam Data (BNO, BEI(i), BXI(i), BLENG(i))

1, 1.429583333e6, 100, 1,

2, 1.429583333e6, 100, 1,

3, 1.429583333e6, 100, 1,

4, 1.429583333e6, 100, 1,

5, 1.429583333e6, 100, 1,

6, 1.429583333e6, 100, 1,

7, 1.429583333e6, 100, 1,

8, 1.429583333e6, 100, 1,

9, 1.429583333e6, 100, 1,

10, 1.429583333e6, 100, 1,

5) Joint Coordinates (JNO, JTI(i) , JTJ(i))

1, 1, 1,

2, 2, 1,

3, 1, 2,

4, 2, 2,

5, 1, 3,

6, 2, 3,

7, 1, 4,

8, 2, 4,

9, 1, 5,

10, 2, 5,

11, 1, 6,

12, 2, 6,

13, 1, 7,

14, 2, 7,

15, 1, 8,

16, 2, 8,

17, 1, 9,

18, 2, 9,

19, 1, 10,

20, 2, 10,

21, 1, 11,

22, 2, 11,

6) Column Connectivity (CNO, CJN(i), CJP(i))

1, 1, 3,

2, 2, 4,

3, 3, 5,

4, 4, 6,

5, 5, 7,

6, 6, 8,

7, 7, 9,

8, 8,10,

9, 9, 11,

10, 10, 12,

11, 11, 13,

12, 12, 14,

13, 13, 15,

14, 14, 16,

15, 15, 17,

16, 16, 18,

17, 17, 19,

18, 18, 20,

19, 19, 21,

20, 20, 22,

7) Beam Connectivity (BNO, BJN(i), BJP(i))

1, 3, 4,

2, 5, 6,

3, 7, 8,

4, 9, 10,

5, 11, 12,

6, 13, 14,

7, 15, 16,

8, 17, 18,

9, 19, 20,

10, 21, 22,

8) Support Condition (SUPNO , SDIR(i,1), SDIR(i,2), SDIR(i,3))

2,

1, 1, 1, 1,

2, 1, 1, 1,

จากตัวอย่างแฟ้มข้อมูลที่ใส่ในโปรแกรมซึ่งจะอธิบายแต่ละหัวข้อของแฟ้มข้อมูล

1) General Data (NSTR, NCOLL, NCOLT, NBEAMT, NCOL, NBEAM, NJOINT)

11, 2, 1, 1, 20, 10, 22,

ในหัวข้อที่ 1 ข้อมูลทั่วไปของโครงสร้างที่วิเคราะห์

NSTR คือจำนวนชั้นของโครงสร้างในตัวอย่างเป็นอาคาร 10 ชั้นแต่จะต้องใส่ในแฟ้มข้อมูลเท่ากับ 11เนื่องจากโปแกรมกำหนดที่พื้นดินเท่ากับ 1

NCOLL คือจำนวนแถวของเสาในโครงสร้าง

NCOLT คือจำนวนชนิดเสาในโครงสร้าง ในกรณีที่เสามีคุณสมบัติไม่เหมือนกัน

NBEAMT คือจำนวนชนิดคานในโครงสร้าง ในกรณีที่เสามีคุณสมบัติไม่เหมือนกัน

NCOL คือจำนวนเสาในโครงสร้าง

NBEAM คือจำนวนคานในโครงสร้าง

NJOINT คือจำนวนจุดต่อในโครงสร้าง

2) mass at joint

0

0

89.5e3

89.5e3

หัวข้อที่ 2 คือมวลรวบจุดของโครงสร้างโดยเริ่มที่จุดยึดรั้ง (Support) โดยแต่ละแถวใส่ค่ามวล 1 จุด

3) Column Data (CNO, CE(i), CI(i), CA(i), CLENG(i))

1, 2.602916667e6, 1, 1, 1,

ในหัวข้อที่ 3 คือคุณสมบัติของเสาโดยอธิบายเรียงตามลำดับดังนี้

CNO คือเสาต้นที่ จากตัวอย่างเท่ากับ1 2.602916667e6

CE คือค่าโมดูลัสยืดหยุ่นของเสา จากตัวอย่างเท่ากับ 2.602916667e6

CI คือค่าโมเมนต์อินเนอร์เชียเสา จากตัวอย่างเท่ากับ 1

CA คือพื้นที่หน้าตัดของเสา จากตัวอย่างเท่ากับ 1

CLENG คือความยาวของเสา จากตัวอย่างเท่ากับ 1

4) Beam Data (BNO, BE(i), BI(i), BLENG(i))

1, 1.429583333e6, 100, 1,

ในหัวข้อที่ 4 คือคุณสมบัติของคานโดยอธิบายเรียงตามลำดับดังนี้

BNO คือคานที่ จากตัวอย่างเท่ากับ1

BE คือค่าโมดูลัสยืดหยุ่นของคาน จากตัวอย่างเท่ากับ 1.429583333e6,

BI คือค่าโมเมนต์อินเนอร์เชียคาน จากตัวอย่างเท่ากับ 1

BLENG คือความยาวของคาน จากตัวอย่างเท่ากับ 1

5) Joint Coordinates (JNO, JTI(i) , JTJ(i))

1, 1, 1,

ในหัวข้อที่ 5 พิกัดของจุดโดย

JNO คือจุดที่เท่าไรของโครงสร้าง

JTI(i) คือจุดมีตำแหน่งอยู่เสาแนวที่

JTJ(i) คือจุดมีตำแหน่งอยู่ชั้นที่

6) Column Connectivity (CNO, CJN(i), CJP(i))

1, 1, 3,

ในหัวข้อที่ 6 คือจุดที่เชื่อมต่อของเสา

CNO คือเสาต้นที่

CJN(i) คือจุดที่ข้างล่างของเสาเชื่อมต่อ

CJP(i) คือจุดที่ข้างบนของเสาเชื่อมต่อ

7) Beam Connectivity (BNO, BJN(i), BJP(i))

1, 3, 4,

คล้ายกับหัวข้อที่ 6 คือจุดที่เชื่อมต่อของคาน

CNO คือคานต้นที่

CJN(i) คือจุดที่จุดต่อคานด้านซ้ายเชื่อมต่อ

CJP(i) คือจุดที่ที่จุดต่อคานด้านขวาเชื่อมต่อ

8) Support Condition (SUPNO , SDIR(i,1), SDIR(i,2), SDIR(i,3))

2,

1, 1, 1, 1,

หัวข้อที่คือการยึดรั้งที่ฐานของโครงสร้าง

โดยบรรทัดแรกกำหนดจำนวนจุดที่มีการยึดรั้ง จากแฟ้มข้อมูลเท่ากับสอง

บรรทัดต่อมาคือรายละเอียดของของจุดที่มีการยึดรั้ง

SUPNO คือจุดที่ยึดรั้ง

SDIR(i,1) คือสภาพการยึดรั้งของจุดในแนวตั้ง ถ้ามีการยึดรั้งใส่1แต่ถ้าอิสระใส่ 0

SDIR(i,2) คือสภาพการยึดรั้งของจุดในแนวนอน ถ้ามีการยึดรั้งใส่1แต่ถ้าอิสระใส่ 0

SDIR(i,3) คือสภาพการยึดรั้งของจุดจากการหมุน ถ้ามีการยึดรั้งใส่1แต่ถ้าอิสระใส่ 0

ตัวอย่างแฟ้มข้อมูลของแรงพลวัตและการเลือกการแสดงผลของโครงสร้าง

General Loading Data (number of dynamic joint load ,ground acceleration)

3, 0,

load dynamic number

1

load joint number and number time step

5, 4,

time step

0 1

0.5 11

1 1.3

3 13

load dynamic number

2

load joint number and number time step

9, 4,

time step

0 1

0.5 11

1 1.3

3 13

load dynamic number

3

load joint number and number time step

13, 4,

time step

0 1

0.5 11

1 1.3

3 13

dynamic data (NMODE, NSTEP, DT, SOLVE,)

10, 1559, 0.02, 1,

damping ratio

0.02

0.02

0.02

0.02

0.02

0.02

0.02

0.02

0.02

0.02

0.02

property TMD (MTMD DAMTMD STIFFTMD)

4220

2076.4071

17038.35128

select file out put(nstory, ncolumn, nbeam)

1, 0, 1,

select story nunber

11

select ncolumn nunber

1

General Loading Data (number of dynamic joint load ,ground acceleration)

3, 0,

ในแถวแรกจำนวนแรงพลวัตที่กระทำกับโครงสร้าง ซึ่งในตัวอย่างนี้แรงพลวัติกระทำที่จุดต่อของโครงสร้างจำนวน 3 แรง และ 0 คือไม่มีแรงแผ่นดินไหว แต่ถ้าเท่ากับ1 คือมีแรงแผ่นดินไหวแสดงในตัวอย่างต่อไป

load dynamic number

1

ในบรรทัดถัดมาคือแรงที่ 1

load joint number and number time step

5, 4,

5, คือแรงพลวัตกระทำในแนวด้านข้างที่จุดต่อที่ 5

4, คือจำนวนช่วงเวลา (time steps)

time step

0 1

0.5 11

1 1.3

3 13

ในบรรทัดต่อมาคือเวลา และขนาดของแรง เช่น 0 1 คือ เวลาที่ 0 แรงเท่ากับ 1ซึ่งแรงพลวัตที่ 2 และ3 ก็ทำเช่นเดียวกัน

dynamic data (NMODE, NSTEP, DT, SOLVE,)

10, 8, 0.02, 1,

หลังจากเขียนแรงครบก็เว้นหนึ่งบรรทัดและบรรทัดต่อมาคือคุณสมบัติทางพลวัตของโครงสร้าง

10, คือจำนวนโหมดที่วิเคราะห์

8, คือจำนวนช่วงเวลาที่วิเคราะห์ (number of time steps)

0.02, คือระยะเวลาในแต่ละช่วง (duration)

1, คือเลือกวิเคราะห์ ถ้าเท่ากับ 0 ไม่คิดมวลหน่วงปรับค่า แต่ถ้าเท่ากับ 1 คิดมวลหน่วงปรับค่า ซึ่งจะต้องใส่ข้อมูลของมวลหน่วงปรับค่า

damping ratio

0.02

ในบรรทัดต่อมาคือค่าอัตราส่วนความหน่วงในแต่ละโหมดที่ผู้ใช้กำหนดให้ครบจำนวนโหมดที่วิเคราะห์ ในตัวตัวอย่างวิเคราะห์ 10โหมด

property TMD (MTMD DAMTMD STIFFTMD)

4220

2076.4071

17038.35128

ในบรรทัดต่อมาถ้าเลือก SOLVE เท่ากับ 1เพื่อวิเคราะห์มวลหน่วงปรับค่าด้วยจะต้องใส่คุณสมบัติของมวลหน่วงปรับค่าแสดงดังตัวอย่าง

4220 คือมวลของมวลหน่วงปรับค่า

2076.4 คือความหน่วงของมวลหน่วงปรับค่า

17038.3 คือสติฟเนสของมวลหน่วงปรับค่า

select file out put(nstory, nbeam, ncolumn)

1, 0, 1,

ในบรรทัดต่อมาคือเลือกจำนวนชิ้นส่วนแสดงผลทางแฟ้มข้อมูล

1, คือจำนวนชั้นที่แสดงการกระจัดที่เวลาใดๆ

0, คือเลือกจำนวนคานที่แสดงผล

1, คือเลือกจำนวนเสาที่แสดงผล

select story nunber

11

ในบรรทัดต่อมาคือเลือกการกระจัดของชั้นเลือกแสดงผลทางแฟ้มข้อมูล ในตัวอย่างเลือกชั้นที่ 10 ซึ่งต้องบอก1 เข้าไปเนื่องจากโปรแกรมกำหนดให้ที่ผิวดินเท่า 1

select ncolumn nunber

1

ในบรรทัดต่อมาคือเลือกเสาแสดงผลทางแฟ้มข้อมูล ในตัวอย่างเลือกเสาต้อที่ 1 โดยเสาและคาน แต่ละชั้นเรียงลำดับเลขจากซ้ายไปขวา

ตัวอย่างแฟ้มข้อมูลของแรงแผ่นดินไหว

General Loading Data

0, 1,

file ground aceleration

Elcentro10

dynamic data (NMODE, NSTEP, DT, SOLVE,)

1, 1560, 0.02, 1,

damping ratio

0.02

property TMD (MTMD DAMTMD STIFFTMD)

6092.7

2253.956367

56145.03382

select file out put(nstory, ncolumn, nbeam)

1, 0, 0,

select story nunber

11

General Loading Data

0, 1,

ในแถวแรกจำนวนแรงพลวัตที่กระทำกับโครงสร้าง ซึ่งในตัวอย่างนี้แรงพลวัติกระทำที่จุดต่อของโครงสร้างจำนวน 0, แรง และ 1, คือมีแรงแผ่นดินไหวแสดง

file ground aceleration

El Centro10

บรรทัดต่อมาคือชื่อแฟ้มข้อมูลของคลื่นแผ่นดินไหว ในตัวอย่างชื่อ El Centro10ในส่วนบรรทัดต่อไปก็เช่นเดียวกับที่อธิบายก่อนหน้านี้

ตัวอย่างแฟ้มข้อมูลของคลื่นแผ่นดินไหว

8

Time(sec) ,acerelation(g)

0.0000 0.0618

0.0200 0.0357

0.040 0.0097

0.060 0.042

0.080 0.0744

0.100 0.1069

0.120 0.0669

0.140 0.0272

8 ในบรรทัดแรกคือจำนวนช่วงเวลาของคลื่นแผ่นดินไหว

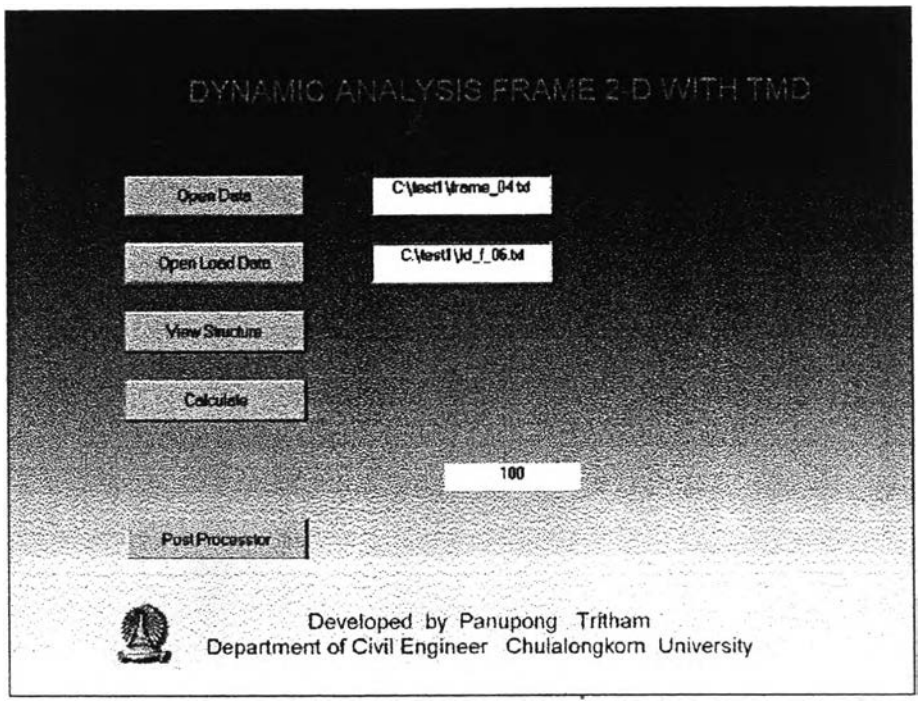
Time(sec) ,acerelation(g)

0.0000 0.0618

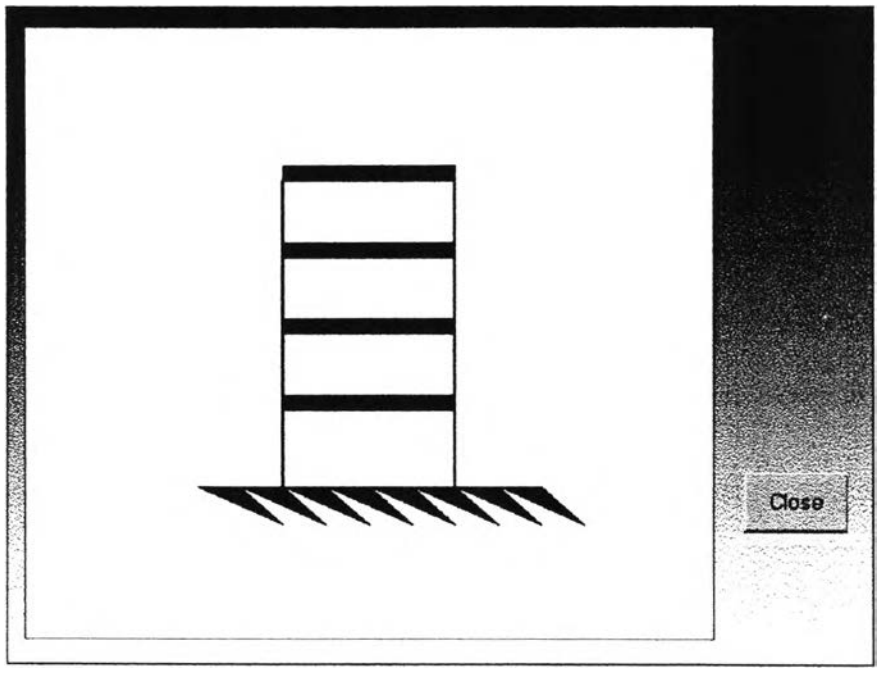
ในบรรทัดถัดมาคือเวลาและแรงของช่วงเวลา(time step)ต่างๆ ตัวอย่างที่ช่างเวลาที่ 1 เวลา0 แรงเท่ากับ 0.0618

ขั้นตอนการใช้โปรแกรม

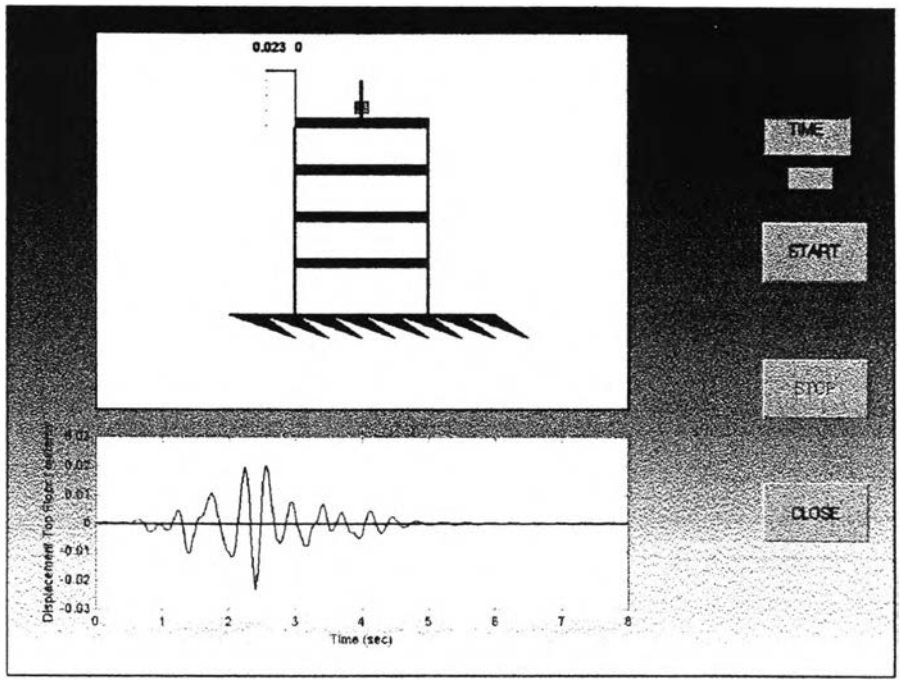
เปิดโปรแกรมโดยเรียก frtmd จาก MATLAB หน้าจอจะแสดงภาพดังนี้



หลักจากนั้นกดปุ่ม Open data เพื่อเลือกแฟ้มข้อมูลของโครงสร้างที่จะวิเคราะห์ โดยชื่อแฟ้มข้อมูลจะแสดงในช่องขาวขวามือ หลักจากนั้นกดปุ่ม Open Load Data เพื่อเลือกแฟ้มข้อมูลของแรงที่จะวิเคราะห์ โดยชื่อแฟ้มข้อมูลของแรงจะแสดงในช่องขาวขวามือ หลังจากนั้นตรวจสอบคุณสมบัติทางคณิตของโครงสร้างโดยกดที่ปุ่ม View structure หน้าจอจะแสดงภาพของโครงสร้าง โดย Close เพื่อปิดภาพ



หลังจากนั้นเลือก Calculate เพื่อวิเคราะห์โดยมีตัวเลขแสดงเปอร์เซ็นต์การวิเคราะห์เมื่อเท่ากับ 100 จะวิเคราะห์เสร็จ ในโปรแกรมสามารถแสดงภาพเคลื่อนไหวของโครงสร้าง โดยกดที่ Post Processtor ซึ่งมีปุ่ม Start เพื่อเริ่มแสดง และ Stop เพื่อหยุด โดยมีกราฟการกระจัดของชั้นบนสุดของโครงสร้างแสดงด้วย ซึ่งมีหน้าจอดังนี้



ภาคผนวก ค
โปรแกรม

main program

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
%                  Linear-Elastic Analysis programming
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
function Responcs(datafile,loadfile)
hdata=get(gcf,'userdata');
hperstring1=hdata(1);
hperstring2=hdata(2);
hperstring3=hdata(3);
hperslide=hdata(4);
set(hperslide,'visible','on');
global NSTR NCOLL NCOLT NBEAMT NCOL NBEAM NJOINT;
global CEI CXI CA CRHO CLENG;
global BEI BXI BA BRHO BLENG;
global STIF;
global MASS mass POS;
global JTI JTJ;
global CJN CJP BJN BJP;
global DOF;       DOF = 2;
global NSUP SUPNO SDIR;
global RESNOD FRENOD;  % RESNOD = restraint node no. FRENOD = free node no.
global CONDEN frenod resnod;
global FORC DISP;
global NLDOF NVRDOF NDOF NSTRJT NSTRDOF STRJTNO JTTB JTLDOF; % for determine lateral dof
global CINFORC BINFORC EQCHK;
global REACT;    % Reaction at support
% CINFORC(1:6,I)          = nodal internal force of column no. I [ l- v- r- l+ v+ r+]
% BINFORC(1:4,I)          = nodal internal force of beam no. i [ v- r- v+ r+ ]
% EQCHK(I)                        = static equilibrium check vector
% Load Data Variable
global LJTNO LUNFNO LPTNO LSDN0;
global LJTN LJTDIR;
global LUNFMB LUNFLEN LUNFBEG LUNFMAG;
global LPTMB LPTLOC LPTMAG;
global LSDJN LSDDIR;
% DEBUGING: temporary variables
global AKE AKER AKES KE KA;
global AME AMER ME MA;
global AA AB BA BB;
global aa ab ba bb;
global kk ka ak aa KK;
global MM MN NM NN
% eigenvalue eigenvecter
global D V TEMP
global n y
% dynamic load
global A NDYL JDYL NST DYLN
% ground accerelation
global IGA NGAST GA GAL GALD
% file load input
global load1 load2 load3 load3 load4 load5 load6 load7 load8 load9 load10 load11 load12
global load13 load14 load15 load16 load17 load18 load19 load20 load21 load22 load23 load24 load25
% file load adjust time step
global LOAD1 LOAD2 LOAD3 LOAD3 LOAD4 LOAD5 LOAD6 LOAD7 LOAD8 LOAD9 LOAD10 LOAD11
LOAD12
global LOAD13 LOAD14 LOAD15 LOAD16 LOAD17 LOAD18 LOAD19 LOAD20 LOAD21 LOAD22
LOAD23 LOAD24 LOAD25
% output data gui
global X Y AC ABM xy story njoint NSTR;
global TIME U SOLVE;
global NMOD NSTEP DT CR STORY BEAM COLUMN NSTORY BEAMOUT COLUMNOUT;
% property TMD
global MTMD DAMTMD STIFTMD
% ***************************************************
%       1.        INPUT PART
% ***************************************************
% Call Input Function
```

```
viewstructure1(datafile);
linput01(loadfile);
%**********************************************************
%  OUTPUT PART
%**********************************************************
fp_1 = fopen('DISP1.txt','w');
% **********************************************************
%  2.    SOLUTION PART
% **********************************************************

% 2.1 Calculate Basic Properties and set the dof no. and location
Lateraldof;
% 2.2 Calculate Element Stiffness Matrix and Assemble to structural stiffness
% Initialize size of Structural stiffness matrix : STIF

STIF = zeros(NDOF);
MASS = zeros(NDOF);

% DEBUGING :temporary variable to store stiffness matrices at each step
AKE = [];                    AKA = [];        AME=[];     AMA=[];

% ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% 2.2.1 COLUMN element stiffness matrix (forming and assembling)
% ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
for i = 1:NCOL,

  % Form the element stiffness matrix

  RIGID = [ CEI(i) CXI(i) CA(i)];
  DIMEN = [ CLENG(i)];
  [CKE, CKA] = localstiff(RIGID,DIMEN,1);
  %  DEBUGING  storing the element stiffness at each step

  AKE   = [ AKE CKE ];
  AKA   = [ AKA CKA ];

  % Assemble to the structural stiffness

  Assemble(CKE,CKA,[CJN(i) CJP(i)],1);

end;


% ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% 2.2.2 BEAM element stiffness matrices (forming and assembling)
% ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

for i = 1:NBEAM,

  % Form the element stiffness matrix

  RIGID  =      [ BEI(i) BXI(i) 0 ];
  DIMEN  =      [ BLENG(i)];
  [ BKE,BKA] =  localstiff(RIGID, DIMEN, 2);

  %  DEBUGING: storing the element stiffness at each step

  AKE   = [ AKE BKE ];

  % Assemble to the structural stiffness

  Assemble(BKE,0.00,[BJN(i) BJP(i)],2);

end;
%
% Assemble mass of structure
%
for i = 1:NJOINT;
  temp = JTLDOF(i);
  MASS(temp,temp) = MASS(temp,temp)+ mass(i);
```

```
end,

%
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++
% 2.3  Partitioning the structural stiffness matrix for known/unknown dof
%
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++
%        2.3.1 Check Restraint dof and determine both restraint and free dof
RESNOD = [], FRENOD = [],frenod=[].
for i = 1:NSUP,
  if SDIR(i,1)==1,
    temp =JTLDOF(SUPNO(i));
  end,
  MN = (SUPNO(i)-1)*DOF;
  temp1=find(SDIR(i,1:3)~=0);
  temp = [temp MN+find(SDIR(i,2:3)~=0)];
  RESNOD = [ RESNOD temp ];
end,
RESNOD = sort(RESNOD);
FRENOD = 1:NDOF;
FRENOD(RESNOD) = [];
FRENOD;
NVRDOF;
temp1 =find(FRENOD>NVRDOF);
frenod=FRENOD(temp1);
NLDOF=size(frenod,2);
resnod=FRENOD;
resnod(temp1)=[;
temp2 = size(resnod,2);
temp4=[];
for i=1:temp2/2
  temp3=2*i-1;
  temp4=[temp4 temp3];
end,
resnod(temp4)=[];
%
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++
%        2.3.2.1 Partitioning structural stiffness matrix into 4 parts
%
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++

AA = STIF(FRENOD,FRENOD);
AB = STIF(FRENOD,RESNOD);
BA = STIF(RESNOD,FRENOD);
BB = STIF(RESNOD,RESNOD);
%
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++
%        2.3.2.2 Partitioning structural mass matrix into 4 parts
%
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++

MM = MASS(frenod,frenod);
MN = MASS(frenod,resnod);
NM = MASS(resnod,frenod);
NN = MASS(resnod,resnod);

%
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++
%   condensation degree zeros mass
%
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++
```

```
kk = STIF(frenod,frenod);
ka = STIF(frenod,resnod);
ak = STIF(resnod frenod);
aa = STIF(resnod,resnod);
n  = (aa\eye(size(aa)));
m  = ak'*(aa\eye(size(aa)))*ak;
KK = kk-ak'*(aa\eye(size(aa)))*ak;

% ++++++++++++++++++++++++++
% 2.4  Form Loading vector.
% ++++++++++++++++++++++++++
% Case of Joint Loading

FORC = zeros(NDOF,1);
if LJTNO > 0
  for i = 1:LJTNO
    id = (LJTN(i)-1)*DOF+1;          % vertical dof
    FORC([id id+1]) = FORC([id id+1])+ LJTDIR(i,2:3)';
        FORC(JTLDOF(LJTN(i))) = FORC(JTLDOF(LJTN(i))) + LJTDIR(i,1);
  end,
end,
% Case of Support Displacement Loading

DISP = zeros(NDOF,1);
if LSDNO > 0,
  for i = 1:LSDNO,
    id = (LSDJN(i)-1)*DOF+1;
    DISP([id id+1]) = DISP([id id+1]) + LSDDIR(i,2:3)';
    DISP(JTLDOF(LSDJN(i))) = DISP(JTLDOF(LSDJN(i))) + LSDDIR(i,1);
  end,
end,

% ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% 2.5  Solving for the unknown displacement and force
% ++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% solving for unknown displacement

DISP(FRENOD) = (AA)\( FORC(FRENOD)-AB*DISP(RESNOD));
FORC(RESNOD) = BA*DISP(FRENOD)+BB*DISP(RESNOD);
fprintf(fp_1,'%.2e\n',DISP);
fclose(fp_1);

%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% 2.6 solving for eigen value problem
%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% solving for eigen value problem
[V D]=eigs(KK,MM,NMOD,0);
tempv=size(frenod,2);
for i=1:NMOD,
V(:,i)=V(:,i)/V(tempv,i);
end,
V;
MODE=sqrt(D);



%-------------------------------------------------
% 2.7 Generate damping matrix
%          Modal damping matrix
%-------------------------------------------------
%generate  modal damping ratio
MODDAM=zeros(NMOD);
for i=1:NMOD,
  MOMA=          V(:,i)'*MM*V(:,i);
  MODDAM(i,i)=2*CR(i)*MODE(i,i)*MOMA;
end,

%-------------------------------------------------
% 2.7 Generate modal mass and stiffness matrix
%          Modal analysis
%-------------------------------------------------
```

```
MODMA=zeros(NMOD);
MODSTIF=zeros(NMOD);
for i=1:NMOD,
  MODMA(i,i)=V(:,i)'*MM*V(:,i);
  MODSTIF(I,I)=V(:,i)'*KK*V(:,i);
end,

%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% 2.7 set initial condition and
%    tranform to genneralize coordinate
%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% set inital displacement is zeros
MODIND=zeros(NMOD,1);
IND=zeros(NLDOF,1);
for i=1:NMOD,
  MODIND(i,1) = (V( ,i)'*MM*IND)/MODMA(i,i);
end,

% set initial velocity is zeros
MODINV=zeros(NMOD,1);
INV=zeros(NLDOF,1);
for i=1 NMOD,
  MODINV(i,1) = (V(:,i)'*MM*INV)/MODMA(i,i);
end,


%
% input number and ramp for each time increment
%Choose solve TMD or dynamic analysis
 if SOLVE == 1

% generate mass structure with TMD

 temp4 =zeros(1,NMOD);
 lasttemp=size(frenod,2);
 for I = 1:NMOD,
   V(lasttemp,i);
   temp4(1,I) = V(lasttemp,I)*MTMD;
 end,
 temp5   = zeros(NMOD,1);
 MODMTMD = [MODMA temp5;temp4 MTMD];

% generate stiffness structure with TMD

 temp6 =zeros(1,NMOD);
 temp7 =zeros(NMOD,1);
 for I=1:NMOD,
   temp7(i,1) = -STIFTMD;
 end,
 MODKTMD = [ MODSTIF temp7 ;temp6 STIFTMD];

% generate damping structure with TMD
 temp8 =zeros(1,NMOD);
 temp9 =zeros(NMOD,1);
 for I=1:NMOD,
   temp9(i,1) =-DAMTMD;
 end,
 MODDTMD = [ MODDAM temp9 ;temp8 DAMTMD],

% set initial displacement and velocity
 INDTMD  = [MODIND;0];
 INVTMD  = [MODINV;0];
 end,
%++++++++++++++++++++++++++++++++++++++++++++++
% atjust time step of dynamic load
%++++++++++++++++++++++++++++++++++++++++++++++
if NDYL > 0.
  for i = 1 NDYL,
    if I == 1
```

```
LOAD1 = zeros(NSTEP,1);
I  = 0;
NST(i);
NSTEP;
DT;
for k  = 1:NST(i),
  if k == NST(i),
    I=I+1,
    LOAD1(I,1)= load1(k,2);
  else
    NTSS = round((load1((k+1),1)-load1((k),1))/DT),
    if NTSS ~=0,
      DSTEP = (load1((k+1),2)-load1((k),2))/NTSS,
      for j = 1:NTSS,
        I = I+1;
      LOAD1(I,1)= load1(k,2)+(j-1)*DSTEP;
      end,
    end,
  end,
 end,
elseif i == 2,
 LOAD2 = zeros(NSTEP,2);
 I  = 0;
 for k  = 1:NST(i),
   if k == NST(i),
     I=I+1;
     LOAD2(I,1)= load2(k,2);
   else
     NTSS = round((load2((k+1),1)-load2((k),1))/DT);
     if NTSS ~=0,
       DSTEP = (load2((k+1),2)-load2((k),2))/NTSS;
       for j = 1:NTSS,
         I = I+1;
       LOAD2(I,1)= load2(k,2)+(j-1)*DSTEP;
       end,
     end,
   end,
 end,
elseif i == 3,
 LOAD3 = zeros(NSTEP,1);
  I  = 0,
 for k  = 1:NST(i),
   if k == NST(i),
    I=I+1;
    LOAD3(I,1)= load3(k,2);
   else
    NTSS = round((load3((k+1),1)-load3((k),1))/DT);
   if NTSS ~=0,
     DTSEP = (load3((k+1),2)-load3((k),2))/NTSS,
     for j = 1:NTSS,
       I = I+1;
      LOAD3(I,1)= load3(k,2)+(i-1)*DSTEP;
     end,
   end,
  end,
 end,
elseif i == 4,
 LOAD4 = zeros(NSTEP,1);
  I  = 0;
 for k  = 1:NST(i),
   if k == NST(i),
    I=I+1;
    LOAD4(I,1)= load4(k,2);
   else
    NTSS = round((load4((k+1),1)-load4((k),1))/DT);
                                  if NTSS ~=0,
     DSTEP = (load4((k+1),2)-load4((k),2))/NTSS;
     for j = 1:NTSS,
       I = I+1;
       LOAD4(I,1)= load4(k,2)+(i-1)*DSTEP;
```

```
        end,
      end,
    end,
  end,

elseif i == 5,
  LOAD5 = zeros(NSTEP,1),
   l  = 0,
  for k  = 1:NST(i),
    if k == NST(i),
     l=l+1,
     LOAD5(l,1)= load5(k,2),
    else
     NTSS = round((load5((k+1),1)-load5((k),1))/DT),
     if NTSS ~=0,
       DSTEP = (load5((k+1),2)-load5((k),2))/NTSS,
       for j = 1:NTSS,
         l = l+1;
         LOAD5(l,1)= load5(k,2)+(i-1)*DSTEP,
       end,
     end,
    end,
  end,

elseif i == 6,
  LOAD6 = zeros(NSTEP,1);
   l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
     l=l+1;
     LOAD6(l,1)= load6(k,2);
    else
     NTSS = round((load6((k+1),1)-load6((k),1))/DT),
     if NTSS ~=0,
       DSTEP = (load6((k+1),2)-load6((k),2))/NTSS,
       for j = 1:NTSS,
         l = l+1;
         LOAD6(l,1)= load6(k,2)+(i-1)*DSTEP;
       end,
     end,
    end,
  end,

elseif i == 7,
  LOAD7 = zeros(NSTEP,1);
   l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
     l=l+1;
     LOAD7(l,1)= load7(k,2);
    else
     NTSS = round((load7((k+1),1)-load7((k),1))/DT),
     if NTSS ~=0,
       DSTEP = (load7((k+1),2)-load7((k),2))/NTSS;
       for j = 1:NTSS,
         l = l+1;
         LOAD7(l,1)= load7(k,2)+(i-1)*DSTEP;
       end,
     end,
    end,
  end,

elseif i == 8,
  LOAD8 = zeros(NSTEP,1);
   l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
     l=l+1;
     LOAD8(l,1)= load8(k,2);
    else
```

```
        NTSS = round((load8((k+1),1)-load8((k),1))/DT);
        if NTSS ~=0,
          DSTEP = (load8((k+1),2)-load8((k),2))/NTSS;
          for j = 1:NTSS,
            l = l+1;
            LOAD8(l,1)= load8(k,2)+(i-1)*DSTEP;
          end,
        end,
      end,
    end,

elseif i == 9,
  LOAD9 = zeros(NSTEP,1);
   l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
      l=l+1;
      LOAD9(l,1)= load9(k,2);
    else
      NTSS = round((load9((k+1),1)-load9((k),1))/DT);
      if NTSS ~=0,
        DSTEP = (load9((k+1),2)-load9((k),2))/NTSS;
        for j = 1:NTSS,
          l = l+1;
          LOAD9(l,1)= load9(k,2)+(i-1)*DSTEP;
        end,
      end,
    end,
  end,
elseif i == 10,
  LOAD10 = zeros(NSTEP,1);
   l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
      l=l+1;
      LOAD10(l,1)= load10(k,2);
    else
      NTSS = round((load10((k+1),1)-load10((k),1))/DT);
      if NTSS ~=0,
        DSTEP = (load10((k+1),2)-load10((k),2))/NTSS;
        for j = 1:NTSS,
          l = l+1;
          LOAD10(l,1)= load10(k,2)+(i-1)*DSTEP;
        end,
      end,
    end,
  end,
elseif i == 11,
  LOAD11 = zeros(NSTEP,1);
   l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
      l=l+1;
      LOAD11(l,1)= load11(k,2);
    else
      NTSS =round((load11((k+1),1)-load11((k),1))/DT);
      if NTSS ~=0,
        DSTEP = (load11((k+1),2)-load11((k),2))/NTSS;
        for j = 1:NTSS,
          l = l+1;
          LOAD11(l,1)= load11(k,2)+(l-1)*DSTEP;
        end,
      end,
    end,
  end,

elseif i == 12,
  LOAD12 = zeros(NSTEP,1);
   l  = 0;
  for k  = 1:NST(i),
```

```
    if k == NS"(i),
      l=l+1;
      LOAD12(l,1)= load12(k,2);
    else
      NTSS = round((load12((k+1),') -load12((k),1))/DT);
      if NTSS ~=0,
        DSTEP = (load12((k+1),2)-load12((k),2))/NTSS;
        for j = 1:NTSS,
          l = l+1;
          LOAD12(l,1)= load12(k,2)+(j-1)*DSTEP;
        end,
      end,
    end,
  end,

elseif i == 13,
  LOAD13 = zeros(NSTEP,1);
  l  = 0;
  for k  = 1:NST(i),
    if k == NS"(i),
      l=l+1;
      LOAD13(l,1)= load13(k,2);
    else
      NTSS = round((load13((k+1),') -load13((k),2))/DT);
      if NTSS ~=0,
        DSTEP = (load13((k+1),2)-load13((k),2))/NTSS;
        for j = 1:NTSS,
          l = l+1;
          LOAD13(l,1)= load13(k,2)+(j-1)*DSTEP;
        end,
      end,
    end,
  end,

elseif i == 14,
  LOAD14 = zeros(NSTEP,1);
  l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
      l=l+1;
      LOAD14(l,1)= load14(k,2);
    else
      NTSS = round((load14((k+1),1)-load14((k),1))/DT);
      if NTSS ~=0,
        DSTEP = (load14((k+1),2)-load14((k),2))/NTSS;
        for j = 1:NTSS,
          l = l+1;
          LOAD14(l,1)= load14(k,2)+(j-1)*DSTEP;
        end,
      end,
    end,
  end,
elseif i == 15,
  LOAD15 = zeros(NSTEP,1);
  l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
      l=l+1;
      LOAD15(l,1)= load15(k,2);
    else
      NTSS = round((load15((k+1),1)-load15((k),1))/DT);
      if NTSS ~=0,
        DSTEP = (load15((k+1),2)-load15((k),2))/NTSS;
        for j = 1:NTSS,
          l = l+1;
          LOAD15(l,1)= load15(k,2)+(j-1)*DSTEP;
        end,
      end,
    end,
  end,
```

```
elseif i == 16,
  LOAD16 = zeros(NSTEP,1),
  l = 0,
  for k = 1:NST(i),
    if k == NST(i),
      l=l+1,
      LOAD16(l,1)= load16(k,2),
    else
      NTSS = round((load16((k+1),1)-load16((k),1))/DT),
      if NTSS ~=0,
        DSTEP = (load16((k+1),2)-load16((k),2))/NTSS,
        for j = 1:NTSS,
          l = l+1,
          LOAD16(l,1)= load16(k,2)+(j-1)*DSTEP,
        end,
      end,
    end,
  end,

elseif i == 17,
  LOAD17 = zeros(NSTEP,1),
  l = 0,
  for k = 1:NST(i),
    if k == NST(i),
      l=l+1,
      LOAD17(l,1)= load17(k,2),
    else
      NTSS = round((load17((k+1),1)-load17((k),1))/DT),
      if NTSS ~=0,
        DSTEP = (load17((k+1),2)-load17((k),2))/NTSS,
        for j = 1:NTSS,
          l = l+1,
          LOAD17(l,1)= load17(k,2)+(j-1)*DSTEP,
        end,
      end,
    end,
  end,
elseif i == 18,
  LOAD18 = zeros(NSTEP,1),
  l = 0,
  for k = 1:NST(i),
    if k == NST(i),
      l=l+1;
      LOAD18(l,1)= load18(k,2),
    else
      NTSS = round((load18((k+1),1)-load18((k),1))/DT),
      if NTSS ~=0,
        DSTEP = (load18((k+1),2)-load18((k),2))/NTSS,
        for j = 1:NTSS,
          l = l+1;
          LOAD18(l,1)= load18(k,2)+(j-1)*DSTEP,
        end,
      end,
    end,
  end,
elseif i == 19,
  LOAD19 = zeros(NSTEP,1),
  l = 0,
  for k = 1:NST(i),
    if k == NST(i),
      l=l+1;
      LOAD19(l,1)= load19(k,2),
    else
      NTSS = round((load19((k+1),1)-load19((k),1))/DT),
      if NTSS ~=0,
        DSTEP = (load19((k+1),2)-load19((k),2))/NTSS,
        for j = 1:NTSS,
          l = l+1;
          LOAD19(l,1)= load19(k,2)+(j-1)*DSTEP,
        end,
```

```
        end,
      end,
    end,

elseif i == 20,
  LOAD20 = zeros(NSTEP,1);
  l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
      l=l+1,
      LOAD20(l,1)= load20(k,2);
    else
      NTSS = round((load20((k+1),')-load20((k),1))/DT);
      if NTSS ~=0,
        DSTEP = (load20((k+1),2)-load20((k),2))/NTSS;
        for j = 1:NTSS,
          l = l+1,
        LOAD20(l,1)= load20(k,2)+(i-1)*DSTEP;
        end,
      end,
    end,
  end,

elseif i == 21,
  LOAD21 = zeros(NSTEP,1);
  l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
      l=l+1;
      LOAD21(l,1)= load21(k,2);
    else
      NTSS = round((load21((k+1),')-load21((k),1))/DT);
      if NTSS ~=0,
        DSTEP = (load21((k+1),2)-load21((k),2))/NTSS;
        for j = 1:NTSS,
          l = l+1;
          LOAD21(l,1)= load21(k,2)+(i-1)*DSTEP;
        end,
      end,
    end,
  end,

elseif i == 22,
  LOAD22 = zeros(NSTEP,1);
  l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
      l=l+1;
      LOAD22(l,1)= load22(k,2);
    else
      NTSS = round((load22((k+1),1)-load22((k),1))/DT);
      if NTSS ~=0,
        DSTEP = (load22((k+1),2)-load22((k),2))/NTSS;
        for j = 1:NTSS,
          l = l+1;
        LOAD22(l,1)= load22(k,2)+(i-1)*DSTEP;
        end,
      end,
    end,
  end,
elseif i == 23,
  LOAD23 = zeros(NSTEP,1);
  l  = 0;
  for k  = 1:NST(i),
    if k == NST(i),
      l=l+1;
      LOAD23(l,1)= load23(k,2);
    else
      NTSS = round((load23((k+1),1)-load23((k),1))/DT);
      if NTSS ~=0,
```

```
          DSTEP = (load23((k+1),2)-load23((k),2))/NTSS.
          for j = 1:NTSS,
            I = I+1,
          LOAD23(I,1)= load23(k,2)+(j-1)*DSTEP.
          end.
        end.
      end.
    end.
elseif i == 24.
  LOAD24 = zeros(NSTEP,1).
  I  = 0;
  for k  = 1:NST(i).
    if k == NS(i).
      I=I+1.
      LOAD24(I,1)= load24(k,2).
    else
      NTSS = round((load24((k+1),1)-load24((k),1))/DT).
      if NTSS ~=0.
        DSTEP = (load24((k+1),2)-load24((k),2))/NTSS.
        for j = 1:NTSS.
          I = I+1.
        LOAD24(I,1)= load24(k,2)+(j-1)*DSTEP.
        end.
      end.
    end.
  end.

elseif i == 25.
  LOAD25 = zeros(NSTEP,1).
  I  = 0;
  for k  = 1:NST(i).
    if k == NS(i).
      I=I+1.
      LOAD25(I,1)= load25(k,2).
    else
      NTSS = round((load25((k+1),1)-load25((k),1))/DT).
      if NTSS ~=0.
        DSTEP = (load25((k+1),2)-load25((k),2))/NTSS.
        for j = 1:NTSS.
          I = I+1;
        LOAD25(I,1)= load25(k,2)+(j-1)*DSTEP.
        end.
      end.
    end.
  end.
end.
%
% atjust time step of ground motion
%
 if IGA >0.
  GAL = zeros(NSTEP,1);
    I  = 0;
   NGAST
   for k  = 1:NGAST.
     if k == NGAST.
       I=I+1;
       GAL(I,1)= GA(k,2);
     else
       NTSS = round((GA((k+1),1)-GA((k),1))/DT).
       if NTSS ~=0.
         DSTEP = (GA((k+1),2)-GA((k),2))/NTSS.
         for j = 1:NTSS.
           I = I+1;
         GAL(I,1)= GA(k,2)+(j-1)*DSTEP.
         end.
       end.
     end.
   end.
```

```
  end,
%
% read load data from file (LOADI) input in load for caculate
RESP   = zeros((2*NMOD),NSTEP),
DYDISP = zeros(NDOF,NSTEP),
DYFORC = zeros(NDOF,NSTEP),
CINDYFORC = [],
BINDYFORC = [],
TIME     = zeros(1,NSTEP),
for j = 1:NSTEP
  if j==1,
    TIME(1,1)=0,
  else,
    TIME(1,j) = TIME(1,j-1)+DT,
  end,
%set hperslide of fig
per=(j/NSTEP)*100,
set(hperstring1,'string',int2str(per)),
set(hperslide,'value',per/100),
% input leteral degree of freedom
if NDYL >0,
  DYLOAD = zeros(NDOF,1),
  for i = 1:NDYL
  if i == 1,
    id = JTLDOF(JDYL(i)),
    DYLOAD(id,1)=LOAD1(j,1),
  elseif i == 2,
    id = JTLDOF(JDYL(i)),
    DYLOAD(id,1)=DYLOAD(id,1)+LOAD2(j,1),
  elseif i == 3,
    id = JTLDOF(JDYL(i)),
    DYLOAD(id,1)=DYLOAD(id,1)+LOAD3(j,1),
  elseif i == 4,
    id = JTLDOF(JDYL(i)),
    DYLOAD(id,1)=DYLOAD(id,1)+LOAD4(j,1),
elseif i == 5,
  id = JTLDOF(JDYL(i)),
  DYLOAD(id,1)=DYLOAD(id,1)+LOAD5(j,1),
elseif i == 6,
  id = JTLDOF(JDYL(i)),
  DYLOAD(id,1)=DYLOAD(id,1)+LOAD6(j,1),
elseif i == 7,
  id = JTLDOF(JDYL(i)),
  DYLOAD(id,1)=DYLOAD(id,1)+LOAD7(j,1),
elseif i == 8,
  id = JTLDOF(JDYL(i)),
  DYLOAD(id,1)=DYLOAD(id,1)+LOAD8(j,1),
elseif i == 9,
  id = JTLDOF(JDYL(i)),
  DYLOAD(id,1)=DYLOAD(id,1)+LOAD9(j,1),
elseif i == 10,
  id = JTLDOF(JDYL(i)),
  DYLOAD(id,1)=DYLOAD(id,1)+LOAD10(j,1),
elseif i == 11,
  id = JTLDOF(JDYL(i)),
  DYLOAD(id,1)=DYLOAD(id,1)+LOAD11(j,1),
elseif i == 12,
  id = JTLDOF(JDYL(i)),
  DYLOAD(id,1)=DYLOAD(id,1)+LOAD12(j,1),
elseif i == 13,
  id = JTLDOF(JDYL(i)),
  DYLOAD(id,1)=DYLOAD(id,1)+LOAD13(j,1),
elseif i == 14,
  id = JTLDOF(JDYL(i)),
  DYLOAD(id,1)=DYLOAD(id,1)+LOAD14(j,1),
elseif i == 15,
  id = JTLDOF(JDYL(i)),
  DYLOAD(id,1)=DYLOAD(id,1)+LOAD15(j,1),
elseif i == 16,
  id = JTLDOF(JDYL(i)),
```

```
      DYLOAD(id 1)=DYLOAD(id,1)+LOAD16(j,1).
    elseif i == 17.
      id = JTLDCF(JDYL(i)).
      DYLOAD(id 1)=DYLOAD(id,1)+LOAD17(j,1);
    elseif i == 18.
      id = JTLDCF(JDYL(i)).
      DYLOAD(id 1)=DYLOAD(id,1)+LOAD18(j,1);
    elseif i == 19.
      id = JTLDCF(JDYL(i)).
      DYLOAD(id 1)=DYLOAD(id,1)+LOAD19(j,1).
    elseif i == 20.
      id = JTLDCF(JDYL(i)).
      DYLOAD(id 1)=DYLOAD(id,1)+LOAD20(j,1);
    elseif i == 21.
      id = JTLDOF(JDYL(i)).
      DYLOAD(id 1)=DYLOAD(id,1)+LOAD21(j,1);
    elseif i == 22.
      id = JTLDOF(JDYL(i));
      DYLOAD(id 1)=DYLOAD(id,1)+LOAD22(j,1);
    elseif i == 23.
      id = JTLDOF(JDYL(i));
      DYLOAD(id 1)=DYLOAD(id,1)+LOAD23(j,1);
    elseif i == 24.
      id = JTLDOF(JDYL(i));
      DYLOAD(id 1)=DYLOAD(id,1)+LOAD24(j,1).
    elseif i == 25.
      id = JTLDOF(JDYL(i)).
      DYLOAD(id 1)=DYLOAD(id,1)+LOAD25(j,1).
     end.
   end.
  end.
%
% ground motion loads for each time step
%
 if IGA > 0.
   GALOAD = zeros(NDOF,1);
   temp2 = size(frenod,2);
      for i = 1:temp2.
        temp3 = frenod(i);
         GALOAD(temp3,1) = MASS(temp3,temp3)*GAL(j,1);
        end.
  end.
%
% tranform force to modal force
%
if NDYL > 0.
MODDYL=zeros(NMOD,1);
   for i=1.NMOD.
     MODDYL(i,1)=V(:,i)'*DYLOAD(frenod,1);
   end.
end.
%
%tranform ground motion to modal
%
if IGA > 0.
  MODGAL=zeros(NMOD,1);
   for i=1:NMOD.
     MODGAL(i,1)=-V(:,i)'*GALOAD(frenod,1);
   end.
 end.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%solve time history by state space vector
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if SOLVE == 0.
  I  = eye(NMOD);
  O  = zeros(NMOD);
 tempA = -inv(MODMA)*MODSTIF;
 tempB = -inv(MODMA)*MODDAM;
  B  = [zeros(NMOD);-inv(MODMA)];
```

```
  A   = [O I;tempA tempB];
  X0  = [MODINC;MODINV];
  if (IGA > 0)&(NDYL>0),
    P = MODDYL+MODGAL;
  elseif IGA > 0,
    P = MODGAL;
  elseif NDYL >0,
    P = MODDYL
  end,
  [n,m]   = size(B);
  PHI     = expm(A*DT);
  GAMMA   = inv(A)*(PHI-eye(n))*B;
  TX      = X0;
  RESP(:,j)= TX;
  TX      =(PHI*TX)+(GAMMA*P);
  MODIND  = TX(1:NMOD,1);
  MODINV  = TX(NMOD+1:n,1);
  TEMPDISP = zeros(NDOF,1);
  for i=1:NMOD
   TEMPDISP(frenod) = TEMPDISP(frenod)+ V(:,i)*MODIND(i,1);
  end
  DYDISP(frenod,j) = TEMPDISP(frenod);
  DYDISP(resnod,j) = -inv(aa)*ak*DYDISP(frenod,j);

% ************************************************
% 3  Determine the internal force for each member
% ************************************************
% (3.1) Determine moment and shear at both two ends of each members
if COLUMNOUT > 0,
  for k = 1:COLUMNOUT,
    i = COLUMN(k);
      id  = (i-1)*4+1;                    % 1st column index for element stiffness [CKER](i) in
[AKER]
      id1 = JTLDOF(CJN(i));               % dof no. for lateral dof of joint CJN(i)
      id2 = (CJN(i)-1)*DOF+1;             % dof no. for vertical dof of joint CJN(i)
      id3 = JTLDOF(CJP(i));               % dof no. for lateral dof of joint CJP(i)
      id4 = (CJP(i)-1)*DOF+1;             % dof no. for vertical dof of joint CJP(i)
      CINFORC([1 3 4 6],i) = AKE(1:4,id:(id+3))*DYDISP([id1 id2+1 id3 id4+1],j);
    CINFORC([2 5],i)                  = (AKA(1,i)).*([1 -1,-1 1]*DYDISP([id2 id4],j));
  end,
end,
CINDYFORC =[ CINDYFORC CINFORC];
if BEAMOUT >0,
  for k = 1:BEAMOUT,
    i = BEAM(k);
      id  = NCOL*4+(i-1)*4+1;             % 1st column index for element stiffness [BKER](i) in [AKER]
      id1 = (BJN(i)-1)*DOF+1;             % dof no. for vertical dof of joint BJN(i)
      id2 = (BJP(i)-1)*DOF+1;             % dof no. for vertical dof of joint BJP(i)
    BINFORC([1 2 3 4],i) = AKE(1:4,id:(id+3))*DYDISP([id1 id1+1 id2 id2+1],j);
  end,
end,
BINDYFORC=[BINDYFORC BINFORC];

%
%  solve structure with tuned mass damper
%
else
  I   = eye(NMOD+1);
  O   = zeros(NMOD+1);
 tempA = -inv(MODMTMD)*MODKTMD;
 tempB = -inv(MODMTMD)*MODDTMD;
  B   = [zeros(NMOD+1),-inv(MODMTMD)];
  A   = [O I;tempA tempB];
  X0  = [INDTMD;INVTMD];
  if (IGA > 0)&(NDYL>0);
    P = [MODDYL;0] +[MODGAL;-MTMD*GAL(j,1)];
  elseif IGA > 0,
    P = [MODGAL;-MTMD*GAL(j,1)];
  elseif NDYL >0,
    P = [MODDYL;0];
```

```
  end,
  [n,m]   = size(B);
  PHI     = expm(A*DT);
  GAMMA   = inv(A)*(PHI-eye(n))*B;
  TX      = X0;
  TX      =(PHI*TX)+(GAMMA*P);
  INDTMD  = TX(1:NMOD+1,1);
  INVTMD  = TX(NMOD+2:n,1);
  % displacement of TMD
  DYDISPTMD(j)=TX(NMOD+1);
  TEMPDISP = zeros(NDOF,1);
  for i=1:NMOD,
   TEMPDISP(frenod) = TEMPDISP(frenod)+ V(:,i)*INDTMD(i,1);
  end,
  DYDISP(frenod,j) = TEMPDISP(frenod);
  DYDISP(resnod,j) = -inv(aa)*ak*DYDISP(frenod,j);


% ************************************************
% 3  Determine the internal force for each member
% ************************************************
% (3.1) Determine moment and shear at both two ends of each members
if COLUMNOUT > 0,
  for k = 1:COLUMNOUT,
    i = COLUMN(k);
      id  = (i-1)*4+1;                          % 1st column index for element stiffness [CKER](i) in
[AKER]
      id1 = JTLDOF(CJN(i));          % dof no. for lateral dof of joint CJN(i)
      id2 = (CJN(i)-1)*DOF+1,        % dof no. for vertical dof of joint CJN(i)
      id3 = JTLDOF(CJP(i));          % dof no. for lateral dof of joint CJP(i)
      id4 = (CJP(i)-1)*DOF+1;        % dof no. for vertical dof of joint CJP(i)
      CINFORC([1 3 4 6],i) = AKE(1:4,id:(id+3))*DYDISP([id1 id2+1 id3 id4+1],j);
   CINFORC([2 5],i)              = (AKA(1,i)).*([1 -1,-1 1]*DYDISP([id2 id4],j));
  end,
end,

CINDYFORC =[ CINDYFORC CINFORC];
if BEAMOUT >0,
  for k = 1:BEAMOUT,
    i = BEAM(k);
      id  = NCOL*4+(i-1)*4+1;          % 1st column index for element stiffness [BKER](i) in [AKER]
      id1 = (BJN(i)-1)*DOF+1;          % dof no. for vertical dof of joint BJN(i)
      id2 = (BJP(i)-1)*DOF+1;          % dof no. for vertical dof of joint BJP(i)
   BINFORC([1 2 3 4],i) = AKE(1:4,id:(id+3))*DYDISP([id1 id1+1 id2 id2+1],j);
  end,
end,
BINDYFORC=[BINDYFORC BINFORC];


end,
end,
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%sent displacement story to GUI
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if SOLVE >0,
  U=[DYDISP(frenod,:);DYDISPTMD];
else,
  U=DYDISP(frenod,:);
end,

% ************************************
%       OUTPUT  FILE
% ************************************

%open output files story
for i = 1:NSTORY,
  fp(i)=fopen(['Story' int2str(STORY(i)) '.txt'],'w');
end,
% open out put files column
temp=NSTORY;
for i = 1:COLUMNOUT
    fp(temp+i)=fopen(['column' int2str(COLUMN(i)) '.txt'],'w');
```

```
end,
% open out put files beam
temp=NSTORY+COLUMNOUT
for i = 1:BEAMOUT,
  fp(temp+i)=fopen(['beam \BEAM' num2str(i) '.txt'],'w');
end,
%write file output story displacement
for i=1:NSTORY,
  temp =find(JTJ==STORY(i)),
  temp1=JTLDOF(temp(1)),
  fprintf(fp(i),'%8s\n','Displacement'),
  for j=1:NSTEP,
    fprintf(fp(i),'%10.4f\n',DYDISP(temp1,j));
  end,
end,
%write file output internal force column
temp=NSTORY;
for i=1:COLUMNOUT,
  fprintf(fp(temp+i),'%8s%25s%15s%18s%18s\n','TIME','FORCE-X1','MOM-Z1','FORCE-X2','MOM-Z2'),
  for j=1:NSTEP,
      k=((j-1)*COLUMNOUT+COLUMN(i)),
      fprintf(fp(temp+i),'%10.4f%20.4f%20.4f%20.4f%20.4f\n',TIME(j),CINDYFORC(1,k),CINDYFORC
(3,k),CINDYFORC(4,k),CINDYFORC(6,k));
  end,
end,
%write file output internal force beam
temp=NSTORY+COLUMNOUT;
for i=1:BEAMOUT,
  fprintf(fp(temp+i),'%8s%25s%15s%18s%18s\n','TIME','FORCE-X1','MOM-Z1','FORCE-X2','MOM-Z2');
  for j=1:NSTEP,
      k=((j-1)*BEAMOUT+BEAM(i)),
      fprintf(fp(temp+i),'%10.4f%20.4f%20.4f%20.4f%20.4f\n',TIME(j),BINDYFORC(1,k),BINDYFORC
(2,k),BINDYFORC(3,k),BINDYFORC(4,k));
  end,
end,
fclose('all');
set(hperslide,'visible','off');

++++++++++++++++++++++++++++++++++++++++++++++++++++++++
%                    Function : Input the data of structures
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
function viewstructure1(fname)
global X Y AC ABM xy story njoint NSTR;
global NSTR NCOLL NCOLT NBEAMT NCOL NBEAM NJOINT,
global CEI CXI CA  CLENG;
global BEI BXI BA  BLENG,
global JTI JTJ;
global CJN CJP BJN BJP,
global NSUP SUPNO SDIR;
global mass POS
DOF = 3;
fp = fopen(fname,'r');
fgetl(fp);
NSTR      = fscanf(fp,'%d');      fscanf(fp,'%c',1),
NCOLL     = fscanf(fp,'%d');      fscanf(fp,'%c',1);
NCOLT     = fscanf(fp,'%d');      fscanf(fp,'%c',1),
NBEAMT    = fscanf(fp,'%d');      fscanf(fp,'%c',1),
NCOL      = fscanf(fp,'%d');      fscanf(fp,'%c',1),
NBEAM     = fscanf(fp,'%d');      fscanf(fp,'%c',1);
NJOINT  = fscanf(fp,'%d');        fscanf(fp,'%c',1),
fgetl(fp);
fgetl(fp);
for i = 1:NJOINT,
  mass(i) = fscanf(fp,'%f',1);
end

fgetl(fp);
fgetl(fp);
for i = 1:NCOL,
      CNO         = fscanf(fp,'%d');      fscanf(fp,'%c',1);
```

```
  for k = 1:NST(i),
      for j = 1:dof,
      load11(k,j) = fscanf(fp2,'%f',1);
      end,
  end,
 elseif i==12,
  for k = 1:NST(i),
      for j = 1:dof,
      load12(k,j) = fscanf(fp2,'%f',1);
      end,
  end,
 elseif i==13,
  for k = 1:NST(i),
      for j = 1:dof,
      load13(k,j) =fscanf(fp2,'%f',1)
      end,
  end,
 elseif i==14,
  for k = 1:NST(i),
      for j = 1:dof,
      load14(k,j) = fscanf(fp2,'%f',1);
      end,
  end,
 elseif i==15,
  for k = 1:NST(i),
      for j = 1:dof,
      load15(k,j) = fscanf(fp2,'%f',1);
      end,
  end,
 elseif i==16,
  for k = 1:NST(i),
      for j = 1:dof,
      load16(k,j) = fscanf(fp2,'%f',1);
      end,
  end,
 elseif i==17,
  for k = 1:NST(i),
      for j = 1:dof,
      load17(k,j) = fscanf(fp2,'%f',1);
      end,
  end,
 elseif i==18,
  for k = 1:NST(i),
      for j = 1:dof,
      load18(k,j) = fscanf(fp2,'%f',1);
      end,
  end,
 elseif i==19,
  for k = 1:NST(i),
      for j = 1:dof,
      load19(k,j) = fscanf(fp2,'%f',1);
      end,
  end,
 elseif i==20,
  for k = 1:NST(i),
      for j = 1:dof,
      load20(k,j) = fscanf(fp2,'%f',1);
      end,
  end,
 elseif i==21,
  for k = 1:NST(i),
      for j = 1:dof,
      load21(k,j) = fscanf(fp2,'%f',1);
      end,
  end,
 elseif i==22,
  for k = 1:NST(i),
      for j = 1:dof,
      load22(k,j) = fscanf(fp2,'%f',1);
      end,
```

```
            x(j+1)=x(j)+temp4;
            y(j+1)=y(j);
          else,
            BNUM=find(BJN==temp2(j));
            temp4=BLENG(BNUM);
            x(j+1)=x(j)+temp4;
            y(j+1)=y(j);
          end,
        end,
      X=[X x];
      Y=[Y y];
    temp5=temp5+temp1(i);
    y(1)=temp5;

    end,
    xy=[X'Y'];
    AC=zeros(length(xy));
    AB=zeros(length(xy));
    for i=1:NCOL,
      ax=CJN(i);
      ay=CJP(i);
      AC(ax,ay)=1;
    end,
    for i=1 NBEAM,
      ax=BJN(i);
      ay=BJP(i);
      ABM(ax,ay)=1
    end,

    function linput
    ++++++++++++++++++++++++++++++++++++++++++++++++++++
    %                Function : input the loading data
    ++++++++++++++++++++++++++++++++++++++++++++++++++++
    function linput01(fname)
    global NMOD NSTEP DT SOLVE CP STORY BEAM COLUMN NSTORY BEAMOUT COLUMNOUT
    global fp2 fp3;
    global LJTNO LUNFNO LPTNO LSDNO;
    global LJTN LJTDIR;
    global LUNFMB LUNFLEN LUNFBEG LUNFMAG;
    global LPTMB LPTLOC LPTMAG;
    global LSDJN LSDDIR;
    global A NDYL JDYL NST DYLN
    global IGA NGAST GA
    global load1 load2 load3 load3 load4 load5 load6 load7 load8 load9 load10 load11 load12
    global load13 load14 load15 load16 load17 load18 load19 load20 load21 load22
    global MTMD DAMTMD STIFTMD
    DOF = 3;
    dof = 2;
    fp2 = fopen(fname,'r');
    fgetl(fp2);
    LJTNO  =fscanf(fp2,'%d');          fscanf(fp2,'%c',1);
    LUNFNO=fscanf(fp2,'%d');           fscanf(fp2,'%c',1);
    LPTNO  =fscanf(fp2,'%d');          fscanf(fp2,'%c',1);
    LSDNO =fscanf(fp2,'%d');      fscanf(fp2,'%c',1);
    NDYL  =fscanf(fp2,'%d');      fscanf(fp2,'%c',1);
    IGA   =fscanf(fp2,'%d');      fgetl(fp2);
    if LJTNO > 0,
      fgetl(fp2);
      for i = 1:LJTNO,
        LJTN(i)       = fscanf(fp2,'%d');          fscanf(fp2,'%c',1);
        for j = 1 DOF,
          LJTDIR(i,j) = fscanf(fp2,'%f');      fscanf(fp2,'%c',1);
        end,
      end,
    end,

    if NDYL >0,
      fgetl(fp2);
      for i = 1 NDYL
```

```
  CEI(CNO)      = fscanf(fp,'%f'),          fscanf(fp,'%c',1),
      CXI(CNO)      = fscanf(fp,'%f');          fscanf(fp,'%c',1),
      CA(CNO)       = fscanf(fp,'%f');          fscanf(fp,'%c',1),
  CLENG(CNO) = fscanf(fp,'%f'),          fscanf(fp,'%c',1),
 end
fgetl(fp),
fgetl(fp),
for i = 1:NBEAM,
  BNO    = fscanf(fp,'%d');          fscanf(fp,'%c',1);
  BEI(BNO)      = fscanf(fp,'%f');          fscanf(fp,'%c',1);
  BXI(BNO)      = fscanf(fp,'%f');          fscanf(fp,'%c',1);
  BLENG(BNO) = fscanf(fp,'%f');          fscanf(fp,'%c',1);
end,
fgetl(fp);
fgetl(fp),
for i = 1:NJOINT,
  JNO                = fscanf(fp,'%d'),     fscanf(fp,'%c',1),
  JTI(JNO)           = fscanf(fp,'%d'),     fscanf(fp,'%c',1),
  JTJ(JNO)           = fscanf(fp,'%d'),     fgetl(fp);
end
fgetl(fp),
for i = 1:NCOL,
      CNO                = fscanf(fp,'%d');          fscanf(fp,'%c',1);
  CJN(CNO)           = fscanf(fp,'%d');     fscanf(fp,'%c',1);
      CJP(CNO)           = fscanf(fp,'%d');     fgetl(fp);
end,
 fgetl(fp),
 for i = 1:NBEAM,
  BNO                = fscanf(fp,'%d'),     fscanf(fp,'%c',1),
  BJN(BNO)           = fscanf(fp,'%d'),     fscanf(fp,'%c',1),
      BJP(BNO)           = fscanf(fp,'%d'),     fgetl(fp),
end,
fgetl(fp),
NSUP   =      fscanf(fp,'%d'),                    fgetl(fp);
SUPNO = zeros(1,NSUP),
SDIR    = zeros(NSUP,3),
for i = 1:NSUP,
  SUPNO(i)      = fscanf(fp,'%d'),     fscanf(fp,'%c',1);
  SDIR(i,1)  = fscanf(fp,'%d'),     fscanf(fp,'%c',1);
  SDIR(i,2)  = fscanf(fp,'%d');          fscanf(fp,'%c',1);
      SDIR(i,3)  = fscanf(fp,'%d'),          fscanf(fp,'%c',1);
end,
fgetl(fp);
fclose(fp);
x(1)=0;y(1)=0;
X=[];
Y=[];
temp5=0;
NSTR;
story=[],
for I=1:NSTR,
  temp=find(JTI==1);
  temp1(i)=CLENG(temp(1));
  temp2=find(JTJ==i);
  story=[story temp2];
  if I~=1,
    XLINT=JTI(temp2(1));
    x(1)=x(XLINT);
  end,
  temp3=size(temp2,2);
  njoint(I)=temp3;
  if I==1,
    temp2=find(JTJ==2);
    for j=1:temp3-1,
      BNUM(j)=find(BJN==temp2(j));
    end,
  end,
    for j=1:temp3-1,
      if i==1,
        temp4=BLENG(BNUM(j)),
```

```
    fgetl(fp2);
    fgetl(fp2);
    JDYL(i) = fscanf(fp2,'%d'),                fscanf(fp2,'%c',1);
    NST(i) = fscanf(fp2,'%d');     fscanf(fp2,'%c',1);
fgetl(fp2);
fgetl(fp2);
if i==1,
  for k = 1:NST(i),
        for j = 1 dof,
        load1(k,j) = fscanf(fp2,'%f',1);
        end,
  end,
  load1
elseif i==2,
  for k = 1:NST(i),
        for j = 1 dof,
        load2(k,j) = fscanf(fp2,'%f',1);
        end,
  end,
elseif i==3,
  for k = 1:NST(i),
        for j = 1 dof,
        load3(k,j) = fscanf(fp2,'%f',1);
        end,
    end,
    load3
  elseif i==4,
  for k = 1:NST(i),
        for j = 1 dof,
        load4(k,j) = fscanf(fp2,'%f',1);
        end
    end,
    load4
  elseif i==5,
  for k = 1 NST(i),
        for j = 1.dof,
        load5(k,j) = fscanf(fp2,'%f',1);
        end,
  end,
 elseif i==6,
 for k = 1:NST(i),
        for j = 1:dof,
        load6(k,j) =fscanf(fp2,'%f',1);
        end,
 end,
elseif i==7,
 for k = 1:NST(i),
        for j = 1:dof,
        load7(k,j) = fscanf(fp2,'%f',1);
        end,
 end,
elseif i==8,
for k = 1:NST(i),
        for j = 1:dof,
        load8(k,j) = fscanf(fp2,'%f',1);
        end,
 end,
elseif i==9,
for k = 1:NST(i),
        for j = 1:dof,
        load9(k,j) =fscanf(fp2,'%f',1);
        end,
 end,
elseif i==10,
 for k = 1:NST(i),
        for j = 1:dof,
        load10(k,j) =fscanf(fp2,'%f',1);
        end,
 end,
elseif i==11,
```

```
    end
   elseif i==23,
    for k = 1:NST(i),
        for j = 1 dof,
        load23(k,j) = fscanf(fp2,'%f',1);
        end,
    end,
   elseif i==24,
    for k = 1:NST(i),
        for j = 1 dof,
        load24(k,j) =fscanf(fp2,'%f',1)
        end,
    end,
   elseif i==25,
    for k = 1:NST(i),
        for j = 1 dof,
        load25(k,j) = fscanf(fp2,'%f',1);
        end,
     end,
   end,
  fgetl(fp2);
  fgetl(fp2);
 end,
end,
if IGA >0,
  fgetl(fp2);
  FNAME = fgetl(fp2);
  fp3   = fopen([FNAME '.txt'],'r');
  NGAST = fscanf(fp3,'%f',1);
  fgetl(fp3);
  fgetl(fp3);
  for k=1:NGAST,
    for j = 1:dof,
    GA(k,j)  = fscanf(fp3,'%f',1);
    end,
  end,
  fclose(fp3);
end,
%*************************
%*** dynamic solve Data ***
%*************************
fgetl(fp2);
NMOD = fscanf(fp2,'%d',1);    fscanf(fp2,'%c',1);
NSTEP = fscanf(fp2,'%d',1);    fscanf(fp2,'%c',1);
DT   = fscanf(fp2,'%f',1);    fscanf(fp2,'%c',1);
SOLVE = fscanf(fp2,'%d',1);    fscanf(fp2,'%c',1);
fgetl(fp2);
fgetl(fp2);
if NMOD > 0,
  for I =1:NMOD,
    CR(I) = fscanf(fp2,'%f',1);
  end,
end,
if SOLVE > 0,
  fgetl(fp2);
  fgetl(fp2);
  MTMD   = fscanf(fp2,'%f',1);
  DAMTMD  = fscanf(fp2,'%f',1);
  STIFTMD = fscanf(fp2,'%f',1);
end,

%*************************
%*** set out put file  ***
%*************************
fgetl(fp2);
fgetl(fp2);
NSTORY = fscanf(fp2,'%d',1);    fscanf(fp2,'%c',1);
BEAMOUT  = fscanf(fp2,'%d',1);    fscanf(fp2,'%c',1);
COLUMNOUT = fscanf(fp2,'%d',1);    fscanf(fp2,'%c',1);
fgetl(fp2);
```

```
fgetl(fp2);
if NSTORY > 0,
  for i =1:NSTORY,
    STORY(i) = fscanf(fp2,'%d',1);
  end,

end,
if BEAMOUT > 0,
  fgetl(fp2);
        fgetl(fp2);
  for i =1:BEAMOUT,
    BEAM(i) = fscanf(fp2,'%d',1);
  end,
end,
fgetl(fp2);
fgetl(fp2);

if COLUMNOUT > 0,
  for i =1:COLUMNOUT,
    COLUMN(i) = fscanf(fp2,'%d',1);
  end,

end,
fclose(fp2);

%**********************
%*** Loading Data ***
%**********************
%* General loading data
%LJTNO                  =       numbers of loaded joint
%LUNFNO          = numbers of distributed load
%LPTNO                  = numbers of point loading
%* Joint Load
%LJTN(i)                = Joint number of loading no i
%LJTDIR(i,j)    =       magnitude of joint loading no i in the direction j
%* Uniformly-distributed load
%LUNFMB(i)      =       loaded member number
%LUNFLEN(i)     =       length of loading
%LUNFBEG(i)     =       beginning point of loading
%LUNFMAG(i)     =       magnitude of loading (intensity)
%* Point Load
%LPTMB(i)               = loaded member number
%LPTLOC(i)      = location of point load
%LPTMAG(i)      =       magnitude of loading
%NDYYL    = number of dynamic loads
%DYLN     = dynamic load number
%JDYL     = dynamic load action at joint
%NTSTP    = number of time step
%K          = number
%KT       = time at time step number k
%FRO      = force at time step number k
```

function Lateral dof

```
++++++++++++++++++++++++++++++++++++
% purpose:to determine Lateral dof in each story line
++++++++++++++++++++++++++++++++++++
function Lateraldof()
% NLDOF  = total number of lerteral dof overall structure
% NVRDOF = total number of vertical & rotational dof overall structure
% NDOF   = total number of all dof overall structure
% NSTRJT  = number of joint in each story line
% NSTRJTNO = joint number for each story level(contained in row vecter for all story line)
% JTLDOF(i)  = lateral dof number of joint no i
global NDOF NLDOF NSTRJT NSTRJTNO JTLDOF;
global NSTR NJOINT;
global JTJ;
global NVRDOF;
NLDOF=1;
JTLDOF=[];
for i=1:NSTR,
```

```
if i==1,
  temp=find(JTJ==i);
  NSTRJT=size(temp,2);
  [temp1 id]=sort(JTJ(temp));
  NSTRJTNO=[NSTRJTNO temp(id)];
  for j=1:(NSTRJT),
    JTLDOF(NSTRJTNO(j))= NLDOF;
    if NLDOF<NSTRJT,
      NLDOF=NLDOF+1;
    else,
      break,
    end,
  end,
  NLDOF=NLDOF+1;  %check number dof  atsupport
  NSTRJT=size(temp,2) ,
  njoint=0;
  % case story not one
else,
  njoint = njoint+(NSTRJT);
  temp2  = find(JTJ==i),
  NSTRJT = size(temp2,2),
  [temp4 id]=sort(JTJ(temp2)),
  NSTRJTNO=[NSTRJTNO temp2(id)];
  for j=(njoint+1) (njoint+NSTRJT),
    JTLDOF(NSTRJTNO(j))=NLDOF;
  end,
  NLDOF= NLDOF+1;
end,

end,

%
%determine number of leteral dof, vertical & rotational dof and overall dof of structure
%
NVRDOF = 2* NJOINT;
 %set lateral dof numberplace after vertical and rotational dof
JTLDOF = JTLDOF+NVRDOF;
NDOF   =NVRDOF+(NLDOF-1);
```

Local stiffness

```
++++++++++++++++++++++++++++++++++++++++++++++
%                  Stiffness  for the frame 2-D element
++++++++++++++++++++++++++++++++++++++++++++++
function [KE,KA ]=localstiff( RIGID,DIMEN,MODE )
EI=RIGID(1);
XI=RIGID(2);
A=RIGID(3);
LENG=DIMEN(1);
%
% stiffness matrix at the local axis
%
KE = [];
c=EI*XI/(LENG^3);
if MODE==1,
KE =[ 12*c      -6*LENG*c   -12* c     -6*LENG*c;...
     -6*LENG*c   4*LENG^2*c  6*LENG*c   2*LENG^2*c;...
     -12*c       6*LENG*c    12*c       6*LENG*c;...
     -6*LENG*c   2*LENG^2*c  6*LENG*c   4*LENG^2*c];

else
KE=[ 12*c      6*LENG*c      -12* c     6*LENG*c;...
    6*LENG*c   4*LENG^2*c     -6*LENG*c  2*LENG^2*c;...
   -12*c      -6*LENG*c       12*c      -6*LENG*c;...
    6*LENG*c   2*LENG^2*c     -6*LENG*c  4*LENG^2*c];


end,
```

```
%*************************************************
%   axial stiffness [KA]
%*************************************************

if MODE==1,

  KA  =  EI*A/LENG;

else
  KA  =  [];

end,
```

function Assemble

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% Purpose: Assemble the element stiffness to the structural stiffness matrix
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
function Assemble(KE,KA,POSIT,MODE)
global STIF JTLDOF DOF;
JN = POSIT(1);
JP = POSIT(2);

if MODE == 1,  % Column Stiffness Assembling

  %  (1) assemble axial stiffness
        MAN             = (JN-1)*DOF+1;         % dof no  for vertical dof of jt  CJN(i)
        MAP             = (JP-1)*DOF+1          % dof no  for vertical dof of jt  CJP(i)
        STIF([MAN MAP],[MAN MAP]) = STIF([MAN MAP],[MAN MAP]) + KA*[1 -1;-1 1];

  %  (2) assemble rotational and lateral stiffness
        MRN             = (JN-1)*DOF+2          % dof no  for rotational dof of jt CJN(i)
        MRP             = (JP-1)*DOF+2;         % dof no  for rotational dof of jt CJP(i)
        MLN             = (JTLDOF(JN));         % dof no  for lateral dof of jt CJN(i)
        MLP             = (JTLDOF(JP));         % dof no  for lateral dof of jt CJP(i)
        STIF([MLN MRN],[MLN MRN]) = STIF([MLN MRN],[MLN MRN]) + KE([1 2],[1 2]);
        STIF([MLP MRP],[MLP MRP]) = STIF([MLP MRP],[MLP MRP]) + KE([3 4],[3 4]);
        STIF([MLN MRN],[MLP MRP]) = STIF([MLN MRN],[MLP MRP]) + KE([1 2],[3 4]);
        STIF([MLP MRP],[MLN MRN]) = STIF([MLP MRP],[MLN MRN]) + KE([3 4],[1 2]);

else,                                   % Beam Stiffness Assembling

  MVN           = (JN-1)*DOF+1;         % dof number for vertical dof of joint BJN(i)
  MVP           = (JP-1)*DOF+1;         % dof number for vertical dof of joint BJP(i)

  % dof number for rotational dof = MVN+1  or MVP+1
  STIF([MVN MVN+1],[MVN MVN+1]) = STIF([MVN MVN+1],[MVN MVN+1]) + KE([1 2],[1 2]);
  STIF([MVP MVP+1],[MVP MVP+1]) = STIF([MVP MVP+1],[MVP MVP+1]) + KE([3 4],[3 4]);
  STIF([MVN MVN+1],[MVP MVP+1]) = STIF([MVN MVN+1],[MVP MVP+1]) + KE([1 2],[3 4]);
  STIF([MVP MVP+1],[MVN MVN+1]) = STIF([MVP MVP+1],[MVN MVN+1]) + KE([3 4],[1 2]);

end,
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% Purpose: Assemble the element mass to the structural mass matrix
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
function Assemblemass(ME,MA,POSIT,MODE)
global MASS JTLDOF DOF;
JN = POSIT(1);
JP = POSIT(2);

if MODE == 1,  % Column mass Assembling

  %  (1) assemble axial stiffness
        MAN             = (JN-1)*DOF+1;         % dof no  for vertical dof of jt  CJN(i)
        MAP             = (JP-1)*DOF+1;         % dof no  for vertical dof of jt  CJP(i)
        MASS([MAN MAP],[MAN MAP]) = MASS([MAN MAP],[MAN MAP]) + MA;

  %  (2) assemble rotational and lateral stiffness
        MRN             = (JN-1)*DOF+2;         % dof no  for rotational dof of jt CJN(i)
        MRP             = (JP-1)*DOF+2;         % dof no  for rotational dof of jt CJP(i)
```

```
        MLN             = (JTLDOF(JN));         % dof no. for lateral dof of jt CJN(i)
        MLP             = (JTLDOF(JP));         % dof no. for lateral dof of jt CJP(i)
        MASS([MLN MRN],[MLN MRN]) = MASS([MLN MRN],[MLN MRN]) + ME([1 2],[1 2]);
        MASS([MLP MRP],[MLP MRP]) = MASS([MLP MRP],[MLP MRP]) + ME([3 4],[3 4]);
        MASS([MLN MRN],[MLP MRP]) = MASS([MLN MRN],[MLP MRP]) + ME([1 2],[3 4]);
        MASS([MLP MRP],[MLN MRN]) = MASS([MLP MRP],[MLN MRN]) + ME([3 4],[1 2]);

else,                                   % Beam mass Assembling

  MVN           = (JN-1)*DOF+1;         % dof number for vertical dof of joint BJN(i)
  MVP           = (JP-1)*DOF+1;         % dof number for vertical dof of joint BJP(i)
    % dof number for rotational dof = MVN+1  or MVP+1
  MASS([MVN MVN+1],[MVN MVN+1]) = MASS([MVN MVN+1],[MVN MVN+1]) + ME([1 2],[1 2]);
  MASS([MVP MVP+1],[MVP MVP+1]) = MASS([MVP MVP+1],[MVP MVP+1]) + ME([3 4],[3 4]);
  MASS([MVN MVN+1],[MVP MVP+1]) = MASS([MVN MVN+1],[MVP MVP+1]) + ME([1 2],[3 4]);
  MASS([MVP MVP+1],[MVN MVN+1]) = MASS([MVP MVP+1],[MVN MVN+1]) + ME([3 4],[1 2]);

end,
function frame2
++++++++++++++++++++++++++++
%   GUI AT START
++++++++++++++++++++++++++++
function fig = frame2(;
clear all,
% FIG-files
global AC ABM xy njoint story NSTR
global U TIME SOLVE datafile loadfile
figure('NumberTitle','off','Name','DYNAMIC ANALYSIS FRAME2-D WITH TMD',
    'colormap',bone(64),'Units','normalized','position',[0 -0.035 1 0.968]),
ha_background = axes('Units','normalized','Position',[0 0 1 1],'box','on');
patch([0; 1; 1; 0],[0; 0; 1; 1],[0; 0; 0; 0] 'cdata',[5 5 2 2],'facecolor','interp');
set(gca,'xtick',[],'ytick',[]);
text(0.52, 0.09,[['Developed by Panupong Tritham'...
        'Department of Civil Engineer Chulalongkorn University']],...
    'color',[1 0 3 0.4],'fontsize',13 5,'horizontalalignment','center');

text(0.2, 0.88,'DYNAMIC ANALYSIS FRAME 2-D WITH TMD','color',[1 0 3 0.4],...
  'fontsize',19)

load pic_logo2;
axes('Units','normalized','Position',[0.1 0.04 0 1 0.1],'color',[0.8698 0 917 0.92]);
image_handle = image([0.1 0.2],[0.03 0.13],a);
set(gca,'xtick',[],'ytick',[]);set( gca,'Xcolor',[0.8698 0.9167 0.9167],'Ycolor',[0.8698 0.9167 0 9167]),
axis('image');
hc_data = uicontrol(..
        'Units','normalized', ...
        'BackgroundColor',[0.752941176470588 0 752941176470588 0 752941176470588],
        'Callback','[datafile] = openfile',
        'ListboxTop',0, ...
        'Position',[0.13 0.70 0.2 0.06],
  'fontsize',10,..
  'String','Open Data');

hc_loaddata = uicontrol(..
        'Units','normalized', ...
        'BackgroundColor',[0.752941176470588 0 752941176470588 0 752941176470588],
        'Callback','[loadfile] = openlfile',
        'ListboxTop',0, ...
        'Position',[0.13 0.60 0.2 0.06],
  'fontsize',10,..
  'String','Open Load Data');

hc_view=uicontrol(..
        'Units','normalized',
        'BackgroundColor',[0.752941176470588 0 752941176470588 0.752941176470588],
        'ListboxTop',0, ...
        'Position',[0.13 0.50 0.2 0.06], ...
  'fontsize',10,...
  'String','View Structure', ...
  'Callback','view1(datafile)');
```

```
hc_calculate = uicontrol(...
        'Units','normalized', ...
        'ListboxTop',0, ...
        'Position',[0.13 0.40 0.2 0.06], ...
  'fontsize',10,...
  'String','Calculate', ...
  'Callback','Responds(datafile,loadfile)');
hperstring1 = uicontrol(...
        'Units','normalized', ...
        'ListboxTop',0, ...
  'Position',[0.48 0.30 .15  .04], ...
  'Backgroundcolor',[1 1 1], ...
  'fontsize',12,...
  'Style','text');
hperstring2 = uicontrol(...
        'Units','normalized', ...
        'ListboxTop',0, ...
        'Position',[0.40 0.70 0.2 0.06], ...
  'Backgroundcolor',[1 1 1], ...
  'fontsize',12,...
  'Style','edit');
hperstring3 = uicontrol(...
        'Units','normalized', ...
        'ListboxTop',0, ...
  'Position',[0.40 0.60 0.2 0.06], ...
  'Backgroundcolor',[1 1 1], ...
  'fontsize',12,...
  'Style','edit');
hperslide = uicontrol( ...
        'Units','normalized', ...
        'ListboxTop',0,  ...
        'Position',[0 40 0.40 0.30  0.06],
        'Style','slider');
hc_post=uicontrol(...
        'Units','normalized', ...
        'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588],
        'ListboxTop',0, ...
        'Position',[0.13 0.20 0.2 0.06], ...
  'fontsize',10,...
  'String','Post Processtor', ...
  'Callback','output');
hdata=[hperstring1 hperstring2 hperstring3 hperslide];
set(gcf,'userdata',hdata);

++++++++++++++++++++++++++++
%  open  data file
++++++++++++++++++++++++++++
function fname=openfile
clear all;
hdata=get(gcf,'userdata');
hperstring1=hdata(1);
hperstring2=hdata(2);
hperstring3=hdata(3);
hperslide=hdata(4);
[ filename, pathname] = uigetfile('*.*','Open file for reading');
fname=[pathname filename];
set(hperstring2,'string',fname);

++++++++++++++++++++++++++++
%  open  load data file
++++++++++++++++++++++++++++
function [loadfile]= openifile()
hdata=get(gcf,'userdata');
hperstring1=hdata(1);
hperstring2=hdata(2);
hperstring3=hdata(3);
hperslide=hdata(4);
[ filename, pathname] = uigetfile('*.*','Open loadfile for reading');
loadfile =[pathname filename];
set(hperstring3,'string',loadfile);
```

```
function view1
++++++++++++++++++++++++++++++
%   GUI   show  geometry  of structure
++++++++++++++++++++++++++++++
function view1(datafile)
viewstructure1(datafile)
global X Y AC ABM xy story njoint NSTR;
a=max(xy),
a=max(a),
x=xy(:,1)/a;
y=xy(:,2)/a,
x(1)-x(njoint(1));
center=0.5*(x(njoint(1))-x(1)),
x=x+(0.5-center),
xy=[x y],

% set linewidth of column
if NSTR < 15,
    cthick = 2;
else,
  cthick =  75;
end,
% set joint of story
intemp=njoint(1)+1,
for i=2:NSTR,
  xp(1,i)=x(intemp),
  xp(2,i)=x(intemp+njoint(i)-1);
  xp(3,i)=x(intemp+njoint(i)-1);
  xp(4,i)=x(intemp),
  yp(1,i)=y(intemp),
  yp(2,i)=y(intemp),
  yp(3,i)=y(intemp)+0.2*y(njoint(1)+1),
  yp(4,i)=y(intemp)+0.2*y(njoint(1)+1),
  intemp=intemp + njoint(i) ;
end,      % for i=2:NSTR,
 x=x;
y=y;
xp=xp,
njoint
NSTR ;
figure('NumberTitle','off','Name','DYNAMIC ANALYSIS FRAME2-D WITH TMD',...
    'colormap',bone(64));
hc_background = axes('Units','normalized','Position',[0 0 1 1],'box','on');
patch([0; 1; 1; 0],[0; 0; 1; 1],[0; 0; 0; 0],'cdata',[5 5 2 2],'facecolor','interp');
set(gca,'xtick',[],'ytick',[]);
hc_close = uicontrol('Style','pushbutton', ...
        'Units','normalized', ...
        'ListboxTop',0, ...
        'Position',[0.85 0.202380952380952 0.1196 0.09], ...
  'fontsize',10,...
  'String','Close', ...
  'callback','close(gcf)');
ha_g0 = axes('Units','normalized','color',[0 0.2 .95],'Position',[0.021 .0381 .793 .93]),
axes(ha_g0);
[xd1 yd1]=gplot(AC,xy);
ha=plot(xd1,yd1);
set(ha,'linewidth',cthick),

% draw story
for i=2:NSTR,
  patch(xp(:,i),yp(:,i),[0 0 0]);
end,

% draw ground
x(1);
tempg=x(1)-0.75*(x(2)-x(1));
tempg1=abs(x(njoint(1))-x(1)+(x(2)-x(1)))/(0.25*(x(2)-x(1)));
l=(0.25*(x(2)-x(1)));
for i=1:tempg1,
```

```
  patch([tempg+(i*1) tempg+(i+1)*l tempg+(i+2)*l],[0 0 -l],[0 0 0]),
end
set(gca,'XLim',[-0.5 1.5],'YLim',[-0.5 1.5],...
set(gca,'xtick',[],'ytick',[]),

function output
%++++++++++++++++++++++++++++++
%   GUI  show  animation of structure
%++++++++++++++++++++++++++++++
function output(action)
global AC ABM xy njoint story NSTR
global U TIME SOLVE x y xp yp

n=length(TIME),
tempn=ceil(TIME(n)),
tempnx=0:tempn,
tempny=zeros(1,tempn+1),
if nargin<1,
  action='initialize',
end,
if strcmp(action,'initialize'),
tmdnjoint=njoint,
a=max(xy),
a=max(a),
xy
x=xy(:,1)/a
y=xy(:,2)/a
center=0.5*(x(njoint(1))-x(1))
x=x+(0.5-center)
xmax=x(njoint(1))-x(1)
xy=[x y]
NSTR
if SOLVE==1
  TMDNSTR=NSTR+1;
  tmdnjoint(TMDNSTR-1)=njoint(TMDNSTR-1)+2,
  tmdnjoint(TMDNSTR)=1;
  temp=[0.5 1. 0.5 1+y(njoint(1)+1);0.5 1+0.55*y(njoint(1)+1)];
  temp1=size(xy,1),
  temp2=zeros(temp1,3),
  temp3=zeros(3,temp1),
  temp4=zeros(3),
  AC=[AC temp2;temp3 temp4];
  xy=[xy;temp];
  temp1=size(xy,1);
  story=(1:temp1);
  AC(temp1-2,temp1-1)=1;
  AC(temp1-1,temp1)=1;
 end,  % if SOLVE==1
NSTR
% set linewidth of beam
if NSTR < 15,
    bthick = 12-((NSTR-1)*0.7);
  else,
    bthick = 2;
end,

intemp=njoint(1)+1;
for i=2:NSTR,
  xp(1,i)=x(intemp);
  xp(2,i)=x(intemp+njoint(i)-1);
  xp(3,i)=x(intemp+njoint(i)-1);
  xp(4,i)=x(intemp);
  yp(1,i)=y(intemp);
  yp(2,i)=y(intemp);
  yp(3,i)=y(intemp)+0.2*y(njoint(1)+1);
  yp(4,i)=y(intemp)+0.2*y(njoint(1)+1);
  intemp=intemp + njoint(i) ;
end,      % for i=2:NSTR,

  if SOLVE ==1,
```

```
  'fontsize',10,
  'string','TIME',
        'Style','edit');
hc_string= uicontrol(
        'Units','normalized',
        'ListboxTop',0, ...
        'Position',[0.88 0.73 0.05 0.035], ...
  'fontsize',10,...
  'Style','text');

UMAX=max(abs(J));
UMAX=max(UMAX,[],2);
UMAX1=UMAX;

if SOLVE==1,
    TEMPU=U(1:NSTR-1,:);
    TEMUMAX=max(abs(TEMPU));
    TEMUMAX=max(TEMUMAX,[],2);
    UMAX1=TEMUMAX;
  end,

ha_g2 = axes('Units','normalized','Position',[0.1 0.1 0.6 0.26],'box','on');
axes(ha_g2);
p0=plot(TIME,U(NSTR-1,:));
set( gca,'fontsize',8);set( gca,'Xcolor',[1 0.3 0.4],'Ycolor',[1 0.3 0.4]);
xlabel('Time (sec)');
ylabel('Displacement Top Floor ( meters)');
hold on;
plot(tempnx,tempny,'k');
set( gca,'xlim',[0 tempn]);
hold off;
ha_g0 = axes('Units','normalized','Position',[0.1 0.4 0.6 0.56],'box','on');
axes(ha_g0);
%draw line
    if SOLVE==1,
      temy =(x(1)-(UMAX1/(3.5*UMAX)));
    else,
      temy =(x(1)-(1/3.5));
    end,
X1=[x(1) temy];
Y1=[1.3 1.3];
X2=[x(1) x(1)];
Y2=[1.31 0.98];
X3=[temy temy];
Y3=[1.31 0.98];
plot(X1,Y1,'k');
hold on;
plot(X2,Y2,'k');
plot(X3,Y3,':k');
text(x(1),1.43,'0','fontsize',8);
text((temy-.05),1.43,num2str(UMAX1,2),'fontsize',8);
[xd1 yd1]=gplot(AC,xy);
ha=plot(xd1,yd1);
set(ha,'linewidth',cthick);
hold off;

if SOLVE==1,
  NSTR=TMONSTR;
end,
NSTR
for i=2:NSTR,
  if (SOLVE==1)&(i==NSTR),
    patch(xp(:,i),yp(:,i),[0.26 0.72 0.73]);
  else,
    patch(xp(:,i),yp(:,i),[0 0 0]);
  end,
end,

% draw ground
x(1);
```

```
        i=TMDNSTR;
        -0.1*(xmax/2),
        xp(1,i)= 0.5-0.1*(xmax/2);
        xp(2,i)= 0.5+ 0.1*(xmax/2),
        xp(3,i)= 0.5+0.1*(xmax/2);
        xp(4,i)= 0.5-0.1*(xmax/2);
        yp(1,i)=1+0.3*y(njoint(1)+1);
        yp(2,i)=1+0.3*y(njoint(1)+1);
        yp(3,i)=1+0.55*y(njoint(1)+1);
        yp(4,i)=1+0.55*y(njoint(1)+1);
      end,
      if SOLVE==1,
        njoint=tmdnjoint;
      end,
end,
% set linewidth of column
if NSTR < 15,
    cthick = 2;
  else,
    cthick = 0.75;
  end,

if strcmp(action,'initialize'),

figure('NumberTitle','off','Name','DYNAMIC ANALYSIS FRAME2-D WITH TMD',...
    'colormap',bone(64),'Units','normalized','position',[0 -0.035 1 0.968]),
hc_background = axes('Units','normalized','Position',[0 0 1 1],'box','on');
patch([0; 1; 1; 0],[0; 0; 1; 1],[0; 0; 0; 0],'cdata',[5 5 2 2],'facecolor','interp'),
set(gca,'xtick',[],'ytick',[]),

hc_stop = uicontrol('Style','pushbutton',
        'Units','normalized',
        'ListboxTop',0, ...
        'Position',[0.85 0.388095238095238 0.1196 0.09],
  'fontsize',10,
  'String','STOP', ...
  'Enable','off',...
  'Callback',['handle=get(gcf,"userdata"),',
    'ha_go=handle(4);',...
    'hc_start=handle(1);',...
    'hc_stop=handle(2);',...
    'set(hc_start,"enable","on");',...
    'set(hc_stop,"enable","off");',...
    'set(gca,"userdata",-1);']);

hc_start = uicontrol('Style','pushbutton', ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588], ...
        'ListboxTop',0, ...
        'Position',[0.85 0.59047619047619 0.1196 0.09], ...
  'fontsize',10,...
  'String','START', ...
'Callback',['handle=get(gcf,"userdata");',...
    'hc_start=handle(1);',...
    'hc_stop=handle(2);',...
    'set(hc_start,"enable","off");',...
    'set(hc_stop,"enable","on");',...
    'output("start");']);

hc_close = uicontrol('Style','pushbutton', ...
        'Units','normalized', ...
        'ListboxTop',0, ...
        'Position',[0.85 0.202380952380952 0.1196 0.09], ...
  'fontsize',10,...
  'String','CLOSE', ...
  'callback','close(gcf)');
hc_time= uicontrol(...
        'Units','normalized', ...
        'ListboxTop',0, ...
        'Position',[0.85 0.78 0.1 0.06], ...
```

```
tempg=x(1)-0.75*(x(2)-x(1));
tempg1=abs(x(njoint(1))-x(1)+(x(2)-x(1)))/(0.25*(x(2)-x(1)));
l=(0.25*(x(2)-x(1)));
for i=1:tempg1,
  patch([tempg+i*l,tempg+(i+1)*l,tempg+(i+2)*l],[0 0 -l],[0 0 0]);
end,

set(gca,'XLim',[-0.5 1.5],'YLim',[-0.5 1.5]);
set(gca,'xtick',[],'ytick',[]),
handle =[hc_start hc_stop hc_close ha_g0 hc_string];
set(gcf,'userdata',handle);
elseif strcmp(action,'start');
  axhndl=gca;
  handle =get(gcf,'userdata');
  hc_start=handle(1);
  hc_stop=handle(2);
  hc_close=handle(3);
  ha_g0=handle(4);
  hc_string=handle(5);
  n=length(TIME);
  UMAX=max(abs(U));
  UMAX=max(UMAX,[],2);

    if SOLVE==1,
    TEMPU=U(1:NSTR-2,:);
    NSTR
    NSTR-1
    TEMUMAX=max(abs(TEMPU))
    TEMUMAX=max(TEMUMAX,[],2)
    UMAX1=TEMUMAX
    else,
    UMAX1=UMAX
    end,

  %draw structure

  get(axhndl,'userdata');
  ha_g2 = axes('Units','normalized','Position',[0.1 0.1 0.6 0.26],'box','on');
  axes(ha_g2);
    if SOLVE==1,
    p0=plot(TIME,U(NSTR-2,:));
    else,
    p0=plot(TIME,U(NSTR-1,:));
    end,
  set( gca,'xtick',[]),
  hold on,
  if SOLVE==1,
    p1=plot(TIME(1),U(NSTR-2,1),'Erasemode','xor');
    else,
    p1=plot(TIME(1),U(NSTR-1,1),'Erasemode','xor');
    end,
  set(p1,'linewidth',3.5);
  set(p1,'color',[1 0 0]);
  set( gca,'ytick',[]),
  set( gca,'Xcolor',[1 0.3 0.4],'Ycolor',[1 0.3 0.4]);
  set( gca,'xgrid', 'on' );
  plot(tempnx,tempny,'k');
  set( gca,'xlim',[0 tempn]);
  hold off;

axes(ha_g0);
%draw line
  if SOLVE==1,
    temy =(x(1)-(UMAX1/(3.5*UMAX)))
  else,
    temy =(x(1)-(1/3.5))
  end,
X1=[x(1) temy]
Y1=[1.3 1.3];
```

```
X2=[x(1) x(1)]
Y2=[1.31 0.98];
X3=[temy temy]
Y3=[1.31 0.98],
plot(X1,Y1,'k'),
hold on;
plot(X2,Y2,'k');
plot(X3,Y3,':k');
text(x(1),1.43,'0','fontsize',8);
text((temy-.05),1.43,num2str(UMAX1,2),'fontsize',8);
[xd1 yd1]=gplot(AC,xy);
ha =plot(xd1,yd1,'Erasemode','background');
set(ha,'linewidth',cthick);

% draw floor
for i=2:NSTR,
  if (SOLVE==1)&(i==NSTR),
     h0(i)=patch(xp(:,i),yp(:,i),[0.26 0.72 0.73],'Erasemode','background');
     else,
     h0(i)=patch(xp(:,i),yp(:,i),[0 0 0],'Erasemode','background');
     end,
end,

% draw ground
        x(1),
        tempg=x(1)-0.75*(x(2)-x(1));
        tempg1=abs(x(njoint(1))-x(1)+(x(2)-x(1)))/(0.25*(x(2)-x(1)));
        l=(0.25*(x(2)-x(1)));
        for i=1:tempg1,
patch([tempg+i*l,tempg+(i+1)*l,tempg+(i+2)*l],[0 0 -l],[0 0 0]);
        end,

play = 1;
set(gca,'XLim',[-0.5 1.5],'YLim',[-0.5 1.5] );
set(gca,'xtick',[],'ytick',[]);
set(gca,'Drawmode','Fast');
set(axhndl,'userdata',play);
get(axhndl,'userdata');

while get(axhndl,'userdata')==play,
  get(axhndl,'userdata');
  for count=1:n,
     set(hc_string,'string',num2str(TIME(count)));

     if get(axhndl,'userdata')~=play,
       break,
     end,

% change coordinate x
NU=U/(3.5*UMAX);
temp=0;
XY=xy;
for i=2:NSTR,
      if i==2,
        njoint(i);
        temp=temp+njoint(i);
        for j=temp+1:temp+njoint(i),
          xy(story(j),1);
          NU(i-1,count);
          XY(story(j),1)=xy(story(j),1)+NU(i-1,count);
          xpdy(:,i)=xp(:,i)+NU(i-1,count);
        end,
      else,
        for j=temp+1:temp+njoint(i),
          XY(story(j),1)=xy(story(j),1)+NU(i-1,count);

          xpdy(:,i)=xp(:,i)+NU(i-1,count);
        end,
      end,
      temp=temp+njoint(i);
```

```
    end,        % for i=2:NSTR

    if SOLVE==1,
      set(p1,'xdata',TIME(count),'ydata',U(NSTR-2,count));
    else,
      set(p1,'xdata',TIME(count),'ydata',U(NSTR-1,count));
    end,
   [xd1 yd1]=gplot(AC,XY),
   set(ha,'xdata',xd1,'ydata',yd1);
   for i=2:NSTR,
    set(h0(i),'xdata',xpdy(:,i),'ydata',yp(:,i));
   end,
   drawnow;
  end,          % for count=1:n
  if count==n,
    set(axhndl,'userdata',-1),
    set(hc_start,'enable','on');
    set(hc_stop,'enable','off');
  end,
end,            % while get(ha_g0,'userdata')==play,
end,      %if
```

ประวัติผู้เขียน

นายภาณุพงศ์ ไตรธรรม เกิดเมื่อวันที่ 8 สิงหาคม พ.ศ.2517 ที่ อำเภอบ้านหมี่ จังหวัดลพบุรี สำเร็จการศึกษาระดับปริญญาตรีวิศวกรรมศาสตร์บัณฑิต จากมหาวิทยาลัยศรีปทุม เมื่อปีการศึกษา 2539 และได้เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2540