

Hybrid Connection Simulation Using Dynamic Nodal Numbering Algorithm

¹Alongkorn Lamom, ²Thaksin Thepchatri and ²Wanchai Rivepiboon

¹Faculty of Engineering, Mahasarakham University, Mahasarakham Thailand, 44150

²Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand

Abstract: Problem statement: The traditional dynamic nodal numbering algorithm was proposed for generating code number to support truss and frame simulation using only one element type, frame element; it facilitates the software development process. The traditional algorithm, however, supports simulation of only two types of connection: Hinged connection and rigid connection. In many cases, the problems consist of mix-connection and hybrid connection which cannot be simulated by the traditional technique directly. **Approach:** In this research, an improvement of dynamic nodal numbering algorithm was proposed to support hybrid connection simulation. It required only one frame element to simulate hinged, rigid and also various hybrid connection types. **Results:** The implemented software following the new algorithm was tested with two hybrid structure examples. The analysis results are shown in term of shear force, bending moment and axial force. **Conclusion:** The proposed algorithm does not only improve the ability of connection simulation, but it is also more convenient to implement, debug and expand the finite element software. Hence it reduces the cost, time and effort in developing the computer software.

Key words: Truss analysis, frame analysis, hybrid connection, nodal numbering algorithm

INTRODUCTION

In the structural analysis, information of forces in a structure is goal. To analyze forces induced in a structure, a multitude of mathematical operations are required which tend to increase dramatically in larger structures. These operations consume much computational time and are susceptible to error. As such, algorithms for structural simulation are used to help eliminate these problems.

In 1963, the code number algorithm for forming the structural equations was introduced by Tezcan (Kassimali, 1999). The algorithm was developed based on the finite element method. It has been widely used since then because the approach meets programming needs in that it is systematic and convenient. Although this method is powerful, it, however, separately implements two components, elements, for truss and frame structural simulation abilities. It is inconvenient for implementation.

In 2005, Lamom and Rivepiboon (2005) proposed the dynamic nodal numbering algorithm. Using this technique, it requires only frame element for implementing software to handle truss and frame structural simulation. The technique facilitates the software development process.

Although the dynamic nodal numbering algorithm can simulate many structural scenarios, it does not have the ability to simulate some special structural cases: like some mix and hybrid connections. An example of such a hybrid connection is the structure composed of a scissor hinged joint as shown in Fig. 1.

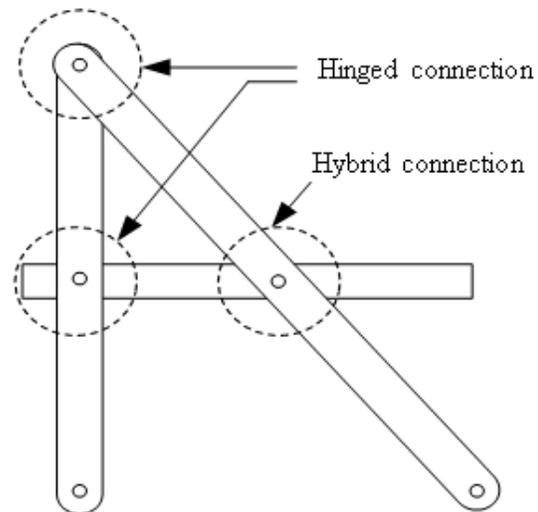


Fig. 1: Hinged and hybrid connections

Corresponding Author: Alongkorn Lamom, Faculty of Engineering, Mahasarakham University, Thailand, 44150
Tel: +66-87-2233355

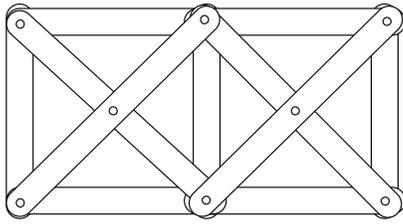


Fig. 2: Structure composed of hybrid connections

In Fig. 1, the structure consists of two types of connection: a hinged connection and a scissor hinged connection. A hinged connection is defined as a connection where the rotational movement of connected members does not depend on each other. A scissor hinged connection refers to a connection, which is built up from two members connected at their centers by a hinge. At the scissor connection, the rotational movement of connected members is dependent on each other.

In the traditional technique, structural simulation is accomplished following the input of the element types. Usually, there are two element types for 2D simulation: The frame element and the truss element. They are provided to simulate two types of connections: Rigid connections and hinged connections. The structures shown in Fig. 1 and 2 consist of a scissor hinged connections which cannot be covered by these two element types. The simulation of such connections is then not possible without some programming modifications. Although there are member release techniques which can be applied to simulate a hinged connection in a frame structure, they cannot be used to represent the scissor connections shown in Fig. 1 and 2 directly.

MATERIALS AND METHODS

To eliminate simulation limitation, a new algorithm for structural simulation is proposed. The functionality of this algorithm is to generate the code numbers which will be used in formulating the structural equations. Using this new algorithm, only one element type, the frame element, is required. The algorithm can easily simulate the hinged, rigid and hybrid connections. The differences between the traditional and the new algorithm are discussed below.

In the traditional technique, structural simulation provides two separate engines to simulate plane truss and plane frame structures (Kassimali, 1999; Schodek, 2004; Kardestuncer, 1974; Johns and Ross, 1998; Reddy, 1993; Hibbeler, 1995; Moaveni, 1999; Sack, 1989; Weaver and Gere, 1990). Figure 3 illustrates calculation flow of this model. The first engine, called

the truss module, is provided to simulate a structure with all connections hinged. The second engine is a frame module which is prepared for simulating a structure with all rigid connections. The truss module is implemented by using the truss element while the frame module is developed by using the frame element.

The first step, when using traditional structural simulation software, is to classify a structure as either a plane truss structure or a plane frame structure. A plane truss structure is a structure of which all members are connected by hinged connections. In a plane frame structure, all members are connected by rigid connections (Lamom and Rivepiboon, 2005).

After classifying the type of the structure, the engine is chosen. Connection simulation which depends on the corresponding engine is then selected. For example, if the frame module is selected, all joints in the structure are automatically analyzed as rigid connections. On the other hand, if the truss module is selected, all joints are automatically analyzed as hinged connections. Although, as previously stated, different types of connection simulation in the same structure are possible with some programming modifications, it is, however, inconvenient to work with a structure that employs hybrid connections.

Although in the member release technique, three new element types are created to simulate various hinged connections in a frame structure, it uses more element types than in new proposed model and it can not completely support simulation of different hybrid connections. As an example, the connections presented in Fig. 1 and 2 cannot be automatically simulated by the traditional technique by the member release technique. To solve this type of problem, a new model and a new algorithm are required.

In order to implement the software which can analyze the truss or frame or hybrid connection structure using only frame element, the new programming model is explained. It is an improvement over the traditional plane frame programming model in that it increases the flexibility of structural simulation. The calculation flow of the new model is shown in Fig. 4. Only one frame element type is used in the computational process. The analyzed structure can be plane truss, plane frame and combined truss and frame members. In addition, connection at each member joint can be easily specified as hinged, rigid or hybrid.

In the traditional model, engine selection affects connection simulation in a structure, while in the new model, it does not. In the new model, the connected member data is added to the system in order to collect information indicating connection format at each member joint. These input data is used to control simulation of the connections.

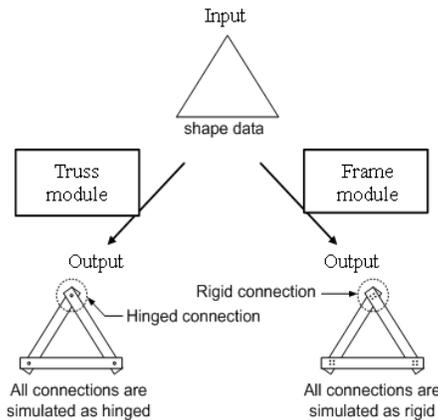


Fig. 3: Calculation flow in the traditional programming model

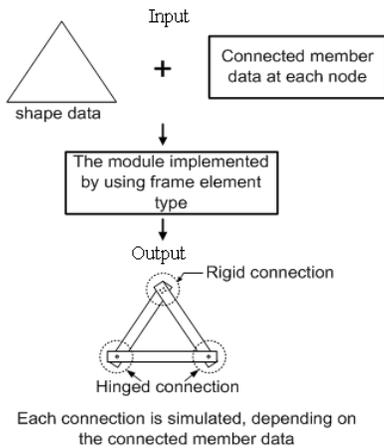


Fig. 4: Calculation flow in the new programming model

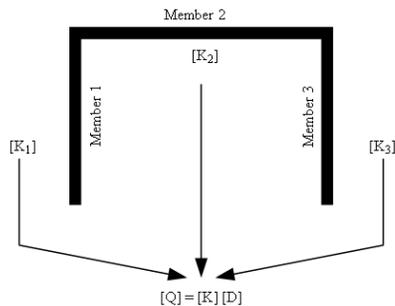


Fig. 5: Equation performing

In the new model of structural simulation, structural shape data and connected member data at each node are input into the module engine, which has the ability to simulate various formats of connection. The engine simulates a structure by forming and

solving the equations according to the data of the structural shape and connected members. However, the new algorithm is based on stiffness method concept for calculation.

Process of numerical calculation using the stiffness matrix method can be divided into two parts. The first part forms while the second part solves the equation.

In the stiffness matrix method, the structural governing equilibrium equation formed in the first part is as follows:

$$[Q] = [K] [D]$$

Where:

[Q] = Structural external force matrix

[K] = Structural stiffness matrix and

[D] = Structural displacement matrix

The above equation represents the force displacement relation of a structure. In forming the structural stiffness matrix [K] (Kassimali, 1999; Johns and Ross, 1998; Reddy, 1993; Hibbeler, 1995; Moaveni, 1999; Sack, 1989; Weaver and Gere, 1990), the structure is divided into finite number of elements called members.

For example, there are three members in the structure shown in Fig. 5. Each member has its own individual stiffness matrix [k], called the member stiffness matrix, which is calculated from the material and geometrical properties of the member. These member stiffness matrices are combined to form the structural stiffness matrix [K] according to code numbers (Kassimali, 1999). These code numbers are generated by the algorithm, called a nodal numbering scheme. The advantages of the new nodal numbering scheme over the traditional one are discussed below.

The new programming model is improved from the conventional plane frame programming model by adding connected member data at each node and redesigning the generating code numbering scheme.

In the traditional nodal numbering scheme, the code numbers of a member are referred from nodes and the member connected at these two nodes. The scheme will generate code numbers at every node in the x, y and z directions. Then the member refers to both nodes at its edge for code numbers.

A code number generated by the new scheme is created in the order of member-node (Fig. 6). Each member consists of two nodes, the beginning node and the ending node. Each node has three degrees of freedom or three code numbers for the displacements in the x, y and z axes. Code numbers are assigned for each member end as shown in the following algorithm:

Define

m = Member number

X_{m-1} = Code number in x-axis of the beginning node of the member m

Y_{m-1} = Code number in y-axis of the beginning node of the member m

Z_{m-1} = Code number in z-axis of the beginning node of the member m

X_{m-2} = Code number in x-axis of the ending node of the member m

Y_{m-2} = Code number in y-axis of the ending node of the member m

Z_{m-2} = Code number in z-axis of the ending node of the member m

1. Assign ordering number to all members.
2. Set code number = 0
3. For all members

If the beginning node already existed then

X_{m-1} = Code number along x-axis from the existing node

Y_{m-1} = Code number along y-axis from the existing node

If the beginning node of member m is connected to the existing node is rigid, then

Z_{m-1} = Code number along z-axis from the existing node

Else

Code number = Code number + 1

Z_{m-1} = Code number

End if

Else

Code number = Code number + 1

X_{m-1} = Code number

Code number = Code number + 1

Y_{m-1} = Code number

Code number = Code number + 1

Z_{m-1} = Code number

End if

End loop

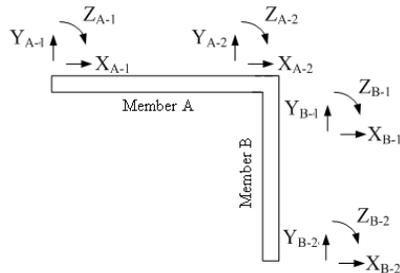


Fig. 6: Dynamic nodal numbering scheme

Figure 7 shows a structure with two members rigidly connected at node 2. Generating code numbers for this structure starts from member A, first at the beginning node 1 and then at the ending node 2 in the direction of x, y and z, respectively. Due to the fact that the beginning node and ending node of member A have not existed before, all code numbers are generated from numbers 1-6. Next, code numbers are generated for member B at the beginning node 2 and ending node 3. Since the beginning node of member B is rigidly connected with the ending node of member A, at which code numbers have already been generated, code numbers in the x, y and z axes of the beginning node of member B are copied from the ending node of member A. At the ending node of member B, code numbers 7 to 9 are assigned because this ending node has not existed before.

An example is illustrated to show how code numbers are generated in a special structure having a non-typical connection. The structure shown in Fig. 9 has 2 members, one is horizontal member and the other one is vertical member. Both members are pin connected at node 2. In computation processes, the horizontal member is divided into 2 members, Member A and B. These two members, of course, are rigidly connected at node 2. For each member, the beginning node and the ending node are assigned in a manner represented by the black arrows in Fig. 8. The head, or tip of the arrows, indicates the ending node of a member while the tail of the arrow refers to the beginning node.

Code numbers start at the beginning node and ending node of member A. These nodes are nodes 1 and 2. Since these two nodes have not existed before, code numbers are generated for the beginning node and ending node of member A (from numbers 1-6) as shown in Fig. 9. Next, code numbers are generated for member B. Because the beginning node 2 is rigidly connected to member A, code numbers for this node are copied from the existing ending node of member A.

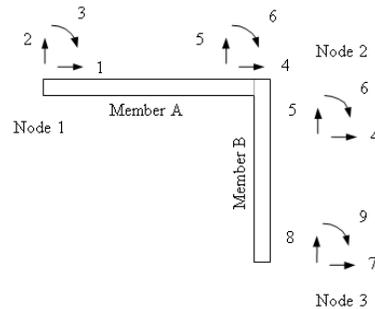


Fig. 7: Code numbers for a plane frame structure

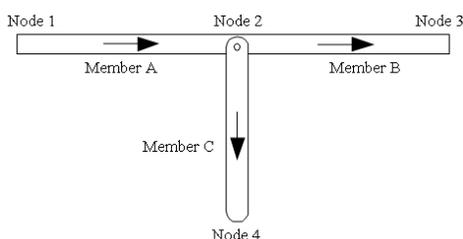


Fig. 8: A special structure

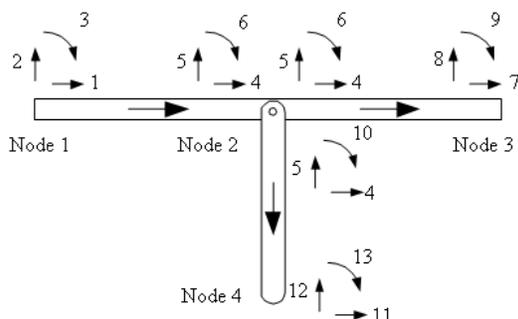
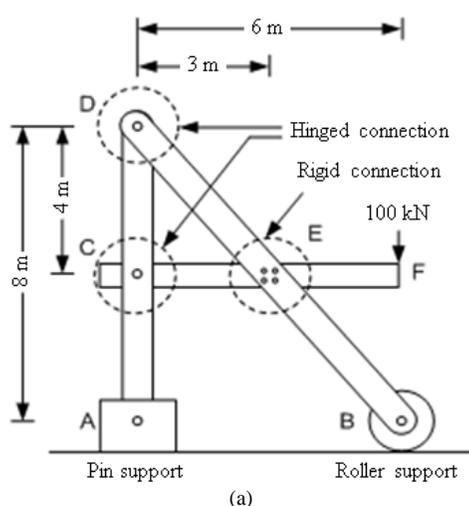


Fig. 9: Code numbers generation of a special structure

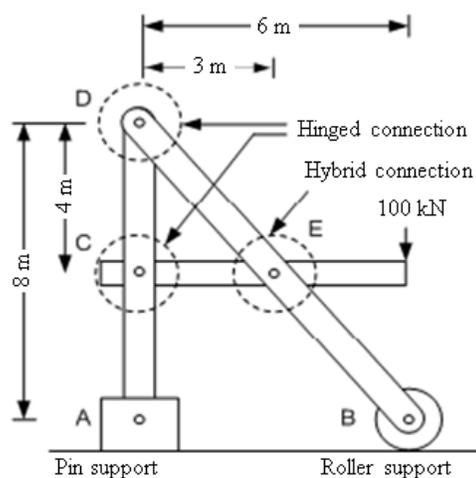
However, new code numbers 7-9 are created for the ending node of member B because this node has not existed before. For member C, the beginning node is pin connected to member A at node 2. Therefore, code numbers 4 and 5, in the x and y axes, are copied from the ending node of member A. In the z axis, due to the independent rotation, a new code number 10 is generated. Next, code numbers 11-13 are assigned at the ending node of member C because this node has not existed before.

RESULTS

To show the analysis efficiency of the new algorithm over the traditional technique, an example demonstrating different connection types is selected. These structures are shown in Fig. 10. In Fig. 10a, joint E is a rigid connection while in Fig. 10b, it is a hybrid one. At the hybrid connection, members joined at this joint will behave like a scissor, i.e., all members will have continuous curvatures at this joint. Both models have same material properties and a load of 100 kN acts vertically at node F. The structural height is 8 and 6 m wide. The problems were solved by SUTStructor (Lamom and Bisarnsin, 2002), a structural analysis software from which the proposed node numbering algorithm has been incorporated. This software has already been extended and used in structural optimize technique research (Lamom *et al.*, 2008). The software is shown in Fig. 11. Since the proposed algorithm automatically generates the code number, it is suitable to use with the software designed as graphic user interface.



(a)



(b)

Fig. 10: Selected example of the structure. (a) Hinged and rigid connection types (b) hinged and hybrid connection types

An example software is graphic user interface. To configure and manage a behavior of connection, user only selects and clicks at node. No command line is needed. The example software can be downloaded for free at <http://www.sut.ac.th/sutstructor>.

Results from the analysis were shown in Fig. 12-14. Shear force diagrams were illustrated in Fig. 12a and b. Bending moment diagrams were shown in Fig. 13a and b. In Fig. 13a, because of the rigid connection at joint E, moments were found only in members EF and BE. In Fig. 13b, on the other hand, extra moments were found in members CE and DE because of the hybrid connection at joint E.

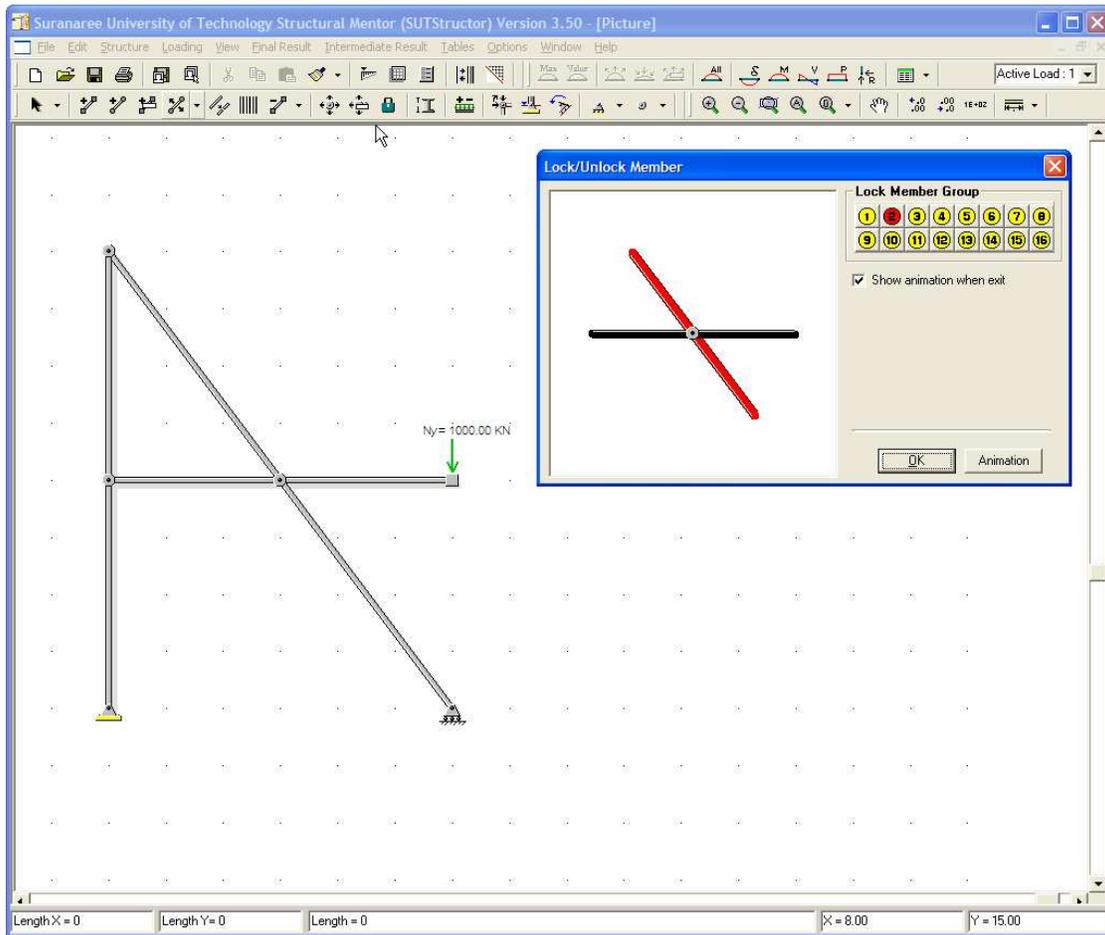


Fig. 11: The computer software implemented based on the proposed algorithm

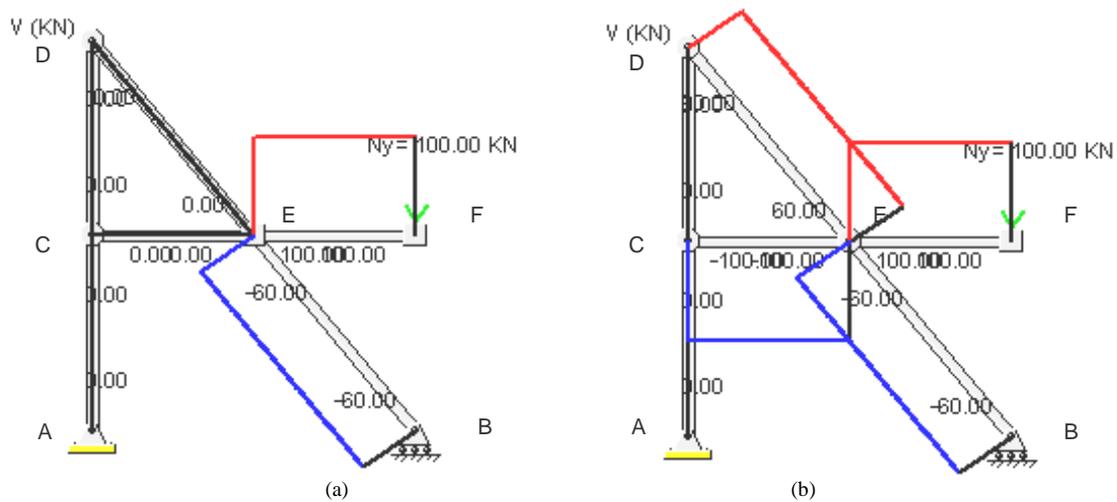


Fig. 12: Shear force diagram of the example from the implemented program. (a) hinged and rigid connection types (b) hinged and hybrid connection types

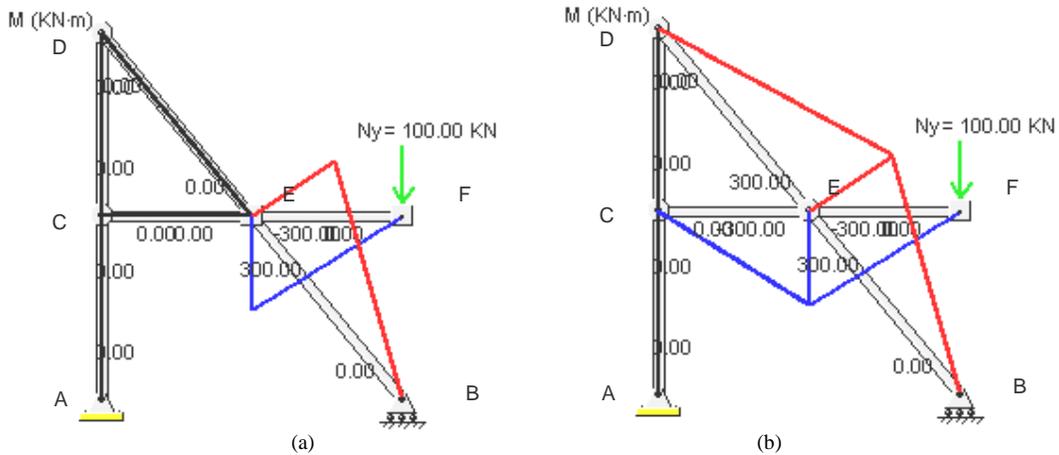


Fig. 13: Moment diagram of the example from the implemented program. (a) Hinged and rigid connection types (b) hinged and hybrid connection types

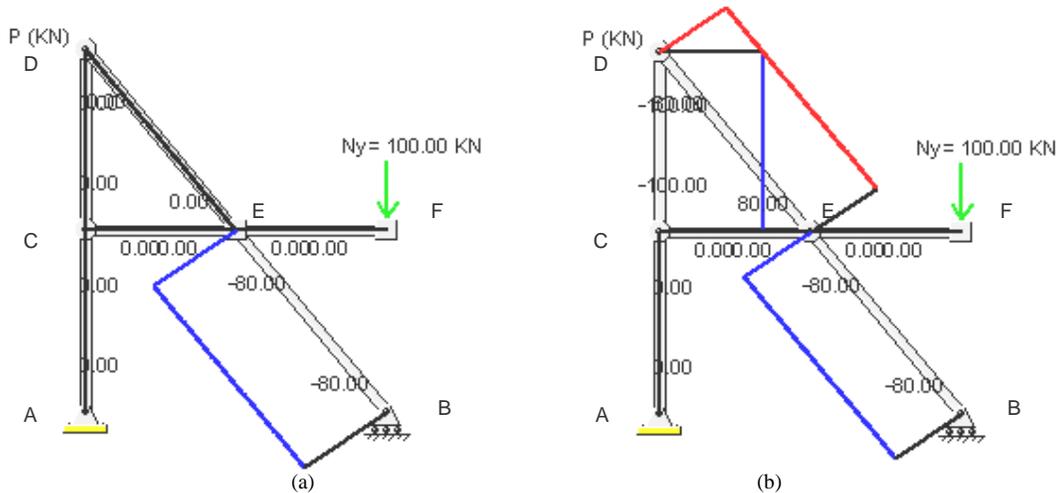


Fig. 14: Axial force diagram of the example from the implemented program. (a) Hinged and rigid connection types (b) hinged and hybrid connection types

An axial compressive force of 80 kN was found only in member BE, Fig. 14a, for the structure with rigid connection at node E. However, in Fig. 14b, rigid with hybrid connection at node E, extra axial forces of -100 kN and +80 kN were found in member CD and DE, respectively.

DISCUSSION

The simulation results show that the proposed technique can handle structures analysis problems consisting of the hybrid connection. It can compute and report answers in term of force diagram such as shear force, axial force and bending moment. Moreover, the

developed software based on this technique also shows that this technique can be actually adopted in the software development. Since the new model requires only one frame module to simulate truss, frame and hybrid connection structure, this technique not only increases simulation ability, but also increases software implementation convenience.

CONCLUSION

The research proposes a new nodal numbering algorithm enabling structural simulation with various formats of connection. This new algorithm is an improvement of the dynamic nodal numbering

algorithm. The proposed algorithm has advantages over the traditional programming model in that it can handle plane truss, plane frame and other structural problems having various formats of connections within one single model. It facilitates the software development process and also increases the structural simulation flexibility.

The algorithm, however, has disadvantage in that its simulation abilities cannot cover all connection types yet. We plan to eliminate this disadvantage in the future.

REFERENCES

- Hibbeler, R.C., 1995. Structural Analysis. 3rd Edn., Prentice-Hall, USA., ISBN: 0023540419, pp: 729.
- Johns, T. and C.T.F Ross, 1998. Computer Analysis of Skeletal Structures. 1st Edn., Spon Press, USA., ISBN: 10: 0419119701, pp: 112.
- Kardestuncer, H., 1974. Elementary Matrix Analysis of Structures. 1st Edn., McGraw-Hill, New York, ISBN: 0070333181, pp: 428.
- Kassimali, A., 1999. Matrix Analysis of Structures. 1st Edn., CL-Engineering, USA., ISBN: 10: 0534206700, pp: 600.
- Lamom, A. and T. Bisarnsin, 2002. Structural analysis software for education. Proceeding of the 8th National Convention on Civil Engineering, (CE'02), IEEE, Thailand, pp: 119-123.
- Lamom, A. and W. Rivepiboon, 2005. A nodal numbering algorithm for 2D structural simulation. ECTI Trans. Comput. Inform. Technol., 1: 108-116.
- Lamom, A., T. Thepchatri and W. Rivepiboon, 2008. Heuristic algorithm in optimal discrete structural designs. Am. J. Applied Sci., 5: 943-951. <http://www.scipub.org/fulltext/ajas/ajas58943-951.pdf>
- Moaveni, S., 1999. Finite Element Analysis: Theory and Application with ANSYS. 2nd Edn., Prentice-Hall, USA., ISBN: 0137850980, pp: 527.
- Reddy, J.N., 1993. Introduction to the Finite Element Method. 2nd Edn., McGraw-Hill Science, New York, ISBN: 10: 0070513554, pp: 896.
- Sack, R.L., 1989. Matrix Structural Analysis. PWS-Kent Pub. Co., USA., ISBN: 0534915647, pp: 329.
- Schodek, D., 2004. Structures. 5th Edn., RR Donnelley and Sons Company, USA., pp: 608.
- Weaver, W. and J.M. Gere, 1990. Matrix Analysis of Framed Structures. 3rd Edn., Kluwer Academic, USA., ISBN: 0412078619, pp: 546.