AIAA JOURNAL
Vol. 33, No. 2, February 1995

Object-Oriented Approach to Modeling and Analysis of Truss Structures

Mitsunori Miki*
Doshisha University, Tanabe, Kyoto 610-03, Japan
and
Yoshisada Murotsu†
University of Osaka Prefecture, Sakai, Osaka 593, Japan

Object-oriented structural and geometric analysis methods for truss structures are proposed. The methods are based on the iterative relaxation of nodal unbalanced forces by moving the nodes. The entities appeared in truss analysis are modeled as classes, and their class hierarchy is established. The knowledge about a truss node, a truss member, a material, and a sectional shape of the member are extracted and constructed as a knowledge base in an object-oriented manner. The object-oriented structural analysis is found to be valid for material and geometric nonlinearities, and the object-oriented geometric analysis where a kinematic problem is converted into a problem for a force equilibrium at the nodes is also found to be effective for determining the geometry of variable geometry trusses. The whole system is written in an object-oriented language, Smalltalk, and several results on structural and geometric analysis are presented. The speed of the computation is discussed along with the size of the structures and the over-relaxation factor. The extension of the knowledge base is also discussed.

Introduction

TRUSS structures have wide applicability to adaptive or intelligent structures since they are able to change their geometry or they suppress their vibrations by using some active and/or passive members.[1–4] The conceptual design of such advanced trusses becomes somewhat complicated since the designers have a lot of options to achieve their goal. Such options can include the use of piezoelectric active members, variable-length telescopic members, plastically displacing members with Coulomb friction, members made of shape memory alloys, members made of viscoelastic damping materials, and so on. In this case, the modeling and analysis of the truss becomes very complicated. A knowledge-base system is one of the solutions to the problem.

Knowledge-base systems work with knowledge bases not with procedures written in the computer programs.[5] The object-oriented approach is a typical method for constructing knowledge bases. Knowledge-base systems yield remarkable benefit in modeling and simulation of engineering systems, but their use in the field of structural modeling and analysis is very rare. The authors believe that knowledge-base systems provide very flexible modeling and analysis capabilities and complete friendliness to the designers during the conceptual design process of structures. Much research on the development of knowledge-based systems in structural engineering field has been done.[6–8] The object-oriented approach also has been introduced in some cases.[9,10] However, those systems are not operated by complete object-oriented manners. In this paper, the object-oriented approach is used completely for all details for the modeling and analysis of truss structures.

The object-oriented analysis of truss structures also provides remarkable simplicity in the structural and geometric analysis of variable geometric trusses with nonlinear response of materials and actuators. The conventional structural analysis of a truss with large deformation or geometric nonlinearity, material nonlinearity, and/or complicated phenomena such as nonlinear thermal expansion and slacks at the joints of truss members requires sophisticated computer codes and, futhermore, such codes are difficult to rewrite to improve the capacity for treating other behaviors.

The objectives of this paper are to propose a new object-oriented method for the structural and geometric analysis of advanced truss structures with many complicated behaviors and to develop a prototype knowledge-base system for designing such trusses. The proposed method is based on a simulation technique, and it gives a unified analysis for solving various complicated nonlinear problems. The scope of this study is confined to static problems in this paper.

Object-Oriented Approach

The object-oriented system[11] is a programming environment in which the fundamental processing paradigm is to send a message to an object rather than using the more traditional approach of calling a procedure to operate on some data. This approach qualitatively enhances the design, creation, and maintenance of a software system. A great deal of this added power derives from modularity.

Every entity in an engineering analysis is considered as an object, each object is an instance of each class, and these classes have a class hierarchy. All of the knowledge is divided into small parts, and they are stored in pertinent classes. Therefore, the primary part of the research involves the development of the class hierarchy and the data structure in each class and specifying the behavior of each class. The class hierarchy and the data structures are static representations of a knowledge, whereas methods by which objects respond to messages are dynamic representations of the knowledge.

The language used in this study is Smalltalk-80,[13] which is a typical object-oriented programming language, and it has the aforementioned characteristics. However, it should be noted that the main object of this research is not to introduce an object-oriented language for analyzing truss structures but to propose a new object-oriented simulation method, which is very suitable for object-oriented programming environments.

Knowledge Representation for Trusses

A truss structure consists of nodes, members, and supports. Each node has data on its coordinates, the relevant support and members, and so on. Each member has data on its material, sectional shape, length, the relevant nodes, and so on. Supports are classified into hinged supports and movable supports, and the movable supprts are further classified into horizontally movable supports and vertically movable supports, and so on. Each material has data on its name, modulus, strength, and so on. Sectional shapes are classified into

Presented as Paper 93-1406 at the 34th AIAA/ASME/ASCE/AHS/ASC Structures,Structural Dynamics, and Materials Conference, La Jolla, CA, April 19–22, 1993; received July 27, 1993; revision received Sept. 9, 1994; accepted for publication Sept. 12, 1994.

*Professor, Department of Knowledge Engineering and Computer Science. Member AIAA.

†Professor, Department of Aerospace Engineering. Member AIAA.

Table 1 Instance variables of the classes for truss analysis

Instance variable name	Content
Truss	
nodes	aTrussNode
members	aTrussmembers
TrussNode	
coordinates	aCoordinates
support	aSupport
members	aTrussMembers
reactionForce	aForce2D
externalForce	aForce2D
nodalForceUnbalanced	aForce2D
TrussMember	
material	aMaterial
sectionalShape	aSectionalShape
originalLength	aNumber
node1	aTrussNode
node2	aTrussNode
memberForce	aForce2D
Active TrussMember	
actuator	anActuator

Note: 'aX' means an instance of class X.

```
Object
    Truss
    Dictionary
        TrussNodes
        TrussMembers
    TrussNode
    TrussMember
        ActiveTrussMember
    Support
        HingedSupport
        MovableSupport
            HorizontallyMobavleSupport
            VerticallyMovableSupport
```

Fig. 1 Class hierarchy for truss analysis, note: italics indicate the existing class in Smalltalk, and the indentation represents the relation between super classes and subclasses.

circular shapes, rectangular shapes, and so on. This static knowledge is represented by a class hierarchy shown in Fig. 1, and, together with instance variables of the classes, shown in Table 1.

The meaning of these data structures is explained as follows.

A truss has collections of nodes and members, and these collections are instances of the class Dictionary which is the existing class in Smalltalk. The instance variable nodes has a collection of an instance of class TrussNode, and the variable members has a collection of an instance of TrussMember. A truss node has instance variables: coordinates, support, members, reactionForce, externalForce, and nodalForceUnbalanced. The variable coordinates holds an instance of class Coordinates which represents the coordinates of the node. The variable support holds an instance of class Support which represents a support condition of the node. The variable members holds the collection of the members connected to the node. The variable reactionForce holds a force vector induced by the support, and externalForce holds a force vector externally applied to the node. The variable nodalForceUnbalanced holds the unbalanced force vector at the node, which is the sum of the forces received from the connected members, the reaction force, and the external force. The principle of the proposed analysis is based on the iterative relaxation of this nodal unbalanced force, which is explained later.

The dynamic knowledge are the methods for these classes. The methods are classified into several categories, such as instance creating, initializing, accessing, calculating, analyzing, displaying, and so on. To build a truss structure, at first its constituent structural elements should be created. In Smalltalk, every class can receive the message new to create its instance. For example, a truss node is created as

TrussNode new (1)

In Smalltalk, the object comes first, followed by the name of the method. The system returns an instance of class TrussNode after evaluating Eq. (1).

The accessing to the instance variables is performed by the accessing methods. For example, to give an instance of Coordinates to a truss node, the following expression is evaluated:

aTrussNode coordinates:aCoordinates (2)

where aTrussNode is an instance of class TrussNode and aCoordinates is an instance of Coordinates. Such an abbreviated form is used in this paper. In this case, the message has the argument aCoordinates which is used as an input data. In Smalltalk, an argument is followed by a colon.

The important methods for the instances of TrussNode and TrussMember are listed as follows:

For a truss node,
- nodalForce
  - Calculate the sum of the forces received from the connected members, the external force, and the reaction force induced by its support.
- forceSensitivity
  - Calculate the sensitivity of nodal force to nodal displacement, which has four elements.
- moveToBalancePoint
  - Move the node to the point where the nodal unbalanced force is expected to vanish.

For a truss member,
- directionOn:aTrussNode
  - Calculate the direction of the member with respect to the given node.
- calculateMemberForce
  - Calculate the axial force in the member.

The last method is composed of three stages: 1) calculate the strain of the member based on the current coordinates of its end nodes, 2) send a message to its material to obtain the stress, and 3) calculate the member force by multiplying the stress and the sectional area of the member. It should be noted that the calculation of the stress is the duty of a material, not the duty of a truss member. A truss member only sends a message to obtain the stress to a material by giving the strain. How to calculate the stress from the strain is a knowledge of a material, and the knowledge of a truss member is independent from the behavior of materials. This is a typical example of the modularity of knowledge in object-oriented systems. The method for calculating the member force is shown in Fig. 2.

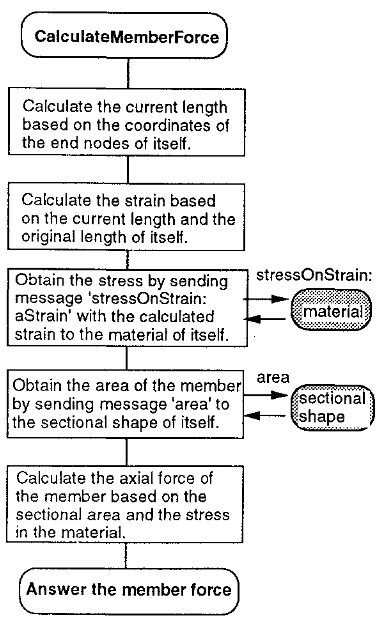

Fig. 2 Method for calculating the member force.

Knowledge Representation for Geometry

Classes Coordinates and SectionalShape are needed for truss analysis. Class Coordinates is the subclass of class Point which is the existing class in Smalltalk and has instance variables x and y, and the instance of Point can receive messages to answer the radius vector and vectorial angle in polar coordinate system.

Class SectionalShape is developed for structual analysis and has the information on the sectional area, moment of inertia with respect to each coordinate, and so on. Class SectionalShape has several subclasses such as Circular-Shape and RectangularShape. Each class has characteristic dimensions.

Knowledge Representation for Materials

Materials are characterized by their names, standards, physical properties, chemical properties, mechanical properties, electrical properties, and so on. The knowledge of the materials can be divided into pertinent classes, and the object-oriented approach provides the maximum flexibility in representing the knowledge.

The important method of materials in truss analysis is to obtain the stress from the given strain, as mentioned before. A perfectly elastic and brittle material has longitudinal modulus of elasticity, and it calculates the stress by multiplying the modulus and the given strain. If the stress is lower than its strength, the material returns the stress, whereas it returns zero stress when the calculated stress is greater than the strength. This is a knowledge for responding to message stressOn-Strain:aStrain, and the material itself has such knowledge.

For materials with nonlinear stress-strain relations, they are to have the knowledge to calculate the stress by using, for example, $\sigma = f(\varepsilon)$, where σ is stress and ε is strain, or a table.

Knowledge Representation for Quantities

In an engineering knowledge-base system, the manipulation of physical quantities such as force, length, mass, stress, and strain is another important aspect. In conventional computer programs, all quantities are treated as numbers and have no data structure. Taking care of wrong calculations, such as a length plus a mass, and the conversion of physical units are left to the computer programmers in this case.

In an object-oriented paradigm, however, every quantity can be modeled and defined as an object having complicated informations and pertinent methods. Miki and Sugiyama[14,15] have proposed a new method of treating physical quantities using an object-oriented approach. By using the method, an instance of class Length, for example, is created as follows:

Length magnitude: 30 ft (3a)

Length magnitude: 3.72 km (3b)

Every physical quantity has its pertinent knowledge, such as length times length yields area or stress divided by modulus yields strain. Furthermore, performing wrong calculations, such as length + mass, yields error messages. This concept provides the maximum friendliness in developing user programs and detects the essential errors in the programs which have not been detected so far.

In this paper, however, many quantities do not have such complicated data structures, and they are treated as instances of class Number which is the existing class in Smalltalk. The reason why quantities are not modeled is the computing time. The object-oriented truss analysis is based on a relaxation method, and it has a relatively long computation time. The introduction of a complicated model of quantity into such an iteration process adds great overhead. The trade-off between exact modeling for objects and computation performance is a subject in object-oriented approaches.

Truss Modeling

A truss structure is produced by the following steps:
1) Create truss nodes.
2) Create the collection of the truss nodes.
3) Create a material or materials.
4) Create a sectional shape or sectional shapes.

```
    fiveMemberTruss
        "Example of truss structure with five members"

01      | aNodes node1 node2 node3 node4
        member1 member2 member3 member4
        member5 aMembers aluminum shape |

02  node1 := TrussNode
        coordinates:(Coordinates x:0 y:0)
        support:Support hinged.
03  node2 := TrussNode
        coordinates:(Coordinates x:0.4 y:0)
        support:Support hinged.
04  node3 := TrussNode
        coordinates:(Coordinates x:0 y:0.3)
05  node4 := TrussNode
        coordinates:(Coordinates x:0.4 y:0.3)
06  aNodes := TrussNodes new
        index:1 node:node1;
        index:2 node:node2;
        index:3 node:node3;
        index:4 node:node4.

07  aluminum:= Aluminum new.
08  shape := HollowCircle innerRadius:9e-3 outerRadius:10e-3.

09  member1 := TrussMember material:aluminum
            sectionalShape:shape
            between:node1 and:node2.
10  member2 := TrussMember material:aluminum
            sectionalShape:shape
            between:node1 and:node3.
11  member3 := TrussMemberWithActuator material:aluminum
            sectionalShape:shape
            between:node1 and:node4.
12  member4 := TrussMember material:aluminum
            sectionalShape:shape
            between:node2 and:node4.
13  member5 := TrussMember material:aluminum
            sectionalShape:shape
            between:node3 and:node4.

14  aMembers := TrussMembers new
        index:1 member:member1;
        index:2 member:member2;
        index:3 member:member3;
        index:4 member:member4;
        index:5 member:member5.

15  ^ Truss new members:aMembers nodes:aNodes
```

Fig. 3 Program for making a five-member truss; note, numbers leftmost of each line not necessary for actual programs.

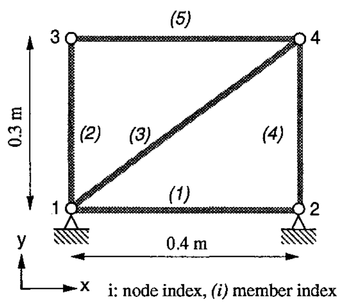

Fig. 4 A five-member truss.

5) Create truss members.
6) Create the collection of the truss members.
7) Create a truss.

Figure 3 shows the user program for making the five-member truss structure shown in Fig. 4. In this case, the material is aluminum, and the truss members are tubes with the inner diameter of 9 mm and the outer diameter of 10 mm. All numeric values are in fundamental SI units.

It can be seen from Fig. 3 that this type of program is very easy to write and to read, and it also provides the document for the job. The documentation of software takes much time and effort, and the introduction of an object-oriented approach is a solution to such a problem.

Object-Oriented Truss Analysis

Structural analysis is generally performed by numerical methods, such as the finite element method. The knowledge-base system proposed here, however, adopts a simulation method in which the nodes of a truss move iteratively to attain the equilibrium state of force. This method is effective for any complicated behaviors, such

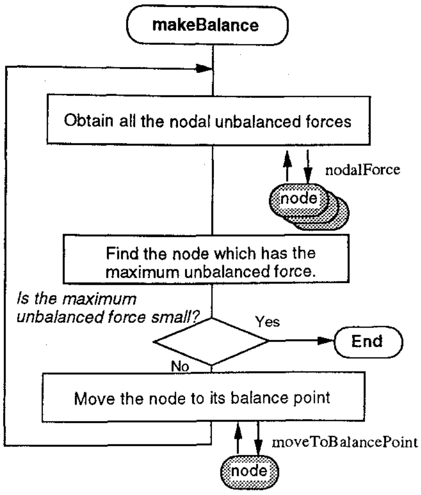

Fig. 5 Flow chart of the object-oriented truss analysis.

as geometric and material nonlinearities, the generation of active forces in active members, and so on, although it takes much time. This simulation method is very suitable for object-oriented systems.

The principle of the object-oriented truss analysis (OOTA) proposed is the iterative relaxation of the unbalanced nodal forces of a truss. When an external force is applied to a node, the unbalanced nodal force occurs, and the node moves such that the unbalanced force vanishes. This process continues until every node is in an equilibrium state, as shown in Fig. 5. The important point is that the relaxation of the nodal unbalanced force is performed at the node which has the maximum nodal unbalanced force.

A truss node has a method named moveToBalancePoint. In this method, a truss node calculates the nodal unbalanced force vector, and it moves to a point where the unbalanced force is expected to vanish. The destination point of this move is calculated from the sensitivity of the nodal force to the nodal displacement. The displacement of the move is obtained as follows:

$$[d] = [s][f] \tag{4}$$

where

$$[d] = \begin{bmatrix} dx \\ dy \end{bmatrix} \qquad [s] = \begin{bmatrix} fx, x & fx, y \\ fy, x & fy, y \end{bmatrix}^{-1}$$

$$[f] = \begin{bmatrix} fx \\ fy \end{bmatrix}$$

where $[d]$ is the displacement of the move, $[s]$ the sensitivity of the nodal displacement to the nodal force, and $[f]$ the nodal unbalanced force.

Sensitivity $[s]$ is obtained from the change in nodal unbalanced force with respect to the small movement of the node. This calculation is performed every time as the node moves, and this makes the analysis effective for material and geometric nonlinearities.

The node moves only once in the method moveToBalance-Point, and sometimes the nodal unbalanced force does not vanish due to several nonlinearities, but the next step is to find the node which has the maximum nodal unbalanced force among all nodes. To obtain the accurate equilibrium state at one node during the relaxation process is meaningless.

This ralaxation process is similar to the Gauss–Seidel iteration for solving linear systems.[16] The Gauss–Seidel method may be expressed as

$$d_j^{k+1} = \frac{1}{a_{ii}}\left(f_i - \sum_{j<1} a_{ij} d_j^{k+1} - \sum_{j>1} a_{ij} d_j^k\right) \tag{5}$$

where $[a]$ is the global stiffness matrix of a truss, κ represents the iteration number, and $(1/a_{ii})$ corresponds to sensitivity $[s]$.

In the Gauss–Seidel method, the relaxation is performed sequentially with respect to the node index, whereas the proposed method makes the relaxation at the node with the maximum nodal unbalanced force which corresponds to the expression in the parenthesis in Eq. (5). Another different point is that Eq. (5) is the expression for one component of the displacement, whereas the proposed method treats the displacement and force vectors.

```
Object
    TrussMember (nodes, members)
        TrussmemberWithActuator (actuator)
    Actuator (trussMember)
        LinearActuator (displacement,
            dispacementLimit, force, forceLimit)
            ForceActuator
            DisplacementActuator
```

Fig. 6 Class hierarchy and the instance variables for active members, note: italics indicate the existing class in Smalltalk.

To increase the speed of convergence, a factor for over relaxation is used. In this case, the displacement of the move of a node is expressed using the over-relaxation factor ω as follows:

$$[d] = \omega[s][f] \tag{6}$$

The over-relaxation factor is said to be in the range of 1.2 and 1.8 on the basis of experimental evidence.[16]

Object-Oriented Geometric Analysis

The determination of the geometry of variable geometry truss structures generally is performed by solving the nonlinear kinematic equations on the nodes. This method has some drawbacks.

1) It should be done apart from the structural analysis. This may introduce some complexity for the problem with the coupling of the axial force of the active member and the induced force of the actuator.

2) It does not have flexibility for remodeling truss structures since the change in the nonlinear kinematic equations becomes complicated.

3) It can be applied only to determinate truss structures. Generally, a variable geometry truss is determinate, but the indeterminate truss with high compliance members could be utilized as a variable geometry truss by introducing active members.

The object-oriented geometric analysis of truss structures proposed here overcomes such problems by converting such a geometric problem into a structural problem where the equilibrium of forces is considered.

The principle of the object-oriented geometric analysis is the generation of axial stress in active members when its actuator produces displacement, and its end nodes do not move. The axial stress produces the nodal force at the nodes, and nodal unbalanced force occurs. The object-oriented structural analysis method mentioned before determines the new positions of the nodes.

Actuators incorporated in truss members generally are linear actuators, and they are classified into force actuators and displacement actuators depending on their actuator mechanisms. A force actuator produces a prescribed axial force, and a displacement actuator produces a prescribed axial displacement within the range of the lower and upper limits of force and displacement. The class hierarchy and the instance variables for actuators are shown in Fig. 6.

The object-oriented geometric analysis of truss structures is capable of analyzing the complicated coupled problem of the action of active members and the axial force of members.

System Description and Results

Effectiveness of Analysis

The proposed system was written in Smalltalk-80 (ParcPlace Systems Objectworks/Smalltalk Release 4.1) which is a typical object-oriented language. The computer used was a typical engineering workstation (Sun SPARCStation IPX).

First, the effectiveness of the object-oriented structural analysis is investigated for the five-member determinate truss shown in Fig. 4. The deformed shape is shown in Fig. 7, and the result shows good agreement with that of conventional analysis. It should be noted that the buckling of a member is not considered here. The extension of the knowledge base to the buckling is explained later.

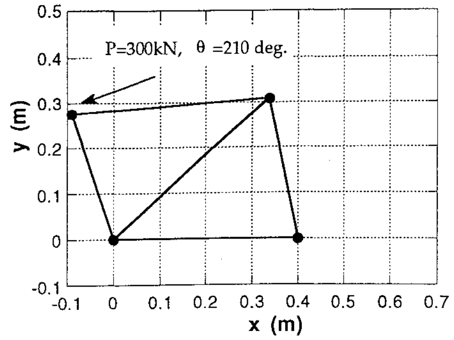

Fig. 7 Result of the structural analysis.

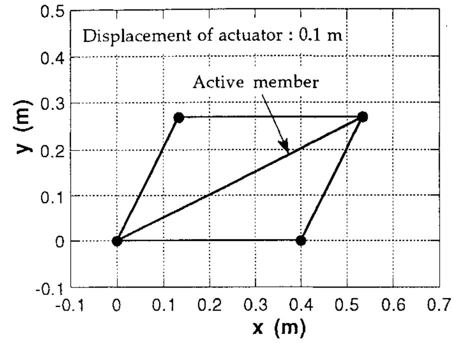

Fig. 8 Result of the geometric analysis.

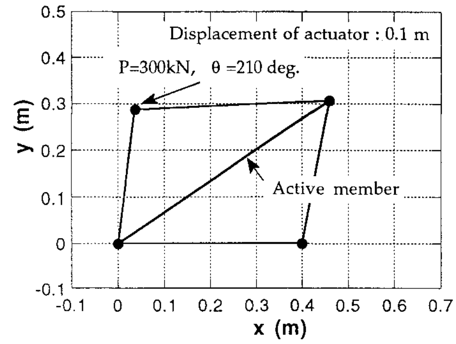

Fig. 9 Simultaneous analysis of geometry and structure.

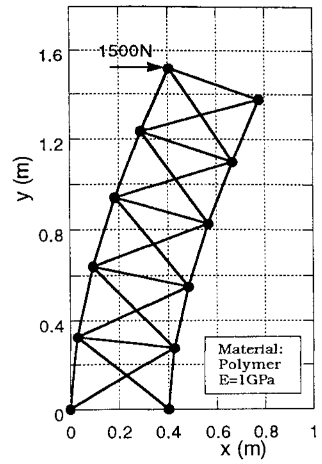

Fig. 10 Result of the structural analysis of a 26-member truss.

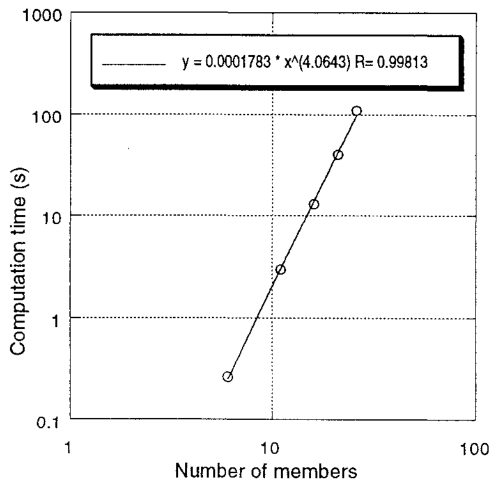

Fig. 11 Computation time as a function of the number of truss nodes or members.

Next, the effectiveness of the object-oriented geometric analysis is investigated for the same truss shown in Fig. 4 except for member 3 which is an active member with a displacement actuator. An actuator is incorporated into member 3, and the actuator produces the displacement of 0.1 m. Figure 8 shows the deformed shape due to the extension of the active member whereas the truss has no external load. The result also shows good agreement with that of a conventional kinematic analysis, and the object-oriented geometric analysis is found to be effective.

Simultaneous Analysis

The object-oriented geometric analysis is based on the object-oriented structural analysis and, therefore, the analysis of structure and geometry can be performed simultaneously. Figure 9 shows the result of simultaneous analysis for the truss shown in Fig. 4 subjected to the same load as in Fig. 7 and to the same increase in the length of member 3 as in Fig. 8.

Computation Time and Speeding Relaxation

The computation time in knowledge-base systems is not short as compared to conventional numerical systems. Furthermore, the proposed method is based on the iterative relaxation of nodal unbalanced forces and, consequently, the analysis takes a relatively long time. To investigate the relation between the computation time and the number of the nodes or members of trusses, the structural analyses of indeterminate trusses with 6, 11, 16, 21, and 26 members are conducted.

The deformed shape is shown in Fig. 10 where the applied force is large so that the deformation can be seen clearly, and it is assumed that the material is perfectly elastic and that buckling does not occur. The relation between the computation time and the number of the members is shown in Fig. 11. It is found from this figure that the results trace a straight line on a log-log paper, and the slope is 4.06; that is, the computation time is proportional to the fourth power of the number of members. Consequently, the proposed method is not effective for large structures, but is effective for small structures with complicated behaviors.

The effectiveness of the over-relaxation factor is investigated for a 16-member truss. The computation time as a function of the over-relaxation factor is shown in Fig. 12, and the appropriate value of the factor is considered to be within the range of 1.6 and 1.8. The factor of 2.0 yields an oscillation of nodes, and the calculation is not converged.

Extension of Knowledge Base

Extension to Nonlinear Materials

The ease of extending the knowledge base is investigated by introducing nonlinear materials. The member forces are calculated as their stresses multiplied by their sectional areas in each truss member. In the proposed analysis, a truss member calculates its deformed length from the nodal coordinates of both ends, and then calculates its strain. The truss member sends message stressOnStrain with the argument of the calculated strain to the material of the member. The material answers the corresponding stress.

If the material is perfectly elastic, it will return just the strain multiplied by its Young's modulus. If the material has nonlinear

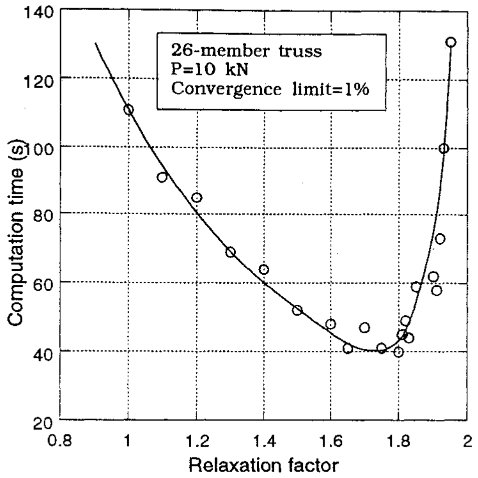

Fig. 12 Computation time as a function of the over-relaxation factor.

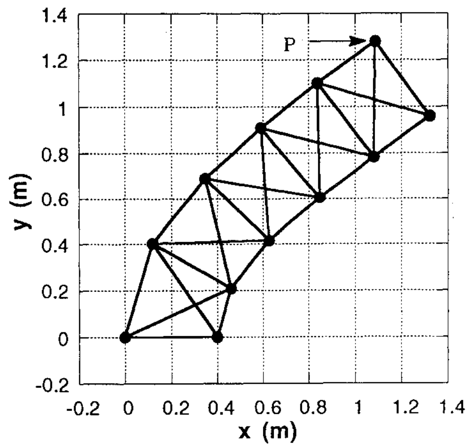

Fig. 13 Result of the structural analysis for the 26-member truss with elastic-plastic materials.

stress-strain behavior, it returns the corresponding stress which may be calculated by a certain strain function. It is important that this procedure is completely confined within the material class, and the extension of a knowledge is performed by only making a new material class which has a nonlinear behavior. It should be noted that it is not necessary for any other knowledge to revise.

The deformation of the 26-member indeterminate truss is analyzed where the materials of the members are all elastic-plastic materials, and the result is shown in Fig. 13 which is a typical example of the problem with material and geometric nonlinearities.

Extension to Buckling and Thermal Expansion

The knowledge of the buckling of a member is not a knowledge of materials, but is a knowledge of the members themselves. In this case, an additional knowledge is constructed, which is a modified method of calculate MemberForce shown in Fig. 2.

The name of the new method becomes calculate MemberForce and the old method becomes calculate MemberForce WithoutBuckling. Figure 14 shows the new method where the member sends the message minimumMomentOfInertia to its sectional shape, and the knowledge for responding to it should be developed in class SectionalShape.

The deformation due to the thermal expansion of the members can be analyzed by adding a small amount of knowledge. The coefficient of thermal expansion is introduced as an instance variable of class Material, and the method for calculating a stress from a strain is modified a little, that is, the stress is calculated from the given mechanical strain minus its thermal strain. The instance variable temperature is incorporated into materials and truss members, then a truss with a temperature distribution can be analyzed.

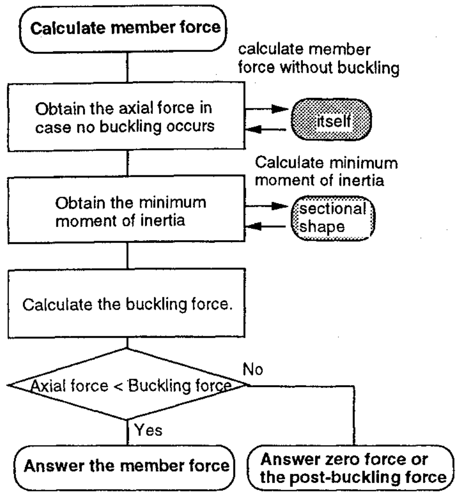

Fig. 14 New method for considering buckling.

Thus, the extension of the knowledge base is found to be very easy and effective. This makes the proposed system useful for considering various complicated phenomena in advanced truss structures.

Concluding Remarks

The object-oriented approach is found to be useful in modeling and analysis of truss structures. The proposed object-oriented structural and geometric analyses are based on the iterative relaxation of nodal unbalanced forces of a truss. The proposed knowledge-base system provides very flexible simulation capability. The knowledge for truss structures and the related objects is established. The designers do not have to take care of fundamental engineering knowledge, and they are able to devote themselves to create new truss structures. The knowledge base is found to be easy to be extended. The system is suitable for conceptual design of advanced truss structures.

The methodology presented in the paper is applicable to various discrete structures that are composed of various elements that have distinct element boundaries. Those elements are modeled as objects, and through those boundaries messages and answers are passed. The limitations are on the number of elements since this approach takes much time. The calculation time is proportional to roughly the number of elements to the fourth power. Therefore, the limitation depends on the computer power. For dynamic problems this method is applicable although a little modification is necessary. Every message should be sent with small time intervals and synchronization. However, it adds a remarkable amount of calculation time to this approach, so the number of elements should be very small in the case of dynamic problems.

The most important utility of this method is the simulation capability for the behavior of discrete structures with very complicated behaviors of their elements. There is no general approach to such complex systems, and even the finite element method requires additional formulation in dealing with such complexities. However, the proposed approach is applicable to almost any complex systems. Furthermore, the proposed method is very suitable for parallel processing and, therefore, it becomes a general approach in dealing with various problems with massively parallel computers.

References

[1]Miura, K., and Furuya, H., "Adaptive Structure Concept for Future Space Applications," *AIAA Journal*, Vol. 26, No. 8, 1988, pp. 995–1002.

[2]Wada, B. K., and Utku, S., "Adaptive Structures for Deployment/Construction of Structures in Space," *Proceedings of the 33rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* (Dallas, TX), AIAA, Washington, DC, 1992, pp. 379–385.

[3]Chen, F. S., and Wada, B. K., "On an Adaptive Truss Manipulator Space Crane Concept," *Proceedings of the First Joint U.S./Japan Conference on Adaptive Structures* (Maui, HI), edited by B. K. Wada, J. L. Fanson, and K. Miura, 1991, pp. 726–742.

[4]Murotsu, Y., Senda, K., and Hisaji, K., "Optimal Configuration Control of an Intelligent Truss Structure," *Proceedings of the First Joint U.S./Japan Conference on Adaptive Structures* (Maui, HI), edited by B. K. Wada, J. L. Fanson, and K. Miura, 1991, pp. 157–175.

[5]Coyne, R. D., Rosenman, M. A., Radford, A. D., Balachandran, M., and Gero, J. S., *Knowledge-Based Design Systems*, Addison-Wesley, New York, 1990.

[6]Paek, Y., and Adeli, H., "STEELEX: A Coupled Expert System for Integrated Design of Steel Structures," *Engineering Applications of Artificial Intelligence*, Vol. 1, Sept. 1988, pp. 170–180.

[7]Grierson, D. E., and Cameron G. E., "A Knowledge-Based Expert System for Computer Automated Structural Design," *Computers and Structures*, Vol. 30, No. 3, 1988, pp. 741–745.

[8]Rogers, J. L., and Sobieski, J., "STRUTEX A Prototype Knowledge-Based System for Initially Configuring a Structure to Support Point Loads in Two Dimensions," NASA CP-3031-PT-1, 1989, pp. 317–331.

[9]Nakai, S., Fukuwa, N., Hirose, K., Katuura, H., and Ebijara, M., "A Knowledge-Based Approach to the Structural Analysis of Large Space Structures," *Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics and Materials Conference*, AIAA, Washington, DC, 1991, pp. 462–469.

[10]Glass, B. J., and Hanagud, S., "Identification of Time-Varying Structural Dynamic Systems: An Artificial Intelligence Approach," *AIAA Journal*, Vol. 30, No. 5, 1992, pp. 1371–1382.

[11]Wagner, P., "Concepts and Paradigm of Object-Oriented Programming," *OOPS MESSENGER*, Vol. 1, No. 1, 1990.

[12]Smith, D. N., *Concepts of Object-Oriented Programming*, McGraw-Hill, New York, 1991.

[13]Goldberg, A., and Robson, D., *Smalltalk-80: The Language and Its Implementation*, Addison-Wesley, Reading, MA, 1983.

[14]Miki, M., and Sugiyama, Y., "ExpertSystem for Composites Design with Object-Oriented Language," *Proceedings of the 5th Japan-U.S. Conference on Composite Materials* (Tokyo), 1990, Japan Society of Composite Materials, Tokyo, pp. 653–660.

[15]Miki, M., and Sugiyamam, Y., "Treatment of Physical Quantities in Structural Engineering Using an Object-Oriented Approach," *Transactions of the Japan Society for Mechanical Engineers*, Vol. 57A, No. 543, 1991, pp. 2830–2835.

[16]Scheid, F., *Theory and Problems of Numerical Analysis*, McGraw-Hill, New York, 1968.