Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte

Band: 40 (1982)

Rubrik: Theme 3: Technical and legal responsibility associated with structural computations: quality assurance procedures

THEME 3

Technical and Legal Responsibility Associated with Structural Computations. Quality Assurance Procedures.

Responsabilité technique et légale associée au calcul des structures par ordinateur. Procédures d'évaluation qualitative.

Technische und rechtliche Verantwortlichkeit in Verbindung mit EDV-Berechnungen. Qualitätsabsicherungs-Verfahren.

Leere Seite
Blank page
Page vide

A Methodology for the Evaluation of Designs for Standards Conformance

Evaluation de projets par rapport aux normes de construction

Methode für die Wahl von Konstruktionen mit Normenübereinstimmung

Steven J. FENVES
Professor
Carnegie-Mellon University
Pittsburgh, PA, USA

SUMMARY

A critical aspect of evaluation of designs is that of evaluating conformance with the governing standards, and other regulatory documents defining acceptable designs. The paper presents a methodology for the formulation and use of standards. The objective of the methodology is to assist developers in formulating clear, complete and unambiguous standards and to provide tools for generating CAD programs.

RESUME

Un aspect essentiel de l'évalution des activités de conception porte sur l'examen de la conformité des résultats à des standards et autres documents normatifs qui définissent les solutions acceptables. L'objectif de la méthodologie présentée est d'assister les concepteurs par une formulation claire et complète de standards et de fournir des éléments utiles à l'établissement de programmes de CAO.

ZUSAMMENFASSUNG

Ein kritischer Gesichtspunkt in der Wahl von Konstruktionen besteht darin, Übereinstimmung mit den öffentlichen Normen und andern Ausführungsdokumenten über annehmbare Konstruktionen zu finden. Dieser Bericht stellte eine Methode zur Formulierung und Anwendung von Normen vor. Das Ziel dieser Methode ist es, Ingenieuren zu helfen, klare, vollständige und eindeutige Regeln zu formulieren und Werkzeuge zur Entwicklung von CAD-Programmen zur Verfügung zu stellen.

1. INTRODUCTION

1.1 Role of Evaluation

To put the paper in proper perspective, a simplified model of the design process is first given. Design of a system, product or artifact in general involves three phases:

- synthesis, where one or more potential solutions are created satisfying a few key design constraints;
- analysis, where the performance of the candidate design(s) is computed and design parameters are selected so that the performance of the candidate design(s) is satisfactory – or even optimal – with respect to a few additional technological constraints; and
- evaluation, where the design judged to perform adequately is further evaluated with respect to all applicable constraints.

The design is considered acceptable if all constraints evaluate to satisfied; if any one constraint evaluates to violated, the design must be revised.

This simple model introduces two key issues.

First, constraints are specified by groups or classes. The generic form of a class of constraints will be called a requirement; the application of that requirement to any particular instance is called a constraint.[3] Thus, the technological requirement in a flow network is:

$$\text{Flow} \leq \text{Capacity}$$

while the constraint on each component i is

$$\text{Flow(i)} \leq \text{Capacity(i)}.$$

An integral part of the design process is to expand the given requirements into specific constraints for each instance of the class they pertain to.

Second, the design phase in which a given requirement is used is entirely up to the designer. The person (or agency) specifying a requirement does not know whether the designer will incorporate the constraints arising out of that requirement as a generative tool in synthesis, as a performance measurement tool in analysis or as a passive checking tool in evaluation. All requirements must therefore be given in a standard form. The most general form is the passive or checking form, that is, a boolean expression evaluating to true or false, which can be interpreted as requirement satisfied or violated, respectively. This form can be used directly for evaluation, or it can be converted by the designer into active forms for use in synthesis or analysis.

1.2 Sources of Evaluation Requirements

Evaluation requirements, and the design constraints which they generate, come from three sources.

Technological requirements arise from the physical principles governing the function of the artifact or system in question, such as conservation of energy, equilibrium of forces, compatibility of displacements, etc. These requirements are the easiest to represent and process, and in a CAD environment are generally incorporated into application programs or procedures.

A second group of requirements are internal to the design process, and represent the owner's objectives and resources (e.g., the requirement "cost ≤ budget") or the designer's intention or style (e.g., "aspect ratio of a beam ≤ 2.0").

A third group of requirements is external to the designer or owner of a project, arising from the standards, codes, design specifications, regulations and other normative documents defining the acceptable performance or required characteristics of a system. Specifically, in an industry as widely dispersed and diversified as the building industry, building standards are viewed as the only "collective memory" of the profession.[4] Increasingly, regulations are introducing similar external constraints into many other design activities.

The remainder of this paper will deal specifically with the external evaluation requirements embodied in standards and codes. However, as the presentation will demonstrate, the methodology is equally applicable to internal requirements.

For the purposes of this paper, the term *standard* encompasses all types of documents used for the evaluation of design and construction, including model and legal codes, consensus standards, and trade association and proprietary specifications.

The lifecycle of a standard begins when the proposed standard is first formulated by groups of knowledgeable people. Upon balloting and resolution it is promulgated by the sponsoring organization (such as ISO, ANSI or ASTM). The adopted standard, in turn, undergoes revisions and updates, either on a fixed schedule or when significant new information or technology becomes available.

1.3 Critique of Present Status.

The present mode of generating, promulgating and using standards suffers from two major deficiencies.

First, there are no recognized formal methods for generating or reviewing the content or the form of proposed new standards or modifications of existing ones. Because standards are so important to industry and because the cost of producing them is high, there is a need for a method, beyond due process, informal peer review, and occasional test comparisons with previous standards for making objective evaluations of the logic and internal consistency of standards.

Second, there are very few tools available for users of standards, that is, the designers responsible for producing designs conforming to the requirements of a standard and the regulatory agencies charged with enforcement of conformance. Both groups of users must exercise considerable effort in interpreting the written expression of a standard to generate their own evaluation procedures. The problem is further compounded in a computer-aided design enviroment, where each organization, starting essentially from scratch, implements its own interpretation of a standard into a program for its own use. Even the slightest change in the standard requires changes, sometimes major ones, in all such programs. Furthermore, such programs frequently incorporate interpretations of the junior members of the organization, because they are the only ones who had learned to program a computer. Neither the designers using these programs nor the persons who have to make judgments on the results generated have any direct way of ascertaining that the programs are based on the correct interpretation of the standard in question.

1.4 Objectives of Methodology

The objective of the methodology to be presented is to improve design practice through better standards and better methods for the use of standards.

For the assistance of standard developers, the methodology applies to two distinct processes:

- *Formulation*, the generation of the information content of the standard; and
- *Expression*, the exposition of the information content in both conventional textual form and in forms adaptable to computer processing of the constraints in the standard.

The methodolody provides some objective measures of two requisite properties of standards:

- *Completeness*, meaning that the standard can be applied to all possible situations within its scope; and
- *Clarity*, meaning that the interpretation of a standard can yield one and only one result when applied in any one situation.

For the use of standards, that is, the interpretation and application of standards in the evaluation of designs in both manual and computer-aided environments, the methodology provides a set of direct and convenient tools, as will be illustrated.

The presentation that follows is a brief summary of concepts developed over a ten-year period, and applied to a number of standards, codes and specifications.[5,6,7,8,9,10,11]

2. A MODEL OF STANDARDS

1.4.1 Provisions

The basic unit of a standard is a provision or normative statement stipulating that a product or process shall have or be assigned some quality. A number of forms and types of provisions fit this definition:

- a performance requirement, e.g., "the system shall maintain an adequate supply of hot water,"
- a performance criterion, e.g., "hot water temperature shall be controlled between 40°C and 50°C,"
- a prescriptive criterion, e.g., "the hot water tank shall have a capacity of 150 liters,"
- a determination or function, e.g., "the flow q = av."

Each provision has the function of assigning a value to a data item or datum. It is useful to recognize two kinds of provisions, distinguished by function:

- *Requirements*, or those provisions that are directly indicative of compliance with some portion of a standard. Such provisions can normally be characterized by boolean data values, with true and false interpreted as satisfied or violated.
- *Determinations*, or all provisions that are not requirements. Such provisions are normally characterized by either numerical or logical values, including boolean, but are not amenable to characterization as satisfied or violated.

1.4.2 Data Items

A data item or datum is a precise identification of an information element occurring in a standard. The status (satisfied or violated) of each requirement is represented by a datum. Each result or variable generated by a determination is a datum. More than one determination may address the same variable, thus the same datum may represent more than one determination. In addition, every other variable referred to in a standard but not explicitly assigned a result by some provision is a datum. For example, the density of a material may be referred to, but not defined, in a standard. Such data are referred to as *basic* or *input* data, and their values are not determined by the standard itself. All data assigned a value by a provision of the standard are termed *derived* data. The list of data is similar to, but much longer than, a conventional list of definitions and symbols found in present standards.

The set of data items plus the systems used to express rules for evaluating and relating them contain all the information necessary to evaluate compliance with a standard.

1.4.3 Decision Tables

A decision table is used to represent the rules for assigning a value to a datum. A decision table is an orderly presentation of the reasoning leading to a decision. It is easily analyzed to assure that the reasoning leads to a unique result in each case and that no possibility exists for encountering an unanticipated situation.

The format and use of decision tables is best illustrated by an example. The following representative requirement is taken from Reference 12:

"1.4.4 Site limitation for Seismic Design Performance Category D – No new building or existing building which is, because of change in use, assigned to Category D shall be sited where there is a potential for an active fault to cause rupture at the ground surface at the building".

Evaluation of this requirement will result in a value of satisfied or violated for the datum "Category D site limitation."

The following data items are used in evaluating the Category D site limitation:

- Seismic performance category (A, B, C, or D),
- Building stage (new or existing),
- Proposed work on existing building (true or false),
- Seismic performance category before proposed work (A, B, C, or D), and
- Potential exists for ground rupture from active fault (true or false).

Data that are used in the evaluation of a given datum are called the *ingredients* of that datum. Likewise, the datum is said to be a *dependent* of each of its ingredients. By itself, the list of ingredients for a datum does not give enough information to evaluate the datum; the decision table is used to collect all the rules for the evaluation of a datum.

TABLE 1 - Decision table for sample provision

	1	2	3	4	E
Conditions					
1. Seismic performance category = D	N	Y	Y	Y	
2. Building stage = new	-	Y	-	N	
3. Proposed work on existing building = change of use and seismic performance before proposed work ≠ D	-	-	Y	N	
4. Potential exists for ground rupture from active fault = true	-	N	N	-	
Actions					
1. Category D site limitation requirement = satisfied	X	X	X	X	
2. Category D site limitation requirement = violated					X

The decision table for the Category D site limitation datum is shown in Table 1. The four parts of the decision table are separated by the broken lines. The *condition stub* in the upper left defines all logical conditions that have a bearing on the outcome, for instance, "1. Seismic performance category = D." The lower left portion of the decision table is the *action stub*, defining all possible actions that can be taken. Here, Action 1 states that the Category D site limitation requirement is satisfied and Action 2 states that it is violated.

The *condition entry* in the upper right-hand portion of the table is divided into a set of *rules*. Each vertical column contains one combination of conditions that defines a rule. For instance, Rule 1, read down the column, applies when Condition 1 is false (N) and the other three conditions are immaterial (...). Rule 2 applies when Condition 1 is true (Y), Condition 2 is true, condition 3 is false (designated by the minus sign; it need not be checked, because it is predetermined to be false by the outcome for Condition 2) and Condition 4 is false. Rule 5, labelled E (for *else*), corresponds to all other combinations of conditions not explicitly included in the preceding rules, such as all conditions being true. The lower right-hand portion of the table, the *action entry*, shows by an X the action appropriate to each rule.

The decision tree generated from the decision table shown in Table 1 is shown in Figure 1. The decision tree provides exactly the same information for Rules 1 through 4 as the decision table, but it also shows two additional combinations of conditions. These additional combinations represent situations included in the else rule of the decision table.

Occasionally a standard contains a single rule for the determination of a value. A decision table for such a datum would contain no conditions. Representation as a single statement, termed a *function*, is adequate.

2.4 Information Network

An *information network* is used to represent the precedence relations among the data in the standard. Each datum corresponds to a node in the network, and the nodes are connected branches that represent the ingredients of each datum. The information network graphically represents the flow of information through the data and thus the decision points in the set of provisions. Figure 2 shows such a network for a small portion of Reference 12. The figure shows that the determination of the required level of seismic analysis depends on the data items: "seismic performance category," "building configuration," "plan configuration," and "vertical configuration," which in turn depend on other data.

The entire information network can be assembled once each datum and its direct ingredients are known. The assembly is easily performed by a computer program.

2.5 Classification System

A *classification system* is used to generate outlines that represent the arrangement and scope of the standard. Requirements and determinations likely to be directly referred to by users are classified according to a model for provisions.

The overall organization of a standard is based on a model structure for provisions and the classification of each provision according to that structure.[13] The model structure of a requirement includes two parts, a *subject* and a *predicate*. The subject may be a physical entity (for instance, a part of a building), a process (for example, design or manufacture), or a participant in the process (for example, a designer, builder, or regulatory agency). The predicate is a particular quality required of a subject (for instance, strength or stiffness of a building part or quality assurance documents from a manufacturer). The list of classifiers pertaining to a particular provision is termed its *argument list* (for example, *design* and *documentation* would be in the argument list for a requirement concerning the submission of engineering calculations).

The classifiers are systematically organized into hierarchies to represent the successively finer subdivisions of the subjects and the required qualities (predicates) falling within the scope of a standard. Figure 3 provides an example of one hierarchy of classifiers; the example includes all the subdivisions of the process of building design, which is one of the subject areas in Reference 12. The provisions coming under a particular classifier are called the *scope list* of that classifier. The scope list can be generated by a computer program that transposes the argument lists for all the provisions.

3. APPLICATION OF METHODOLOGY

As indicated in the Introduction, the methodology is applicable both to the development of new or revised standards and to the use of existing standards. For the former, a distinction is made between formulation, that is, the generation of the information content, and expression, that is, the presentation of that content. These applications are briefly described in the following sections.

3.1 Applications in Formulation

Decision tables representing proposed provisions can be readily checked for <u>completeness</u> (all possible combinations of condition entries are included as rules), <u>lack of ambiguity</u> (no two rules can be matched simultaneously) and <u>redundancy</u> (two or more rules resulting in the same action). Of these, lack of completeness is most typical in early drafts of a provision.

The else rule is a major tool in the analysis of provisions for completeness. Each combination of condition values included in the else rule must be reviewed to see whether a single action, such as Action 2 in the example shown earlier is appropriate, or whether the table is incomplete and needs additional rules to cover the scope of the provision completely.

The information network is useful in the analysis of the formulation of a standard because it clearly shows the impact of each datum on other data. The complete information network can be used to:

- Determine the dependents of each datum,
- Trace the global ingredients of a particular datum (that is, all the data that have any possible influence on the datum in question), and
- Trace the global dependence of a particular datum (that is, all the data that might be influenced by the datum in question).

The information networks can be checked for completeness (absence of detached nodes or subnetworks) and the presence of loops ("circular definitions," where the evaluation of a datum requires the known value of one of its dependents).

In a similar fashion, the classification system can be checked for completeness (all provisions are classified in each of the relevant hierarchies) and for the property of consistency, that is, that uniform technical and logical bases are provided for comparable provisions.

3.2 Applications in Textual Expression

The purpose of expression is to present the information content of a standard in a form convenient for use. For manual use, this means producing a textual form that is clear, consistent and easy to use.

To a limited extent, decision tables can be used to write the text of individual provisions, for example, by writing the text for simple or more common rules before that for the more complex or less frequent rules.

The information network is a major tool for organizing the text of a standard. The global ingredients can be used to order the written expression of a set of provisions. Each branch in the network corresponds to a link or reference that must be represented in the text. Any branch not represented by close juxtaposition of the two data at either end of the branch automatically becomes a cross-reference between the two portions of the standard where the data are located.

Furthermore, two strategies of textual organization are possible. In the top-down strategy, the text is organized by giving the highest-level requirements first, followed in turn by the lower-level requirements down to the determinations and eventually the basic data items; this gives the expert user the option to read only as far as he needs to, skipping those provisions which are familiar or known not to apply. In constrast, a bottom-up strategy defines basic data first, then their dependent determinations, followed by higher-level determinations and eventually the requirements; this provides a "foolproof" step-by-step recipe which would be useful to the novice but would undoubtedly be repetitious and boring for the expert.

Finally, the classification system provides the major tools for the synthesis of the organization of a standard. Outlines can be developed by successively appending trees of classifiers from the hierarchies to produce a tree of headings resembling a table of contents. Different outlines can be obtained by varying the order in which the trees are appended. Several trial outlines can be generated and the one best suited for the intended use of the standard retained. Indexes are generated with classifiers as headings, usually in alphabetical order, and the scope list for each classifier provide a reference to the relevant provisions.

3.3 Applications for Computer-Aided Use

A number of existing or proposed standards and design specifications have been documented in the format described in above, that is:

- a comprehensive list of data items;
- decision tables and functions defining the derivation of individual data items;
- an information network showing the precedence or evaluation sequence of derived data items; and
- a classification scheme identifying key data items.

Formulations in this class include those for the AISC Specification for Steel Design[5], the ACI Concrete Code[14], the Tentative Criteria for LRFD Steel Design[15] and the Tentative Seismic Design Provisions[16]. Unfortunately, these formulations suffer from the fact that they have not been updated to reflect modifications introduced in the original written standards.

The representation of standards in the form of networks of decision tables can be applied to CAD at four levels. At the lowest level of CAD application, the decision table formulations provide a convenient basis for programming segments of standards by conventional manual techniques, e.g., by coding the provisions in a procedural language such as FORTRAN. The primary advantages of using these formulations instead of the original written standard are first, that questions of individual interpretations are largely eliminated and second, that the required program logic - both for individual provisions and for their interrelations - is made much clearer.

At the next level (although to the author's knowledge this has not been done in a production environment) decision table preprocessors could be used directly. These preprocessors accept as input a combination of decision tables and procedural statements and produce as output source code resulting from an optimal conversion of the tables into sequences of IF-statements.[17]

At the third level, efficient processors can be developed for checking conformance with standards provisions. Input consists of the data list, decision tables, functions and the network represented by the ingredience lists of each derived datum. Just as in textual expression, two execution strategies are possible.[6,18] In the top-down strategy, the program attempts to evaluate the topmost requirement specified by the user. If any of the ingredients are as yet undetermined, the program recursively descends and attempts to evaluate the missing ingredient. If a basic data item is needed for the evaluation, it is requested from the user. Eventually, the program backtracks until it terminates by evaluating the topmost requirement. This mode is primarily suitable for selective interactive "spot checking" of completed designs. By contrast, in the bottom-up strategy, the basic data items are entered first and the derived data items are evaluated in sequence, without backtracking, until the topmost requirement is evaluated. This mode is more suitable for routine evaluation of repetitive components in a batch mode.

The fourth level addresses the issue brought out in the Introduction, namely, that at the designer's option selected passive evaluation criteria need to be converted into active assignment procedures for use in synthesis or analysis. Thus, a simplified requirement on stress limitation in a structural element may be stated in a standard as

$$f = P/A \le F$$

where f = actual stress
P = force on element
A = area of element
F = allowable stress.

A designer choosing an element area for a structural element for given P and F can do so subject to $A \ge P/F$. At other stages of design, the designer assigning a capacity to an element given A and F can do so subject to $P \le FA$. In other words, at different stages of design any of the data items appearing in a constraint expression may be designable subject to conformance with the requirement. Methods of symbolic manipulation can be used to convert networks of requirements and determinations into expressions for bounds on designable data item as a function of the remaining data items.[19] The resulting expressions can be evaluated interactively, or they may be compiled into subprograms of CAD systems. It is worth emphasizing that the result is not automated design: the designer must still choose (or program the choice of) an actual value within the bounds allowed by the requirements of the standard.

In closing this section, it is to be reiterated that nothing in the methodology presented or CAD tools described is specifically predicated on external evaluation requirements embodied in standards; internal requirements representing the designer's or owner's "standards" can be cast in the format presented and processed accordingly.

4. STATUS OF WORK

4.1 Aids for Formulation and Expression

The methodology for the analysis of standards was developed and refined over a number of years by working with individuals and committees drafting various standards.[10,20] The main shortcomings experienced were: first, the analysts did not have sufficiently flexible computer-based tools to respond to the rapid pace of drafting and modifications; and second, there was a lack of long-term storage for the data (data item lists, decision tables, networks, classification hierarchies and outlines) between successive versions of a standard.

As a result of this experience, the National Bureau of Standards (NBS) has commissioned a major software system, Standards Processing Software (SPS) which provides a convenient user interface to enter, modify and display data, analysis capabilities (generation of decision trees, information networks, outlines and indexes), and a database for flexible storage and access.[21] NBS intends to provide training sessions and tutorial material for the use of the SPS system, and will make access to the system available to specification writing bodies.

4.2 Aids For CAD Use

Prototype programs have been developed for the top-down and bottom-up execution of networks of decision tables[18] and for the symbolic reformulation of passive checking requirements into expressions for the bounds on designable data items.[19]

Both sets of programs accept a "high-level" description of the applicable standard, namely a network of decision tables. Thus, when the governing standard is updated or modified, only the resulting new decision tables are needed to re-generate the programs.

Both sets of programs are limited by the fact that, in the terminology of the introductory section they deal with requirements, not constraints. That is, they deal with generic data items such as "the force P," rather than specific instances, such as "the force P(i,j,k) on segment i of element j in loading condition k."

Work is in progress to develop general techniques whereby requirements can be "mapped" into constraints applied to instances of data residing in a database.[22] The major consideration is that such techniques be largely independent of the actual organization of the database. Modern database management tools, particularly the relational database model, can provide a large measure of this independence.

5. CONCLUSIONS

Standards, codes and design specifications embody hundreds, if not thousands of evaluation criteria which govern the acceptability of systems, artifacts and products, particularly in the building industry where codes have the force of law, intending to safeguard public health, safety and welfare. Furthermore, designers may choose key criteria for a priori generation, rather than a posteriori evaluation, of candidate designs.

Standards and codes embody much of the "collective memory" of what has worked in the past; every major structural failure precipitates a search for code provisions which need to be added or modified to avoid similar failures in the future. Yet, designers overwhelmingly view standards as an imposition or impediment, frequently because of their awkward format and difficulty of interpretation, rather than their intent or content.

In this paper, a formal representation of standards and a methodology for the for the use of that representation has been presented. The methodology has two distinct applications:

- in the development of new or modified standards, it can assist in the formulation, by checking proposed standards for completeness and clarity, and in the expression of the content.
- in the use of existing standards, it can assist in the generation of CAD programs incorporating evaluation and design procedures based on the standards.

ACKNOWLEDGEMENTS

Portions of this work were supported by the National Science Foundation and the National Bureau of Standards. The contributions of the author's colleagues R.N. Wright and J.R. Harris and of numerous students is gratefully acknowledged.

REFERENCES:

3. Wright, R.N., Boyer, L.T., and Melin, J.W., "Constraint Processing in Design," Journal of the Structural Division, Vol.97, No.ST1, (New York: American Society of Civil Engineers, 1971), pp.481-494.

4. Siess, C.P., "Research, Building Codes, and Engineering Practice," ACI Journal, Vol.56, No.11, (Detroit American Concrete Institute, 1960), pp.1105-1122.

5. Fenves, S.J., Gaylord, E.H., Jr., and Goel, S.K., "Decision Table Formulation of the 1969 American Institute of Steel Construction Specification," Civil Engineering Studies, No.SRS 347, (Urbana: University of Illinois, 1969).

6. Fenves, S.J., "Representation of the Computer-Aided Design Process by a Network of Decision Tables," Computer and Structures, Vol.3, No.5, (New York: Pergamon Press, 1973), pp.1099-1107.

7. Nyman, D.J., and Fenves, S.J., "An Organization Model for Design Specifications," Journal of the Structural Division, Vol.101, No.ST4, (New York: American Society of Civil Engineers, 1975), pp.697-716.

8. Fenves, S.J., K. Rankin, and H. Tejuja, The Structure of Building Specifications, Building Science Series 90, (Washington: National Bureau of Standards, 1976).

9. Fenves, S.J., and R.N. Wright, The Representation and Use of Design Specifications, Technical Note 940, (Washington: National Bureau of Standards, 1977).

10. Fenves, S.J., "Recent Developments in the Methodology for the Formulation and Organization of Design Specifications," Int. J. of Engineering Structures, Vol.1, No.5, London: IPC Science and Technology Press, pp.223-229 (1979).

11. Harris, J.R., S.J. Fenves and R.N. Wright, "New Tools for Standard Writers," Standardization News, Vol.8, No.7, (Philadelphia: American Society for Testing and Materials, 1980), pp.10-17.

12. Applied Technology Council, Tentative Provisions for the Development of Seismic Regulations for Buildings, Special Publication 510, National Bureau of Standards, Washington, DC, June 1978.

13. Harris, J.R., "Organization of Design Standards," Unpublished Ph.D. Thesis (Urbana: University of Illinois, 1980). To be republished as NBS Technical Note, 1982.

14. Noland, J.L., and Feng, C.C., "American Concrete Institute Building Code in Decision Logic Table Format," Journal of the Structural Division, Vol.101, No.ST4, (New York: American Society of Civil Engineers, 1975), pp.677-696.

15. Nyman, D.J., J.D. Mozer and S.J. Fenves, "Decision Table Formulation of the Load and Resistance Factor Design Criteria," Report R-77-6, Department of Civil Engineering, (Pittsburgh: Carnegie-Mellon University, 1977).

16. Harris, J.R., S.J. Fenves and R.N. Wright, Analysis of Tentative Seismic Design Provisons for Buildings, NBS Technical Note 1100, (Washington: National Bureau of Standards, 1979).

17. Pollock, S.L., Decision Tables: Theory and Practice, (New York: Wiley, 1971).

18. Stirk, J.A., "Two Software Aids for Design Specifications Use," Unpublished M.S. Thesis, (Pittsburgh: Carnegie-Mellon University, 1981).

19. Holtz, N.M. and S.J. Fenves, "Using Design Specifications for Design," Computing in Civil Engineering, (New York: American Society of Civil Engineers, 1980, pp.92-101.

20. Harris, J.R., S.J. Fenves and R.N. Wright, "Logical Analysis of Tentative Seismic Provisions," J. of the Structural Division, Vol.107, No.ST8, (New York: American Society of Civil Engineers, 1981), pp.1629-1641.

21. Fenves, S.J., "Software for Analysis of Standards," Computing in Civil Engineering, (New York: American Society of Civil Engineers, 1980), pp.81-92.

22. Rasdorf, W.J., "Structure and Integrity of a Structural Engineering Design Database," Design Research Center Report, DRC-02-14-82, (Pittsburgh: Carnegie-Mellon University, 1982).

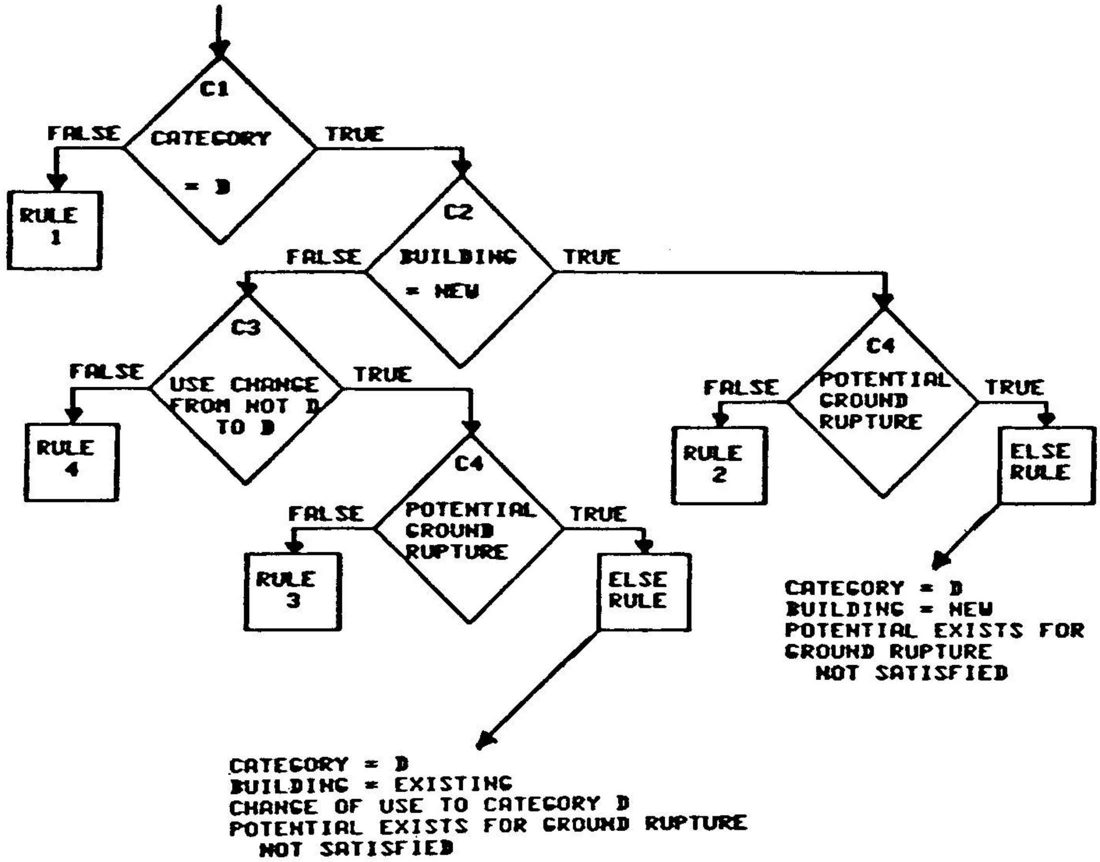


Figure 1. Decision Tree.
Each path represents one column of Table 1.

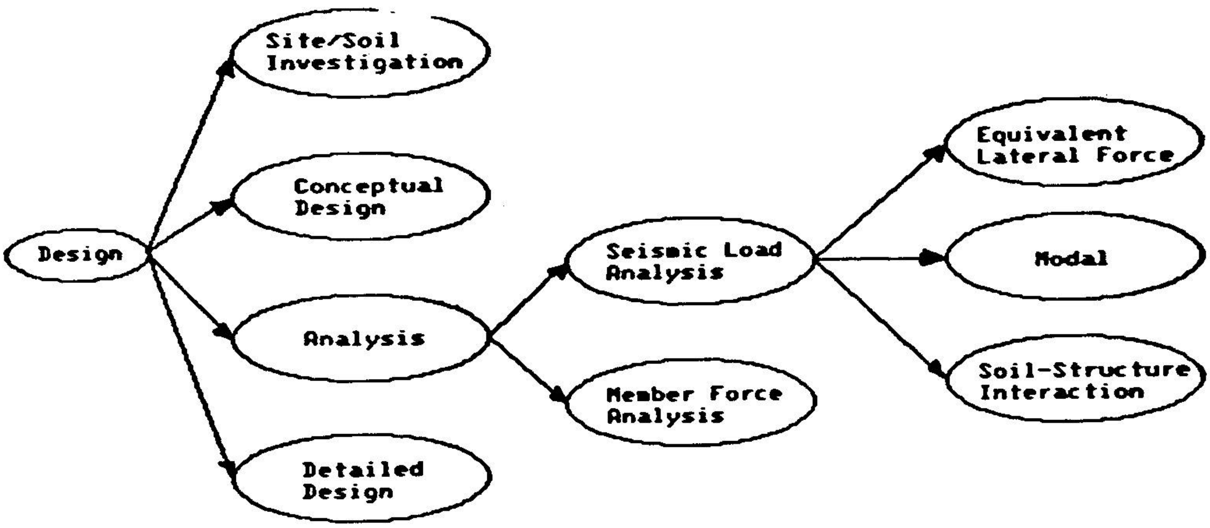


Figure 3. Classification Hierarchy for the Process of Design.

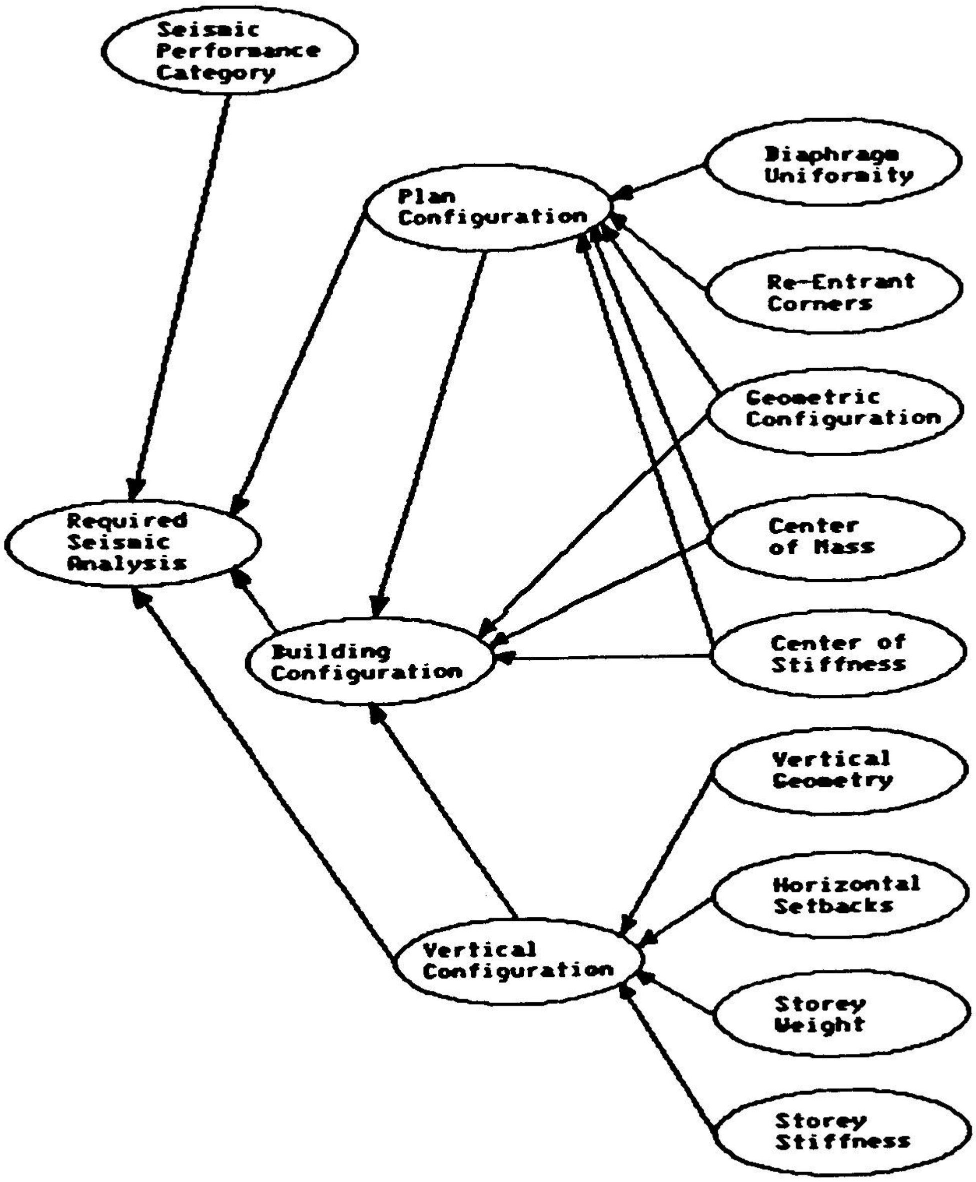


Figure 2. Information Network.
Branches are directed from the ingredient datum to the dependent datum.

Validation of Computations: A Synopsis of Criteria

Critères de vérification de calculs par ordinateur

Kriterien zur Überprüfung von Computerberechnungen

Dieter D. PFAFFINGER
Dr. sc. techn.
Fides Trust Company
Zurich, Switzerland

Dieter D. Pfaffinger obtained his civil engineering and his doctorate degree at the Swiss Federal Insitute of Technology in Zurich (ETHZ). He Has been working with computers in structural engineering since 1964. As a senior consultant, he has been involved in numerous structural projects and in software developments. He is also a lecturer at the ETHZ.

SUMMARY
The most important checks and considerations to validate computations from an engineer's point of view are discussed. A set of general criteria is given which apply to every structural analysis. In addition special checks for dynamic problems are presented. The outlined criteria are illustrated by examples.

RESUME
Les tests et les considérations les plus importants permettant la vérification, du point de vue de l'ingénieur, de calculs effectués par ordinateur sont présentés. Des critères applicables pour tous les types de calcul de structures sont énumérés. Pour les problèmes dynamiques, des tests spécifiques sont proposés. Ces critères sont illustrés par des exemples.

ZUSAMMENFASSUNG
Die wichtigsten Tests und Überlegungen zur Überprüfung einer Computer-Rechnung vom Standpunkt des Ingenieurs aus werden diskutiert. Ein Satz allgemeiner Kriterien wird angegeben, welche für jede Tragwerksberechnung Gültigkeit haben. Für dynamische Probleme werden spezielle Tests zusammengestellt. Die Ausführungen werden durch Beispiele illustriert.

1. INTRODUCTION

Computerized structural analysis has developed rapidly over the past two decades. Powerful computers and a large variety of software packages permit the efficient solution of many static, dynamic and field problems. The scope and complexity of the problems which can be solved as well as the accuracy which can be achieved have steadily increased over the years. New facilities such as computer graphics, CAD/CAE and still more automated, stable and efficient numerical techniques have made the use of computers in structural analysis very attractive.

Today, computerized analysis is no longer a domain of highly specialized engineers. More and more structural analysts with little knowledge of the underlying numerical methods are taking advantage of the existing facilities. All computations, however, have to be verified before the results are further used. It ist therefore indispensable that the analyst is familiar with the validation criteria, a synopsis of which is given in this paper. Furthermore, it is required that the software in use supports this validation by furnishing the appropriate information and also automatically performs certain checks as far as possible and feasible.

2. EQUATIONS OF MOTION

The analysis of a structure is done on an analysis model which contains simplifications and idealisations, but has to reflect the essential physical behaviour of the structure. Today, analysis models usually are built up from finite elements. As sketched in Fig. 1, on each node acts a resulting internal force $\vec{F}$, an inertia force $\vec{T}$, a damping force $\vec{D}$ and an external force $\vec{P}$, which have to be in equilibrium. Representing the forces of all nodes by the vectors $\{F\}$, $\{T\}$, $\{D\}$ and $\{P\}$, respectively, the equilibrium equation

$$\{T\} + \{D\} + \{F\} + \{P\} = \{0\} \quad (1)$$

must hold. In an actual numerical analysis equ. (1) will only be satisfied within a certain accuracy. Introducing

$$\{P\} = \{Q\} + \{R\} \quad (2)$$

where $\{Q\}$ denotes the external loads and $\{R\}$ the reactions, equ. (1) appears in the form

$$\{T\} + \{D\} + \{F\} + \{Q\} + \{R\} = \{\varepsilon\} \quad (3)$$

with the residual nodal forces $\{\varepsilon\}$. If the problem is formulated in constrained nodal displacements $\{q\}$, $\{R\}$ will disappear in equ. (3).

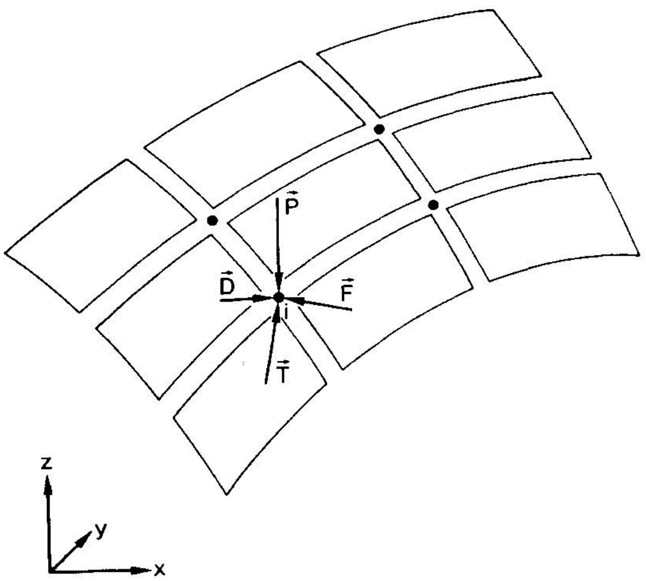


Fig. 1 FE-Model

In a linear analysis, the displacements of the structure depend on the total loads only and not on the loading history. Thus the problem can be formulated in total displacements. In constrained displacements $\{q\}$ the equilibrium equation becomes

$$[M]\ \{\ddot{q}\} + [C]\ \{\dot{q}\} + [K]\ \{q\} = \{Q\} \qquad (4)$$

with the mass matrix $[M]$, the viscous damping matrix $[C]$and the stiffness matrix $[K]$. In addition, two initial conditions exist. Deleting the inertia and the damping term, the basic equation for static analysis is obtained. Dropping only the inertia term, the governing equation of a number of field problems including heat transfer analysis results.

In the nonlinear case, it is advisable to formulate the equations of motion in an incremental form. Usually the internal forces are thereby obtained from a linearised stiffness matrix. Equ. (3), however, must hold for the nonlinearized expressions. Thus a solution obtained from linearized equations eventually has to be improved iteratively until $\{\varepsilon\}$ in equ. (3) is sufficiently small.

The complete time-dependent solution of the equations of motion is obtained by discretisation in the time domain. As shown in Fig. 2, $\{q(t)\}$ is replaced by displacements $\{q_i\}$ at discrete times t_i and a polynomial interpolation in-between. The basic form of the equation for the integration is

$$[A]\ \{\Delta q\} = \{\bar{Q}\} \qquad (5)$$

The integration matrix $[A]$ depends on the mass, damping and stiffness matrix and on the integration time step Δt. $[\bar{Q}]$ is a known effective load vector which also contains $\{\dot{q}\}$ and $\{\ddot{q}\}$ at the beginning of the time step. Equ. (5) permits the step by step integration of the equations of motion in the linear and nonlinear case.

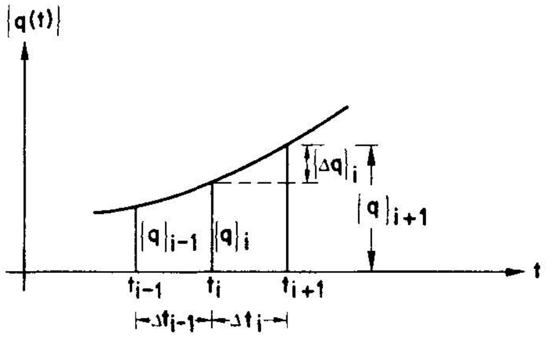


Fig. 2 Discretisation in the time domain

3. VALIDATION OF THE ANALYSIS MODEL

In a first step the analysis model has to be designed conceptually before it is defined numerically. This work is of central importance for the reliability of the analysis. All essential physical properties of the real structure must be reflected in the model. Considerations have to be made on the type of the analysis, the discretisation including the properties of the elements used, the required accuracy, the available computing facilities, the numerical methods to be used and last but not least on the time schedule and costs. In this phase primarily engineering knowledge and experience is required, supported in special cases by preliminary numerical investigations.

Once the model is defined conceptually, it will be described numerically as input data for a particular computer program. In Table 1 the most important checks to validate the model are listed. The geometry and topology of the discretisation are verfied graphically. Modern interactive graphic mesh generators permit the definition and validation in one step. The numerical values of the cross sectional properties and of the material properties have to be checked. For a displacement model only geometric constraints have to be considered. In the general case they appear as linear constraint equations

$$\sum_k a_{ik} - d_i = o \qquad (6)$$

where the a_{ik} denote constraint coefficients, d_i is a fixed displacement and the

□ Geometry and topology
□ Cross sectional properties
□ Material properties
□ Constraints
□ Applied loads
□ Number of modes
□ Time or load steps
□ Convergence criteria
□ Simple static load case

Table 1 Validation of the analysis model

q_k are degress of freedom. Linear constraint equations serve for instance to model rigid parts of the structure or to represent generalized tying conditions. The simple constraint $q_i = o$ is a special case of equ. (6). Each linear constraint equation leads to a reaction which is distributed to the degrees of freedom according to the constraint coefficients. These coefficients as well as the position of the constrained degrees of freedom have to be checked. Finally the applied loads have to be validated with respect to magnitude and position.

In a modal dynamic analysis the number of modes has to be chosen according to the frequency content and the participation factors of the loads. In dynamic or nonlinear analyses time steps or load steps have to be selected. The integration time step Δt is critical for the accuracy of the solution. As a rule of thumb, in an unconditionally stable algorithm Δt should satisfy the condition

$$\Delta t \leq \frac{1}{20\ f_{max}} \tag{7}$$

where f_{max} [Hz] denotes the highest frequency of interest. This leads to approximately 2 % numerical damping in f_{max}. It also should be noted that stability limits of unconditionally stable algorithms usually have been derived for the linear case and may have to be modified for nonlinear analyses. Finally the convergence criteria for iterative solution techniques (eigenvalue extraction, nonlinear analysis) have to be validated or adapted to the problem.

It is always a good idea first to subject a complex model to a simple load case such as gravitational loading and to run a linear static analysis. The inspection of the results frequently leads to the uncovering of hidden errors in the model and thus can save the analyst from useless major computations.

4. VALIDATION OF RESULTS: GENERAL CRITERIA

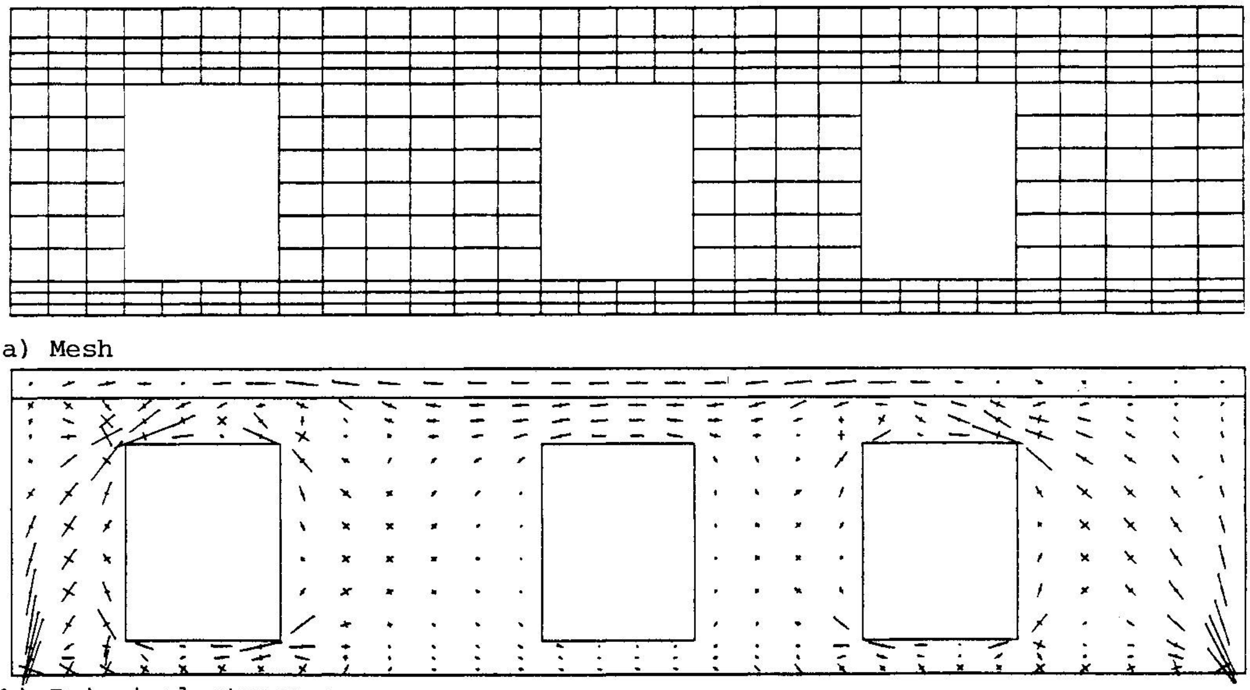
Table 2 shows the most important general criteria to validate results. These criteria are basically applicable to linear and nonlinear static and dynamic analyses as well as to field problems. First of all, global and local equilibrium has to be satisfied. Thus the residual forces according to equ. (3) have to be small for the solution without linearisation. In the case of direct integration, the equilibrium equations may look different depending on the integration

- Equilibrium
- Geometric constraints
- Static constraints
- Stability
- Energy
- Convergence criteria
- Plausibility checks
- Experimental results

Table 2 General criteria

algorithm used. Equilibrium also means, that the momentum and moment of momentum theorems etc. are satisfied. It would thus be useful to obtain the resultant vectors of momentum, moment of momentum, damping forces and external forces at selected times from the program. Violation of equilibrium can indicate a program error, a not converged solution, to few modes or an ill-conditioned system matrix.

Geometric constraints are verified by inspecting the reactions and the deformed shape of the structure. It is necessary that the program also calculates the constraint forces of linear constraint equations. Violation of geometric constraints usually stems from input errors. Static constraints, on the other hand, are automatically satisfied in displacement models in the sense of the under-



a) Mesh

b) Principal stresses

Fig. 3 Wall with openings

lying energy expressions. Thus the quality of satisfaction of prescribed stress conditions is an indication of the quality of the mesh near the respective boundaries. Fig. 3 shows as an illustration the mesh and the principal stresses at the centroids of the elements of a supporting wall with three rectangular openings. It is seen that the trajectories reflect well the static boundary conditions along the stress free edges.

Global instabilities will occur when the determinant of the stiffness matrix (static analysis) or of the integration matrix in equ. (5) (direct integration) becomes very small or changes sign. The determinant is easily obtained as product of the diagonal terms of the triangular factor. An unstable solution can indicate a real, physical instability of the structure or may be caused by numerical reasons. Examples of numerical instabilities are static analyses with high differences in the stiffness coefficients or dynamic analyses with an only conditionally stable integration algorithm. Thus care must be taken to identify the causes of an instability.

In a static analysis, the strain energy of the elements is always greater than or equal to zero. A negative strain energy usually stems from erroneous material coefficients. The strain energy per element should be a slowly varying function. This requirement leads to criteria for the mesh quality. In a dynamic analysis, the kinetic energy, the dissipation energy and the work of the external forces are useful quantities to validate the results.

Iterative solution procedures such as eigenvalue extraction or a number of nonlinear techniques are controlled by convergence parameters. The satisfaction of the convergence criteria has to be checked in the solution.

Every analysis should be validated by plausibility checks. In simple cases, global checks suffice. It is important that the analyst is familiar with the appropriate methods such as for instance the Rayleigh quotient for eigenvalues or the limit theorems of plasticity for the determination of collaps loads. In more complex situations, a detailed counter analysis using different methods and/or a different model can clarify questions about a solution.

Finally, the comparison of numerical results with experiments may give further evidence of the validity of a solution. In the machine building industry, tests are frequently possible on prototypes before production starts, whereas in civil engineering the tests usually can be performed only after the completion of the building. Such a posteriori tests, however, are still very useful to calibrate the analysis methods. In all comparisons between numerical and experimental results, the accuracy of the experiment has to be included in the considerations.

5. VALIDATION OF MODAL ANALYSES

There exist a number of additional criteria for the modal analysis of linear dynamic problems which are listed in Table 3. First, the eigensystem of the undamped structure has to be determined. Assuming a harmonic motion, equ. (4) reduces to

$$([K] - \omega^2 [M]) \{\bar{q}\} = \{o\} \tag{8}$$

with the eigenfrequency ω and the eigenvector $\{\bar{q}\}$.

□ Error bounds of eigenvalues
□ Sturm sequence check
□ Shape of eigenvectors
□ Orthogonality
□ Completeness of modal loads

Table 3 Criteria for modal dynamic analyses

In the numerical calculations, equ. (8) will only be satisfied within a certain accuracy. By calculating the corresponding residual vector it is possible to establish error bounds on the eigenvalues $\lambda_i = \omega_i^2$. Thus the accuracy of the eigenvalues of the analysis model can be verified.

It is important, that no frequencies have been missed in the interval of interest. This is particularly important if the number of degrees of freedom of the model has been reduced by condensation. Shifting the origin of the eigenvalue axis to a shift-point Λ (Fig. 4), equ. (8) becomes

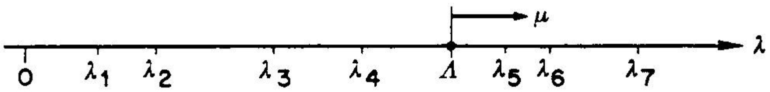


Fig. 4 Shift of eigenvalues

$$([\bar{K}] - \mu[M])\ \{\bar{q}\} = \{o\} \tag{9}$$

with

$$[\bar{K}] = [K] - \Lambda[M] \tag{10}$$

The Sturm sequence check states, that the number of negative terms on the diagonal of the triangular factor of $[\bar{K}]$ is equal to the number of eigenvalues below the shift-point. Applying the check to the uncondensed system at different values of Λ determines the number of eigenvalues in the corresponding intervals.

The discretisation of the structure has to be such that the analysis model can assume the mode shapes corresponding to the eigenvalues of interest. Thus the shapes of the eigenvectors permit a judgement of the quality of the model. In particular, if the spacial wave lengths are of the order of the dimensions of the finite elements, the eigenvector usually reflects properties of the analysis model rather than of the real structure.

The eigenvectors are orthogonal with respect to the stiffness and mass matrix. Thus the quality of satisfaction of the orthogonality conditions is a measure for the quality of the set of eigenvectors.

In a modal analysis, the load vector $\{Q\}$ in equ. (4) is represented by its modal contributions

$$P_i(t) = \{\bar{q}_i\}^T\ \{Q\} \qquad i = 1,...n \tag{11}$$

where n denotes the number of modes included in the analysis. It is important to chose n such that $\{Q\}$ is properly represented with respect to time and space. Usually n is much smaller than the number of degrees of freedom of the analysis model which leads to the omission of high-frequency contents of the solution. If the load is properly represented by the n modes, the neglected structural response will be a quasi-static response. By such considerations the solution can be further improved.

6. CONCLUDING REMARKS

Many structural analysis programs perform certain validation checks automatically or provide options to initiate such tests. The software developers should always keep the validation aspect of an analysis in mind and must make dedicated efforts to enhance the corresponding program capabilities.

It can be expected that the reliability of computerized analysis will still further increase during the next years, especially in the field of nonlinear problems. In particular, self-adaptive discretization and solution techniques in combination with interactive graphics will greatly facilitate the validation of computations in the future. From the experimental side, more and better test data will permit a still better calibration of the numerical procedures as well as for instance of complex material models such as reinforced concrete. All these developments will make computerized structural analysis a still more powerful tool in the hands of the knowledgable and experienced engineer.

REFERENCES

1. BATHE, K.-J. and WILSON, E.L.: Numerical Methods in Finite Element Analysis. Prentice Hall, Inc., Englewood Cliffs, N.Y., 1976

2. ODEN, J.T. and BATHE, K.J.: A Commentary on Computational Mechanics. Applied Mechanics Review, Vol. 31, 1978

3. PFAFFINGER, D.: Praktischer Einsatz von FE-Systemen. Tagungsbericht "Anwendung der Finite-Element-Methode im Bauwesen" vom 6.2.1980 in Nürnberg, Kernforschungszentrum Karlsruhe, KFK CAD 151, 1980

Nonlinear Computations From The User's Point of View

Calcul non linéaire des structures: point de vue d'utilisateur

Nichtlineare Berechnungen vom Standpunkt des Anwenders

Miloš MARINČEK
Professor
Univ. of Edv. Kardelj
Ljubljana, Yugoslavia

Miloš Marinček, born 1918, obtained his civil engineering degree in Ljubljana and Ph. D. in Vienna. He founded the university institute for metal structures and was director until 1961. He was later involved in theoretical research in inelastic behaviour and safety of structures.

SUMMARY
Some suggestions are given regarding the more useful application of nonlinear computations of structures using dimensionless load-displacement diagrams for structures with standardised dimensionless inelastic material and cross-sectional properties.

RESUME
L'article fait état de quelques suggestions propres à améliorer l'application du calcul non linéaire des structures en recourant à des diagrammes charge-déplacement sans dimension, ainsi qu'à des lois de matériaux inélastiques et des propriétés de sections standardisées.

ZUSAMMENFASSUNG
Um eine bessere Anwendung von nichtlinearen Computer-Berechnungen zu erreichen, werden einige Vorschläge gegeben, indem dimensionslose Belastungs-Verschiebungs-Diagramme mit standardisierten dimensionslosen Material- und Querschnittseigenschaften benützt werden.

1. INTRODUCTION

The nonlinear computations of structures have among many possible aspect of the informatics in the field of structural engineering an important role in the further development and successful practical use of the limit states design of structures. So much better assessment of the safety of structures and therefore more rational use of the material in structures is possible.

Owing to his responsibility the design engineer is much interested to know the safety resp. reliability of structures: relating the appearance of undesirable phenomena at the working state, regarding the beginning of unacceptable damage due to the inelastic displacements at overloading, and concerning the ultimate carrying capacity, which is connected with the inelastic behaviour of material too.

The structural behaviour is best represented with the characteristic load-displacement and /or time-displacement relationship. Normalising the load-displacement relationship using the elastic limit state as the norm, a very useful generalisation of the nonlinear behaviour of structures is possible.

"Exact" nonlinear computations of structures represent the simulation of the real structural behaviour and replace more and more the experimental work. However the so called deterministic tests remain very important in order to confirm the theory.

Using computers the consideration of the influence of different parameters is very simple (differnt material behaviour, mechanical and geometrical imperfections). "Exact" nonlinear computations are also very useful for the assessment of various approximate methods. They serve, further, for the elaboration of differnt design aids regarding nonlinear behaviour of structures, which simplify the every day work of the design engineer. For this purpose a choice of appropriate representative normalised (dimensionless) shapes of the material laws, geometries and loadings is neccessary.

Reliability of nonlinear computations is very important. It is the condition for an understanding between suppliers of computer programs and users. The activity of a user is not computer programing, but to solve technical problems, to deside what to do with stresses, strains and displacements obtained. He must rely that the stress equilibrium, strain compatibility and material low are adequately applied in the structural computer programs. On the other hand he is much more interested in a clear and simple enough input and output.

A complete confidence in nonlinear computations (regarding accuracy and costs) can be obtained with systematic computer solutions of basic examples of structural mechanics as benchmarking cases, using representative shapes of the material laws and geometries.

The paper deals with metal structures. The structures made of other materials can be treated similarly.

2. REPRESENTATIVE LOAD-DISPLACEMENT DIAGRAM OF THE STRUCTURE

It is easy to asses the nonlinear behaviour of a structure tracing the representative load-displacement diagram for a given loading path, obtained either experimentally or with the computer simulation. Hovever, the great advantage of the computer simulation is a very simple separation of influences of different parameters. So it is easy to divide geometrical nonlinearities from material nonlinearities, the primary behaviour from the additional one. It is simple to judge the influence of different material laws, the influence of different geometrical and/or material imperfections etc. In this way, with the deterministic knowledge regarding the influence of various parameters and with the knowledge relating their statistical distribution, the rational probabilistic predictiction of the real behaviour of structures is possible. In other words, the deterministic knowledge of the real behaviour of structures until the exhaustion is the condition for the successful probabilistic treatment of the safety resp.reliability of the structures.

It must be emhasised that not only the strength but also the inelastic deformability plays an important role in the assessment of structures. Fig.1 shows different load-displacement diagrams, F-U, with the same ultimate strength. In Fig. 2 the working state (WS), the inelastic deformation limit state (IDLS) and the ultimate limit state (ULS) are presented. While ULS is always on the peak of the curve, the IDLS as the second criteria for the determination of the allovable working state never can be defined uniquely. It depends too much on specific requirements for the given function of individual structures.

It is suitable to treat the inelastic structural behaviour from the point of view of three characteristic plastic strain regions: compression instability with small strains, tension plastic instability with medium strains and fractures with large strains.

The load-displacement diagram of the primary behaviour of a stocky compression component has usually the shape shown in Fig. 3. Material and geometrical imperfections like residual stresses and initial bow can substantially reduce the ultimate limit load. Additional reduction is possible due to lateral or local buckling. The ductility requirements of the material for compression components are low. They are more technologically conditioned.

The load-displacement diagram of the primary behaviour of a tension component with the constant cross-section has a typical shape according to Fig. 4. It has the full similarity to the course of the stress-strain diagram obtained with the tensile test, until the tensile strength is reached at the finish of the uniform elongation. Therefore besides the elastic modulus and the yield stress the tensile strength and the uniform elongation are the leading material properties for tension components. Residual stresses, initial geometrical imperfections and lateral loads have no effect on the ultimate limit state of a tension component if there is no condition for an earlier fracture. Only in the region of small strains the inelastic deformability can be more expressed.

Ductile fracture behaviour with large strains can be observed on bending or torsion test besides usual tensile test at necking. The large plastic rupture strains have an immense influence at sharp strain concentrations like cracks.

In general the primary load-displacement diagram of a component or of a structure can be shortened (decreasing of the carrying capacity and of the ductility) due to the lateral or local buckling at compressed parts or due to sudden fracture or stable crack growth at tension parts. At the stable crack growth the decreasing of the load carrying capacity with the increasing of the crack area has to be considered as the decreasing of the safety with the time, Fig. 6.

Besides the influence of the triaxiallity of the stresses on the deformability and fracture the inelastic behaviour of the material depends also on the temperature and strain velocity (impact, creep). There is an infinite number of possible working diagrams of the materials, with further complications: the anisotropy, the physical nonhomogeneity (different working diagrams) and the geometrical nonhomogeneity (defects).

For the computer simulation of the inelastic structural behaviour the assumption of the linear distribution of strains through the thickness of bars, plates and shells enables a considerable simplification. However , the eventual plastic strain reversal , e.g. at strong seismic loading, together with the influence of Bauschinger-Effect, makes again the computer simulation very complex and costly, also for the research.

3. NORMALISED LOAD-DISPLACEMENT DIAGRAM AND THE USE OF REPRESENTATIVE DIMENSIONLESS PARAMETERS

To overcome the complexity of inelastic computer simulations of individual cases there are two measures at disposal.

First measure is the use of normalised (dimensionless) load-displacement relationship with the elastic limit state as the norm for the load and the corresponding displacement as the norm for the displacement. In this way every individual computation (or experimental test) is generalised, because it is valid for all geometrically similar structures with similar shape of the working diagram of the material.

The second much more effective measure is the choice of typical dimensionless stress-strain diagrams of the material, cross-sections, components and systems, when neccessary also representative dimensionless material and geometrical imperfections can be used. So a lot of generalised computations of typical examples can be made in advance, once for ever. Of course an international cooperation for this purpose is necessary.

Such generalised computations also enable a quantitative classification of structures regarding their inelastic behaviour (ductility, plastic reserve) and the creation of many design aids in the form of technical data sheets as an addition to the future international structural codes, which can be prepared very efficiently with the help of computers.

As an example Fig. 7 shows the dimensionless "horizontal force- horizontal displacement" for a cantilever column. The results of the parametric nonlinear computations of this simplest case are very useful in the judgment of the seismic behaviour of framed structures. In advance prepared moment-curvature relationships for a given type of the cross-section and material certainly considerably reduce the price of the computations.

4. GENERALISED REPRESENTATIVE STRESS-STRAIN DIAGRAMS OF THE MATERIAL

There are two types of the working diagrams for metals: with the continuous strain hardening after the linear behaviour (e.g. aluminium alloy, austenitic steel) and with the additional plastic plateau (ferritic steel). While the plastic plateau clearly represents the yield stress, there is a need of a redifinition of the yield stress for the continuously hardening materials. The yield stress defined with 2‰ inelastic strain should be replaced with the new yield stress defined with the equality of elastic and plastic strain (Fig. 8). In this way the well known Ramberg-Osgood stress-strain relationship, normalised with this yield stress and the corresponding elastic strain, has a simple oneparametric expression

$$\bar{\varepsilon}' = \bar{\sigma}'(1 + \bar{\sigma}'^{N})$$

and $(1+\bar{\sigma}'^{N})^{-1}$ represents the dimensionless secant modulus. The exponents N=4, 10 and 30 can be taken as representative.

For small strain problems the above Ramberg-Osgood equation is generally valid. For medium and large strain problems the elastic limit strain $\varepsilon_e = \sigma_e : E$ as an additional parameter has to be taken into account for the dimensional treatment. Fig. 9 taken from /1/ shows normalised natural ($\bar{\sigma}' - \bar{\varepsilon}'$) and engineering ($\bar{\sigma} - \bar{\varepsilon}$) stress-strain diagrams for chosen representative N=4, 10, 30 and for the practical limits of ε_e from 0,001 to 0,01.

For small strain problems the additional plastic plateau (without supplemental difficulty with the upper yield stress) makes a lot of truble because of its extreme complexity due to the inhomogeneous spreading of plastic strains (Lüder's bands). Mathematically simple horizontal line is physically exceptionally complicated.

It is still open question how to choose typical dimensionless working diagrams for the materials with plastic plateau. In any case the Lüder's strain is an additional parameter for such materials.

5. APPROXIMATE NONLINEAR COMPUTATIONS

Besides the "exact" nonlinear computations there is a need for different approximate nonlinear computations of structures (geometrical and/or material). Of course, a simplified method should not be too uneconomical and should be on the safe side. It is appropriate to integrate the approximate and more accurate computations in one programing system.

Approximate computations can serve also for the preliminary design. However it is important to consider their assumptions and limitations. E.g. the plastic hinge theory can be useful for the predominant bending, but it is on the unsafe side for the predominant compression. For the predominant compression an inelastic bifurcation computation gives a physically clear upper limit of the carrying capacity if no geometrical imperfections are taken into account.

It has to be mentioned that the possibility of an automatic jump from linear to the nonlinear computation should be introduced in the successful structural programs, corresponding to the suitable criteria.

6. RELIABILITY OF NONLINEAR COMPUTATIONS

The best assurance of the reliability of nonlinear computations is a large public use of the corresponding software, selected on the base of a sound competition, taking into account exclusively the criteria important for the structural engineering profession as whole. Such programs are optimal regarding the portability, maintenance and updating. After a certain time their mistakeability is negligible, which is the most important property.

Computer graphics with possible interactive guidance of the nonlinear computation and tracing the corresponding load-displacement diagram, including an eventual additional information regarding the convergence resp. the change of the determinant, is the best tool for the reliable nonlinear computation.

As an example Fig. 10 shows the graphical result of the zero determinant search for an inelastic bifurcation computation for a truss. The decreasing of the inelastic bending rigidity of the cross-section with the increasing axial force is taken from ECCS-column buckling curve "c" (fictitous rigidity, which includes the effect of geometrical imperfections besides the residual stresses). The thin curve represents the verification with the elastic bifurcation calculation, using the inelastic rigidities at the inelastic bifurcation as constant elastic rigidities.

Fig. 11 gives three inelastic bending rigidity curves for axially loaded aluminium alloy cross-section according to ECCS (effective without the influence of geometrical imperfections, fictitious with this influence).

Finally, the confidence into the inelastic computer simulations can partacularly be obtained with the systematic computer solutions of the basic examples of the structural mechanics, using internationally recognised representative dimensionless stress-strain curves of the material. Such examples are: tension of the round bar with the necking, torsion of the round and of the rectangular bar, bending of the bar, tension of a strip with the hole, shear and tension of the fillet weld, Brinell hardness test, Griffith's crack, penny shaped crack, single and multiple pores, buckling of the bar with the Shanley-Effect etc. Besides the normalised load-displacement diagrams the change of typical stresses and strains has to be given, if the fracture criteria have to be considered.

Nonlinear computations of this kind, verified by parallel computing with differnt programs and eventually tested with deterministic experimental tests, would have a historical value for structural mechanics and engineering. For the beginning naturally a homogeneous and isotropic material has to be used. Later representative cases of the nonhomogeneity and anisotropy will have to be considered. Such activity should awake the interest for the cooperation between many international assotiations (IABSE, IUTAM, RILEM and the corresponding assotiations for mechanical engineering and metallurgy).

The need for an international cooperation also regarding the practical use of the results of nonlinear computations can be illustrated with the Fig. 12, taken from /2/. For a typical I-profile (IPB_v340 and geometrically similar) and for the exponent N=4 of the Ramberg-Osgood material the ductility functions K_φ (in this case dependet of normalised bending moment $\bar{M}=M:M_e$ only) have been determined (see Fig.13 for axial force $\bar{N}=N:N_e=0$). K_φ is the ratio between the total bending curvature and the corresponding elastic curvature. It represents the reciprocal value of the normalised secant modulus. Then for the beam with the uniformly distributed load, clamped on both ends, the ductility functions K_δ of the load-displacement relationship have been computed in a semi analytical way

$$K_\delta = \frac{6}{\bar{M}_1+\bar{M}_2}\int_{\bar{M}_2}^{\bar{M}_1} \bar{M}\cdot K_\varphi \cdot d\bar{M}$$

with the condition

$$\int_0^{\bar{M}_2} \frac{\bar{M}\cdot K_\varphi}{\sqrt{1-\frac{\bar{M}}{\bar{M}_2}}}\cdot d\bar{M} = \int_0^{\bar{M}_1} \frac{\bar{M}\cdot K_\varphi}{\sqrt{1+\frac{\bar{M}}{\bar{M}_2}}}\cdot d\bar{M}$$

for the relation between the negative moment $\bar{M}_1$ and positive moment $\bar{M}_2$, fulfiled with the iteration. In the normalised load-displacement diagram in Fig. 12 many different limit states are presented. They are defined either with different procentage of unrecoverable displacements (1-7), actual (8,9,10) or linear stress limits (11,12) and also according the criteria of the plastic hinge theory (13-16). The index 0,2 belongs to the 0,2 yield stress and the index 2 to the yield stress with the equality of elastic and inelastic strain. It is clear that the problems are not so much in the correct use of the nonlinear computations but in the establishment of the generalised criteria for the practical application of results.

7. CONCLUSION

Nonlinear computations have to be added to the usual linear structural software systems using contemporary possibilities of pre- and postprocessing for CAD, CAM and public computer network including computers from micros to supers, with the suitable data bases /3/.
A suitable standardisation on an international level can help essentially to the further successful development of the structural engineering profession in the arising informatic society /4/.

REFERENCES

/1/ M.Marinček: Resistance of aluminium weldments and the corresponding mechanical properties, 2nd international conference on aluminium weldments, May 1982, Munich

/2/ M.Marinček: Flexural behaviour of aluminium alloy structures, ECCS-TC16, 1977

/3/ P.Marcal: Application of computers in pressure vessel calculation and design, 2nd summer school for fracture mechanics, June 1982, V.Plana, Yugoslavia

/4/ F.Takino: Considerations on proper usage of design programs, IABSE colloquium on interface between computing and design in structural engineering, 1978, Bergamo

/5/ J.J.Servan-Schreiber: Le defi mondial, 1980

FIGURES

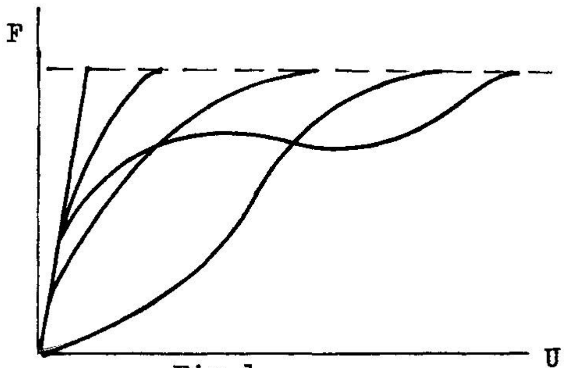


Fig.1

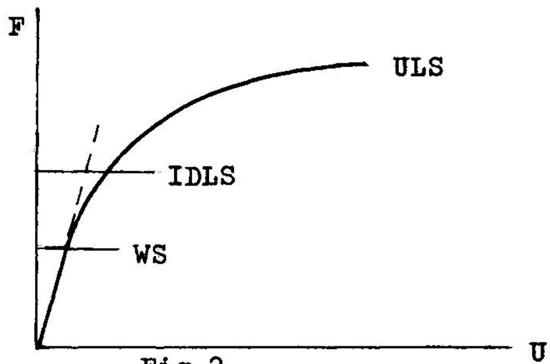


Fig.2

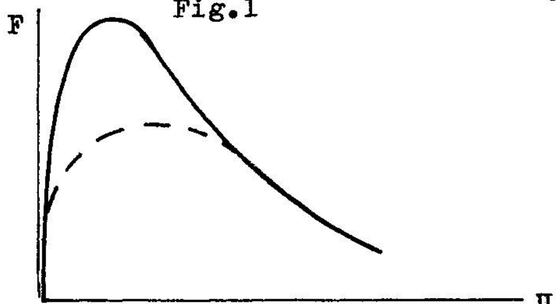


Fig.3

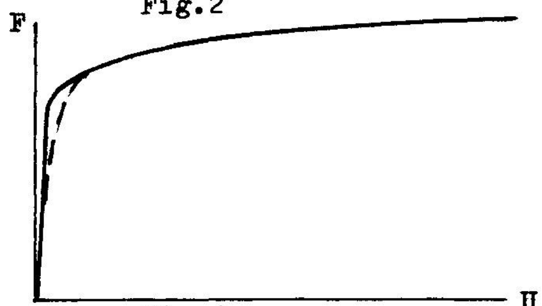


Fig.4

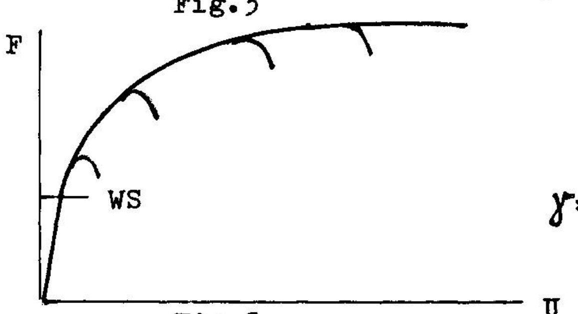


Fig.5

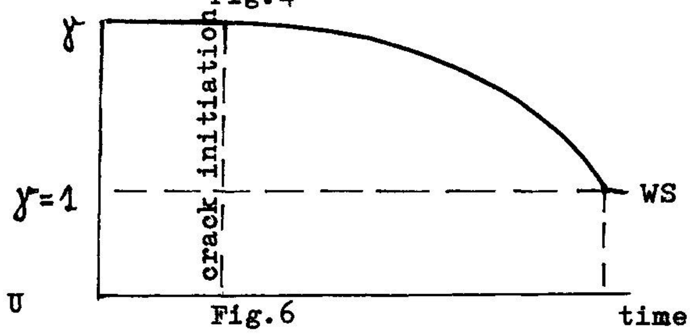


Fig.6

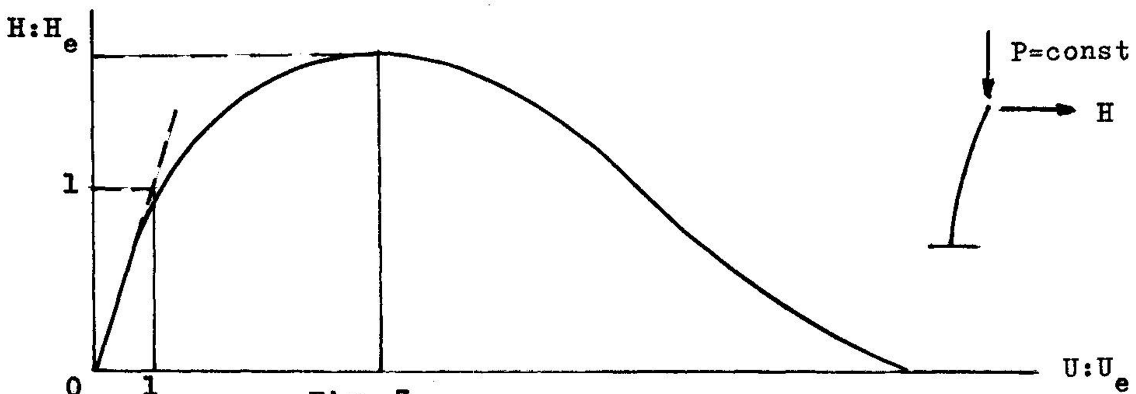


Fig. 7

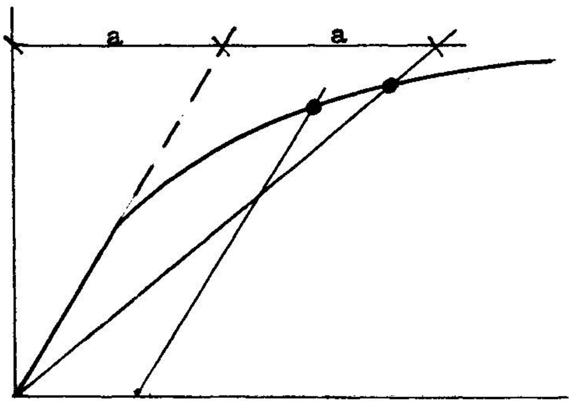


Fig. 8

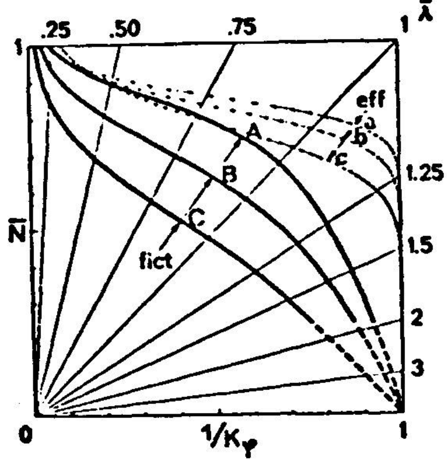


Fig. 11

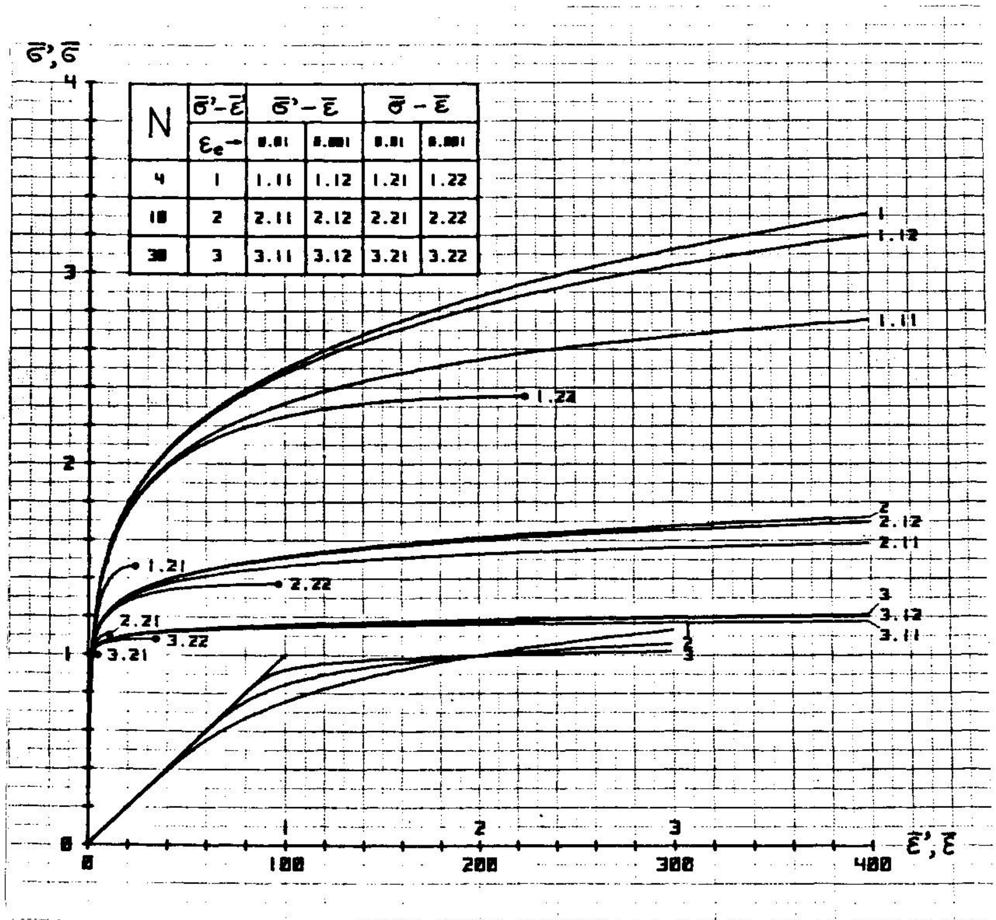


Fig. 9

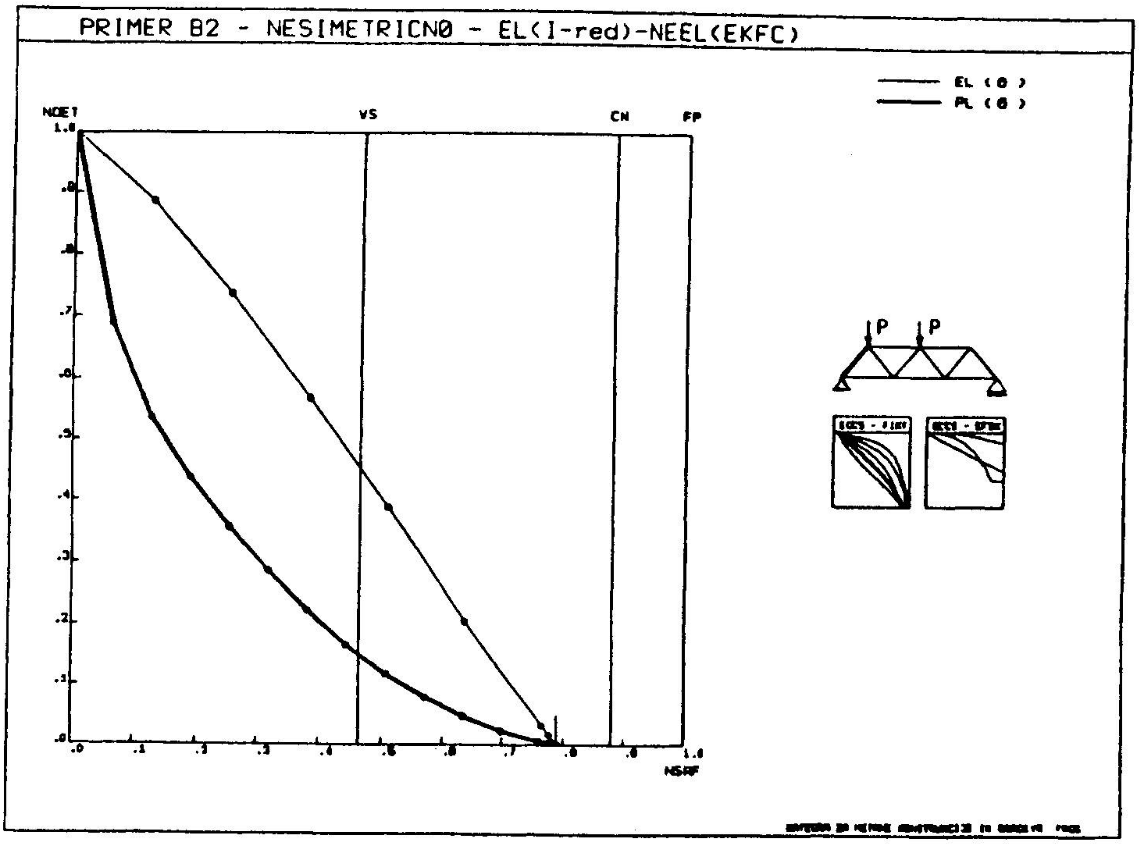

PRIMER B2 - NESIMETRICNØ - EL(I-red)-NEEL(EKFC)
EL (8)
PL (8)
NOET
VS
CN
FP
NSRF
P
P

Fig. 10

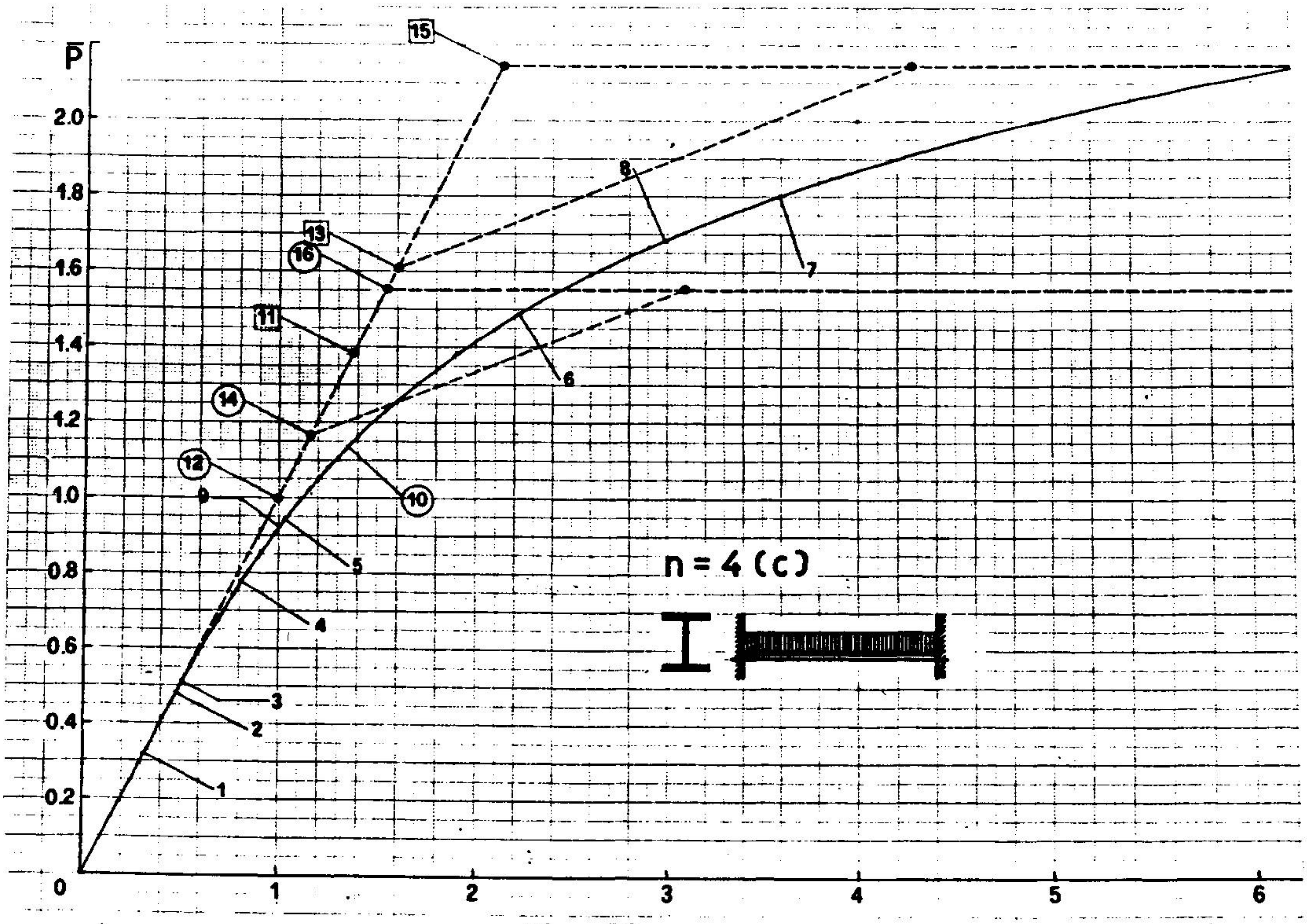
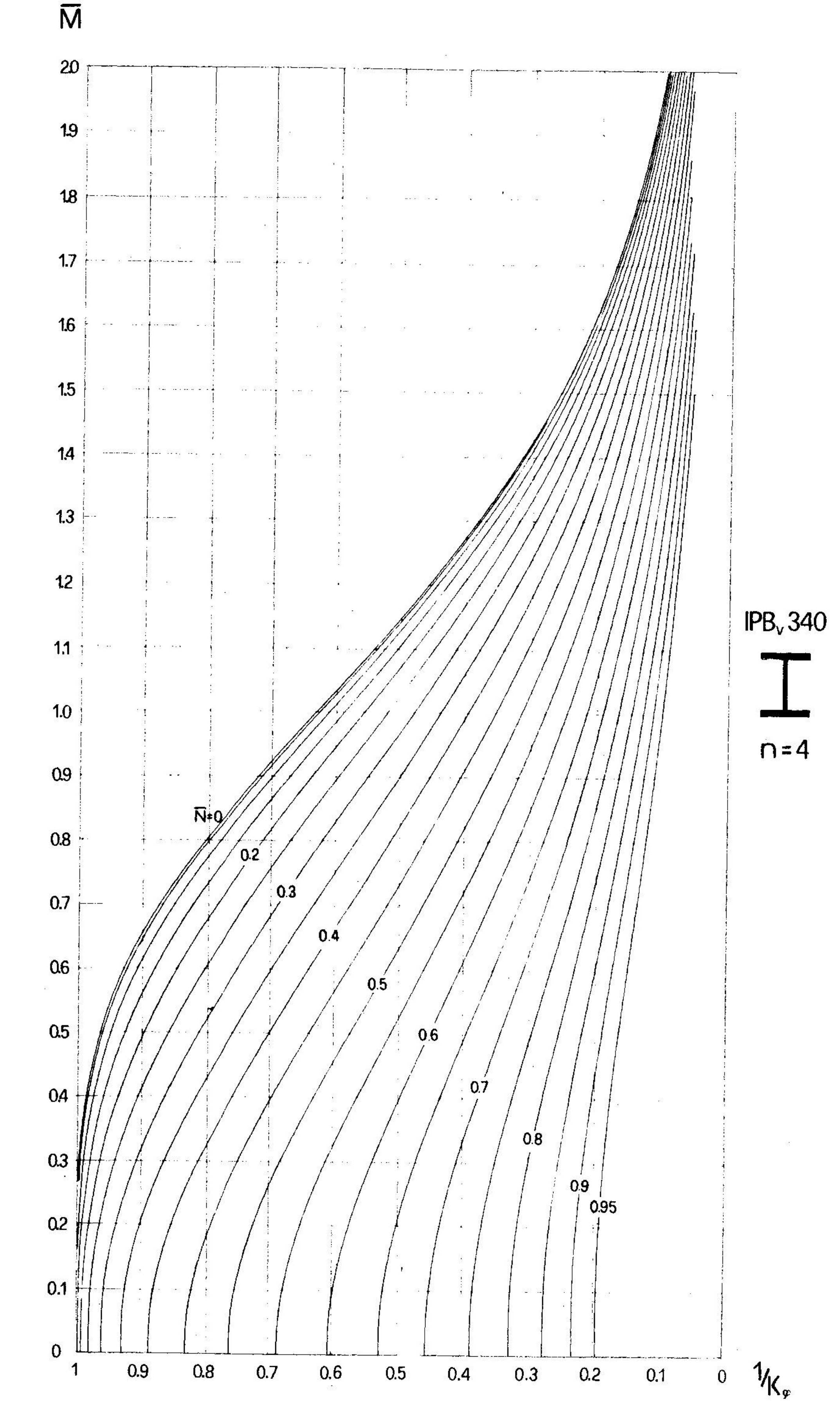

P̄
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
15
13
16
11
14
12
9
10
8
7
6
5
4
3
2
1
n = 4 (c)

Fig. 12



$\overline{M}$
2.0
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
$\overline{N}=0$
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.95
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
$1/K_\varphi$
IPB_v 340
n = 4

Fig. 13

Computer Usage in Building Structural Design

Utilisation de l'ordinateur dans la conception des structures

Computer-Anwendungen im Entwurf

Matsuo KUWAGATA
Staff Engineer
Nippon Telegraph & Telephone
Tokyo, Japan

Matsuo Kuwagata, born 1933, obtained his architectural engineering bachelor's degree at the Tokyo Institute of Technology. Now, he is engaged in the development of DEMOS-E library programs for building engineering. He is a member of the Architectural Institute of Japan.

SUMMARY
This paper describes the present situation of computer usage in the field od structural design and presents several problems in computer application, such as ill effects of computer usage, program reliability and quality assurance, education in computers, and so on.

RESUME
L'auteur expose la situation actuelle de l'utilisation de l'ordinateur dans le domaine de la conception des structures et des bâtiments. Il met en évidence plusieurs problèmes liés à l'application de l'ordinateur, notamment: certains échecs qui peuvent lui être attribués, la confiance et l'assurance du niveau de qualité qui peuvent être accordés aux programmes, ainsi que la formation des utilisateurs.

ZUSAMMENFASSUNG
Dieser Bericht behandelt die gegenwärtige Situation im Entwurf und zeigt einige Probleme bei Computer-Anwendungen, wie z.B. schlechte Auswirkungen infolge Computergebrauchs, Programmverlässlichkeit, Qualitätsabsicherung, Ausbildung auf dem Computer, etc.

1. INTRODUCTION

Based on the experience and knowledge of building failures caused by earthquakes which occurred in these decades in Japan, research and development of a-seismic technology have been well in progress. Depending on the technological progress, the Japanese Building Code was revised in 1981, and accordingly, building structural design procedures are remarkably changed. In addition to examining that the member working stress is less than the material allowable stress, several other checks are needed under specified conditions. For example, the building ultimate resisting capacity to external forces should be estimated. Because of the new requirements, the amount of structural calculation is nearly doubled, and structural engineers can hardly carry out their work without the help of computers.

With such background, the structural calculation programs and computers capable of dealing with such new design requirements as mentioned above, are increasingly in demand. Therefore, the subscribers of TSS service, the users of computer bureaus, and mini- and micro-computers in house are remarkably increased for structural calculation usage. For instance, the number of subscribers of architectural and building engineering firms for DEMOS-E, which is a representative of TSS services in Japan, are over one thousand in January 1982. Further, computer bureaus, which give services of evaluated structural design programs, are on the increase. Micro-computers, excellent in operation and graphical presentation, are remarkably widespread among design firms and their number may exceed two thousands.

Through the diffusion and popularization of computer systems and structural design programs, a growing interest has been created in the problems on computer application to structural design, which are now under review among engineer circles. Namely, they are the problems on ill effect of computer usage, the problems on program quality assurance, the problems on responsibility associated with computer application and so on. In order to discuss and investigate these problems, AIJ (Architectural Institute of Japan) organized a preparatory committee in 1978 and set up a permanent committee in 1980. Under the auspices of this committee, symposiums are held every year. The one held in spring, 1982 was the fourth, in which these problems were keenly discussed, especially program reliability issue in the panel discussion.

In this paper, the present situation of computer usage in the field of building structural design is described first, and then problems on computer application, which are focussed through the discussion in AIJ committee and other circles, are presented.

2. THE FEATURE OF COMPUTER APPLICATION ON BUILDING STRUCTURAL DESIGN

2-1 Popularization of Computer Usage in Building Structural Calculation

As the result of legislative requirements of calculation of building ultimate resisting capacity based on collapse mechanism and due complication of structural calculation procedure, the amount of calculation done by structural engineers has definitely increased. The calculation amount may be doubled in comparison with that required under the conditions of conventional earthquake coefficient method and during elastic and allowable stress analysis. Further, because of analytical conditions complexity, such as three-dimensional effect depending on eccentricity, the soil deformation effect under the footing and the effect of column-girder connection behavior, calculation work has a tendency to increase steadily. There has been a growing demand for computer usage in order to manage the calculation work efficiently and economically. The fact is confirmed by the increasing trend of TSS subscribers, computer bureau users and micro-computer sets. According to the recent investigation of computer application on structural design, the computer usage ratio in structural calculation ranges from 30% to 90%, and it may be recognized that computers are widely used in structural calculation work.

2-2 Development of Overall and Automated Structural Calculation Programs by Big and Competent Design Firms

Many big design firms, which have technological and economic potential to make up large software, set about developing the overall automated structural calculation programs for supplying the above mentioned computational demand. Most of these programs are qualified by the evaluation procedure and meet the administrative requirements. Up to 1981, the number of evaluated programs have become twenty-four series/thirty-nine programs. In these, six series/nine programs are supplied to the public users. Accompanying this program developing race, several issues on program making, such as efficient programming technique, program module parts, and so on, have been eagerly discussed. In parallel with the matters, such criticism has arisen that a few stereotyped programs will limit the diversity of structural planning and make structural design monolithic.

2-3 Popularization of Small Computer

In these days, the popularization of small computer is amazing, because of its performance improvement and lowering price due to electronics development. Supposedly, many architects and building engineers firms possess mini- and micro-computers. This tendency will be much more accelerated, as the building structure automated calculation programs for small computers are evaluated and easily accepted to administrative confirmation. While small computers can function as intelligent terminals connected to a big computer system, it can control its own terminals, such as graphic display equipment, plotters, and digitizers. Utilizing these multifunctions of small computers, construction of integrated and efficient computer system composed of small and big are also tried in design firms. For developing various possibility of small computers in building engineering, a sub-committee is organized in AIJ computer usage committee and starts to study the matters of small computers.

2-4 Distribution of Information, Data, and Programs

With the advance in computer hardware usage environment, there have been growing needs for comprehensive information, and a strong tendency toward open distribution of data and program in building engineering circles. However, the matter of information distribution is at the beginning stage, and it has many problems to be solved. For instance, evaluation of software and data, legal protection of program copyright, standardization of program documents, arrangement of distribution mechanism, and so on. For the sake of treating the issues of information distribution, a sub-committee has been organized in AIJ computer usage committee. The sub-committee now conducts an investigation of data and programs, such as meteorological, earthquake wave data, and building engineering calculation programs, and plans to promote the distribution of these kinds of information.

2-5 A Growing Concern in the Problems on Computer Usage

Concerning such pivot issues that main procedure becomes black-box in the automated structure calculation programs and how engineers cope with this matter, there have occurred various discussions. They are such problems as responsibility of engineers and programs, abuse and misuse of programs, and education on computer usage in schools and firms. On the issue of program reliability, it was chosen as the main theme of the panel discussion in 1982 AIJ computer application symposium, and discussed from the various standpoints of developers, operators, users, and administrative officers. The important arguing points in the theme are as follows: highly reliable program making technology, confirmation method of program reliability and feasibility of social checking system, way of avoiding program misuse, knowledge of how to use and check programs, and responsibility limits between developers, agents and users.

3. PROBLEMS SUBMITTED ON COMPUTER APPLICATION

3-1 Ill Effects of Computer Usage

3-1-1 Abuse and Misuse of Programs

It is practically impossible for users completely to check and understand the functions and details of building structure automated calculation programs, because these sorts of programs have very complicated and highly developed contents, and besides, they are very big in size. (For example, DEMOS-E BUILD-1 has eleven hundreds kilo statements in FORTRAN) The black-box phenomena of programs cause its misuse because of deviating from the basic preconditions and scope of the programs. Meanwhile, it is a matter of deep concern that unexperienced engineers use automated calculation programs blindly and design structures directly under instructions from the computers. This means abuse of programs. Furthermore, there is another abuse of programs such as applying programs unreasonably after unnatural modeling of design objects.

3-1-2 Monolithic Standardization of Design and Obstruction of Engineer Advancement

Over dependency on programs produces a tendency to do structural design fitting in with the program function and scope, and to confuse means with ends. Consequently, there increases uncharacteristic structure with poor consideration by blindly depending on the evaluated programs. Engineers may become skillful in applying programs. However it is hard for them to develop design ability, such as, of modeling and analyzing structure correctly, and getting at the essence of its behavior.

3-2 Program Reliability and Quality Assurance

3-2-1 Reliability

On program reliability, it is recognized that there exist two facets: (1) correctness of program itself, (2) correctness of program using design. First problem deeply depends on developers.

For decreasing mistakes at designing and programming stages, highly reliable program designing technique, systematic program testing, and exact program documentation are necessary. Second problem is mainly related to users. It is necessary that program users avoid program misuse and check computing results. For correct usage of programs, reliability of users' manual and support of consulting engineer skillful to the programs are important. In Japan, the evaluation system performs an important role to confirm the automated structure calculation program reliability. With regard to program documentation, the sub-committee of document standardization in the AIJ committee has investigated various issues and prepared a guide book "Documentation for program development, maintenance, and usage", which is publicized at 1982 symposium.

3-2-2 Quality Assurance

It is being understood that one hundred percent debugging of programs is impossible in practice, but consciousness of users is severe on program bugs yet. Program users require always to be informed on bugs, and sometimes, to go back to the past and check bug influence. In general, programs debugging and quality improvement are completed the faster, the more frequently programs are used. Furthermore, it can be said that programs service, together with consulting engineer's service, is the best quality assurance.

3-3 Responsibility on Program Usage

An idea that the responsibility on program usage belongs to users, is being fixed. In case of computer bureaus, such contract between user and bureau is becoming popular in Japan that the responsibility of the bureaus on program usage is reasonably limited to, at most, computing charges.

3-4 Program Development

3-4-1 Input Issue

Input data mistakes frequently occur, because the sort and quantity of input records are many, and input regulation is rather complicated, and further, users manual explaining input records is liable to be not understandable. It is said that input data correction frequency necessary to be accepted to programs is from two or three times at least, to five or six times at most. Recently, micro computer is used for input data generation and syntax check especially applying its graphic display function. This technique of input data preparation is very efficient.

3-4-2 Output Issue

As for program output, it is not readable as a structural calculation document, because of a large amount of output and lines of mere characters. According to the use of document, output items should be arranged as serviceable as possible. Even in case of printers, it is necessary that readable output presentation, such as graphically arranged output similar to building plan and plane frame shape is thought out. It is a matter of welcome that graphic and diagram output such as by graphic displayer and plotter is developed.

3-4-3 Program

Self development is ideal, but it is economical to use others' programs when we consider developing and maintenance cost. In case of using others' programs, the programs which are of usually overall automated structure calculation systems, are package type, and cannot accept users' option, such as linkage of users' sub-programs to the host programs. From this viewpoint, a module type program wherein users can easily link his own module programs to host system and compose a tailor-made program is desirable. Furthermore, most of automated structure calculation programs are generally big in size, and they are redundant in case of rough check at primary design stage. Therefore, such CAD type programs are desirable wherein engineers can freely insert his judgment into the program through man-machine communication in the middle of calculation.

3-5 Evaluation of Programs

It is a general viewpoint that the program evaluation system is not almighty, but effective to guarantee reliability and quality of programs and this is a kind of necessary evil. Especially, building officials strongly support the evaluation system, for the reason that it is difficult to check correctness of the program submitted to officials for building permits. In case of structural calculation being done by computers, building officials usually insist on the use of the evaluated programs. Against this point of view, there are quite a few whose opinions are that responsibility of structural design is not on computer programs but on engineers. Further, there is such another opinion that the procedure of program evaluation is not so clear and accurate.

3-6 Education on Computers

There are such strong opinions and demands that a subject on computer application technology should be added to the curriculum for building engineering. However, education on computers in schools is on a trial stage and far from systematization. Now, it stays mainly in the training level of programming languages, such as FORTRAN. People expect that reasonable computer education should be given in schools for engineers to acquire ability of choosing applicable programs and using computers fast and correctly in practical work. In AIJ, a sub-committee on computer education has been organized and started various activities, such as surveying the state of the art on computer education in schools, setting up a standard curriculum, and editing textbooks for various levels.

4. CONCLUSION

In this paper, the problems on computer usage, focussed on structural design, are discussed, and the demerits of computer usage are exclusively emphasized, but these negative effects never reduce the merits of computer usage at all. On the contrary, it is believed that computer usage, by overcoming the demerits, can contribute to the advancement and rationalization of structural design. The presentation of the problems mentioned in the paper were initially triggered by building officials about ten years ago, and the discussion was held from administrative point of view. At present, the problems are widely investigated in the field of AIJ committee. Few practical solutions are presented in the paper, however steady countermeasures have been considered and they will produce fruitful results in near future.

ACKNOWLEDGEMENT

The author wishes to thank Professor M. Fujimoto, Chairman of the Computer Usage Committee in Architectural Institute of Japan, for his kind advice on the preparation of this paper.

Appendix A

Activity of Computer Usage Committee in AIJ

1. Computer usage committee is composed of four sub-committees and their activities are as follows:

(1) Sub-committee of information distribution
 a) Objects: Facilitation of information transfer, such as programs, data on architecture and building engineering.
 b) Results: Surveying the state of the art on computer usage, programs and data in architectural and building engineering firms, and the results being publicized.

(2) Sub-committee of program documentation
 a) Standardization of program documents.
 b) Preparation and publicity of program documentation guide book "Documentation for program development, maintenance and usage". (Ref. Appendix B.)

(3) Sub-committee of education on computer usage
 a) Survey of computer education in architectural and building engineering course of technical school, college and university, and proposal of computer disciplinary curriculum.
 b) Publicity of the surveying results for various educational organizations, and preparation of textbook "Computer application series for architecture and building engineering".

(4) Sub-committee of small computer application technology
 a) Survey and investigation of the matters peculiar to small computer, such as the state of the art of hardware and software, documentation, algorithm, computer aided design, and control technology.

2. Symposiums held by the committee are as follows:

No.	Year	Participants	Papers	Themes of Panel Discussion
1	1979	659	104 (32)*	—
2	1980	600	86 (33)	Information distribution on computer application in building engineering.
3	1981	466	61 (27)	The present situation and the future of education on computer usage.
4	1982	448	64 (27)	1) Investigating the issues on new aseismic structural design code 2) Program reliability

Note. * The figure in parentheses indicates the number of papers on structural design.

Appendix B

Program Documentation for Program Development, Maintenance, and Usage (How to make a good program) (Draft)

Sub-committee of program documents has proposed a guide book on program documentation for improving program reliability and facilitating program distribution. In the following, the table of contents is shown.

1. Necessity of program documentation
2. Variety of documents
3. Documents at program developing
   - 3.1 Surveying and planning stage (Development plan briefing)
   - 3.2 Basic design (Basic design specification)
   - 3.3 Composition design (Composition design specification)
   - 3.4 Detail design (Detail design specification)
   - 3.5 Coding (Program list)
   - 3.6 Module program test (Module test report)
   - 3.7 Integrated program test (Integrated test report)
4. Program manual
5. Users manual
6. Description of program summary

   Standard program description form for common knowledge and program information transfer.

REFERENCES

F. Takino, Considerations on Proper Usage of Design Programs. IABSE COLLOQUIUM BERGAMO, August 1978 PROCEEDINGS.

SESSION IV

DISCUSSION

October 8, 1982 - Morning

Chairman: G. DEPREZ (Belgium)

J.P. RAMMANT - I do have a question to the last two speakers concerning checking programs. I have got experience that, for a non-linear analysis, I had to struggle against some Regulatory Commissions and they proved that my program was functioning wrong, because their program was functioning all right. Is this a valid method? I ask you because you didn't mention it for checking. Programs should not be used to check other programs. I think it is a serious question.

D.D. PFAFFINGER - May I have a short comment on that question? I happened to have the same experience. We were forced some time ago to verify a program by recalculating against another program, which has been accepted by the Authorities. We were just forced to do that. I think it is not a very good way to do it, because, if you compare the results from several programs, even in the simplest case, you usually get a slightly different result from every program. So I would say that it is not a good practice of checking. But there is little you can do about it.

M. KUWAGATA - Is your questioning point a way of checking a program? Usually we use a benchmark test, because there are many similar function programs in Japan, so we can compare the results on the same model of a structure. And that is a most reasonable way. A second way of checking programs is to chose a simple model. In that case we can get the result of mathematical analysis methods, so we can compare the result of analysis and program computation result. Of course there are many other ways, but these are most frequently used method for checking a program.

M. FANELLI - Speaking about program validation, or verification, I think that a way, that is both very sound in principle and very appealing to the practicing engineer, is the validation against experimental results, if the experimental results are properly obtained of course. And, in this connection, I would like to mention that it seems to me that there is a trend now in all branches of engineering, where experimental data are both precious for validation of theory (and so also computer programs) and difficult to obtain; a trend, say to try to establish, through International Organizations, international databases of experimental results, which can be put at the disposal of people interested, for instance, in validation on computer programs, trying to simulate the thing that is done by experimental results. This has been done, for instance, in the field of fracture mechanics that I know of, and I am sure it is being done in several other fields. At least in one instance, I am directly concerned, not in the field of civil engineering but in the field of hydraulic engineering. I feel that this could be a good way to share the knowledge and to promote workshops or benchmark tests to compare different programs and to validate them. If you do that, sometimes you get some very unsettling experience, even on very simple cases. Different programs give very different results and you discover that the programs can be very sensitive to the basic models that are incorporated into them, or to the basic analytical procedure that you use, expecially so in dynamic problems. So I would plea for a wider diffusion of this practice on international databases of experimental results, connected with the practice of organizing international workshops, or benchmark tests to compare and validate computer programs. If someone would comment on that, I think it would be useful.

E.ANDERHEGGEN - To Prof. Fenves: I was impressed by the fact the you could convince the old people of the Committee to use computer programs and somehow to produce codes, if I understood you correctly. Could you tell us exactly how this program works, what is the input, what is the output? Finally you have to produce a text; do you have text editing facilities incorporated? How does the thing really work in practice?

S. FENVES - Concerning your first comment, if you show people that you can help them, they are usually quite willing to do so. Specifically, in the case of the AISC steel specifications, it was George Winter (whom many of you know through IABSE) who looked at our first document - a formal representation of the specification that was just passed in 1969 - and who came to me afterwards and said, "This was a nice dry academic postfacto exercize; what can you do to help us while we are drafting the specification?". Much of my work since then has been the direct result of that comment.
To the second question, we haven't started text writing yet, but I have a couple of students who are interested in doing that, purely as an exercise in artificial intelligence. All you need is a random number generator that generates a few simple variants on the sentence "should not exceed", "shall be less than", "shall be limited to", and you can produce texts.
The program that is coming out of the U.S.National Bureau of Standards accepts descriptions of data items and of decision tables. For the decision tables, it can generate trees and identify missing, redundant or contradictory rules, and simply reports that result; the user can go back, add rules, change entries and redo the analysis. At the network level, it accepts, as part of the definition of the data items, the local precedence among them and assembles all of that together (it is a cannibalized CPM program, nothing more than that; a partial precedence ordering among branches of a directed graph) and checks to see if the graph is connected and is acyclic. If the graph is not acyclic, the program outputs the list of nodes that comprise a cycle. The user can look at it, go back, change the definitions and so on. For outlining purposes, you can ask for a display of a spanning tree. Any spanning trees of the network is an ordered sequence; all the links that are not in the spanning tree become cross references in the text, pointing to things that have been previously defined. You can always order the tree so that everything points one way, so that you don't have the usual shifting in the text where something will be defined fifteen pages later, while some other thing has been defined seven pages earlier. Finally, at the outline level, classifiers are attached to individual provisions, so that you can generate trial outlines, and change them around as you like it. That's all the program does at the present. As I said, some people are interested in expanding it into production of text.

J. BLAAUWENDRAAD - A question for Prof.Fenves and a second one for Dr.Pfaffinger. However firstly a comment. We heard that there were roughly two reasons to do the work that you explained us. One was that several States in the United States have so different codes, I understood (at least there are differencies, and you try to cover that) and, on the other hand, just to make your codes better and more complete. That would even hold if you had one code for all States, isn't it? I think we have similar problems in Europe but we try, stimulated by the European Communities, to harmonize our codes first and then try to get a better text. We have a committee which is studying this. You said you are in contact with CEB. I do not know if they are feeding in the European Community Committee, but it would be nice to have you in contact with that Committee. And my question is: these codes are growing and growing and get more and more detailed. There may be a danger that you make it easier to go even further in this way? What do you think about it? Do you stimulate it by using decision tables, or don't?

S. FENVES - That question has been brought up before. You can talk to any Committee member that has been through the exercise that we put them through when they ask us to cooperate with them, and they will be the first ones to vote for simplicity, conciseness and compactness, because we make them work twice as hard as they would normally work. Maybe Brook's chart is correct, writing a specification is much like writing a program. To do it our way may multiply the work by a factor of nine, as Brook indicates about programs. The people that we have contact with and that we have educated by us would be the last ones to be tempted to add more regulations.

D.D. PFAFFINGER - To Prof. Fenves. The final result will be written text to the public. Will you also give the decision tables to the programmers, who have to convert the written text again into Fortran statements?

S. FENVES - In the previous studies that we did, such as AISC69 and AISC81, the decision tables have been published as separate documents. At that time, we could not get the original committees to review and approve the tables, but a lot of people are using them directly as a source document for coding and a lot of students are using them. I understand that after a very complicated lecture on buckling provisions and columns in the steel specification, the students sneak down into the library and look at the decision table to find out what the lecture was about.

J. BLAAUWENDRAAD - Another question to Dr.Pfaffinger. I liked your presentation and I think it is very useful. My question is, is it all validation on the user's level when using a program? Or is part of it in fact a check on the validity of the program at the moment when it is brought to the market? A couple of things may be done, especially you plea for inserting automatic checks and things like that.

D.D. PFAFFINGER - Infact it's both of it. I would say: most of the validation checks have to be asked by the user, but the programs have to provide the means for those checks and things, that can be done automatically, should be done by the program; so the ball is also with the software developer. But as the things are now, most of the validation has to be done at the user's level.

H. PIRCHER - My question is how clear is the responsibility in case of bad mistakes and in case of big damage, and if somebody can find the mistake in the program and the calculation. How clear is the responsibility?

D.D. PFAFFINGER - The general position is that the only responsible man is the engineer. He has the final responsibility. There is no responsibility on the data center, or the software developer or whatsoever. The final responsibility is of the engineer and, if there is something wrong, he will be the one who has to defend himself.

H. PIRCHER - Is it clear also for the Law? In the case of a damage what will be the happening for the law?

D.D. PFAFFINGER - It is clear in ordinary cases, with the exception of gross negligence. If it can be shown that there has been gross negligence then you will be in bad shape, but that's usually not the case. But I am talking like a lawyer, I am not.

G. DEPREZ - It is sure that the engineer is responsible of results. He has to supply correct computation and good design. The informatic field has a responsibility of means. Now it is clear for everybody that it is impossible to give guarantee that a program is safe and reliable. At present we can be more sure, but not absolutely sure that a program is reliable. For the lawyer, from time to time these ideas became more clear, but it is sure too that all the firms which sell programs has to let know clearly every detail to their customers. If they don't use this normal way, they can be considered like people who wanted to sell something different from what they sell. They could have the responsibility to have not informed their customers on what they exactly sold them.

H. PIRCHER - What can we do, so that also persons out of the engineers community get this opinion?

M. KUWAGATA - As for the responsibility of computation results, when I read a paper of the proceedings of the colloquium held in 1978, I had a very strong impression. I feel sorry, I forgot the name of the author of that paper. It said that there is a famous sentence on the program manual face: "This program has been tested to the best of our knowledge, any responsibility in connection with the use of this program must however be declined". Such a sentence, or such philosophy is not valid in Japan now. But, anyway the responsibility is very clear, great and heavy to the program owner. Therefore, we and other big software developers, like Japan IBM, have the limitation of the responsibility to the computation charge.

H. PIRCHER - I have to say that all this is my opinion too, but I have some experiences that this opinion is not a common opinion to others. I have an example: we had to do with a very unexperienced client and we had to do calculation and, at the end, the calculation was wrong and stupid due to three mistakes. A first mistake was that the client prescribed nine prestressing cables for a box-girder bridge (and we know that the number of prestressing cables should be two, four, six, eight or ten, not nine); this was the first mistake. The second mistake was a normal mistake in input prepared by us and the third mistake was an error in praparing data for loading, done by the client. So we had three mistakes and the calculation was very very wrong, but the client needed three weeks to discover it and they said they had a three weeks work to be remade and it was necessary because our input data was wrong. So we had three mistakes, three weeks occuped and we needed insurance to fight against them. It was very difficult to manage the situation and I think the common opinion is not so clear, and I think that it is a problem for the user of programs and especially for smaller programs. That is also a problem for education.

S.J. FENVES - I have found that the opinion stated was very common. The case that you described about a court judgement in Japan, I cannot see how that kind of a judgement could be rendered by any Court in the United States, awarding damages based on a second party misusing, or not knowing, how to use the tools that he contracted to deliver. I cannot see that in a similar situation any Court would exonerate the designer from mistakes in the program.

D.D. PFAFFINGER - May I just say one sentence? I think that situation is not a question of opinion but a question of formulating a contract with the client correctly and you have explicitly to state what you are liable for and what you are not. Then you are covered in all ordinary cases with the exception of gross negligence.

G. SCHMIDT-GOENNER -If you are not able to use a pocket calculator you should not use it. You cannot make the man who sold you a pocket calculator responsible for the bad results you get out of it. I think, if you bring it down to that level, it should be clear who is responsible for the results of an engineering task.

P. LENGYEL - My problem is - I fully agree - that the user should have the responsibility for using programs, but, if we are thinking of the basic goals of structural engineering, then, by this mean, only one goal is achieved and I am thinking of two aims. The first one is to achieve all results always with less effort that is something we can achieve this way: by using computer programs and checking the results with traditional methods, with results achieved by traditional methods or by other programs. However, I am not sure we can achieve this way the other aim, which is to get more economic structures, that is that all new results during the development of new models will in any cases differ from the previous results and, if he takes this solution, it is not guaranteed. I think the way out may be what Prof. Fanelli has advised for achieving this aim and this would be partly a question also if I may quote it. Doesn't Prof. Fenves think that somehow the judgement of computer sooner or later, may be later, must be included and regulated in the standards because of the second aim?

S.J. FENVES - If you are talking about standards in the European terminology, definitely yes. If you are talking about standards - namely performance specification in U.S. and CIB terminology - the answer is no. What designers want in performance specifications is less and less prescription. A pure performance specification cannot possibly address the tool that you use to derive the results. A performance specification says the light intensity in this room shall be so many lumens per square meter. How you achieve that level, is nobody's concern. Whether you use a computer to calculate that intensity has nothing to do with that performance standard. So, if you are talking about performance standards, there is no place in them for mentioning the computer. If you are talking about procedures or prescriptions on how to do things, then computation is definitely involved.