Zeitschrift: IABSE congress report = Rapport du congrès AIPC = IVBH Kongressbericht

Band: 13 (1988)

Artikel: Computers and building codes: enemies forever?

Autor: Syvertsen, Tor G. / Sandvik, Rune

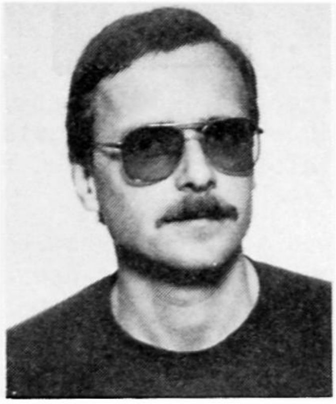
DOI: https://doi.org/10.5169/seals-12993

Computers and Building Codes – Enemies Forever?

Informatique et codes de construction – ennemis pour toujours?

Computer und Normen – Feinde für immer?

Tor G. SYVERTSEN
Scientific advisor
SINTEF
Trondheim, Norway



Tor G. Syvertsen, born 1949, received his MSc in Structural Mechanics at The Norwegian Institute of Technology (NTH). Working in consulting and research, he has been involved in many aspects of computer-aided structural engineering. He is now senior lecturer at NTH and scientific advisor to the research organisation SINTEF.

Rune SANDVIK
Scientist
Norw. Council for Build. Stand.
Oslo, Norway

Rune Sandvik, born 1952, received his MSc at the Institute of Geophysics, Meteorology, University of Oslo. He has been working at The Norwegian Council for Building Standardization incorporating later results of Building Climatology research in national and international standardization.

SUMMARY
The traditional formulation and representation of codes and standards constitute serious obstacles for incorporation in a computer integrated engineering environment. New models and methods for formulation of standards has been proposed and prototyped, but practical applications are lagging far behind the needs. An intermediate approach based on a functional specification for computer implementation of an existing standard for snow loading is proposed and discussed, and some pertinent problems are illustrated. Alternative approaches for computerized representation and processing of standards are briefly discussed.

RÉSUMÉ
La formulation traditionelle des codes et des normes crée des obstacles à leur incorporation dans un environnement d'ingéniérie informatisé. De nouveaux modèles et méthodes de procédures d'homologation ont été développés mais leurs applications pratiques sont inexistantes. Une approche intermédiaire d'une formulation fonctionelle pour le développement informatisé de normes actuelles sur les charges de neige est proposée et quelques problèmes concrets sont donnés en exemple. Différentes approches pour le développement informatisé des codes sont rapidement discutées.

ZUSAMMENFASSUNG
Die traditionelle Gestaltung der Normen bildet ein ernsthaftes Hindernis bei der Berücksichtigung in computerintegrierten Ingenieursystemen. Neue Modelle und Methoden für die Normentwicklung werden vorgeschlagen und erprobt, aber die praktischen Anwendungen entsprechen noch in keiner Weise den Bedürfnissen. Eine vorläufige Formulierung, die auf einer funktionellen Formulierung für die Computerentwicklung einer bereits existierenden Norm der Schneebelastung basiert, wird vorgeschlagen und diskutiert. Einige aktuelle Problemstellungen sind als Beispiele angeführt. Alternative Verfahren für die Computerverarbeitung und die Entwicklung von Normen werden kurz diskutiert.

1. COMPUTER INTEGRATED ENGINEERING AND BUILDING CODES

We are facing a new computer revolution. In the next decade, computers "will grow more powerfull by at least an order of magnitude and become a ubiquitous intellectual utility" [1]. The introduction of the personal computer made the accessibility and convenience of computers increase even faster, and Apple's Macintosh exhibits how the computer can be designed to fit human behaviour. Adding databases, networking and communications, the technological foundation is laid for a computer integrated engineering environment, where the engineer from her workstation will have access to a variety of tools, functions and services for creation and manipulation of information related to the structural system and objects under work.

Building codes are immense repositories of knowledge and experience which ought to be incorporated in the computer-based engineering information environment. Research in the area of representation and processing of standards has been in progress for two decades [2], and general models, methods, and techniques for computerized formulation and treatment of standards have been proposed and prototyped [3,4,5]. The practical effect on the formulation of standards is unfortunately moderate; revised and even new standards are still old-fashioned, and poorly suited for use with computers. In order to keep up with the increasing demands for efficient programming of standards, an approach based on a functional specification could be feasible.

2. A PRAGMATIC APPROACH: REQUIREMENT SPECIFICATION

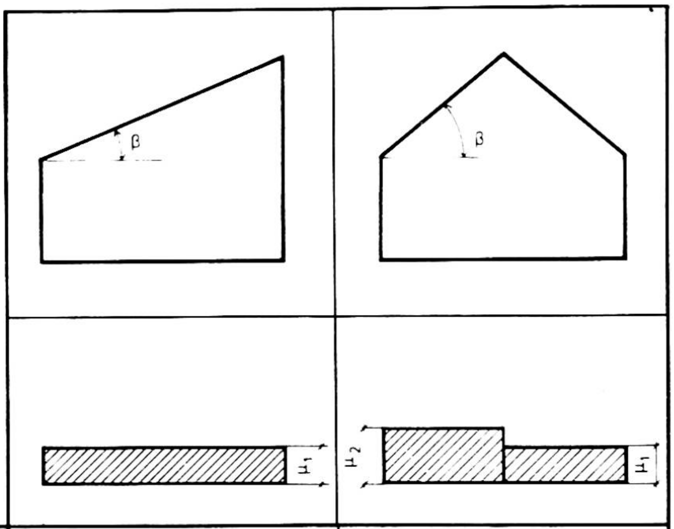
2.1 Design of Building Structures - Snow Loads

The Norwegian Standard NS 3479 "Design of Building Structures - Design Loads" [6] is well suited for manual use, but requires both engineering judgement and adjustments when applied to a particular structure, and is hence a poor candidate for automated computations. A commitee appointed by the Norwegian Council for Building Standardization (NBR) has worked out a recommendation for development of software for computation of snow loading on roofs [7]. The aim of this work has been to provide software developers with an efficient and sound basis for implementation of the requirements of the standard. The target readership of the recommendation is programmers, which differs from the users of the standard.

2.2 A general numerical formulation

NS 3479 identifies seven typical roof profiles, mono-pitch, duo-pitch, arch, and so on. For each roof profile, the form factors for snow loading are defined and referenced to the geometry of the profile, but the profile types have individual reference frames. Fig. 1 illustrates the manner of form factor definition. A computer implementation based on this definition would require entirely different data representations for each profile type, and individual or manual mechanisms for linking of snow load to the structural model. A common reference model for geometry and form factors is necessary, preferably with a rather simple link to the structural system.

The proposed, common geometric model for different roof profiles is illustrated in Fig. 2. The roof points are referenced to a global coordinate system, and the roof segments are connected to neighbouring points. The monotonically increasing numbering sequence of points and segments provides an implicit topology of the system (segment numbers are encircled in the figure). Segments may be furnished with outward ends, to model eaves (not shown on figure). Wall segments may be included, to allow the same model be used for wind loading. To each segment is attached a local coordinate system for definition of form functions. A form function is assumed to vary linearly between the form points, which are always located at the ends of a segment and optionally in between (Fig. 2).



$0° \leq \beta \leq 15°$	$\mu_1 = 0,8$	$\mu_2 = \mu_1 = 0,8$
$15° < \beta \leq 30°$		$\mu_2 = 0,8 + 0,4 \frac{\beta - 15}{15}$ $\mu_1 = 0,8$
$30° < \beta < 60°$	$\mu_1 = 0,8 \frac{60 - \beta}{30}$	$\mu_2 = 1,2 \frac{60 - \beta}{30}$ $\mu_1 = 0,8 \frac{60 - \beta}{30}$
$\beta \geq 60°$	$\mu_1 = 0$	$\mu_2 = \mu_1 = 0$

Fig. 1 Form factors (μ) for mono-pitch and duo-pitch roofs [6].

2.3 Basic load case and derived load case

A basic load case for a roof profile consists of the corresponding maximum loads for all roof segments. A particular roof profile has a prescribed number of basic load cases. The form functions for a particular roof segment is defined by the same number of form points in all the basic load cases.

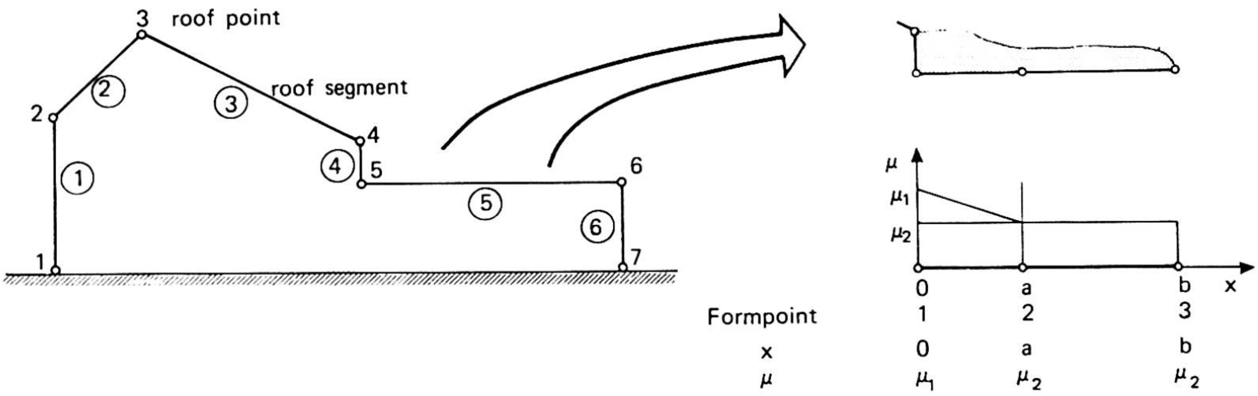
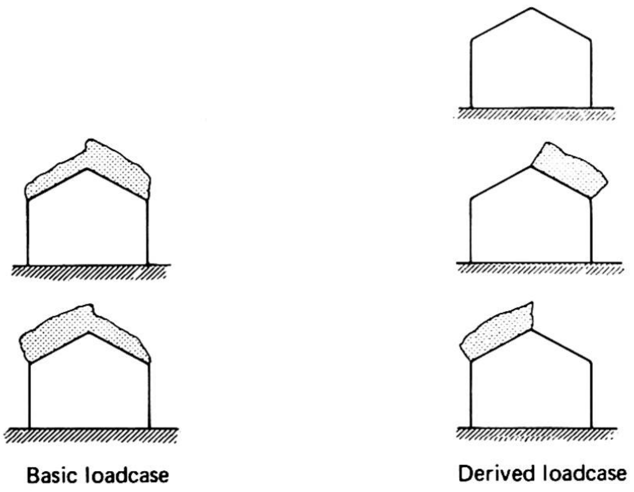


Fig. 2 Geometric model for roof profile and form function for roof segment



A derived load case represent a reduction of load on one or more segments or segment parts (Fig. 3). The concept of derived load case provides a mechanism for generating any number of load cases, as prescribed by NS 3479, cf section 3.2. This mechanism is, however, not included in the general numerical formulation.

Fig. 3 Basic and derived load cases

2.5 Functional specification

Based on the general numerical formulation for profile geometry, form factors and load cases, a software package for computation of snow loads is specified. The specification comprises a modular structure (Fig. 4), a data structure, and interfaces for the individual modules.

The package is linked to an application program by a single, main routine which administrates the different roof profile routines. Input data to the main routine is geometric data description, profile type identifier, characteristic snow load and some control parameters. Output is form point coordinates, form factors for all for points and all basic load cases, and a status variable (error flag). The main routine is called once for each roof segment. The specification does not define how the application program provides or exploites data.

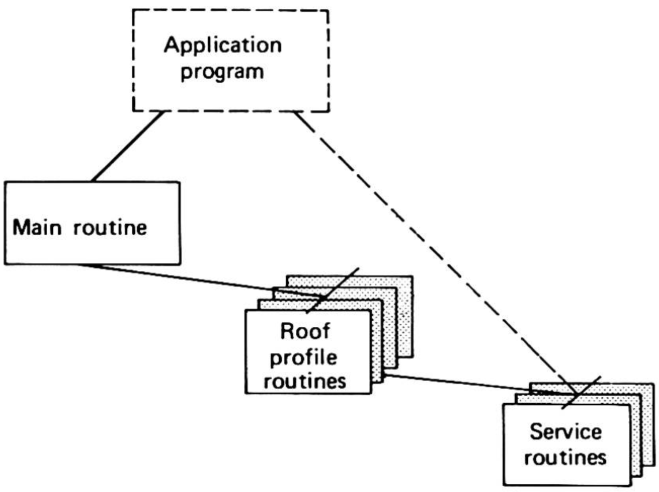


Fig. 4 Module structure for software package

The interface specification for each module defines purpose, input data, output data and error handling. The specification is independent of programming language, and may easily be converted to e.g. Pascal or FORTRAN.

The service routines offer computation of some control variables, form function value in an arbitrary position along a roof segment, error messages, etc. These routines may optionally be invoked from the application program.

The functional specification provides a common basis for programming of the standard's provisions, and the single-routine interface will make application programs far less sensitive to changes in the standard than when "hard-coded" into the program.

2.6 Algorithmic description

The programming of a routine package according to the requirement specification implies the interpretation and translation of the verbal, formulae and graphical information of the standard (Fig. 1) into detailed instructions for the computer. This is a tedious and error-prone task, and in order to relieve the implementators of much distress and reduce the risk of misinterpretation, an algorithmic description was worked out and presented as pseudo-code based on the principles of structured programming.

3. SOME PERTINENT PROBLEMS

The draft recommendation for computer implementation of the snow load part of NS 3479 comprises some 40 pages. The provisions of NS 3479 occupies 7 pages, and the work with detailing of the contents revealed a lot of principal and practical problems related to the formulation of the standard. Some examples may illustrate the incompatibilities between traditional standards and computers.

3.1 Completeness

NS 3479 recognizes seven distinct roof profiles or types. Other types are not covered, nor are combinations of the basic profiles. The problem space of the real world is infinite and continous, while the solution space defined by the standard is finite and discrete. Hence a mapping is needed, but no mapping function or procedure is prescribed.

A basic feature of computers is the generality; a single program may be applied to any problem within its scope by defining the problem in terms of appropriate data describing the problem. With NS 3479 it is difficult to utilize this property. The problem must be manually mapped onto one of the recognized profiles, which in most cases is significantly different from the structural model used for analysis where the load is to be applied.

3.2 Uniqueness

NS 3479 prescribes that under certain circumstances, the roof shall be checked for snow load on any part whatsoever of the roof with no snow load anywhere else on the roof. In principle, this clause gives rise to an infinite number of load cases. For a particular roof, the application of this rule depends on factors as covering material, heat penetration, snow catchers, snow clearance, sub-structure, etc. The solution space prescribed by the standard is in principle infinite and continous, even if the standard's problem space is finite and discrete. Again, the mapping function is missing, and left to the user.

3.3 Correctness

It is evident that when the properties of completeness and uniqueness are missing in a standard, correctness is hard to obtain. Correctness is closely related to the meaning, the result intended by the standard writers. When this meaning is not completely and uniquely expressed, the result is hard to predict and incorrect use is probable.

When converting the provisions of a standard into computer code, another source of error is introduced, namely software errors. Quality assurance and validation of software becomes an important area, which standard writers could consider. providing not only methods and procedures, but also solutions to a standard set of problems.

4. THE CHALLENGES

The hostile relations between building codes and computers will inevitably lead to severe problems. Developers of engineering software are torn between the vast potential of information technology and the old-fashioned standards. If not bridged, this gap will undermine the authority and reliance of established codes and practises.

There are two major problem areas; first, the formal representation of standards and next, the computer implementation of a standard in a vast number of computer programs. Models of design standards exist, but are not perceived by code writers. One such model proposed by Fenves et al [3] consisting of four components (data items, decision tables, information network, organizational system) provides a framework for the representation of certain standards. Lacking expertise among code writers may be supplied by computer-based tools for analysis and synthesis of a code with regards to the formal requisites, like the support system for the Australian Model Uniform Building Code [4].

The solution to the implementation problem might be a generic standards processor [2,5], which treats the standard as data instead of coded instructions, and can be used to link a structural design program with any specific standard. This approach is well suited for expert system technology, which unfortunelately is still unmature and lacks standardisation like good, old FORTRAN!

The functional specification approach for algorithmic programming as described in this paper, is easily followed if the standard is formulated with computer implementation in mind.

Remoulding of existing standards into a fairly computer-compatible form seems to be feasible to day by employment of knowledge which is available to the engineering society. In some areas like the generic standards processing, links between knowledge-based systems and databases or algorithmic software, a lot of research is still necessary to establish models and methods which are convenient for use in standards.

The major challenge of today should be to apply what we already know. The major challenge for the future should be to upgrade the education (and reeducation) of structural engineers to also be masters of the information technology.

REFERENCES

1. Scientific American, Vol. 257, No. 4, October 1987.
2. GARRET J. H., FENVES S.: Summary of the Research in the Area of Representation and Processing of Standards. Carnegie-Mellon University, Dept. of Civil Eng., Pittsburg PA, 1984.
3. GARRET J. H., FENVES S.: A Knowledge-Based Standards Processor for Structural Component Design, Engineering with Computers 2, 219-238 (1987).
4. AMUBC System. Final Report. Dept. of Architectural Science, University of Sydney, 1984.
5. LOPEZ L. A., ELAM, S.L.: SICAD User's Manual. Technical Report, University of Illinois, 1985.
6. NS 3479 Design of Building Structures. Design Loads. Norwegian Standards Association, Oslo, 2nd edition, February 1981.
7. Recommendation for NS 3479. Instruction for development of computer programs for calculations of snow loading according to NS 3479. The Norwegian Council for Building Standardisation, Oslo, to appear in 1988. (In Norwegian).