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

**Band:** 13 (1988)

**Artikel:** Informatics in civil engineering education

**Autor:** Anderheggen, Edoardo / Prater, Edward Gerald

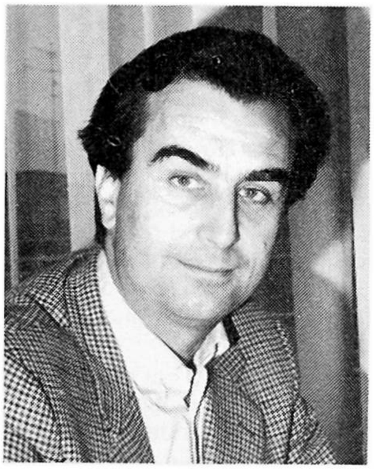
**DOI:** https://doi.org/10.5169/seals-12995

# Informatics in Civil Engineering Education

Informatique dans l'enseignement du génie civil

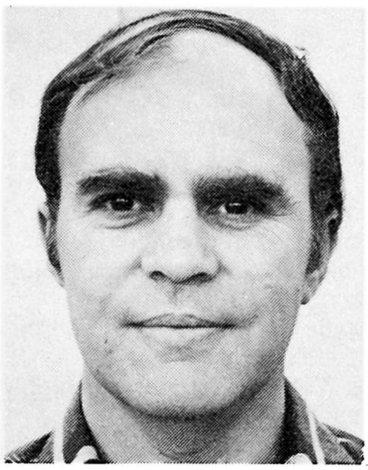
Informatik in der Ausbildung von Bauingenieuren

**Edoardo ANDERHEGGEN**
Professor of Informatics
Swiss Federal Inst. of Tech.
Zurich, Switzerland



E. Anderheggen, born 1939, obtained his Dipl. Ing. and Dr. Ing. in Civil Eng. at the ETH-Zurich. Since 1976 Professor of Applied Informatics, teaching basic informatics and finite element analysis.

**Edward Gerald PRATER**
Senior Research Engineer
Swiss Federal Inst. of Tech.
Zurich, Switzerland



E. G. Prater, born 1941, graduate and Ph.D. of Univ. Wales, Swansea. Lectured in Soil Mechs. at Univ. Sheffield 1966–68; since 1971 at ETH-Z: research in geotechnical earthquake eng. and since 1986 at the Institute of Informatics.

**SUMMARY**
The authors trace briefly the developments in informatics at the Federal Institute of Technology, Zurich, over the past 30 years. In particular, the aims, necessity and contents of introductory courses in informatics for civil engineering students are discussed, together with the rôle of computer programs and CAD in teaching structural analysis.

**RÉSUMÉ**
L'article résume les principaux développements de l'informatique à l'Ecole Polytechnique Fédérale de Zurich au cours des 30 dernières années. Il discute ensuite les problèmes concernant l'enseignement de l'informatique aux étudiants du génie civil soit au niveau de base, soit en relation avec la statique des constructions.

**ZUSAMMENFASSUNG**
Die Autoren zeigen kurz die Entwicklung der Informatik während der letzten 30 Jahre an der Eidgenössischen Technischen Hochschule, Zürich. Insbesondere werden die Ziele, die Notwendigkeit und der Inhalt eines Einführungskurses in der Informatik für Studenten der Bauingenieurabteilung, sowie der Stellenwert von Computer- bzw. CAD-Programmen in der Baustatiklehre diskutiert.

## 1. INTRODUCTION

There is little doubt today that the developments in computer hardware and software over the last 30 years have given birth to a new epoch in the history of our civilisation. With a computer anyone can "build his own machine" [1]. In fact, simply by writing a program, which is essentially a text, one is able to transform a general-purpose computer into a special-purpose machine capable of solving a specific problem. All that is needed is logical thinking and a knowledge of an easy-to-learn formal programming language. The computer itself takes care of the rest. This new way of building machines has proved to be extremely well suited to man's basic natural abilities and, in many fields, much more flexible and efficient than any other problem solving procedures previously devised. As we all know, this basic capability has led to the spectacular development of countless computer applications which directly influence many aspects of our professional and private life.

It is obvious that technical education at the university level must take into account these developments. The point to be stressed, however, is that because of the fundamental significance of informatics - whose historical importance one dares to compare with that of the differential and integral calculus developed in the 17th and 18th centuries - students at university level must understand not only *what* a computer can do, but also *why* it can do it. This is certainly also true in civil engineering: it is not enough to demonstrate all the new possibilities offered by today's computers and assess their impact on the way specific civil engineering problems are solved. Because of the broader scope of higher education, civil engineering students must understand at least the most fundamental aspect of informatics, namely that computers are programmable. Therefore, although students are not intended to work as professional programmers, they must understand what programming involves independent of any practical application.

With these general thoughts in mind a "case study" is discussed in the present paper. It concerns the development of informatics at the Swiss Federal Institute of Technology (ETH) in Zurich and the way informatics is taught in the department of civil engineering. Some questions concerning the teaching of computer-based methods in structural engineering will also be discussed.

## 2. INFORMATICS AT THE SWISS FEDERAL INSTITUTE OF TECHNOLOGY, ZURICH

In the field of informatics our university can look back on a rather illustrious past. Without going into detail some historical highlights might be worth mentioning:

- already in 1953 an electrical relais computer manufactured during the war by the German civil engineer K. Zuse was installed at the ETH.
- the ERMETH computer (with vacuum tube hardware, drum-storage, no operating system or compilers, etc.) was built at our university and operated from 1955 to 1964
- the professors of applied mathematics E. Stiefel and H. Rütishauser greatly contributed to the development of the programmimg language ALGOL60
- in 1971 a large computing centre with hardware costing some 30 Mio. Swiss Francs (in 1971) was installed in a new, specially designed building.
- in the years 1970-73 Professor N. Wirth developed the highly successful programming language PASCAL, to be followed more recently by MODULA-2.
- in the early seventies introductory courses on computer applications and programming were introduced in most engineering departments, including civil engineering.
- in autumn 1981 a separate department of informatics was established. After four and half years of study the first students of our university graduated in informatics in early 1986 (i.e. some 10 years later than in many other European and American universities)
- in spring 1984 the Swiss parliament approved a special funding for providing the Federal Institutes of Technology in Zurich and Lausanne with some 3000 personal computers for students (approximately 1 personal computer for 5 students in all disciplines) in the next 5 years. Today approximately 450 personal computers for students (or about 1/4 of the planned number) have already been installed at the ETH in Zurich.
- of course, many more large and small computers are installed in the institutes where they are used for research purposes.

These few remarks show that a great interest in computers and informatics has always existed both within our university and outside of it. However, the establishment of a new department of informatics still met with opposition and was long delayed. This was due not only to bureaucratic reasons or to the fact that a new department obviously meant cutbacks in other departments. But even as recent as 5 or 6 years ago, a considerable number of people could not understand why a nation like Switzerland with no computer industry of its own needed a new breed of specially educated professional computer experts. It was argued that civil or mechanical engineers, who are the only ones who know how to build structures or machines, should also be the ones to write the programs needed in their professions. There was little appreciation of the fact that the skills required for devising computer based solutions are not only quite difficult to acquire but also very different from the skills required of a civil or mechanical engineer. The differences between operating software and application software were not always clear to everyone. There was even some confusion between numerical mathematics and informatics. Six years after the very successful establishment of the new department of informatics (which today has twice as many students as civil engineering) this kind of discussion is no longer encountered. Informatics is now well established as an independent field of applied science and the general appreciation of some of the problems mentioned above has certainly improved. There is also a broad consensus of opinion that students in all fields of engineering should take some introductory courses in informatics. However, there is still room for improvement in the syllabus of these courses for engineers, including the civil engineering department, in which the authors have direct involvement.

## 3. BASIC EDUCATION IN INFORMATICS FOR CIVIL ENGINEERING STUDENTS

At the ETH in Zurich today's first semester civil engineering students are given an introductory course in informatics comprising 2 hours of lectures and 2 hours of exercises each week over a period of 16 weeks. For these exercises (and also for word processing and other applications, including playing games) the students have free access to a number of Apple Macintosh personal computers (about 1 PC to 5 students). The exercises consist of small Pascal programs or parts of programs to be written and tested by the students using the MacPascal interpreter.

The fundamental concepts of the programming language Pascal are discussed in the lecture and a number of typical algorithms and data structures are explained and demonstrated. Some additional topics concerning hardware, operating systems, different programming languages, networks, etc. are also briefly discussed. It should be stressed, however, that the aim of this lecture is not really to teach programming. In fact, it is realised that the great majority of civil engineers do not need to know how to program. The basic aim of this lecture was explained above: like mathematics or physics, informatics is a fundamental science, which has to be taught to all students of all technical disciplines at a preparatory level and independent of specific applications. Because programming is certainly the most important aspect of informatics, students must understand what it really involves. Later, in professional life, this basic knowledge will be essential for assessing the feasibility of computer based procedures, which in civil engineering, like in most technical disciplines, are bound to have a profound impact on our day-to-day professional work.

This also explains why the programming language Pascal was chosen for this course: because we do not want to teach programming at a professional level, it does not matter if the language students learn is not the language they are most likely to encounter in practice. What really matters is that they learn the basic concepts of programming and Pascal is certainly ideally suited for this purpose. In some other departments of the ETH and also at the University of Zurich the programming language Modula-2 has been used in similar lectures instead of Pascal over the past 2 years. The reason for this is that, although Modula-2 is little used in practice, it represents a natural evolution of Pascal and is considered to be better for understanding modular programming techniques and state-of-the-art software engineering procedures. Again what matters are the fundamentals of informatics and not the practical ability to write professional programs. Other popular programming languages are considered to be inadequate for this purpose: Basic is too simplistic; Fortran, which suffers from an uncoordinated development over a 30 year period, has too many artificial conventions and insufficient built-in checks; the Unix language C is too cryptic for non-professional programmers, etc.

In many fields of civil engineering computers may be regarded as basic professional tools. Therefore, the question arises whether, in addition to the more fundamental aspects of informatics mentioned above, a general introduction to the practical use of computers should also be offered to our students in their first year of study. The answer to this question is not so obvious because such an introduction would have to be, to a large extent, application independent. Furthermore, computers are certainly difficult to program in a proper way, but very easy to use as soon as one understands the problem to be solved and a well-written application program is available. Of course, understanding the specific civil engineering problems to be solved by computer-based methods as well as the choice of the corresponding software cannot be part of an introductory course. Nevertheless, there are a number of important applications which are not specific to any of the classical civil engineering disciplines, but should at least be briefly explained and demonstrated to our students. These include word and document processing, use of spreadsheets, computer-aided drawing, setup and use of databases, mail and data transmission via local and wide area networks, etc. In fact, all these applications may actually turn out to be important working tools and also serve as an excellent introduction to more specific computer applications encountered in later years of study.

Our civil engineering students have to take the mandatory course mentioned above and solve the corresponding exercises. However, they do not have to pass a corresponding examination. We consider this to be a serious deficiency of our curriculum. Like mathematics or physics, informatics is a fundamental discipline made up of a number of relatively difficult concepts, which can only be understood and assimilated by hard work on the part of the individual. Experience shows (and common sense tells) that students will not be willing to invest the necessary effort in a field which, though it looks interesting and is certainly recognised as important, yet is not treated as such within the curriculum leading to their diploma.

## 4. COMPUTERS IN STRUCTURAL ENGINEERING EDUCATION

After the introductory courses in the early stage of their studies students have to learn the specifics of their future profession. It would be far beyond the scope of this paper to list and comment on all computer applications in civil engineering. Therefore, as an example, only some thoughts concerning structural engineering will be briefly discussed.

Structural analysis is the field of civil engineering in which computer based methods (above all the finite element method) had the earliest and probably greatest impact. The way displacements, forces, moments and stresses are computed today bears, in most circumstances, no resemblance to the procedures used 20 or 30 years ago. Of course, these developments, and above all the finite element method, should be taken into due account when teaching structural analysis. Old fashioned hand methods such as the Hardy-Cross method or the use of Fourier series for plate bending analysis definitely no longer belong in the structural engineering syllabus. There is, however, a more important point to be made.

The spectacular advances we have witnessed in structural analysis also have some important drawbacks. In the (good) old days the engineer who had the responsibility of designing a structure generally also defined the mathematical model needed for the analysis and did the corresponding numerical calculations himself (or at least he always knew exactly how to do them). Today structural analysis is often done by "black-box" programs whose internal workings and the sophisticated mathematical models used for simulating reality are not always completely understood by their users. In fact, due to the sophistication of today's structural analysis programs, it has become quite difficult to be at the same time a good designer and a good numerical analyst. This, of course, leads to problems. In some cases designers lose all confidence in the programs they use, and sometimes they even go back to hand calculations. Therefore, when teaching structural engineering the problem of combining modern analysis methods with sound design concepts deserves (at least at undergraduate level) much attention. It is certainly necessary that students gain direct experience in the use of analysis programs when solving practical design problems. These programs should preferably run on user-friendly personal computers and be as simple as possible (i.e. as less general as possible) in order to demonstrate the essentials of the algorithms they use with a minimun of unnecessary overheads. Programs specially written for teaching purposes, although they might not be those one would use in practice, can certainly be very useful.

With this goal in mind a number of programs have been written either strictly for teaching purposes or expressly so as to optimise the understanding of their internal workings (which is also a very welcome feature among practicing engineers). The following programs are used today at the ETH for teaching structural analysis:

SMIS : an iteractive MS-DOS Pascal program for simple matrix manipulations, based an old batch program developed at the University of California, Berkeley. This program is used for teaching the fundamentals of static, dynamic and stability analysis by the finite element method.

MacBeams : A highly interactive Modula-2 program for the MacIntosh with mouse input and graphic output illustrating displacement and moment distribution of plane frame structures (see Fig. 1)

STATIK: a simple yet very general linear frame analysis program including calculation of cross-sectional properties and prestressing. This program is now widely used both in practice and at our university for solving relatively simple day-to-day design problems.

FLOWERS : a large, very general finite element program for static and dynamic, linear and nonlinear analysis of many kinds of structures. This program was also written with teaching in mind (see [2]) and is regularly used for teaching purposes.

Finally, it must be clearly stated that computers have certainly had a great impact on analysis but, so far, very little on conceptional design, which, of course, is what structural engineering students are really supposed to learn. CAD-systems (where "CAD" in most cases stands for "Computer Aided Drawing" possibly with automatic storage and retrievial of graphical and other design data) just like other procedures known under the label of "office automation" will no doubt improve the working efficiency of structural engineering offices considerably. Few people believe, however, that CAD-systems lead or will lead in the near future to conceptually better structures. Therefore, the question of how (or if) to teach CAD as an additionai subject (i.e. in addition to computer-oriented analysis and conceptual design) is still debatable. Probably, as suggested above, it might make sense to explain and demonstrate some well established, simple CAD procedures as a typical example of computer applications in a general, application-oriented basic informatics course in the first year of study.

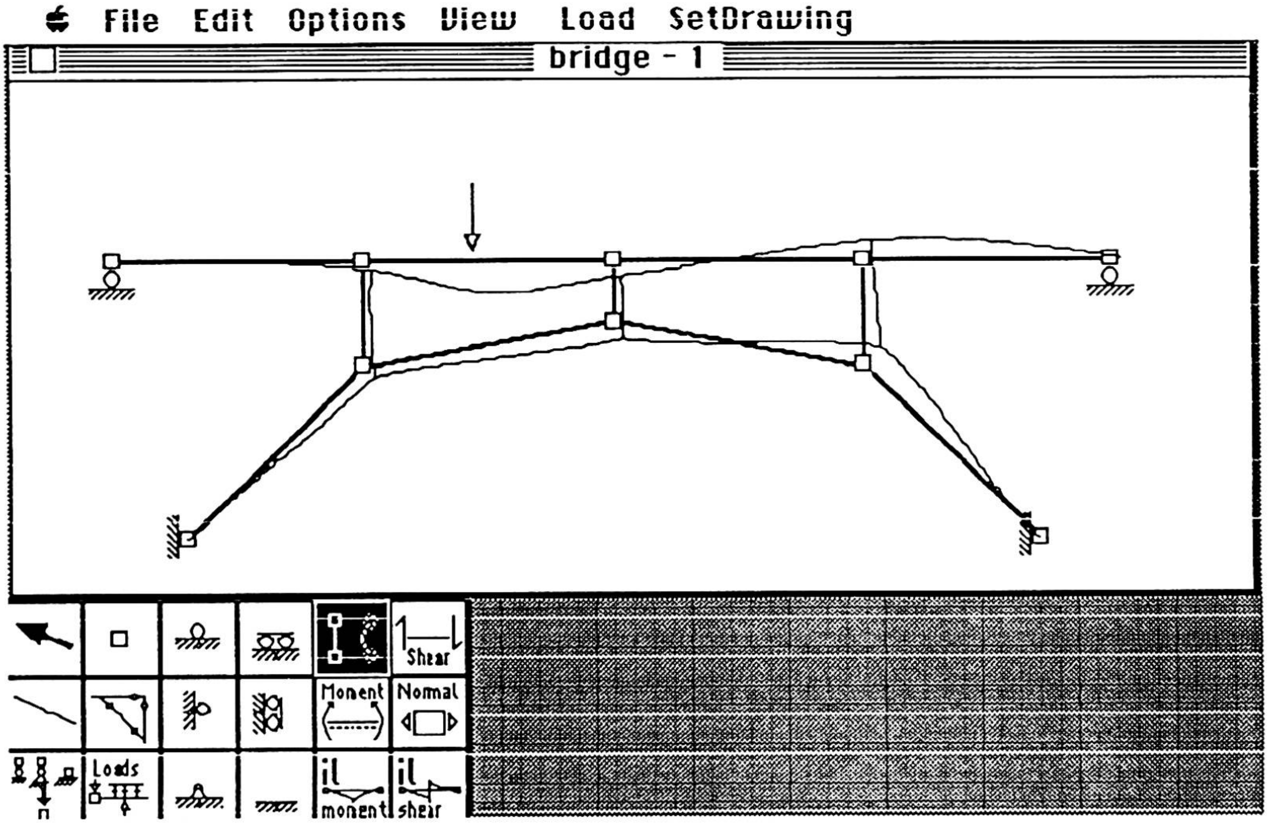



Fig.1 Screen dump of MacBeams Program showing menus, structure and deformed shape

## 5. CONCLUSIONS

This paper outlines some thoughts concerning the teaching of informatics-oriented subjects in our civil engineering department, firstly at preparatory level and then in the field of structural engineering. Obviously, the ideas presented reflect both personal experiences and the special situation of our university. Nevertheless, while computers are rapidly changing our lifes, universities have to be somewhat conservative in order to provide their students first and foremost with the well-established fundamentals of their chosen profession. Because of these conflicting aims it is felt that some of the problems discussed above must also arise, in similar forms, in other universities. It is the scope of the paper to provide a contribution to the discussion of education related problems, which certainly also ought to be of concern to practicing civil engineers.

## REFERENCES

[1] N. Wirth: "Data Structures and Algorithms", Scientific American, Sept.1984, Vol. 251 No.3, pp.48-57

[2] E. Anderheggen: "Nonlinear Structural Analysis: the Glass-Box Approach", Proceedings of the 12th congress of the IABSE in Vancouver, Canada, Sept. 3-7, 1984, pp. 455-462.

Leere Seite
Blank page
Page vide