TITLE: Computing in the Civil Engineering Curriculum: Needs and Issues

KEY WORDS: Academic Survey; CAD; CAE; Civil Engineering Curriculum; Computer Hardware; Computer Software; Computing Curriculum; Computing Survey; Computing Tools; Education; Education Committee; Engineering Courses; Professional Survey; Pros and Cons of Computing Courses; Task Committee on Computing Curriculum;

ABSTRACT: The rapid advances occurring in computer software and hardware have provided the engineers with a powerful means of processing, storing, retrieving, and displaying data. This has made computer science a growing and essential part of nearly every engineering discipline.

The effective use of computers in engineering processes and applications is recognized by many as the key to increased individual, company, and national productivity. This paper presents the results of a survey that was conducted by the ASCE Task Committee on Computing Education to assess the computing needs in the civil engineering profession and to assess the current undergraduate computing curriculum in civil engineering education.

The survey results presented in this paper can be used as a guide by educators who wish to enhance the computing aspects of their engineering curricula and by engineering firms that are evaluating the educational profile of new graduates. The goal is to enable students to possess, in addition to their engineering skills, both the ability to evaluate and use production software and the ability to organize and supervise the development of software.

REPRINT SUMMARY: Computing in the Civil Engineering Curriculum: Needs and Issues, by Kincho Law, William Rasdorf, Mohammad Karamouz, and Osama Abudayyeh. The history of the ASCE Education Task Committee on Computing Curriculum is provided. The results of a survey to assess computing educational needs are presented along with a commentary on the results. Different scenarios for incorporating computing into the civil engineering curriculum are discussed. A compilation of general comments from both surveys is also provided.

REFERENCE: Kincho Law, William Rasdorf, Mohammad Karamouz, and Osama Abudayyeh, "Computing in the Civil Engineering Curriculum: Needs and Issues," *Journal of Professional Issues in Engineering,* ASCE, Accepted for publication.

Contents

Computing in the Civil Engineering Curriculum: Needs and Issues

Kincho H. Law, [1] Member, ASCE
William J. Rasdorf, [2] Member, ASCE
Mohammad Karamouz, [3]. Member, ASCE
and
Osama Y. Abudayyeh, [4] Student Member, ASCE

ABSTRACT

The rapid advances occurring in computer software and hardware have provided the engineers with a powerful means of processing, storing, retrieving, and displaying data. This has made computer science a growing and essential part of nearly every engineering discipline.

The effective use of computers in engineering processes and applications is recognized by many as the key to increased individual, company, and national productivity. This paper presents the results of a survey that was conducted by the ASCE Task Committee on Computing Education to assess the computing needs in the civil engineering profession and to assess the current undergraduate computing curriculum in civil engineering education.

The survey results presented in this paper can be used as a guide by educators who wish to enhance the computing aspects of their engineering curricula and by engineering firms that are evaluating the educational profile of new graduates. The goal is to enable students to possess, in addition to their engineering skills, both the ability to evaluate and use production software and the ability to organize and supervise the development of software.

1 Introduction

In the future, an integrated combination of computer-aided analysis and design tools and techniques will be developed for all types of engineering design problems. This will require applying computer science principles and practices to a variety of engineering systems in order to determine their response to external influences. The implications of this

[1] Assistant Professor, Stanford University, Civil Engineering Department, Stanford, CA 94305-4020, (415) 725-3154

[2] Associate Professor, North Carolina State University, Civil Engineering and Computer Science Departments, Raleigh, NC 27695, (919) 737-2333

[3] Associate Professor, Pratt Institute, Civil Engineering Department, 200 Willonghby Avenue, Brooklyn, NY 11205, (718) 636-3436

[4] PhD Candidate, North Carolina State University, Civil Engineering Department, Raleigh, NC 27695, (919) 737-2333

requirement for the academic community are clear: we must prepare our students to use computer methods and applications as a part of their fundamental engineering education [Rasdorf84]. Computing should be seen as an integral part of civil engineering education. It should not continue to be a curriculum option or specialization, but should be taught as a central medium for communication, analysis, and synthesis throughout the engineering process.

One of the dilemmas in engineering education today is how future engineers can best assimilate the advanced, yet fundamental, knowledge of computing technologies appropriate for their professional engineering career [Comfort81, Rasdorf87, Rehak83]. This paper suggests that the role of the academic community must be to prepare engineering students to use computer methods and applications as a part of their fundamental civil engineering education. It is the responsibility of universities to incorporate contemporary computing fundamentals into their academic curriculum to improve the professional qualifications of their engineering graduates, since it has been well recognized that the traditional introductory programming courses are no longer sufficient for enabling students to become familiar with current computing technologies.

Considering the above observations, the American Society of Civil Engineers' Education Committee of the Technical Council on Computing Practices proposed and started a study on computing education requirements for the civil engineering curriculum. One of the tasks in this study was to assess the current teaching practices and course offerings in civil engineering departments of various universities and to assess the educational requirements, relative to computing, of professional engineering firms. The task committee designed a survey and sent it to both academicians and practicing engineers to solicit responses. The responses to the survey have been accumulated and analyzed. This paper presents and discusses those results and proposes a few guidelines to incorporate computing into the civil engineering curriculum.

The paper is organized as follows: Section 2 provides a discussion of the history of the ASCE Education Task Committee on Computing Curriculum and on the survey conducted; Section 3 lists the results of the survey in tabular and graphical forms; Section 4 provides a commentary on the results; Section 5 discusses different scenarios for incorporating computing into the civil engineering curriculum; and, Section 6 provides a summary. Appendix A is a compilation of general comments that were accumulated from both surveys. The surveys themselves are provided as Appendices B and C.

2 Background

Adequate computer resources have been a subject of concern in engineering education. In 1986, the education committee of the Technical Council on Computer Practices conducted a survey to determine the availability of computing resources in civil engineering departments and to determine the attitude of faculty towards computing in civil engineering education [CE87, Comput87]. The committee received over 93 responses from the civil engineering departments and 550 responses from faculty across the country.

The survey indicated that while almost all universities require a programming course, most civil engineering departments do not require, nor do they have, any additional computing courses beyond the first programming language course. The faculty responding to the survey strongly agreed that some type of a computing technology course should be a fundamental course within a civil engineering degree granting program. Furthermore, the faculty strongly agreed that the civil engineering curriculum should allocate course time to train students to use computers even if the result is an increase of 3-6 hours in the overall number of courses required to complete the undergraduate civil engineering degree. Three specific areas were identified by the task committee that civil engineering students need to be exposed to:

1. The technology of computers - how they work and how to program them.
2. Computers as problem solving tools - how to use spreadsheets, graphics, CAD, databases, etc. to solve engineering problems.
3. Computers as engineering simulators - how to use software to illustrate and evaluate a variety of solutions or alternatives.

Because there is a lack of consistency in computing education to address these areas, there is clearly a need for a comprehensive review of the computing aspects of the civil engineering curriculum.

The formation of a task committee for assessing the status of the civil engineering curriculum with respect to computing was first initiated in 1987. A proposal was written, submitted, and approved by the Technical Council on Computer Practices of ASCE. The proposal called for developing guidelines and recommendations for a computing curriculum in civil engineering education. Activities involved:

1. Assessing the current computing curriculum in civil engineering education by sending out surveys to academicians (Appendix B) that are teaching computing courses in different schools across the nation;
2. Assessing the computing needs of the civil engineering profession by sending out surveys to practicing engineers (Appendix C) in both the public and private sectors in different civil engineering firms across the nation; and
3. Developing recommendations and suggesting formal guidelines for a modified civil engineering computing curriculum.

The proposal called for completion of these activities in a two-year period. This paper addresses these three issues.

3 Survey Results

To assess current computing curriculum needs in civil engineering the task committee members, represented by the authors of this paper, conducted a survey in the Spring of

1988. Two survey mailing lists were approved: one for the professional candidates and one for the academic candidates. The questionnaire was sent to each candidate and the responses were collected, tabulated, and analyzed. The results are presented here.

The results of the academic survey are tabulated in Tables 1 and 2 and displayed in Figures 1 and 2. The results of the professional survey are tabulated in Tables 3 and 4 and displayed in Figures 3 and 4. In the tables, the number of responses shown indicates how many survey responses, out of the total number of responses (given in the table header), selected the item shown. Also, the percentages shown represent the ratio of the number of responses to the total number of responses (given in the table header).

Figure 1 is a bar chart of the prioritization of the different subject areas in civil engineering that were provided by the participants of the academic survey. Table 1 lists the titles of the subject areas displayed in Figure 1. Figure 2 is a bar chart of the number of computing courses offered at the universities represented by the academic survey participants. Table 2 lists the actual number of computing courses displayed in Figure 2. The "no course" percentage (36%) item in Figure 2 and Table 2 includes both those institutions where there is no computing course offering (30%) and those institutions where there are future planned courses (6%).

Figure 3 is a bar chart of the prioritization of the different subject areas in civil engineering that were provided by the participants of the professional survey. Table 3 lists the titles of the subject areas displayed in Figure 3. Figure 4 is a bar chart of the civil engineering computing application areas that were provided by the participants of the professional survey. Table 4 lists the titles of the application areas displayed in Figure 4.

The comments provided by the participants of both surveys are summarized in Appendix A. These comments have been taken into account for developing the recommendations and suggested guidelines for a modified civil engineering computing curriculum presented in Section 5.

Referring to Appendix A, one can conclude that there is a general consensus among academicians and practicing engineers with respect to the need for an increased emphasis on computing in existing civil engineering courses. This conclusion is based on such an indication from 42% of the responses from the academic survey and 28% of the responses from the professional survey, where these percentages were the highest on both surveys.

The second highest percentage on the academic survey was different than that of the professional survey. Academicians expressed a strong belief in the importance of acquiring good software and hardware and maintaining a balanced computing education curriculum that puts equal emphasis on computing and upon obtaining a basic understanding of the underlying civil engineering principles; practicing engineers believe that it is more important to emphasize the use of personal computer based software packages that relate to civil engineering and to teach students how to judge and evaluate computer program results.

The third highest score on the academic survey emphasized that there is a need for educating faculty members on computing concepts, techniques, and utilization. The third highest score on the professional survey was given to the fact that students must learn when it is appropriate to use computers and when it is appropriate to use other problem solving tools.

4 Survey Commentary

From Figures 1 and 3 and Tables 1 and 3 one can observe that the following subjects were the top highest five subjects that both academicians and practicing engineers believed to be the most promising and rewarding to the civil engineering profession:

1. Spreadsheets,
2. CADD (Computer Aided Design and Drafting),
3. Programming,
4. Graphics, and
5. Databases.

From the above list one may conclude that both academicians and practicing engineers realize the need for adopting computing concepts and technologies and integrating them into the civil engineering educational program. It was encouraging to note that both surveys were in almost total agreement on the prioritization of the above list, which may indicate that these subjects are in fact emerging as substantial prerequisites for civil engineering practice and research.

From Figure 2 and Table 2 one can observe that civil engineering departments clearly realize the need for adopting courses to teach computing concepts and techniques. 64% of the schools represented in the survey already offered at least one computing course and an additional 6% are planning to offer their first in the near future.

From Table 4 and Figure 4 one can conclude that practicing engineers overwhelmingly favor (76% of the participants) the design, management, analysis, and drafting application areas to the other areas. This should convey a message to academicians about which existing courses should receive the most attention and provide the largest professional payoff relative to computing.

The following guidelines are based on the comments provided in Appendix A as well as the observations made in this section. These guidelines are not intended to be complete or exhaustive but they do represent somewhat of a consensus distillation of the survey results. Participants generally suggest that colleges and universities should:

1. Emphasize computing in existing civil engineering courses by requiring the use of computers in junior and senior level courses.
2. Acquire and maintain quality hardware and software.
3. Maintain a balanced computing education curriculum that gives increased emphasis on computing concepts as well as on civil engineering principles.

4. Organize workshops for educating faculty members on computing concepts and computer use.

5. Teach students how to use computing concepts effectively and how to judge and evaluate computer program results. The best way to accomplish this is to design homework problems that require students to interact with the computer rather than to use it as a black box.

The need to take actions like those suggested here was underscored by the responses to question 3 of the survey sent to professional engineers. 40% of those responding said yes and 53% said no (7% did not respond) when asked if current graduates have received a sufficient computing education to effectively use computers in civil engineering practice. Clearly then, a majority feel that there is a need for strengthening current computing education in civil engineering.

5 Curriculum Development

Presently, the practice of computer utilization in civil engineering courses requires that the students either write their own computer programs on specific subjects, or use rigidly structured "canned" or tutorial programs supplied by their instructors. Developing even a small program (for example, to solve a system of equations or linear programs) or modifying existing programs to solve a specific problem are time consuming and require a significant amount of time on the part of both the instructors and the students. This program development approach does not provide the students with a better understanding of the nature of the actual engineering problem and, in fact, often detracts the students from learning the essential engineering concepts to the extent desired. On the other hand, most "canned" and tutorial programs that are currently in use by engineering instructors focus a disproportionate amount of the students' effort on the entry of data. Students may spend the majority of their time in preparing the data and end up spending little time understanding the problem and the solution process itself. Furthermore, most tutorial programs are not used in engineering practice.

What students need are a set of computing tools that can be used to facilitate their understanding of the engineering problem solving process and to calculate it to the extent that computers are able to do so. However, to incorporate the use of these computing tools in our undergraduate civil engineering programs and to teach students the fundamentals underlying their use requires modifications to our present curriculum. We identify three scenarios for the incorporation of these problem-solving computing tools and concepts into the civil engineering curriculum:

1. Designate one course to introduce the necessary computing tools and techniques early in the program of study, say in the sophomore year.

2. Incorporate the teaching of computing into existing civil engineering courses as needed.

3. Designate one course to introduce the application of computing to specific civil engineering problems late in the program of study, say in the senior year.

In this paper we are not concerned with recommending one specific approach over another; each choice is highly dependent on the educational environment of the host university and department. Rather, in the following discussion we briefly summarize some considerations for the selection of a particular approach.

The following lists some of the pros and cons of adding a computing course early in the program of study, somewhat as a prerequisite to other engineering courses.

1. A formal computing course would allow centralized control of the basic content or syllabus of the course. An appropriate faculty forum would then exist for discussion and review of course content and objectives.

2. Only one faculty is needed to assume responsibility for direct instruction of the computing course material.

3. The students would be able to master the tools early in their program of study and apply them in subsequent civil engineering courses.

4. Engineering courses could then be able to take advantage of the existing computing skills of the students and focus on the subjects of interest, although individual faculty must be knowledgeable of the computing needs and applications in their civil engineering subject area.

5. The students will not be aware of all of the potential applications of the computing tools and may thus lose interest in the course. (This phenomenon is similar to current problems faced by those teaching calculus and computer programming to freshman engineering students.)

6. Faculty members may resist the use of computing tools in other engineering courses. A mutual agreement among faculty members on the type of tools and techniques that must be taught is necessary in this approach.

7. Changes to the existing (and rigorous) curriculum may be needed in order to incorporate the new computing course; furthermore, existing engineering courses also need to be modified to present the problem solving concepts that utilize these computing tools.

One variation on this scenario is to introduce this course as a self paced study course. The pros and cons of doing so are essentially the same as those listed above for a more formal course. The only difference would be that instead of direct instruction, the supervising faculty member need only provides overall organization of the course structure and guidance on an as needed basis.

The advantages and disadvantages of the second scenario, that of introducing computing tools into existing engineering courses on an as needed basis, are summarized below.

1. No changes in the existing curriculum, with respect to the introduction of new course requirements, are necessary.

2. Students can clearly see the application of the computing tools to various subjects because they are taught the computing principles and the use of the tool at the time that they are needed.

3. Existing engineering courses need to be modified to accommodate/ incorporate the ever-changing computing technologies into the subject areas which are now distributed over many courses.

4. Introducing computing tools into a target course will clearly consume some amount of valuable course time, which otherwise would be used for the engineering subject itself.

5. Nearly all faculty must be knowledgeable about some aspect of computing since they are responsible for teaching the individual subjects and for insuring that the students gain fundamental knowledge in two areas.

In the third scenario, a computing course is introduced at the senior level, after the student has already gained a fundamental understanding of the civil engineering application for which computing will be used. Its purpose, in addition to enhancing the students' computing literacy, is to demonstrate the applicability of the tools in the civil engineering domain. This approach has the following potential advantages and disadvantages.

1. A formal computing course would again allow centralized control of the basic content or syllabus of the course and be subject to formal departmental review procedures.

2. No changes are needed in the other courses in the curriculum.

3. Only one faculty member is needed to assume the responsibility for the computing course; other faculty need not participate at all, nor are they required to issue computing oriented assignments within their courses.

4. Students already understand the engineering concepts; this will increase their interest in learning the computing tools and concepts because they will be able to see how they are used in various engineering computational applications.

5. There is no opportunity to apply the computing tools and techniques that are learned by the students in their other courses.

6. Faculty who would like to incorporate computing in their courses must do it independently, as in scenario two. This would take up valuable course time that could be used for the engineering subject itself and potentially introduce redundancy into the curriculum.

The decision to adopt a specific approach to incorporating computing into the civil engineering curriculum (like those enumerated above) depends in part on the availability of software and hardware resources, and in part on faculty consensus with respect to the relative importance of computing in the education of civil engineering students. For more computing-oriented schools, where computing resources are available and agreeable consensus among faculty can be reached, various combinations of the three approaches described above can be employed (for example, see Reference [Fenves88].)

6 Related Issues

The needs of the engineering profession with respect to computing are many and varied. Key among them, relative to curriculum issues, are the following:

1. Software developers of engineering application-specific software,
2. Managers of hardware and software systems in engineering environments, and
3. Users of generic software tools.

Translating these needs into a course sequence to satisfy all of them is not within the resource realm of most civil engineering departments. The current survey indicated that the top priority item was to satisfy the educational needs of the engineering computer user. Clearly this is an important emphasis area but it does not and should not preclude the others.

It should be emphasized that civil engineering education must strive for a balance between training in computing technology and generic engineering analysis and design principles. This training must not only meet the current patterns of practice and employment but also it must be concerned with anticipating and defining the civil engineering profession of the future. Care must be taken to teach fundamentals. This is particularly crucial in computing, where certain topics emerge as high priority areas for a short period of time. A curriculum that emphasizes populist topics would quickly lose its credibility. And, as in any other educational tradeoff, due consideration must be given to what it takes in terms of the curriculum to produce future civil engineering leaders in addition to producing staff engineers. The university must have a commitment to both.

In order to incorporate into the curriculum the computing technologies described herein, attention must also be focused on faculty development as stated in a 1983 report issued by the Building Research Board as follows. "The university must give serious attention to the continuing educational needs of its own faculties and the professions they serve. The advances of CAD/CAM (CAE) technology are evolving so rapidly that the professions cannot wait for a new generation of faculty to embrace the technology and to teach it with conviction and understanding. Neither can the design/build professions benefit very long from an ad hoc entrepreneurial or "self-taught" approach to CAD/CAM (CAE) [BRB83]." This comment remains a key to computing education in civil engineering.

7 Summary

This paper presented the results of a survey that was conducted by the ASCE Task Committee on Computing Education to assess the role of computing in the civil engineering curriculum. The survey sought to determine the depth of course offering in civil engineering departments of various universities, to assess the computing education specifications of educators, and to assess the computing educational requirements, relative to computing, of professional engineering firms. The motivation for the survey and the origins of the ASCE Education Task Committee were described. The results of the survey were provided and discussed. Conclusions were drawn based on the accumulated results and a commentary on the results was provided and guidelines for incorporating computing fundamentals into the civil engineering curriculum were outlined and discussed.

8 Acknowledgment

The authors would like to express their gratitude to everyone who participated in this survey for their cooperation and subjective comments. In addition, ASCE is acknowledged for supporting the task committee in its charge, for providing the resources to conduct the survey, and for providing the forum to convey the results to the civil engineering community. The former task committee on computing resources was chaired by Professor Darrell Fontane of Colorado State University, whose encouragement and suggestions were very helpful in completing the study described here.

The opinions expressed in this paper are the authors and do not necessarily reflect the views of the American Society of Civil Engineers or the affiliated institutions of the authors.

References

[BRB83] Building Research Board. (1983). Report from the Workshop on Advanced Technology for Building Design and Engineering, National Research Council, August.

[CE87] Civil Engineering Magazine. (1987). June.

[Comfort81] Comfort, W.J., et al. (1981). "Technology Transfer and Development of Computer-Aided Engineering with the University Community," Presented at the Winter Annual Meeting of the American Society of Mechanical Engineers Technology and Society Division, American Society of Mechanical Engineers, Washington, D.C., November.

[Comput87] Computing Training of Civil Engineers, Panel Session at the National Spring Convention. (1987). American Society of Civil Engineers, Atlantic City, NJ, April.

[Fenves88] Fenves, S.J., Hendrickson, C.T., Maher, M.L., Rehak, D.R., and Thewalt, C. (1988). "Two Undergraduate Courses in Computer-Aided Engineering," Engineering Education, American Society of Engineering Education, Pages 122-126, November.

[Rasdorf87] Rasdorf, W.J. and Storaasli, O.O. (1987). "Educational Fundamentals of Computer-Aided Design," International Journal of Applied Engineering Education, Volume 3, Number 3, Pages 247-254.

[Rasdorf84] Rasdorf, W.J. (1984). "Computer Programming in the Civil Engineering Curriculum," Journal of Professional Issues in Engineering, American Society of Civil Engineers, Volume 3, Number 4, Pages 141-148, October.

[Rehak83] Rehak, D.R. (1983). "The Use of Powerful Personal Computers in Engineering Education," Proceedings of the American Society of Engineering Education Annual Convention, American Society of Engineering Education, Volume 1, Pages 165-169, June.

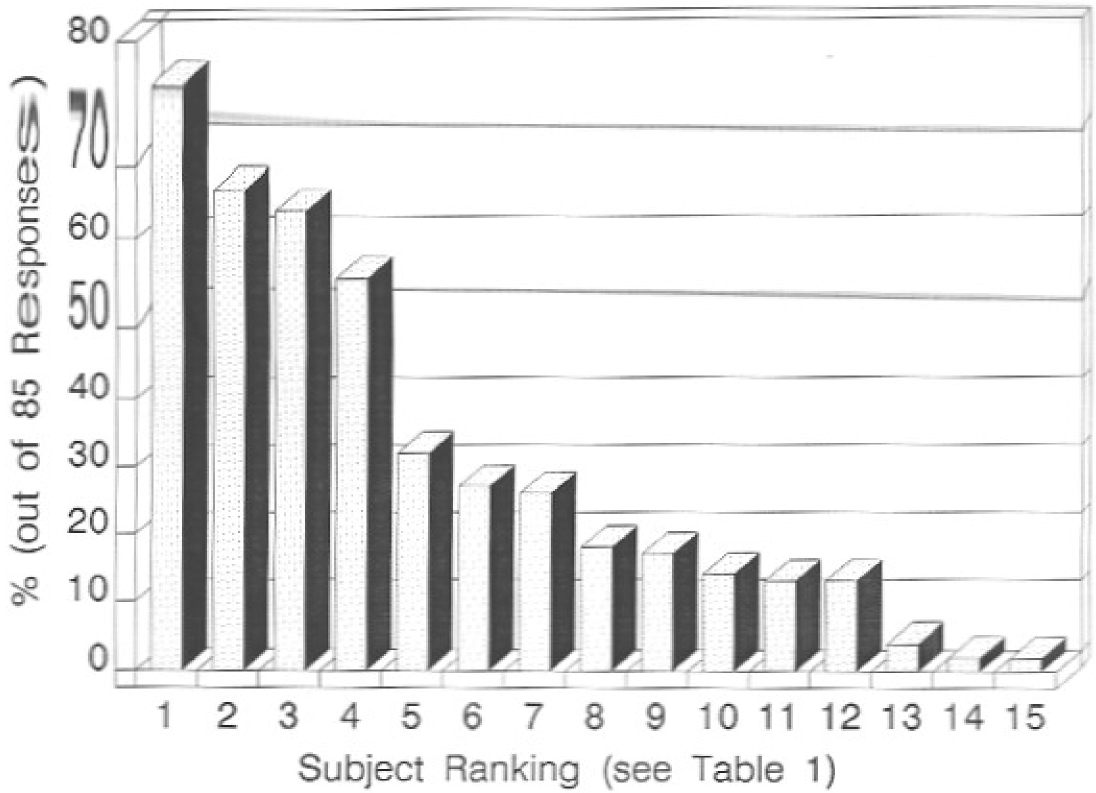

Figure 1: Academic Survey Subject Results.

Table 1: Academic Survey Subject Results (85 Responses).

Rank	Subject	Responses	%
1	Spreadsheet	63	74
2	CADD	57	67
3	Programming	55	64
4	Graphics	46	54
5	Databases	27	32
6	Wordprocessing	23	27
7	Software Packages	22	26
8	Expert Systems	15	18
9	Software Engineering	14	17
10	Geometric Modeling	12	14
11	Numerical Analysis	11	13
12	Operating Systems	11	13
13	Networking	3	4
14	Finite Element Methods	2	2
15	Simulation	2	2

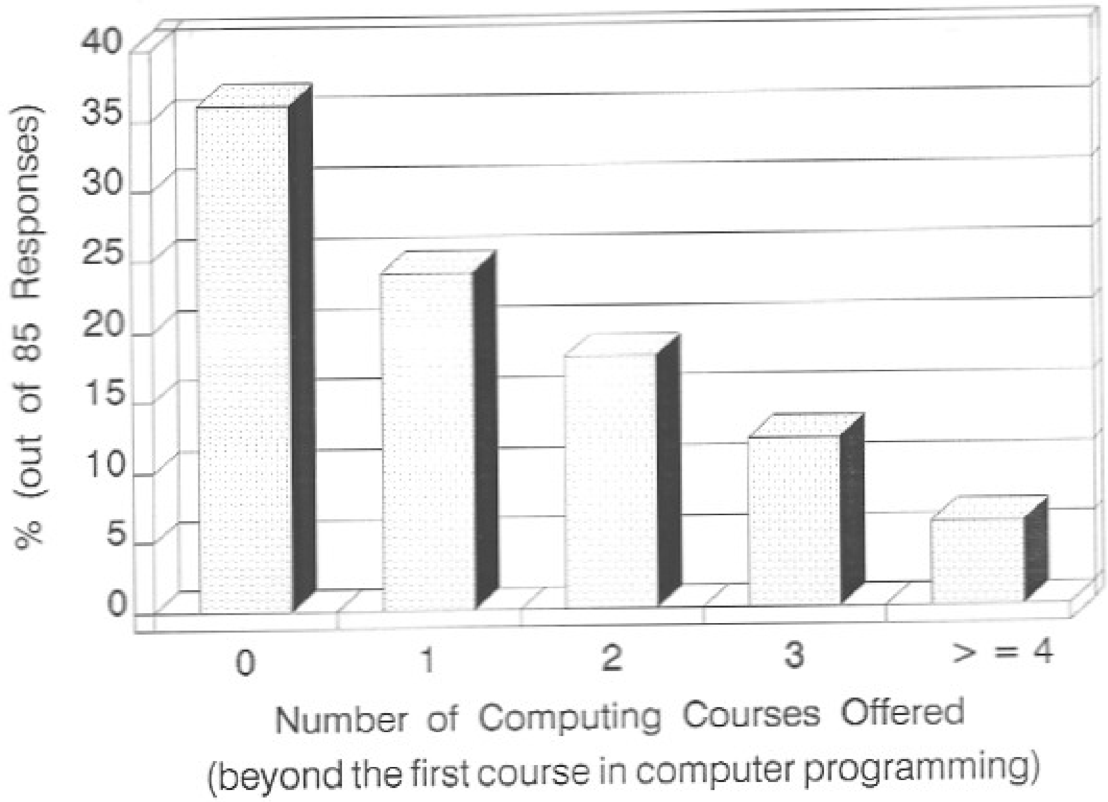

Figure 2: Academic Survey Course Offering Results.

Table 2: Academic Survey Course Offering Results (85 Responses).

Course Offering	Responses	%
no courses	31	36
1 course	20	24
2 courses	18	21
3 courses	12	14
4 or more	4	5

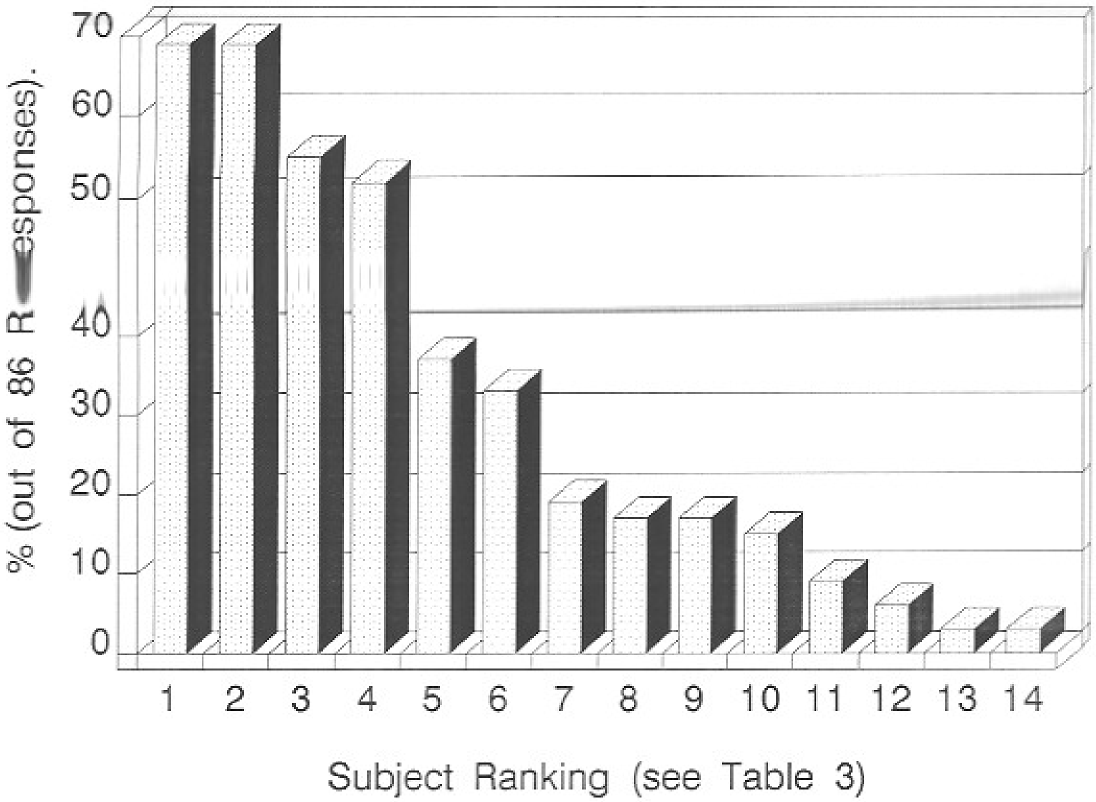

Figure 3: Professional Survey Subject Results.

Table 3: Professional Survey Subject Results (86 Responses).

Rank	Subject	Responses	%
1	CADD	59	69
2	Spreadsheets	59	69
3	Databases	47	55
4	Programming	45	52
5	Graphics	32	37
6	Wordprocessing	28	33
7	Expert Systems	16	19
8	Software Engineering	15	17
9	Software Packages	15	17
10	Operating Systems	13	15
11	Communications	8	9
12	Finite Element Methods	5	6
13	Numerical Methods	3	3
14	Hardware	3	3