Check for updates

The Relationship between Pre-college Mathematics and the Undergraduate Computer Science Curricula

Vicki L. Almstrum
Department of Computer Sciences
University of Texas at Austin

Abstract

In this paper, we will explore the role of mathematics in both the undergraduate and pre-college curricula. We will begin with a brief survey of existing undergraduate computer science curricula, with particular emphasis on the recommendations and controversies related to mathematics topics. Next, we will explore standards for pre-college mathematics, including the Advanced Placement Courses for Computer Science, the new **Curriculum and Evaluation Standards for School Mathematics**, and the recommendations from Project 2061. Finally, we will explore the relationship between the pre-college standards and the computer science curricula of the future.

Introduction

Since 1968, several recommendations for academic undergraduate programs in the computing sciences have been published. Each of these curricula has included mathematics as a key topic, and each was designed with some consideration (whether implicit or explicit) of the pre-college training expected of students entering the prescribed degree programs.

We begin with a retrospective of the key undergraduate curriculum recommendations, from Curriculum '68 to the new Computing Curricula 1990. Emphasis is given to the recommendations for mathematics in each of these curricula. The recommendations from the MAA report on discrete mathematics at the undergraduate level are compared to recommendations in the computer science curricula. Finally, pre-college computer science and mathematics standards are presented, with particular emphasis on the NCTM's **Curriculum and Evaluation Standards for School Mathematics**. The concluding remarks reflect on the potential influence the pre-college standards will have on future undergraduate computer sciences curricula.

Undergraduate CS Curricula

Any survey of computing-related curricula must begin with Curriculum '68 (Atchison, 1968). Curriculum '68 was the culmination of five years of work by a subcommittee of twelve computing educators, set up by the Education Committee of the Association for Computing Machinery (ACM). Consultants were used extensively in this effort, and the result was a set of computer science and mathematics courses which were the core courses for a major in computer science. Curriculum '68 acted as a catalyst, providing a basis for discussion as well as a developmental model for existing and budding computer science degree programs. However, as explained by Pollack (1982), Curriculum '68

> "... also had a dichotomizing aspect: Its basically mathematical orientation sharpened its contrast with more pragmatic alternatives. Most computer science educators agreed that the proposed core courses included issues crucial to computer science. However, the curriculum brought to the surface a strong division over the way in which these issues should be viewed. In defining the contents of the courses, Curriculum '68 established clearly its alignment with more traditional mathematical studies, giving primary emphasis to a search for beauty and elegance." (p. 41)

During the decade following the introduction of Curriculum '68, a number of alternate curricula appeared, each of which answered some aspect of the objections to

Curriculum '68. These new curricula included the areas of management information systems, software engineering, biomedical computer science, information science, applied mathematics (with emphasis on the mathematics of computation), computing center management, and computer engineering (Pollack 1982, p. 43). In the mid-70s, the ACM initiated a new curriculum effort, intended to answer the increased demand for professionally-focused computer science programs. This culminated in what has come to be called Curriculum '78 (Austing, 1979).

Curriculum '78 was criticized by many for simply reflecting the status quo in computer education, rather than providing a forward-looking model. Berztiss (1989) observes that Curriculum '78, instead of successfully integrating the theoretical and practical developments which occurred between 1968 and 1978, stressed the practical side and thus tended to lend a vocational spirit to computer science education. Ralston and Shaw (1980) point out that the mathematics components in Curriculum '78 were essentially the same as those in Curriculum '68, except weaker. (Curriculum '68 required a total of eight mathematics courses, while Curriculum '78 required only five.) Since the mathematics of central importance to computer science had changed drastically during the intervening decade, their prediction was that this would lessen the impact of the entire report.

A second professional organization which has made important contributions to computing-related curricula is the Computer Society of the Institute of Electrical and Electronic Engineers (IEEE). In 1976 and 1983, the IEEE Computer Society published model programs in computer science and engineering (IEEE, 1976; IEEE, 1983). These curricula were specified in the form of subject areas rather than courses and, with regard to aspects of the curriculum outside of computer science and engineering, deferred to the standards of the Accreditation Board for Engineering and Technology (ABET). Shaw (1985) has observed that, although it was strong mathematically, the IEEE program was also disappointing. Her point is that, while the claim was that the IEEE curriculum would be suitable for computer science, it had a heavy bias toward hardware and failed to expose the basic connections between hardware and software (p. 19).

In 1985, the Computer Science Accreditation Commission (CSAC) of the Computing Sciences Accreditation Board (CSAB) established criteria for accreditation of undergraduate computer science degree programs. These criteria are currently undergoing modification (CSAC/CSAB 1990), but with respect to mathematics requirements have undergone very little change. The current draft states that

> The curriculum must include at least one-half year of mathematics. This material must include discrete mathematics, differential and integral calculus, and probability and statistics, and may include additional areas such as linear algebra, numerical analysis, combinatorics, and differential equations. Some of this material may be covered **in courses other than mathematics courses.** (p 21)

There are two differences between the wording of the 1990 draft and the original draft, proposed in 1984 (Mulder and Dalphin, 1984). The first is the softening of the wording regarding the additional areas ("may include" as opposed to "at least one of"); the second is the replacement of the optional area *modern algebra* by the optional area *combinatorics*. Common to both drafts is the recognition that mathematics material may be covered in courses which are not explicitly mathematics.

The latest effort to establish a modern undergraduate computer science curriculum is being carried out by cooperating task forces of the ACM and the IEEE Computer Society. While the final version of the Computing Curriculum 1990 report has not yet been published, intermediate proposals have been distributed for review. In the latest version of the report (Tucker, 1990), a minimum of one-half year of mathematics courses is required of computer science students. This minimum coverage includes topics from discrete mathematics and calculus, as well as at least one of the four subjects *probability*, *linear algebra*, *advanced discrete mathematics*, and *mathematical logic*. The report observes that students in professionally-oriented programs or intending to pursue graduate studies should be advised to take significant mathematics beyond the minimum required (p 27).

All of these computer science curricula can be considered in light of the Mathematical Association of America (MAA) report on discrete mathematics at the undergraduate level (Ralston, 1989). The MAA report gives the experiences of six colleges and universities which had been supported by the Alfred P. Sloan Foundation under a program to foster "... the development of a new curriculum for the first two years of undergraduate mathematics in which discrete mathematics [would] play a role of equal importance to that of the calculus" (p. 1). The intention of the Sloan program was to make recommendations for revision of the first two years of the mathematics curriculum for everyone: mathematics majors, physical science and engineering majors, social and management science majors, as well as computer science majors (p. 91). Ralston explains that, in studying the mathematics requirements included with existing curricula (essentially those described in the previous paragraphs), it was felt that the recommendations for a freshman level discrete mathematics course given in the IEEE model program were probably the most demanding (p. 94). The final recommendations of the MAA Committee were as follows (p. 91):

1. Discrete mathematics should be part of the first two years of the standard mathematics curriculum at all colleges and universities.
2. Discrete mathematics should be taught at the intellectual level of calculus.

3. Discrete mathematics courses should be one year courses which may be taken independently of the calculus.
4. The primary themes of discrete mathematics courses should be the notions of proof, recursion, induction, modeling and algorithmic thinking.
5. The topics to be covered are less important than the acquisition of mathematical maturity and of skills in using abstraction and generalization.
6. Discrete mathematics should be distinguished from finite mathematics, which, as it is now most often taught, might be characterized as baby linear algebra and some other topics for students not in the "hard" sciences.
7. Discrete mathematics should be taught by mathematicians.
8. All students in the sciences and engineering should be required to take some discrete mathematics as undergraduates. Mathematics majors should be required to take at least one course in discrete mathematics.
9. Serious attention should be paid to the teaching of the calculus. Integration of discrete methods with the calculus and the use of symbolic manipulators should be considered.
10. Secondary schools should introduce many ideas of discrete mathematics into the curriculum to help students improve their problem-solving skills and prepare them for college mathematics.

In comparing the Computing Curriculum 1990 draft requirements to the points of this list, one sees that the amount and kinds of mathematics required are somewhat weaker than those implied by the MAA recommendations. The joint recommendations reflect the current status of the ongoing controversy over which (and how many) mathematics courses should be required of computer science majors, as well as the debate of "calculus vs. discrete mathematics". The MAA recommendations suggest that there is gathering momentum behind proposals to elevate discrete mathematics in the curricula of both mathematics and computer science programs. Audience comments during a panel on the Computing Curriculum 1990 at the Twenty-First SIGCSE Technical Symposium on Computer Science Education (SIGCSE, 1990) provided evidence that this controversy is far from resolved.

Pre-college Curricula

From the point of view of university curricula, high school mathematics has traditionally been a bit of a "black box". A common description of the mathematics prerequisite for entering an undergraduate computer science program is "readiness for calculus" (Werth, 1990). Studies to identify factors which could be used to predict success in computer sciences programs have typically included number of high school mathematics classes taken, kinds of mathematics classes taken, grades in high school mathematics courses, and SAT math scores (e.g., Campbell & McCabe, 1984; Butcher & Muth, 1985). The key issue is what high school graduates can be assumed to *know* about mathematics.

The Advanced Placement (AP) Course Description for Computer Science explicitly describes student prerequisites as:

- familiarity with mathematical notation at the level of a second course in algebra
- experience in problem-solving
- appreciation of the need to structure and develop a given topic in a logical manner
- competence in written communication (The College Board, 1990, pp 4-5)

The curricular organization necessary to prepare a high school student to enter an AP computer science course is essentially compatible with that recommended to prepare students for the AP course in calculus. It is important to note, however, that calculus is not a prerequisite for the Advanced Placement computer science courses.

A problem with using Advanced Placement descriptions to define what high school students (should) know about mathematics or other topics is that these definitions have commonly been derived from the college curriculum. For the purposes of this paper, it is desirable to consider pre-college learning independent of what is expected at the college level. High school mathematics curricula are normally determined at the state and district levels. The recommendations of professional organizations such as the National Council of Teachers of Mathematics have traditionally influenced and guided the local mathematics offerings. However, before 1989, there did not exist an accepted and well-defined national standard for the core topics which high school mathematics should encompass. This lack of a standard can be hypothesized as one of the causes of the poor state of American education as reported in, for example, **A Nation at Risk** (National Commission on Excellence in Education, 1983).

Partly in response to studies like **A Nation at Risk**, the National Council of Teachers of Mathematics (NCTM) developed the **Curriculum and Evaluation Standards for School Mathematics**. This document, which was first published in 1989, was the culmination of a tremendous cooperative effort. Fifteen national mathematical science organizations joined with NCTM as endorsers of the **Standards**; twenty-five professional organizations are listed as supporters of the **Standards**; and twenty organizations, including the IEEE, are listed as allies to the effort described in the **Standards**. As a result, the **Standards** express "... the consensus of professionals in the mathematical sciences for the direction of school mathematics in the next decade" (National Council of Teachers of Mathematics, 1989, p vi).

Five general goals permeate the **Standards**; that all students (1) learn to value mathematics, (2) become confident in their ability to do mathematics, (3) become mathematical problem solvers, (4) learn to communicate mathematically, and (5) learn to reason mathematically (NCTM, 1989, p. 5). Organizationally, the **Standards** are broken up into four sections: curriculum standards for each of the three grade levels K-4, 5-8, and 9-12, and evaluation standards for all grade levels. A total of fifty-four standards are spread among these four sections, and delineate the goals, focus, and approach for the various content area topics.

An important theme of the **Standards** is mathematical opportunity for all students. As a result, the various standards reflect mathematics topics which *all* students should have the opportunity to learn. Individual differences among learners are to be addressed by differences in the depth and breadth of coverage. The **Standards** take into account the importance of modern technology, and include many suggestions for appropriate instructional use of computers and graphics calculators.

At the high school level, the **Standards** include a core program intended for all students and additional topics intended primarily for college-intending students. The **Standards** recommend that all high school students should have three years of mathematical study, with four years required of all college-intending students. The curriculum for grades nine through twelve encompasses the following fourteen areas:

1. Mathematics as Problem Solving
2. Mathematics as Communications
3. Mathematics as Reasoning
4. Mathematical Connections
5. Algebra
6. Functions
7. Geometry from a Synthetic Perspective
8. Geometry from an Algebraic Perspective
9. Trigonometry
10. Statistics
11. Probability
12. Discrete Mathematics
13. Conceptual Underpinnings of Calculus
14. Mathematical Structures

Each of these fourteen standards builds upon the foundation which was established in the standards for the earlier grade levels. Several topics are common to all three grade levels: *Mathematics as Problem Solving*, *Mathematics as Communication*, and *Mathematics as Reasoning*. In addition, the topic *Mathematical Connections* appears at every level, with the intention of stressing the rich network of relationships (or, in the words of Computing Curriculum 1990, recurring concepts) among seemingly diverse topics in mathematics.

The presence of a separate standard for discrete mathematics (standard 12) is especially interesting from the point of view of preparing students for undergraduate computer science. It is not intended that discrete mathematics be a separate course offering, but rather that topics of discrete mathematics be integrated throughout the mathematics curriculum. The goals are that:

> In grades 9 through 12, the mathematics curriculum should include topics from discrete mathematics so that all students can--
>
> - represent problem situations using discrete structures such as finite graphs, matrices, sequences, and recurrence relations;
> - represent and analyze finite graphs using matrices;
> - develop and analyze algorithms;
> - solve enumeration and finite probability problems;
>
> and so that, in addition, college-intending students can--
>
> - represent and solve problems using linear programming and difference equations;
> - investigate problem situations that arise in connection with computer validation and the application of algorithms. (NCTM, 1989, p 176)

Ralston (1989) points out that the MAA Committee on Placement Examinations is currently attempting to isolate the skills needed by students taking discrete mathematics. While this effort will not necessarily result in advanced placement for discrete mathematics in the immediate future, Ralston notes that "... it will help to explain what might be the appropriate preparation for a successful experience in such a course" (p 97).

Another nationwide project which promises to influence the pre-college education of American youth is Project 2061, being carried out by the American Association for the Advancement of Science. Project 2061 is a long-range, multi-phase effort to address scientific literacy. The year 2061 refers to the return of Halley's Comet, and the hope that the recommendations which emerge from the project will be a reality by that time. Phase I is complete and has resulted in a report which identifies the "understandings and habits of mind which are essential for all citizens in a scientifically literate society" (American Association for the Advancement of Science, 1989, p. 3). The report specifically addresses science, mathematics, and technology as aspects of scientific literacy, as well as the values, attitudes, and skills which all scientifically literate persons should have. The recommendations of Project 2061 provide a natural extension to those described as part of the the NCTM standards for mathematics. The project is currently in Phase II, in which teams of educators and scientists are creating alternative curriculum models based on the recommendations, as well as blueprints for reform related to all aspects of pre-college education. Phase III (implementation), the final phase of Project 2061, is expected to begin about 1992 and to last well into the 21st century.

Concluding Remarks

The Denning Report's definition of computing as a discipline provides clear evidence of the deep interconnections between mathematics and computer science (Denning, 1989). The theoretical aspects of the Denning Report's nine subareas of computing are particularly mathematical; for example, computability theory and computational complexity are elements of the subarea Algorithms and Data Structures; formal languages and semantics are elements of the subarea Programming Languages; and boolean algebra, statistics, and probability are elements of the subarea Architecture. In addition, many of the elements of abstraction and design rely on mathematical maturity. For example, the abstraction aspect of the subarea Human-Computer Communication includes smoothing and ray tracing algorithms as elements, and the design aspect of the subarea Artificial Intelligence and Robotics includes design techniques for logic programming, theorem proving, and rule evaluation as elements.

It is vital that the study of computing include these specific mathematical topics. However, the need for a mathematical basis goes beyond mathematics content. Strong underlying themes in the literature regarding computer science curricula have included the need to emphasize skills in problem solving (Berztiss 1987, Henderson 1990, Shaw 1985), symbolic manipulation (Gries 1990), and reasoning (Wirth 1990, Myers 1990). Since these skills cannot be learned in a one semester remedial course at college, they must be have been taught much earlier, as a part of the pre-college training students receive. The skills of problem solving, symbolic manipulation, and reasoning are strongly related to the first three common strands of the National Council of Teachers of Mathematics Curriculum (NCTM) Standards: *Mathematics as Problem Solving*, *Mathematics as Communication*, and *Mathematics as Reasoning*. They are also important themes in the Project 2061 recommendations.

Ralston (1989) reports that while he was teaching the first year Introduction to Computer Science course, he kept track of mathematics topics he would have liked the students to have had before, or at least concurrently with, his course. Many of the ideas he listed were topics that students *should have had* in four years of the traditional high school curriculum (p 93). Many computer sciences instructors experience this same gap in their students' backgrounds, and are obliged to make up for it by covering "elementary" skills in their lectures. Unfortunately, this dilutes an already-full curriculum and reduces the opportunity to thoroughly cover "new" material. It follows that the requisite skills should be gained *before* students enroll in undergraduate computer science courses.

The effective nationwide implementation of the NCTM **Standards** and of the recommendations in Project 2061 will positively affect preparedness of entering college students. This in turn will result in an evolving mathematics content in undergraduate programs. However, this change will not be evident for many years. The theme of the 1991 SIGCSE Technical Symposium is "Keep the Information Flowing". One aspect of this information flow is the creation of a cohesive curriculum for grades K-16. Our challenge as educators and computer scientists is to not only be aware of this information flow between grade levels, but to be a part of the process of enabling its implementation.

Acknowledgements

I am indebted to John Werth and David Young for their guidance to important resources and background information. I would like to thank Ralph Cain, Nell Dale, Jeff Brumfield, and three anonymous referees for their comments and encouragement.

References

American Association for the Advancement of Science. (1989). Science for all Americans: A Project 2061 report on literacy goals in science, mathematics, and technology. AAAS Publication 89-01S.

Atchison, W. F. (chair). (1968). Curriculum '68: Recommendations for academic programs in computer science. *Communications of the ACM*, 11(3), 151-197.

Austing, R. H. (chair). (1979). Curriculum '78: Recommendation for the undergraduate program in computer science. *Communications of the ACM*, 22(3), 147-166.

Berztiss, A. (1987). A mathematically focused curriculum for computer science. *Communications of the ACM*, 30(5), 356-365.

Butcher, D. F. and Muth, W. A. (1985). Predicting performance in an introductory computer science course. *Communications of the ACM*, 28(3), 263-268.

Campbell, P. F., and McCabe, G. P. (1984). Predicting the success of freshmen in a computer science major. *Communications of the ACM*, 27(11), 1108-1113.

CSAC/CSAB. (1990). *Criteria for Accrediting Programs in Computer Science in the United States*. Draft for Review and Comment. August 1990.

Denning, P. J. (chair). (1989). Computing as a discipline. *Communications of the ACM*, 32(1), 9-23.

Graham, R. L., Knuth, D. E., & Patashnik, O. (1989). *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley Publishing Company.

Gries, D. (1990) Teaching calculation and discrimination early in the curriculum. Presented at the IFIP WG3.2 workshop on Informatics Curricula for the 90's at Brown University, 5-7 April 1990.

Henderson, P. (1990). Discrete mathematics as a precursor to programming. *SIGCSE Bulletin*. 22(1), 17-21.

IEEE Computer Society Education Committee (Model Curriculum Subcommittee). (1976). *A Curriculum in Computer Science and Engineering*. IEEE Computer Society.

IEEE Model Program Committee, Educational Activities Board. (1983). *The 1983 IEEE Computer Society Model Program in Computer Science and Engineering*. IEEE Computer Society.

Mulder, M. C. and J. Dalphin. (1984). Computer Science Program Requirements and Accreditation. *Communications of the ACM*. 27(4):330-335. Also in *Computer*. 17(4):30-35.

Myers, Jr., J. P. (1990). The central role of mathematical logic in computer science. *SIGCSE Bulletin*. 22(1), 22-26.

National Commission on Excellence in Education. (1983). *A Nation at Risk*.

National Council of Teachers of Mathematics. (1989) *Curriculum and Evaluation Standards for School Mathematics*.

Pollack, S. V. (Ed.). (1982) *Studies in Computer Science*. The Mathematical Association of America.

Ralston, A. and Shaw, M. (1980). Curriculum '78 -Is computer science really that unmathematical? *Communications of the ACM*, 23(2), 67-70.

Ralston, A. (Ed.) (1989). *Discrete Mathematics in the First Two Years*. The Mathematical Association of America.

Shaw, M. (Ed.). (1985). *The Carnegie-Mellon Curriculum for Undergraduate Computer Science*. New York: Springer-Verlag.

SIGCSE. (1990). Twenty-First SIGCSE Technical Symposium on Computer Science Education. Washington D.C., February 22-23.

Tucker, A. B. (Ed.) (1990). *Computing Curricula 1990: Report of the Joint Curriculum Task Force*. Review Draft, September 15, 1990.

Werth, J. (1990). Private conversation.

Wirth, N. (1990) Perspectives on computer science education. Presented at the IFIP WG3.2 workshop on Informatics Curricula for the 90's at Brown University, 5-7 April 1990.