Computers in Education, O Lecarme and R Lewis (eds)
© IFIP, North Holland Publishing Company, 1975

Invited Author

IMPACT OF COMPUTERS ON CURRICULUM IN THE SCHOOLS AND UNIVERSITIES

Patrick Suppes

Institute for Mathematical Studies in the Social Sciences
Stanford University
Stanford, California, U.S.A.

The impact of computers on curriculum is considered, with special attention to what can be done to improve the quality of education in the face of budget constraints and declining student enrollments in the United States. The central thrust of the paper is that the use of computers for instruction is one of the few ways in which we can hope to make substantial improvements in the next several decades. A number of detailed examples from current experience in computer-assisted instruction are described, as a partial guide to what we may expect in the reasonably near future. Some important conceptual and technical problems that need to be solved are examined.

1. INTRODUCTION

My intention in this paper is to survey the kinds of impacts that computers are already having or will have in the near future on curriculum in the schools and universities. On other occasions I have tried to analyze the general issues connected with the use of computers for instruction (for example, Suppes and Morningstar, Chapter 1, 1972). Here I shall avoid, by and large, the general issues and aim at particular examples and the implications of these examples for the future.

In analyzing various examples I want to emphasize three aspects of the use of computers for instruction. The first is the aspect of diversity, especially in terms of content and entire subject matters. With extensive use of computers, students can be offered a diversity of courses far beyond what is feasible on other grounds. The second is the sense of individualization that can be achieved by computer-assisted instruction, both in terms of actual rate of progress of the student and also in terms of the convenience of time and place for the student.

The third is the matter of productivity of faculty in the face of declining budgets, especially in the fact of faculty sizes that have little hope of being increased for the remainder of this century. In the United States anyway this is certainly true of both the schools and the universities. The implications of this have hardly yet begun to be appreciated by many people in education. However, the recent prediction by the Department of Education of the State of New York that the number of students in colleges and universities in 1990 will be 23% less than the number currently enrolled has been a sufficiently dire prediction to arouse broad interest. The associated prediction that about half of the private colleges in New York will be closed by that date because of financial difficulties has also been a subject of sharp discussion. It will not be my concern in this paper to try to make the case for the exact percentage prediction in New York but rather to accept the broad facts on which it is based, namely, the demographic data on number of students and the anticipated bounds on future budgets. The central question becomes how we can continue to enrich the educational opportunities of students in the face of declining faculty sizes.

I begin with the schools and then turn to the universities.

2. THE SCHOOLS

I divide my discussion of curriculum in the schools into three parts, first dealing with basic skills for disadvantaged students, next considering problem solving and programming, and third, special programs for gifted students. I emphasize once again that what I have to say deals almost entirely with the situation in the United States, because I am not deeply familiar with activities in other countries and because I would judge that up to the present the bulk of the use of computers for instruction has taken place in the United States, although I would assume that matters will be changing rapidly in this respect over the next several decades.

2.1. Basic Skills of Disadvantaged Students

My own earliest work in computer-assisted instruction focused on basic skills of disadvantaged students. In particular I concentrated on elementary mathematical skills, and my colleague Richard C. Atkinson concentrated on basic reading skills. Since this work first began at Stanford University in 1963 a large number of research and evaluation studies have been published, and I shall not attempt to review them here. A history of the effort is to be found in the two books, Suppes, Jerman and Brian (1968) and Suppes and Morningstar (1972). Out of those efforts has grown a fairly widespread use in the United States of computer-based supplementary courses in elementary mathematical skills and concepts, basic reading skills and language skills. These courses have been aimed especially at disadvantaged students who have been a central concern of elementary education in the United States since the passage of the Elementary-Secondary Education Act of 1965 in the Johnson Administration.

The impact of the work at Stanford, the subsequent commercial developments at Computer Curriculum Corporation with which I have been closely associated, and the work by other people in the United States on these basic skills as a primary focus of education for students in the age range 6 to 12 years have not led to radical changes in the content of curriculum. Indeed, it was not the intent of those developing these computer-based courses to lead to changes in the content of curriculum.

Furthermore, whatever happens to change the content of the curriculum in the elementary schools or primary schools over the next 50 years, I would be enormously surprised if the basic elementary language, reading and mathematics skills do not continue to be an important part of the curriculum.

The impact of using computers to develop these skills and to provide a powerful approach to remediation has been to change the style and not the content of the curriculum. What is especially important about this change in style is the presentation to each student of exercises at his current level of accomplishment, and the chance to move ahead or to slow down independently of the progress of other students. The computer is being used in this case to fine tune curriculum that has been taught in one form or another since systematic education began, but to fine tune it, in a way that has not been previously possible, to the needs of individual students.

2.2. Problem Solving and Computer Programming

A more radical change in curriculum content has appeared in the use of computers, best exemplified by the work of Seymour Papert and his colleagues at the Massachusetts Institute of Technology. In a variety of imaginative ways, they have been teaching young students to program, especially in the language LOGO, and in the process to tackle and solve a variety of problems, from writing music to playing games. A number of other people around the world have also become involved in this kind of work, and there is little doubt that it will in time have a major impact on curriculum in the schools. The impact as yet is relatively small, but only because it takes time for computer facilities to be available in a large number of schools and, even more important, for teachers to feel at home with these new approaches to education.

A good reference is the delightful article by Papert and Solomon (1972).

2.3. Logic for Gifted Students

My own interest in teaching nonstandard curriculum to gifted students goes back nearly 20 years and predates my interest in computers. In the fall of 1956 I brought into my college logic course a group of able students in the age range 11 to 14 from neighboring schools, and I have in one way or another been continuously involved in the teaching of elementary mathematical logic to students in the schools since that time. An early report on our work is to be found in Suppes and Binford (1965). As we moved from the first experiment at Stanford in 1956 to organizing classes in schools surrounding Stanford, it soon became apparent that the central difficulty was not in the capacity of young students to master the skills and concepts of formal logic but rather in the lack of training and, in some cases, aptitude for these matters on the part of their teachers. Despair at being able to bring a rich supplementary curriculum to gifted students in the schools was one of the things that motivated me to become involved in the use of computers for instruction in 1963.

Our first demonstration programs were running on the PDP-1 at Stanford in 1963 and embodied a full proof checker for elementary logic without quantifiers. This program was available not long after the very first of the proof checkers had been written anywhere in the world. On and off over the past decade we have engaged at Stanford in a variety of efforts to bring formal logic to able students in the schools. A fairly detailed description of the earlier efforts was given in my survey of computer-assisted instruction at Stanford in Bordeaux, France, several years ago (Suppes, 1972).

More recently, we have been working with a group of extremely gifted students in the age range 10 to 15 years. By extremely gifted I mean a group that has been selected with IQs of 170 or more. Our approach to these students has been different from any of our approaches in the past. We have placed terminals in their homes, and during the evening and on weekends they dial up the PDP-10 system in the Institute at Stanford for work in several courses but especially courses in logic. They begin with an elementary logic course aimed at able students of age 12, and when that is completed move on to the computer-based logic course at Stanford, which I discuss in more detail below. In addition, a number of the gifted students have also used their terminals at home to take courses in LOGO and in BASIC.

One of the striking results has been the enormous variability within what is on the surface a homogeneous group of highly gifted students. The amount of interest, the amount of work done and the rate of learning between the best student and the most indifferent student in the group is astoundingly large. This is what one is to expect when individualization without constraints, as in the case of these home-based terminals, is left to follow its own natural course. The very best student, for example, completed the relatively difficult introductory course in logic at Stanford in about four weeks, faster than any Stanford student taking the course at the same time. Other gifted students have found that they were not interested in this direction of work at all and have asked that their terminals be removed after a few weeks. In order to get a sense of the variability in interest we have left the whole process almost completely unstructured in the initial experiment.

To accomplish the most positive results, on the other hand, it is quite clear to us from our current efforts and from a variety of experimentation on similar self-paced and nonformal educational settings that the imposition of some minimal structure is probably almost always beneficial.

One difficulty with doing more with these highly gifted students is that their programs are enormously full already, at least within the framework of school- and parent-suggested activities of the kind frequent in California. These students are busy from morning to night in a variety of programs, from music lessons to jewelry making to competitive athletics. Unless some change is made in the standard school regime there is in fact a relatively small amount of time available to them to pursue in any deep sense supplementary work in mathematics or computer programming. The capacity is there but the social organization is relatively restricting. In other countries and in other settings the situation may well turn out to be very different.

3. THE UNIVERSITIES

Because of the constant or slightly declining enrollments predicted on the average for the next decade and a half in American universities, I would like to concentrate on the question of how computers may be used in universities to offer a full range of instruction with a constant or declining number of faculty.

A second general point I want to emphasize is that, contrary to my own earlier work that was concerned with courses that would be used by a very large number of students, I now see the first important and significant use on a broad scale of computers at the university level may well be in offering a large number of courses that ordinarily have a small enrollment. I give some examples of this phenomenon from several areas of university teaching, but the basic point is worth general consideration. If a professor of some seniority offers a course to a class size ranging from 2 to 6 students, then it must necessarily be an expensive course to teach in terms of the total number of student hours. For example, let us assume that there are four students in the class and that the class constitutes, as it might at Stanford, half of the teaching load for one of the three terms of the year. Assuming the professor's salary is $20,000 per year and the class meets 3 days a week for 11 weeks, then the cost of instruction is $25.25 per student hour, just in terms of the professor's salary and excluding any other costs.

Those of you familiar with the current financial situation, at least of American universities, will recognize quickly enough that courses in this cost range will simply not be offered for any substantial number of years. The pressures to eliminate such courses are growing and will continue to increase.

On the other hand, if, by using computer facilities, a faculty member can offer four such courses simultaneously then the cost of instruction is reduced to $6.31 per student hour, which for advanced work is more reasonable, but is still a figure that is easy to reach even using a sophisticated computer system.

If, in contrast, we tackle the cost of teaching a class of 200 students on the same basic assumptions of a professional salary of $20,000, this course being one-sixth of a teaching load and the class meeting 3 times a week for 11 weeks, the bare cost of instruction is about $.51 per student hour, which is a figure that is not easy to match by use of computer facilities. These simple computations show why I think the important use of computers in teaching is in the presentation of courses of a moderate degree of specialization and at a moderate level of difficulty, rather than at the level of large elementary courses.

With these economic considerations in mind I would like now to turn to a survey of some of the efforts to implement such courses in three areas: the teaching of foreign languages, the teaching of computer programming and the teaching of mathematically oriented courses.

3.1. Foreign Language Teaching

Over the past ten years we have had experience at Stanford in teaching French, German, Russian and Chinese by computer. The highly experimental work in German is described in Levine (1973). The most extensive work by far has been in the teaching of Russian and has been under the direction of Professor Joseph Van Campen, Chairman of the Department of Slavic Languages at Stanford. The initial effort was aimed at the teaching of elementary Russian, and a positive evaluation of this work is reported in Suppes and Morningstar (1969). The more significant developments in line with the computations of costs given above have occurred in the past several years. Van Campen and his younger colleague, Professor Richard Schupbach, have embarked on the extensive use of the Institute's computer facilities to teach specialized courses in Russian language and linguistics that are ordinarily taken by only two or three students. The course in Old Church Slavonic, developed by Van Campen and which is now about 80% computer based, is described in Van Campen (1973). A corresponding course on the linguistic history of the Russian literary language is described in Schupbach (1973). Van Campen and Schupbach have plans to develop approximately 10 to 12 courses in Slavic linguistics that they can offer on a regular basis each year and with a staff consisting just of the two of them, together with some teaching-assistant help. In considering their development of these courses it is to be emphasized that this is about the only way they will be able to develop such courses, because given the present enrollment in Slavic languages at Stanford and given the current and continuing financial constraints on the University there is no hope of adding an additional faculty member to their group to aid in the teaching of these courses.

3.1. Computer-Programming Courses

The reflexive principle of using the computer to teach courses about computer programming will undoubtedly have wide application. At Stanford, thus far the computer-assisted instruction courses in computer programming at the university level have been restricted to a short introductory course in BASIC (Barr, Beard and Atkinson, 1974) and part of the introductory course on LISP taught by John McCarthy and Cordell Green. It is our intention in the future to offer considerably more in this direction, and I anticipate that this is one of the ways that we shall be able to offer a sufficiently wide variety of specialized courses to meet the increasing demands of students interested in the many different aspects of programming.

One of the central difficulties of deeper moves into teaching computer programming entirely by computer is the problem of program verification. The writing of programs that understand the programs students write or even the weaker requirement of having a program checker that students are required to use to verify that their programs are correct is still relatively difficult even for simplified proper subsets of widely used programming languages. I will have more to say about this below in discussing the corresponding problem for mathematical proofs. Suffice it to say at this point that the difficulties are greater for verification of computer programs than for verification of mathematical proofs, because we have for first-order logic Gödel's completeness theorem that provides an exact match between syntax and semantics: one sentence is a logical consequence of another if and only if it can be derived by use of syntactical rules of inference from the first. We can formalize axiomatic set theory in first-order logic and thereby expect to have powerful syntactical methods for checking the validity of proofs, methods that are more on the surface than are the corresponding methods required for verification of program correctness.

3.2. Mathematically Oriented Courses

The most extensive effort in using computers for instruction at Stanford is currently centered around the following two courses, Introduction to Logic and Axiomatic Set Theory. The course in logic has been developed over a number of years and is a direct outgrowth of the work in logic for the schools mentioned above, which first began in 1963. The course in set theory has in turn grown out of the university logic course.

Introduction to Logic. The logic course is one in which the students do essentially all of their work at computer terminals. There are no lectures, but a teaching assistant is available during a number of hours of the day for consultation. This course has been running for several years and is a regular part of the curriculum; by now it is the only version of introductory logic taught at Stanford. It is offered every term, three times a year, and the enrollment currently runs between 50 and 80 students a term. From a programming standpoint, the most important feature of the course is the proof checker. The students give proofs in first-order logic and the program checks their correctness on a step-by-step basis. The system of natural deduction embodied in the proof checker is essentially that given in Suppes (1957).

The proof checker used in the logic course has evolved over a number of years. The original one, begun in 1963, was written essentially by Dow Brian. A later version of the logic course for the schools was produced by Roulette Smith. The proof checker for the college course and the programming for the college course were initially done by Adele Goldberg (1973). The present version was written by Robert Smith, Lawrence Markosian, Lee Blaine and Vesko Marinov. I am continuing the revision and improvement of the curriculum in collaboration with my younger colleagues, especially Edward Bolton and Lawrence Markosian.

Because the structure of the logic course up to the present and the empirical results obtained have been described in Goldberg and Suppes (1972, 1974), I shall not enter into details here.

Axiomatic Set Theory. In many respects, the logic course is deceptive as a model of how to approach mathematically oriented courses in terms of computer-based instruction. The deception arises from the fact that it is a relatively straightforward matter, though nontrivial, to have available a complete proof checker for inference rules in first-order logic. The difficulty of having reasonable rules of proof is at least an order of magnitude greater once we leave the framework of elementary logic and move to the first level of mathematics in which proofs are always given in informal style. At this next level, proofs are given informally because of the tedious and inordinate length of formal versions of the proofs. The human checker of informal proofs as given by students is tolerant and possessed of many good intuitions for evaluating arguments that are not fully worked out. Matters are otherwise in the case of the computer acceptability of such proofs. In the case of the course in set theory, by far the main effort has been devoted thus far to getting semi-reasonable proof procedures that can be used by students interactively with the program. A somewhat detailed account of where we now stand is described in a paper by my younger colleagues (Smith, Graves, Blaine and Marinov, 1975).

As far as we have been able to determine, there has not been much prior investigation of the many features that are needed to move from a first-order logic proof checker to a reasonable set of procedures for giving informal proofs in ordinary mathematics at the level of complexity of the first course in axiomatic set theory. At the present stage of our work, it seems highly unlikely that any single discovery will lead to a sharp breakthrough in how to handle informal proofs within a computer framework. On the other hand, it is evident that there are a large number of practical constructive things that can be done and that are feasible to implement on a computer system of the size of our PDP-10. Let me give some examples of problems that we have solved and also of some that we have not.

In a first-order proof checker the first move of liberalization is from successive to simultaneous substitution for universally quantified variables. The next practical step and one that leads to a rule that is used extensively is to make the substitution of terms for variables in specification for universally quantified variables automatic, that is, by, for example, making the inference from '(Ax)(F(x) → G(x))' and 'F(a)' to 'G(a)' without indicating to the program the specification of 'a' for 'x'. Such a rule of implication is also readily strengthened to include any tautological implication that is desired as well. More elaborate examples, whose formal complexity is ignored in first-order logic, are appropriate procedures for replacing one formula by another in natural and intuitive ways without taking a large number of steps to do so, or more importantly, by not having to specify a large number of conditions that must be satisfied. It is probably especially the latter aspect of matters that needs most attention. In informal proofs, 'obvious' conditions are left implicit. Making such conditions internally explicit in the program-representation of the proof is probably the most crucial aspect of having a good computer-based proof checker for informal proofs. It amounts, of course, to having a program that in mose cases will do what you expect and will only raise questions when the situation is in some way abnormal.

Another current feature of the program is that students may call a fast, restricted theorem prover that runs for a limited amount of time and then stops if a proof is not found. Students use this to cover routine gaps in their informal proofs. Because the method of attack of a resolution theorem prover seems so psychologically different from human ways of thinking about proofs I am doubtful that we shall be able to extend in any very deep way this feature of the program, although it is playing a useful role at what might be called a low level of complexity.

Another useful feature of the program is a subroutine to decide any Boolean expression. For example, any identity made up of set variables and the operation symbols for union, intersection and set difference can be evaluated for validity very quickly by this subroutine, and students may call it at appropriate points in informal proofs.

On the other hand, our proof procedures are still awkward in many cases and it is clear that practical lines of improvement can be found. For instance, our development of cardinal numbers is based upon Tarski's axiom that the cardinal numbers of two sets are identical if and only if the sets are equipollent. In this framework the informal proof that cardinal addition is associative follows trivially from the associativity of the union of sets, and in a textbook-informal proof a one-line argument is used. At the present time it is still tedious to convert this one-line intuitive argument into an informal proof accepted by the program. Another similar example is the proof that the definition of exponentiation for cardinal numbers is justified--justified in the sense that the operation is uniquely defined for all cardinal numbers. Again, the line of proof is completely straightforward from facts already established about corresponding operations on sets, but passing from the level of arbitrary sets to the level of cardinal numbers is not as easy and natural as it should be in the proof procedures currently available.

Improvements of a similar sort are needed in order to do in a natural way the theory of ordinal numbers, especially the rather complex and somewhat difficult proofs justifying definition by transfinite recursion. I would summarize our present view of these matters by saying that we have procedures that are workable and that are being used by students in the course. But we anticipate being able to make substantial improvements and will be very disappointed if we cannot make substantial improvements over the next couple of years.

I have not said much about the curriculum in the course in axiomatic set theory. It follows closely the content of my earlier book (Suppes, 1960). After general developments of elementary set theory, the course concentrates on finite and infinite sets, the theory of cardinal numbers, the theory of ordinal numbers, and the axiom of choice. It seems likely that this classical content will be a part of the curriculum in logic and the foundations of mathematics for many years to come, and therefore we are confident that as we perfect the course we will not need to plan to change in any radical way the curriculum itself in the near future. Admittedly this situation is much different when one is developing curriculum in an area of mathematics or science that is currently under rapid change. This would, for example, be true in a graduate course in set theory that would be concerned with the many important results obtained in the past decade and a half, but the first introductory course does not reach the point at which the many new results can be seriously considered and presented to the students. It is also the case that the informal proof procedures needed for a more advanced course would be beyond our reach and too difficult to develop at the present time. The proofs would move too rapidly and it would be too much for us to close the gap between the standard informal proofs and the internal formal representation in the computer program. We believe that the level of difficulty of a course like the one in set theory I have been describing is the feasible level for current efforts.

Other courses. We are currently planning other courses at this same level of difficulty: a course in proof theory, a course in the foundations of probability, and a course in the foundations of measurement aimed especially at social scientists. These additional courses will require proof machinery at about the same level of difficulty and complexity of the set theory course.

What we are especially interested in determining is whether the effort of extension to these new courses will be sublinear. I mean by this whether we shall find that the effort required to make the extension to these new courses will be definitely less than the effort that has been required to organize the set theory course in the first place.

Final remark on productivity. After having discussed on several previous occasions problems of productivity in the schools and the absence of increased productivity in education through most of this century, I have decided it is best to analyze the problem in terms of my own teaching and see what can be done about my own productivity. By taking responsibility for the two computer-based courses mentioned above and by offering them every term, I have doubled my teaching load at Stanford. In addition, I have adopted on principle the stance that I no longer give lecture courses. My own teaching is divided into computer-based courses in basic subjects like logic and set theory, and advanced seminars on specialized topics. I find this division extremely satisfying personally. I enjoy very much the personal and informal contact of seminars, and do not see their elimination by computer-based instruction in any foreseeable future. As far as course offerings go, I see no reason not to believe that I shall be able to double once again my course load over the next three or four years. In doing so, as indicated earlier, I am not aiming at large courses with large enrollments but at offering a diversity of courses at a medium level of difficulty. I have begun to sample the students at Stanford in terms of such courses and am finding that there are a number of courses in which they would be interested and which we do not currently offer but that we should be able to do a reasonably good job with in the kind of computer-based framework I have been describing. But teaching loads consisting entirely of small seminars are not practical in any university except for a few faculty members.

A mix of (i) large lecture classes for elementary courses, (ii) computer-based instruction for courses of intermediate difficulty and of a rather specialized character, and (iii) informal seminars at the advanced level seems attractive both in terms of productivity and personal satisfaction for faculty in the disciplines I have mentioned. However, it is too early to tell how many will share this opinion.

4. REFERENCES

Barr, A., Beard, M., and Atkinson, R. C. (1974), A Rationale and Description of the BASIC Instructional Program (Tech. Rep. 228, Psych. & Ed. Series), Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, Calif.

Goldberg, A. (1973), Computer-assisted Instruction: The Application of Theorem-proving to Adaptive Response Analysis (Tech. Rep. 203, Psych. & Ed. Series), Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, Calif.

Goldberg, A., and Suppes, P. (1972), A Computer-assisted Instruction Program for Exercises on Finding Axioms, Educational Studies in Mathematics, 4, p. 429.

Goldberg, A., and Suppes, P. (1974), Computer-assisted Instruction in Elementary Logic at the University Level (Tech. Rep. 239, Psych. & Ed. Series), Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, Calif.

Levine, D. R. (1973), Computer-based Analytic Grading for German Instruction (Tech. Rep. 199, Psych. & Ed. Series), Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, Calif.

Papert, S., and Solomon, C. (1972), Twenty Things to Do with the Computer, Educational Technology, 12(4), p. 9.

Schupbach, R. (1973), Toward a Computer-based Course in the History of the Russian Literary Language (Tech. Rep. 221, Psych. & Ed. Series), Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, Calif.

Smith, R. L., Graves, W. H., Blaine, L. H., and Marinov, V. G. (1975), Computer-assisted Axiomatic Mathematics: Informal Rigor, to appear.

Suppes, P. (1957), Introduction to Logic, Van Nostrand, New York.

Suppes, P. (1960), Axiomatic Set Theory, Van Nostrand, New York.

Suppes, P. (1972), Computer-assisted Instruction at Stanford, in Man and Computer (Proceedings of international conference, Bordeaux 1970), Karger, Basel, p. 298.

Suppes, P., and Binford, F. (1965), Experimental Teaching of Mathematical Logic in the Elementary School, The Arithmetic Teacher, 12, p. 187.

Suppes, P., Jerman, M., and Brian, D. (1968), Computer-assisted Instruction: Stanford's 1965-66 Arithmetic Program, Academic Press, New York.

Suppes, P., and Morningstar, M. (1969), Computer-assisted Instruction, Science, 166, p. 343.

Suppes, P., and Morningstar, M. (1972), Computer-assisted Instruction at Stanford, 1966-68: Data, Models, and Evaluation of the Arithmetic Programs, Academic Press, New York.

Van Campen, J. (1973), A Computer-based Introduction to the Morphology of Old Church Slavonic (Tech. Rep. 205, Psych. & Ed. Series), Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, Calif.

NOTE: The research reported herein was supported by the National Science Foundation under NSF Grant EC-43997 to Stanford University.