DOCUMENT RESUME

ED 257 659 SE 045 755

AUTHOR Tinker, Robert F.

TITLE Science and Mathematics Software Opportunities and Needs (SAMSON) Project. Final Report.

INSTITUTION Technical Education Research Center, Cambridge, Mass.
SPONS AGENCY Department of Education, Washington, DC.
PUB DATE Aug 84
CONTRACT 400-82-0022
NOTE 167p.; For the related executive summary, see ED 241 320. Appendix C contains marginally legible print.
PUB TYPE Reports - Research/Technical (143) -- Reference Materials - Directories/Catalogs (132)
EDRS PRICE MF01/PC07 Plus Postage.
DESCRIPTORS *Computer Software; Courseware; Elementary Secondary Education; Higher Education; Instructional Material Evaluation; *Material Development; Mathematics Education; *Mathematics Instruction; *Microcomputers; Science Education; *Science Instruction; *Teacher Education
IDENTIFIERS Mathematics Education Research; Science Education Research; *Software Evaluation

ABSTRACT

This study analyzed the extent to which oppportunities created by computer technology addresses the needs in school science and mathematics instruction and determined what the appropriate microcomputer responses might be to those needs. Information was gathered by obtaining descriptions of most available software; reviewing published software evaluations, grant-supported software development projects, and a broad selection of software; and by consulting experts in the field, school personnel, and software developers. Findings are reported as they relate to the state-of-the-art (considering software development, availability, topic coverage, evaluation, information dissemination, and acquisition and use) and to software potential (examining software for increasing learning, for greater achievement, and for improved teaching productivity and reduced costs). Three findings are these: although software development is expensive and risky, software production is high; schools allocate inadequate resources for software acquisition; and software can increase the range of science and mathematics successfully covered. Recommendations based on these and other findings focus on teacher training and support, software development (particularly software that promotes problem-solving skills), classroom implementation, research, and ways to disseminate information about software (such as microcomputer resource centers and software reviews). The report concludes with an extensive bibliography followed by three appendixes: (1) a list of names and addresses of math and science software vendors, (2) a paper by Harold G. Peters entitled "The Future of Computers in Science Education: An Imaginary Dialogue," and a directory of sources of software information. (JN)

Contract No. 400820022

Office of Research and Educational Development
U.S. Department of Education

SCIENCE AND MATHEMATICS SOFTWARE OPPORTUNITIES AND NEEDS (SAMSON) PROJECT

Final Report

Submitted by:
Technical Education Research Centers
1696 Massachusetts Avenue
Cambridge, Massachusetts 02138

August 1984

Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the U.S. Department of Education.

TABLE OF CONTENTS

SCIENCE AND MATHEMATICS SOFTWARE OPPORTUNITIES AND NEEDS

Robert F. Tinker

Technical Education Research Centers

I. INTRODUCTION

The current state of mathematics and science instruction in U.S. schools is inadequate to the needs of a technological society. At the same time, there are major developments underway in the form of increasingly powerful microcomputers and sophisticated educational software. To what extent can the opportunities created by computer technology address the needs that are being uncovered in school mathematics and science instruction?

This question is the subject of the following report, a summary of research performed for the U.S. Department of Education between September 1982 and May 1983. In addressing the topic, within the province of mathematics and science instruction, we have obtained descriptions of most available software, reviewed many published software evaluations, acquired a broad selection of software, reviewed most grant-supported software, and interviewed a wide range of teachers, supervisors, school administrators, school support personnel, and software developers.

There seems to be general agreement that instruction in math and science at the precollege level is in serious trouble in the United States today. At the same time, microcomputers are being installed in schools in record numbers. Will these microcomputers have a significant effect on improving math and science education?

In order to investigate this question, we have carefully examined most of the software available and also interviewed a large number of computer-using teachers in math and science. All available evidence seems to indicate that software can improve learning, lead to greater achievement on the part of students, improve teaching productivity, and reduce costs. As software developers gain more experience with educational applications and

the available hardware, the quality and educational value of the software seems to rise. Products which are entering the market have been carefully developed, often with Federal funding, and represent solid additions to the resources available to teachers.

We feel that the more creative software, such as simulations, microcomputer-based laboratories and databases, have the greatest potential to substantially increase the amount of learning that can take place at the precollege level. Software of this sort can address process-oriented goals, such as problem-solving and scientific thinking, as well as give students an increased understanding of math and science topics.

The State of the Art

A surprisingly large amount of software is available for use on microcomputers and is appropriate for math and science educational applications, K-12. We have found that, as of May, 1983, there were 971 commercially available software titles in math and 739 in science, and that approximately 100 new titles were being added each month. To this must be added the large number of public domain and locally developed software that is available to educators. Significantly, more than half of the software runs only on the Apple, especially in the more advanced areas of math and science where the market is thin.

In spite of the large number of titles, there is far from uniform coverage of the math and science topics that could be taught at the precollege level. There is almost no elementary science software, and many high school math topics have no supporting software. For instance, software in biology is dominated by games and simulations, many in ecology, and 14 titles treat genetics. On the other hand, equally important biology topics, such as human physiology and medicine, are each treated by only one title.

Dissemination

It was hard enough for us with Federal funding to locate most of the available math and science software; it is putting a tremendous burden on local teachers to expect them to reach the same level of expertise. Not too surprisingly, we found that many teachers desperately want more information about software and are often unaware of software that already exists. Teachers

find out about software through many routes, but most importantly, through magazines and journals.

It is often quite difficult for schools to purchase software, since there is a lack of appreciation of the importance of good software, uncertainty as to its quality, and no history of a budget for software. It was no surprise, then, that the most popular way that math and science teachers used computers involved no software at all, but rather involved teaching how to program in BASIC. This statistic is probably due in part to the fact that many math and science instructors are unusually computer-literate and are called upon to teach computer programming courses. Also many teachers feel that the most educationally sound way of using the microcomputer is to have students solve problems by doing their own programming.

For teachers using software, drill and practice was most popular. Seventy-five percent of the respondents used drill and practice software. About two-thirds used educational games and computational tools, and about half used simulated labs and other simulations. In keeping with these use statistics, we find that an even higher percentage of the commercially available titles are either in tutorial or drill and practice style, including about 90% of those in mathematics.

Teachers have difficulty integrating software into their classroom activities. A large fraction of the software is single topic; i.e., it is designed to explicate a narrow group of ideas in a particular discipline or area. Because the set of single topic software does not provide uniform coverage, teachers must be opportunistic at using software when and if it fits the material they want to cover. On the other hand, there are some comprehensive software packages that cover a range of topics over an extended period. However, comprehensive packages can be even more difficult to use in the classroom because any given package may not address the topics the teacher wants to cover at the reading level and concept level that is deemed appropriate for the particular classroom. The very size of these packages make them difficult to review.

The use of microcomputers in mathematics is advanced compared to its use in science. Mathematics teachers tend to have brought microcomputers into middle and upper schools and often have more microcomputers available for their use. Science teachers may have only one or a few microcomputers accessible to them, and thus tend to emphasize applications such as demonstrations that require only a single computer per class. Thus the large number of tutorials and other applications that assume one or a few students per computer cannot readily be utilized by most teachers at this time.

The Software Potential

We have found it convenient to divide software into two categories on the basis of its educational style and teaching strategy. One style teaches material explicitly through tutorials, dialogs, or drill and practice. This style is by far the most popular, best represented in the available products, and the most researched. This style is probably most effective when there are a lot of facts or procedures to be learned. Research tends to indicate that software in this style does lead to faster learning and, particularly in remedial applications, better learning. Much of the research that supports these conclusions is based upon mainframe computers communicating with Teletypes, a mode which is much less attractive than the current generation of microcomputer-based software that uses graphics, animation, and quick interaction. Thus, we expect that explicit instructional materials prepared today with the best software would show up even better in terms of reduced time on task and increased achievement.

The second style of software includes a number of different teaching strategies, all of which have the student learn through exploration or use of the computer as a problem solving tool. We have termed these implicit styles, since the material to be taught is implicit in the software but not expounded explicitly. Examples of software in this style include:

- Microworlds: Cybernetic environments in which students can explore and solve problems.
- Games in which the material to be learned is an intrinsic part of the game and must be mastered to improve your score.
- Microcomputer-based laboratories in which the computer is turned into a powerful instrument students can use to analyze, display, and save data from experiments.
- Databases: Large collections of data that students can access.
- Tools: Software that solves specific computational or display problems such as graphs and equation solvers.
- Computer languages: General purpose software that

students use to program solutions to problems.

- Simulations: Models of real situations that provide an opportunity for students to learn about systems that can't be brought into the classroom because of cost, time, danger, or other reasons.

Software of this sort can address process-oriented goals, such as problem solving and scientific thinking, as well as giving students an increased understanding of math and science topics. It is difficult to definitively establish the effectiveness of this kind of software because it is both hard to compare with other approaches and hard to measure process goals. However, there is some evidence and considerable expert opinion that attests to the value of well-designed software of this kind.

Cost Savings

Computers are sometimes justified on the basis of their ability to lower costs by allowing faculty to reach a larger number of students. There is little evidence to indicate that this is the case with present software, and little hope that savings of these sorts will be realized in the future in math and science instruction. Most teachers and administrators that we have spoken to feel that the introduction to computers actually makes the teacher's job more complicated. While raising the quality of the instruction, it does not follow that this quality can be diluted over a larger number of students.

In certain specialized cases, cost reductions have been achieved. When the computer is used to manage already existing self-paced programs, some clerical costs can be reduced. When equipping a new laboratory, it may be less expensive to equip it with lab-interfaced computers than to purchase the equivalent electronic test instruments. Finally, in certain situations where there is a lot of drill and practice required, especially in remediation, the use of the computer can free some staff to make more effective use of its time. These are the only cases where we found microcomputer software could result in cost savings. However, it is possible that with the next round of technology which will permit much more complex software, and through the use of expert systems and computer-managed dialogue, there could be much broader areas of cost reduction.

A New Scope and Sequence

The most significant impact of microcomputer software on education will be through the changes it both requires and makes possible in the appropriate scope and sequence of topics covered in the school curriculum. Microcomputers allow for less emphasis on certain topics currently covered in the curriculum and make possible the introduction of new ideas and new topics which previously were not covered at a particular grade level. There can be much less emphasis on arithmetic computations and on rote memorization. Plane geometry can be introduced much earlier and its important concepts can be taught in far less time. The idea of proof by theorem, which is often linked with geometry, can be introduced in another context, making geometry itself more accessible. Numerical techniques that are used to solve differential equations can be introduced as soon as students have completed the equivalent of a first-year course in algebra, long before they know what a derivative is.

In the space created by these changes, a number of additional topics can be covered. The empirical basis of much of high school science can be introduced in the fourth grade with microcomputer-based laboratories. High school students can solve college-level problems using numerical techniques. Students can be introduced through the computer to psychology, physiology, perception, nutrition, health, oceanography, geophysics, and many other topics. Little is known about the appropriate educational levels for the introduction of some of these ideas through the computer. Some anecdotal evidence indicates that well-designed software has the ability to make some very abstract ideas quite concrete and accessible at much earlier stages of intellectual development than was ever previously thought possible.

Recommendations

Microcomputer hardware, together with appropriate software, has the potential of making a major impact on math and science instruction in the United States. However, this will not happen automatically, but rather requires sustained major commitment at the federal, state, and local levels to insure that this potential is turned into a reality.

Teacher Training

The highest priority area is in teacher training, both pre-service and inservice. Teachers have to pick up a whole new set of skills to use the computer and to select appropriate software, but more importantly, to learn new approaches to teaching permitted by the software and to learn new topic areas which can now be taught with the aid of microcomputers. This implies that every math and science teacher and every elementary teacher needs to be reached, not once, but several times through in service training, workshops, summer institutes, conferences, or TV courses. Because of the cost of such a massive amount of in service training, most of it will have to be borne by the local school districts through workshops and training provided by local resource people. However, there is a tremendous need to prepare materials for these resource people, to train the trainers, and to provide them with state-of-the-art materials. At the same time, we should explore the utility of a technology-based teacher training mechanism such as cable TV and videodisks. A similar effort should be applied to the pre-service training of teachers. It is crucial that the next generation of teachers be aware of microcomputer software that is available and ways of using it creatively in the classroom.

New Software

While there are a large number of software titles being produced, there is a need for much more software, particularly programs that focus on problem-solving skills and allow students to undertake their own investigations. There is a particular need for major funding that underwrites creative but high-risk developments for the long term. Logo would not have been developed by a commercial vendor in today's market. It required massive Federal assistance and long-term support. It seems reasonable to expect that there are other equally important ideas that await comparable funding for their development and exploitation. We do view the current commercial market for educational software to be quite healthy and able to produce creative, innovative software, particularly on the part of vendors serving important but small market areas. This process should be enhanced by increasing the amount of software that is reviewed and the amount of information about software that is disseminated to software users.

There is a particular need to develop new software in math and science that is designed for special populations, such as Spanish-speaking, physically disabled and learning disabled students.

Software Implementation

All the software in the world is of no use unless it is actually used in the classroom. Schools should budget for software and should provide sufficient hardware so that all students can augment their math and science instruction through the use of computer software. Schools can increase the impact of software by coordinating the schoolroom use with the computers that are or soon will be available in most homes. This connection can be made by having the school support group purchases of hardware and software, by establishing a software circulating library, and by loaning or renting hardware to homes that do not own appropriate computers.

Many teachers report that it is not easy to incorporate microcomputer software into their curriculum. They need assistance and suggestions for specific ways of augmenting the topics they teach or would like to teach with microcomputer software, and of integrating the software into the classroom.

Research

There is an urgent need for research on computer-mediated science and math learning. Much of the available software does not have a theoretical basis. It is difficult to know whether the software is even effective, and much more difficult to know what learner characteristics are necessary for its effectiveness. Research is needed on exactly how radical a departure from the traditional math and science curricula is possible with microcomputers. Finally, it is of utmost importance to research how equitably microcomputer software is being made available to all members of our society.

II. SAMSON PROJECT BACKGROUND

A. Approach

The goal of this study was to analyze the needs in science and math education and to see what the appropriate microcomputer responses might be to those needs. Neither the time nor resources available permitted a statistical sampling approach to these issues.

Our approach was formative and primarily qualitative. Our information came from experts in the area and from school-based users of microcomputers. In locating interviewees we strove for breadth in terms of geography, computer experience, educational level, and educational role.

B. Methodology

We began with a literature search and an analysis of available software. We then assembled software reviews and surveyed a large number of practitioners. Our work was augmented by expert consultants and an able advisory board.

Literature Search

The ERIC database served as the point of entry for an exhaustive search of the literature published from 1978 through early 1983. We studied the broad issues of teacher needs and educational trends as well as the specifics of microcomputer use in the math and science curricula. Over 300 pertinent articles, conference proceedings, unpublished papers and theses were identified in the ERIC Current Journals in Education and Resources in Education and from a search of the RICE and SPIF databases. The bibliographies in these sources often led us to

more material, and our search continued throughout the term of the project. A complete bibliography of pertinent references appears at the end of this report.

Software Search.

Over 600 commercial vendor names were drawn from the TERC-Dresden Associates joint publication, the Software Finder (Haven, 1983) and the directory, Microcomputers in Education: A Resource Handbook (Woodruff, 1982). Others were identified in citations in articles and reviews, advertisements and brochures. Of the 719 companies contacted by mail, 160 respondents did produce math and science microcomputer software, 34 did not have relevant software, and 425 did not reply at all. A complete list of vendors supplying math and science software is included in the Appendix.

In order to locate non-commercial sources of software, grantees from the National Science Foundation, the Minority Institutions Science improvement Program, the Apple Education Foundation, and the Atari Institute were contacted by mail and encouraged to report on their projects. We learned of over 60 grant-supported software programs, many still under development or without a distributor. We established a card file with descriptive information on these programs, including expected completion date if known.

These software data were assembled into an electronic database we call the Electric Software Finder (ESF). This database is written in dBASE II and runs under CP/M on an S-100 system with dual 8" disks. Remote access to this database was possible during the project. All data are current as of May 31, 1983.

Out of the hundreds of software descriptions received, a number of programs seemed innovative enough to warrant closer examination. In response to our written requests we received for viewing over 100 software programs from 13 vendors.

Reviews and Evaluations

We have assembled an exhaustive collection of review and evaluation materials. Five hundred math and science software reviews indexed in the Winter 1982 issue of School Microware Reviews (Haven, 1982) and covering the period 1979-1982 are on file. The RICE database, 1979-May 1983, and the Microcomputer Index 1981-1983, were searched and printouts made of all math and science software review entries.

Networking

Twenty-three professional groups were contacted by mail. The associations selected have national constituencies and provide support for their members through meetings, journals, and newsletters. To ensure the currency of our selections in the rapidly expanding educational computing field, the Encyclopedia of Associations, 17th Edition (Gale Research, 1983), was consulted for new organizations not yet known to us.

Survey

In order to assess the state of the art, interviews were conducted with 70 developers, teachers, and administrators associated with the production, dissemination, or use of mathematics or science software. Interviews were based on questionnaires tailored to each category of software user. Sessions characteristically ran 40 minutes to an hour. We contacted urban, inner city, suburban, and rural school personnel.

One hundred sixty developers and school-related personnel involved in math and science projects were selected from the Microcomputer Directory: Applications in Educational Settings

(Gutman, 1982). These people were surveyed through mailed questionnaires. Two regional groups - EDCO in Massachusetts and the Regional Education Media Centers of Michigan - distributed questionnaires to their members. The respondents were selected to include policy makers, administrators, and teachers and to represent various levels of experience wth microcomputers, a range of grade levels, and different social settings from communities throughout the U.S. Table II-1 lists the number of respondents in each of these categories. Questions appropriate for individuals in each cell were developed. Table II-2 contains aggregate data from Table II-1, showing the overall balance among various categories. These statistics do not reveal the geographic spread of respondents, who ranged from Cambridge, Mass. to Joshua Tree, California, from Maple City, Michigan to Austin, Texas. Of the 217 respondents, 188 were from outside Massachusetts, 174 were outside New England, and 67 were west of the Mississippi.

Two roundtable discussions held at North Carolina State University brought together area users and developers of math and science software as well as people attending a TERC workshop being held at the University. A description of this project and questionnaires were mailed to invited participants prior to the meetings and formed a useful focus for the lively and informative discussions.

A cross-section of software developers was also surveyed by telephone. The characteristics of this group are summarized in Table II-3.

These project-related activities were supplemented by extensive informal contacts at our workshops, conferences, and microcomputer resource center.

Advisory Board Meetings

Two meetings were held with the Project Advisory Board This distinguished group was selected on the basis of its members' breadth of experience with schools, microcomputers, and/or math and science education. The members of the Project Advisory Board are listed below.

Advisory Board Members

Stacey Bressler
Educational Specialist
Apple Computer, Inc.

Alan Cromer
President
EduTec, Inc.

Robert C. Hayden
Special Assistant to the Superintendent
Boston, MA Public Schools

Carolee Matsumoto
Asst. Superintendent for Curriculum & Instruction
Concord, MA Public Schools

Richard Riley
Educational Computer Consultant
Maine Dept. of Education & Cultural Services

Judah Schwartz
Professor of Engineering Science & Education
Massachusetts Institute of Technology

Thomas L. Sears
General Manager
Compress

The November 22-23, 1982 meeting of the Advisory Board provided an overview of the larger issues, defining the subject and scope of the survey and recommending a more specific analysis of currently available software. The April 21, 1983, meeting focused on the procedures, products, and recommendations that flowed from the survey results. The group discussed the conclusions and recommendations as they would best appear in this report.

Consultants

The project profited enormously from a group of consultants, listed below, who assisted in different ways. For example, Bob Haven assembled software reviews, supplied a database of software titles, and analyzed his database for the project. Hal Peters contributed a position paper which is quoted extensively in this report and reprinted in whole in the Appendix.

Consultants

Robert Haven
Managing Director
Dresden Associates

Hal Peters
Director
CONDUIT

Adeline Naiman
Director
HRM Software

Karl Zinn
Research Scientist
Center for Research on Learning & Teaching
University of Michigan

C. Restriction to Mathematics and Science

In this report we refer to mathematics and science exclusively. All other uses of computers, such as computer science, lanaguage instruction, and instructional assistance in other disciplines are excluded. All interviews were with teachers, school staff, and vendors concerned with math and science instruction. Thus, our conclusions and recommendations refer exclusively to the math/science part of the educational microcomputer field.

Table II-1 Survey Respondents by Category

TEACHERS

Grade Level		Experienced (2+ yrs)									Inexperienced (0-2 yrs)									
		Elementary			Secondary			K-12			Elementary			Secondary			K-12			Totals
Subject		M/S	M	S	M/S	M	S	M/S	M	S	M/S	M	S	M/S	M	S	M/S	M	S	
TELEPHONE	I	0	0	0	0	1	0	0	2	3	2	0	1	0	1	0	0	0	0	10
CONTACT	U	0	0	1	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	4
	S	3	0	0	1	1	4	0	0	0	1	0	1	0	3	9	0	0	1	24
	R	1	3	0	0	0	1	1	0	0	2	0	0	0	2	3	0	0	0	12
MAIL	I	0	0	0	1	2	2	1	1	0	0	0	0	1	1	0	0	1	1	11
CONTACT	U	1	0	0	3	6	2	3	6	2	0	0	0	3	3	0	0	0	0	29
	S	4	2	0	3	8	6	5	2	0	1	0	0	2	5	4	0	3	1	46
	R	3	0	0	4	8	6	3	0	0	1	1	0	4	5	3	5	4	0	47
Totals		12	5	1	12	27	21	13	11	5	7	1	2	10	20	21	5	8	3	184

COMPUTER RESOURCE PERSONNEL

	Elementary	Secondary	K-12
I	1	1	2
U	0	4	2
S	3	6	3
R	3	2	0

27 Respondees whose subject matter was computers. Not separated by phone and mail.

Codes Used:
- I = Inner City
- U = Urban
- S = Suburban
- R = Rural
- M/S = Math and Science
- M = Math
- S = Science
- C = Computer Sciences or Computer Resources

Table II-2 Summary of Respondents

Grade Level	Elementary			Secondary			K-12			
Subject	M/S	M	S	M/S	M	S	M/S	M	S	Totals
Inner City	2	0	1	2	5	2	1	4	4	21
Urban	1	0	1	6	10	4	3	6	2	33
Suburban	9	2	1	6	17	23	5	5	2	70
Rural	7	4	0	8	15	13	9	4	0	60
Totals	19	6	3	22	47	42	18	19	8	184
Computer Resource Personnel:										27
SEA Personnel:										6
Total Respondees:										217

Number of Respondees by Geographical Distribution:

Massachusetts	29
New England (including Massachusetts)	43
East (excluding New England)	107
West of Mississippi	67

Table II - 3

Characteristics of Developers Interviewed

SIZE[1]	LOCATION[2]	SUBJECT	STYLE	MICROCOMPUTER
small	SE	math	tutorial, drill & practice	Apple
small	NE	math	drill & practice	Commodore
small	NE	science	tutorial, drill & practice	TRS-80
small	SE	ma. & sc.	test generator	TRS-80
medium	W	math	game, tutorial	Apple
medium	W	math	tutorial, drill & practice	Apple, Atari, Commodore, IBM
medium	W	math	drill & practice	Apple, Atari, Commodore, TI, TRS-80
medium	W	math	tutorial	Apple, Commodore, TRS-80
medium	NE	science	tutorial, simulations	Apple
medium	MW	ma. & sc.	game, tutorial, drill & practice	Apple, Commodore, TI, TRS-80
medium	MW	ma. & sc.	tutorial, drill & practice	Apple
large	MW	math	totorial, drill & practice	Apple, Atari
large	SW	science	simulations	Apple

1. Size
   small - home business
   medium - national distribution
   large - major national publisher

2. Location
   SE - Southeast
   NE - Northeast
   W - West
   MW - Midwest
   SW - Southwest

III. SURVEY OF NEEDS

We must remember that educational software should be introduced not because it exists, but because it fills real needs. Therefore, we must begin our analysis by reviewing and systematizing those needs.

A. Needs: From the Experts

Several studies have been carried out in the past decade to determine the state of math and science education in the United States. Each one has pointed up the paradox that as our society becomes more and more dependent on sophisticated technology, our schools become less and less able to provide the needed math and science foundation.

- We are raising a new generation of Americans that is scientifically and technologically illiterate... Improving science and math achievement among our young people requires a joining together of efforts by educators, parents, the private sector, and all levels of government.... (National Academy of Sciences, Science and Mathematics in the Schools, 1982)

- Deficiencies in numbers and qualifications of mathematics and science teachers are exacerbated by classroom conditions, including inadequate instructional time, equipment, and facilities.... (National Science Board, Commission on Precollege Education in Mathematics, Science and Technology, Today's Problems, Tomorrow's Crises, October, 1982)

- Adequate support for materials, equipment and teacher time must be available for schools to maintain quality science instruction. (National Science Teachers Association, Position Statement, 1981)

- Improved preservice and in-service teacher

education was given top-priority....Problem solving was consistently ranked high in priority....There is consistent and strong support for increasing the emphasis on applications of mathematics throughout the curriculum.... (National Council of Teachers of Mathematics, Priorities in School Mathematics, 1981)

B. Needs: From the Field

We asked people in the schools to articulate their major needs in math and science other than those needs directly related to microcomputers. Many of the responses indicated needs that were unique, but other needs were frequently mentioned and seemed to be independent of variables such as the size of the school, age level of students, or type of population served.

Ninety-three of the respondents (54%) cited need in the area of materials and curriculum development. Needs in this area included individualized materials (15 times), problem-solving materials and ideas (14 times), relevant texts at different reading levels (12 times), new curriculum (10 times), new teaching materials (8 times), applications-oriented material (6 times), more advanced courses for gifted students (5 times), and more experiential or manipulative materials (4 times).

One comment we heard repeatedly (15 times) was either, "I need motivated students," or "I need something to motivate my students!" It would be easy to dismiss this by saying anyone who has motivated students can be a good teacher. But that ignores the fact that many topics in both the math and the science curricula are either outdated, irrelevant or have no activities planned to demonstrate the applicability of abstract concepts.

The expressed concern for individualized instruction is noteworthy. Teachers feel unable to meet the many and diverse needs of their students. They especially need materials that are interesting and relevant to the older students whose conceptual ability is more advanced than their reading abilities. Also needed are units for advanced students who want to go into a topic in depth, as well as material for students with learning disabilities.

The most cited need (27 times) was for more teachers who are certified and qualified to teach math and science. Frequently mentioned by science teachers (7 times) was the need for summer workshops to bring them up to date in their subject area. Seven administrators stressed the need for retraining of present

teachers so they could be transferred to math or science positions. In all, 36 of the 172 respondents (21%) mentioned training as a major need.

The need for increased budgets was mentioned explicitly by only 12 respondents. Others did cite needing increased lab space (8 responses), modern lab equipment (10 responses), and more time for curriculum development (6 responses). These are all needs which would be met by increased school budgets, so that the total number citing budgetary needs amounted to 23% of the respondents.

Both teachers and administrators expressed concern for the state of science instruction in the elementary schools. Most elementary teachers have not received adequate preservice training in science and most elementary schools do not have either science laboratories or science specialists. As a result the classroom teachers feel inadequate to the task and very little attention is paid to science at the age level when children are "natural scientists." The other concern is that much of what is done in the name of science alienates the children so that they do not choose science courses when they get to high school.

At both the elementary and secondary level there was a need for more laboratory activities. Four respondents (2%) stated this as the major problem in science education. A laboratory activity requires planning, preparation time, experiment time, clean-up time, and follow-up discussions or activities. It is easy to see why laboratories are prime targets for elimination in an era of budget cutbacks and expanding curricula when there is not enough time to cover the needed material. But most educators expressed a belief in the value of experiential learning and stressed the need for updating and expanding the role of hands-on laboratory activities.

IV. THE STATE OF THE ART

We made a major effort in the SAMSON Project to find out what software was available and used in the area of math and science. We attacked this problem from both the developer's and the user's perspective. Our analysis follows the software from its development through its use, and asks how it is developed, how much of it is available, how it is reviewed and evaluated, how teachers find and acquire it, and how they actually use it in the classroom.

A. Software Development

Commercial Development

There is a significant amount of educational software being produced for math and science in spite of the relatively large cost of development. In response to a letter to over 700 vendors, more than 160 returned catalogs or product descriptions which included at least one software product that could be used in math or science instruction. Although this is a large number, it may be inflated. We are not sure whether all of these vendors actually have products that they can deliver. It should be noted that people in small companies often advertise products before they have been completed, and misjudge the time and expense involved in bringing a product to market. We were not able to evaluate products from all of these developers, and thus we cannot say with certainty that there are 160 vendors offering available software.

We estimate that there are 100 pieces of new math and science software being developed each month at the present time. This is a very approximate estimate, based upon the number of titles in the Software Finder and School Microware Directory over time, combined with our finding that the latter directory covered approximately two-thirds of the available products. The number of math and science titles in this directory seems to be growing exponentially, with a doubling time of slightly under two years. That means that two years ago there were a little less than half as many titles in the

math and science areas, and the rate of production was a little less than half the current rate. If historical trends hold, that means that there will be approximately 225 new titles a month in math and science by May of 1985.

These statistics on the rate of production can be somewhat misleading, since they lump together large packages under a single title with very small, single-concept programs. For the purposes of this study, a title available on two or more computers counted as only one title. On the other hand, closely similar software packages with different titles which come from different vendors did get counted as separate packages. These methodological issues aside, the 100 per month number seems reasonable, based on the number of developers. No developer is content with a static catalog. Most want to add a number of titles every year. Thus, the growth of offerings combined with a reasonable growth in the number of vendors would certainly lead to an increase on the order of 100 new titles a month.

Production Costs

Cost data are difficult to obtain because of the natural disinclination of commercial publishers to reveal sensitive information. However, it may safely be said that quality software is extremely expensive to produce. In order to get an idea of software costs, we have to do it on a case-by-case basis. This sort of information can be very misleading, for a number of reasons. First of all, the software costs vary enormously from one type of software to another. A drill and practice might represent a weekend's work, whereas a sophisticated tool might cost $100,000, as Hal Peters estimates ARBPLOT did. (See his position paper in the Appendix). Secondly, much of the cost of software goes into conceptualization, testing, and the development of general-purpose software tools. This might result in the cost of the second package in a particular format being far less expensive than the first.

A few more data points can be gleaned from TERC's own software development experience. We developed a series of energy software packages for approximately $5,000 each and are currently modifying them as a result of classroom testing, for approximately another $2,000 each. On the other end of the spectrum, we have a major, original development project which requires sophisticated software in a new style, which will cost approximately $70,000 to develop, and which will result in the publisher spending a like sum to produce.

One of the major problems facing educators and producers is the wide variety of microcomputers available and the expense and difficulty of producing software for each of these variants. While it is possible to write software that can run on a variety of different machines, a number of practical and economic factors have made this very uncommon in education. In order to run the same software on many machines, there is inevitably a sacrifice in the performance of the software, particularly in the sophistication of the graphics and the speed of its execution. In addition, there are economic disincentives for the manufacturers: they want to encourage development of software that only runs on their hardware.

However, this transportability problem is not as much a barrier as it may seem at first glance. Many educational applications have two characteristics that are important in this regard. First, they are often not terribly complex. Secondly, the cost of actually writing the code is a small part of the total development costs, which also include student trial, documentation development, and revision. Thus, reprogramming the application to run on a different machine is not nearly as expensive as bringing out a product in the first place. Two developers reported that this involved one-quarter of the cost of producing the original product.

Software Pricing

Software is most often priced in the $10-25 range, although significant amounts of software are priced up to $100. (See Figure IV 1.) There is a slight difference between math and science pricing, with relatively more higher priced science software. The detailed breakdown by grade, Figure IV-1 and Tables IV-1 and IV-2, reveals a slight but significant trend toward higher price at higher grades, especially in science.

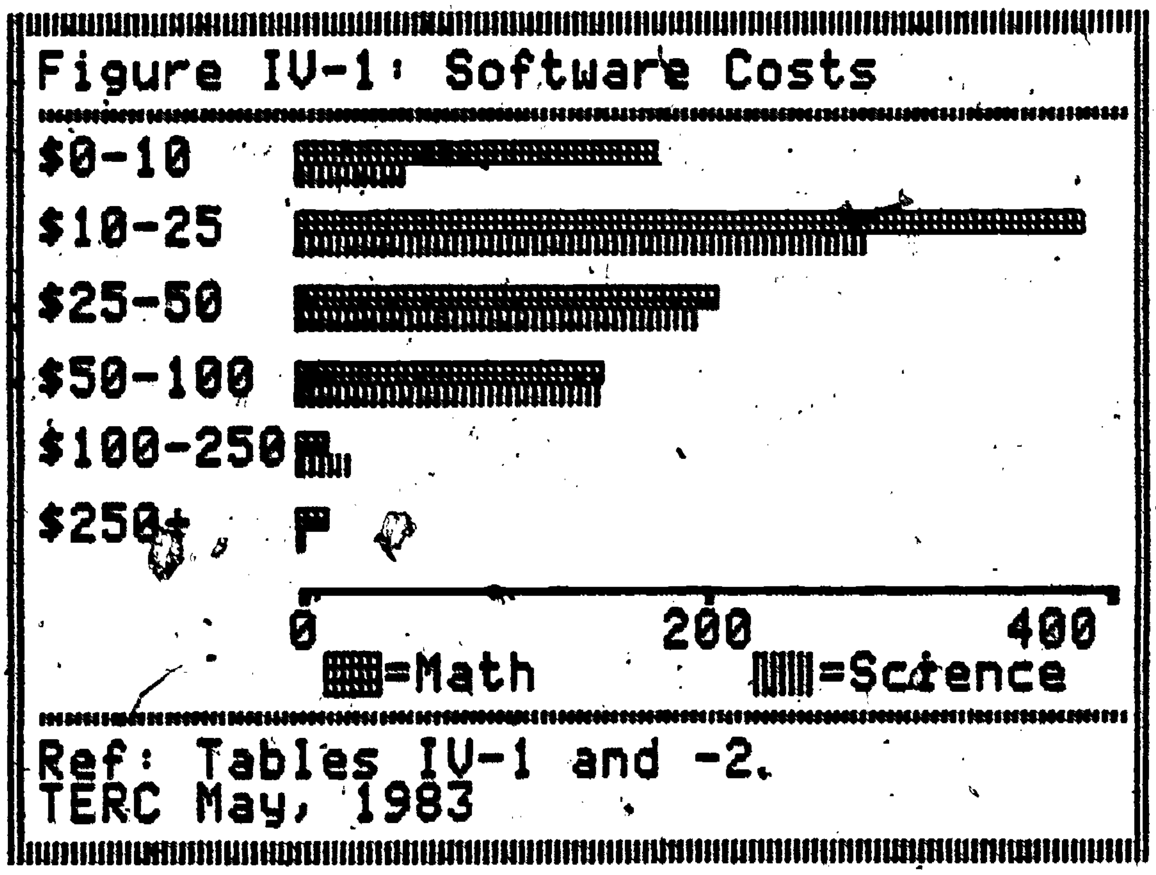


Figure IV-1 Summary of Software Costs
The entries represent numbers of titles.

Piracy

Software piracy is a major problem for commercial software developers and distributors. Electronic Learning reported the results of a survey of 15 producers in which it found copying was the number one problem reported by these producers. Copying was seen as hurting revenues, quality, and price. It is interesting to note that the Minnesota Educational Computing Consortium (MECC) has recently altered its three-year program of giving licensees unlimited copying rights. It found that this produces an atmosphere that condones copying and that licensees do not respect either MECC's legal rights nor those of other software producers. MECC henceforth will produce copy-protected software which will be available to licensees at approximately the cost of a disk, but it will also aggressively protect its rights under the license agreements and punish unauthorized copying.

Many commercial developers feel that educators will, as a matter of course, produce unauthorized copies. While we have

Topic	No. Titles	D&P	Sim	Game	Data	Demo	Tutr	Comp	Test	Mult	None
		PROGRAM STYLE									
Chemistry -- All Topics	171	85	37	2	1	4	39	7	1	1	5
Chemistry (Acid-Base Chemistry)	18	8	4	0	0	2	3	1	0	0	0
Chemistry (Atomic Struct. and the Periodic Table)	31	20	3	1	1	0	4	1	0	0	1
Chemistry (Chemical Composition)	6	0	5	0	0	0	0	0	0	0	1
Chemistry (Compounds and Formulas)	22	17	1	0	0	0	2	1	0	0	1
Chemistry (Miscellaneous)	6	1	2	0	0	0	3	0	0	0	0
Chemistry (Nucleonics)	3	2	1	0	0	0	0	0	0	0	0
Chemistry (Reactions/Equilibria)	25	8	8	1	0	0	8	0	0	0	0
Chemistry (Organic Chemistry)	15	3	1	0	0	0	10	0	0	0	1
Chemistry (Solutions)	10	8	0	0	0	0	1	1	0	0	0
Chemistry (States of Matter)	20	7	9	0	0	1	2	1	0	0	0
Chemistry (Stoichiometry)	9	7	0	0	0	0	0	1	0	0	1
Chemistry (Multiple Topics)	7	1	1	0	0	0	3	0	0	1	0
Chemistry (No Topic)	9	2	1	0	0	1	3	1	1	0	0
Physics -- All Topics	210	55	60	8	0	11	51	17	0	1	2
Physics (Electricity)	52	19	13	0	0	0	15	0	0	0	0
Physics (Heat)	15	5	2	0	0	0	4	4	0	0	0
Physics (Light)	18	8	4	1	0	2	2	1	0	0	0
Physics (Mechanics)	59	12	24	3	0	1	13	4	0	0	2
Physics (Nuclear Physics)	11	0	8	0	0	0	1	2	0	0	0
Physics (Relativity)	0	0	0	0	0	0	0	0	0	0	0
Physics (Sound)	7	4	0	0	0	0	2	1	0	0	0
Physics (Waves)	24	4	7	0	0	4	9	0	0	0	0
Physics (Multiple Topics)	2	0	2	0	0	0	0	0	0	0	0
Physics (No Topic)	22	3	0	4	0	4	5	5	0	1	0
Astronomy	31	2	9	2	1	2	11	4	0	0	0
Basic Skills	34	3	1	3	0	1	25	0	1	0	0
Biology	133	6	55	5	1	1	52	12	1	0	0
Earth Science	36	7	7	1	0	0	20	1	0	0	0
Ecology/Environment	38	0	34	0	1	1	1	1	0	0	0
General Science	18	3	2	5	0	0	6	2	0	0	0
Geology	0	0	0	0	0	0	0	0	0	0	0
Multiple Subjects -- Science	5	0	0	0	0	0	1	1	0	1	2
Natural History	6	0	0	0	0	0	6	0	0	0	0
Physical Sciences	22	0	0	0	0	0	21	0	0	0	0
Technology Education	3	0	1	0	0	0	2	0	0	0	0

Table IV-II Science Topics By Software Style
See text for an explanation of the software style categories.

found that most educators do respect copyright laws, it takes only a few people producing unauthorized versions to give a bad name to the entire profession. Illegal copying makes publishers wonder whether they can recover their development costs in the educational market. They either go into other kinds of software development, or raise their prices so that they can recover this cost quickly before illegal copying eliminates the market.

We are not aware of any underground network that distributes unauthorized educational software in a wholesale manner. Whether or not software piracy is, in reality, an important factor, the fear of it is real and has the clear effect of inhibiting production and dissemination, limiting innovation, and driving up prices. It limits the amount of capital and time developers are willing to invest in a piece of software; it increases the wariness of distributing free software for review; and it makes the developers want to recover their costs quickly by raising the price.

The Development Process

Many of the developers of educational software are extremely small companies, often a single individual who is at the same time teaching in high school or college. This person has typically developed software for use in her or his own teaching and has polished the software and provided some documentation so that it might be of use to colleagues in similar situations. This sort of configuration provides a very efficient, low overhead means of generating and disseminating software. The original teacher did not need an extensive classroom test methodology; the software grew out of classroom needs and was continually used and tested by typical students. The company overhead is extremely low, being run on spare time out of the home. Dissemination is simple, since the teacher usually has professional contacts. By giving talks, or by distributing a Xerox sheet, or through a donated workshop, this person can easily reach a number of colleagues who might buy the software. If the initial software offerings are accepted and purchased, the teacher often produces similar material on related topics. In this way, an individual might develop a number of titles of similar form and substance. This product line specialization helps the small vendor inexpensively develop brand identification and product loyalty.

Most commercial software has been written by a single individual on a royalty basis without advance. Typical royalties paid currently range between 10 and 20% of sale income. Of course, the large number of very small vendors represents the attempt on the part of individual authors to avoid this low

percentage by attempting to handle all of the advertising, sales, and distribution.

The Home Market

There is great commercial interest currently in developing products for the home educational market. There is a growing understanding that many microcomputer buyers have some educational objective in purchasing their hardware. There is also the lure of a two-billion household computer market as opposed to the three hundred thousand microcomputer market currently in schools. Many companies are attracted to the home educational market because of the possibility of placing their software in read-only memory cartridges, which are relatively safe from unauthorized copying, at least by amateurs.

It seems that, at least in some cases, software marketed for the home can be successfully used in the school environment. The converse is rarely the case. Very few educational products aimed at the schools are marketed for home use. This property of the market has led many publishers to focus educational software on the home market and then extend the marketing effort into the schools.

The Role of Grant Support

It is surprising how many commercial software products have originated with some form of government or private foundation funding. The early funding that used mainframes has, of course, spawned BASIC and Logo. Less well known are the Huntington I and II simulation projects, which have inspired dozens of microcomputer implementations of their simulations. The PLATO Project at the University of Illinois is responsible for the initial development of Stan Smith's organic chemistry and general chemistry courses, of Darts, a popular estimation game that is available in many forms, and of a large number of elementary mathematics programs. National Science Foundation (NSF) funding to CONDUIT was responsible for the production and dissemination of a large amount of teacher-originated software which has now been made available on microcomputers. More recent NSF and Apple Foundation funding has been responsible for the initial development of such products as Rocky's Boots, Discovery Learning in Trigonometry, Algebra Drill and Practice, and many, many more titles. Alan Cromer, an Advisory Board member, acquired some Apples for his undergraduate physics laboratory through an Institutional Scientific Equipment Program (ISEP) grant. One condition of the grant

required him to write some software, and this led to the formation of a new company, EduTech, that has now grown into an important supplier of educational software.

There is some evidence to indicate that there are significant new software projects developed with funding support that will result in products reaching the market within the next 12 months. In part, these projects were funded by the National Science Foundation, Education Directorate (prior to its reorganization and downgrading to an office), and by the Apple Foundation. Other promising new software is based on substantial private funding.

School-Generated Software

In addition to commercial software, which 95% of respondents use, non-commercial software is also important in schools. Three quarters of respondents use software generated by their school faculty or students. In some schools this is the only source of applications software. This is especially true when the computers used have only limited graphics and use cassette tapes for storage, since under these circumstances the performance of much of the commercial software can be duplicated by beginning programmers. In some cases, advanced programming classes in a school actually solicit educational programming projects from other teachers and use advanced students to develop educational software with faculty supervision. This provides worthwhile programming projects, helps create a sense of community within the school, and fosters the development of new relationships between students and teachers.

Public Domain Software

Another major source of new software is public domain software distributed through journals or swapped by clubs, dedicated individuals, in-service trainers, and special projects. One-half of respondents reported using software from one of these sources.

One of the most active educational public domain projects is Softswap, operated by the Computer Using Educators (CUE) (Lathrop, 1983). They offer over 200 titles on 65 disks for 7 different microcomputers, including a 33-disk authoring system called BLOCKS. Anyone can copy the programs at the CUE Microcomputer Center or at county school offices throughout

California. The software can also be ordered at $10 per disk. Over half of the titles in this collection cover mathematics or science topics. Most computer clubs and many schools maintain collections of public domain software with some part of the collection being of interest to educators. CompuServe, a public-access database utility, has the ability to accept (upload) and transmit (download) microcomputer software which is used to swap short educational programs.

The problem with all of these public domain sources is quality. Some programs may be excellent, others may not run, others may execute but contain factual errors. They usually come without documentation, so it is necessary to preview large numbers of titles to find software for a particular application. With no documentation, each teacher using the software must generate any related student materials and must fit the software into the curriculum.

Another source of public domain software is Project Seraphim (Moore, 1983). This NSF-sponsored project collects chemistry software, evaluates it, and makes it available with documentation for $4 per disk. Most computer magazines also publish public domain programs, usually with good documentation. Notable in this regard is Dr. Dobbs' Journal, which has been giving away significant software almost since microcomputers were first commercially available.

Low-Cost Software

There are alternatives to commercial and public domain distribution of software that are important to mathematics and science educators. These alternatives are important to consider because they can provide low-cost quality software.

The Minnesota Educational Computing Consortium (MECC) was the software source most often cited by our respondents. MECC originally distributed teacher-generated software, but now employs professional programmers to develop new software and translate MECC software to other machines. MECC has an institutional membership plan whereby a school or group of schools is licensed to receive the entire MECC library of software. Whether it is the low cost of the software, the license agreements, or their early entry into educational computing, MECC software is a major force in educational computing.

Another less well-known model for low-cost software distribution is the Atari Program Exchange (APX). A separate company founded and controlled by Atari (Warner Communications),

its primary mission was to enhance Atari computer sales by publishing user-contributed software. Programs are sold with reasonable documentation in the $20-$50 range with the developer receiving a royalty and retaining copyright. A wide range of software is available including explicitly educational titles and others, like graphing utilities, of educational significance. Pascal, Forth, and a sophisticated symbolic algebra/calculus package are all available at surprisingly low cost through the APX.

Non-Commercial Software

A great deal of software is developed by non-commercial sources. Individual teachers, advanced programming classes, districts, schools of education, and state agencies all produce educational software. Whatever the source, it is usually tailored to a particular teaching environment. The motivation for this in-house development is either to learn programming, to save money, or to fit a perceived need--sometimes quite narrow--which is not being met through other software.

Authoring Systems

Surprisingly, we found that authoring systems--software designed to simplify the preparation of educational packages--were not widely used. A wide selection of authoring systems are available, from simple "shells" - easily-prepared fixed format drill and practice - to full-fledged languages like PILOT that allow nearly complete flexibilty in the finished software. It is not clear why these authoring systems are not in wider use. Perhaps the simpler systems are not well-known or judged to produce trivial software while the more sophisticated ones are seen as being as difficult as BASIC to learn to use, with less commercial significance. Publishers are quite interested in authoring languages because they simplify the software production process.

B. Availability

Quantity

A substantial amount of software is available, although the coverage is by no means uniform in terms of quality, grade level, topics, or type of microcomputer. We found 974 separate software titles in mathematics and 739 in science. We estimate that this represents approximately one-fourth and one-sixth, respectively, of all commercial software. Most of these titles cover single topics, although a few are large, comprehensive packages that cover one or more years of a curriculum.

While the total of 1,713 titles seems like a large number, if they were uniformly divided between math and science, among five microcomputers, and among the grades there would be fewer than 15 topics covered per course on any one computer. In fact, the coverage is anything but uniform, with great concentrations of software for certain microcomputers in some topic areas and a complete lack of software for other microcomputers and other topics.

Grade Distribution

The grade distribution is quite different for math as compared to science, as shown in Figure IV-2 and Table IV-3.

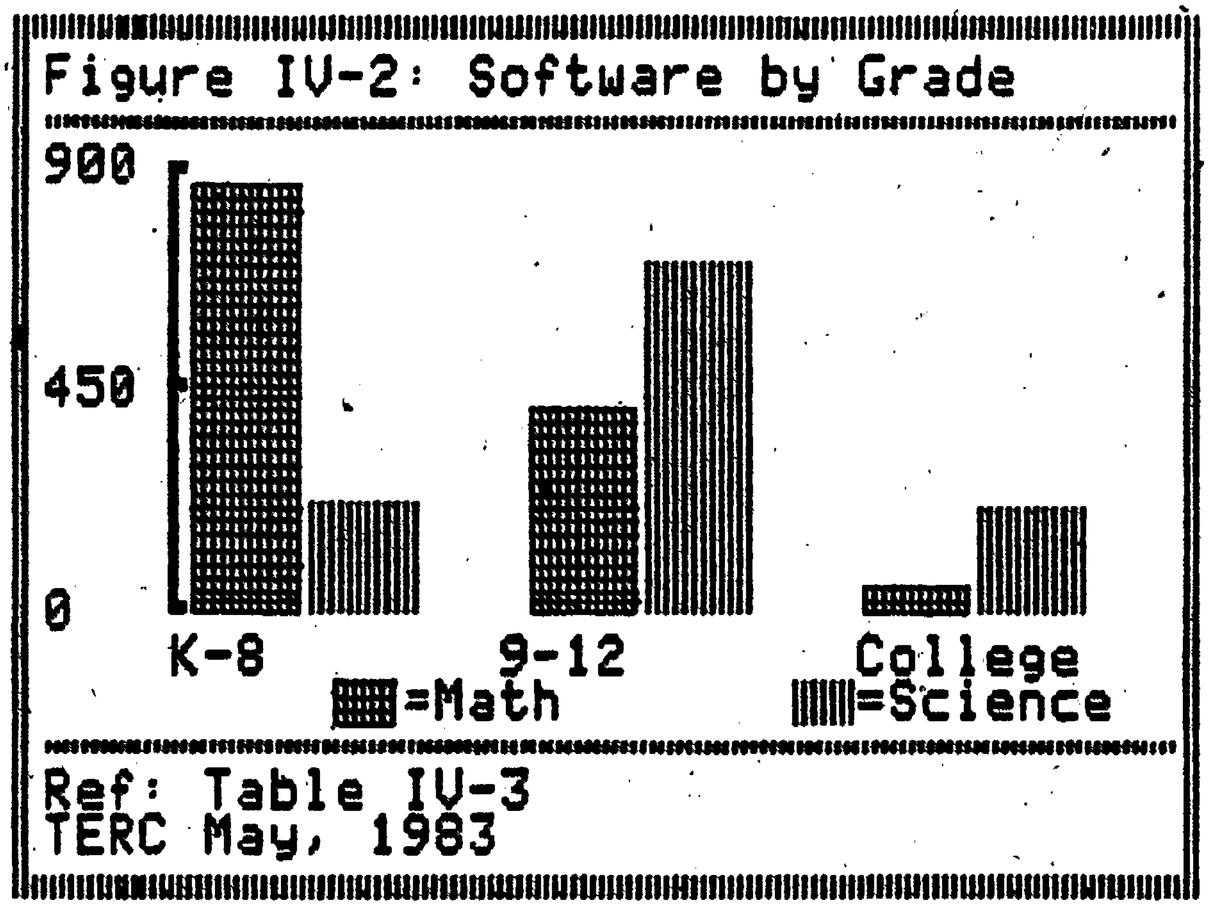


Figure IV-2 Software Titles by Grade

Math software is strongly oriented toward elementary grades. For math, the most titles are available at sixth grade, with large numbers for grades 3 through 8. There are a total of 863 (89%) titles that apply to at least one of the grades K-8, while there are only 415 (49%) for grades 9-12 and only 47 (5%) for college level.

Science software, on the other hand, is highly concentrated at the high school level. The largest number of titles is available at the twelfth grade; the number increases gradually from Kindergarten to grade 12, but drops off sharply at the college level. There are a total of only 216 (29%) titles that apply to at least one of the grades K-8, while 703 (95%) can be used in one of the grades from 9 through 12. A far larger proportion of science software (28%) than math software (5%) applies to the college level.

Table IV-3 Software Titles by Grade Level

Grade Level	Science	Math	Total
Kindergarten	0	110	110
First Grade	7	232	239
Second Grade	15	330	345
Third Grade	29	394	423
Fourth Grade	32	502	534
Fifth Grade	47	492	539
Sixth Grade	70	535	605
Seventh Grade	112	412	524
Eighth Grade	188	443	631
Ninth Grade	304	352	656
Tenth Grade	406	301	707
Eleventh Grade	620	226	846
Twelfth Grade	638	211	849
Kindergarten to Eighth Grade	216	863	1079
High School	703	415	1118
College	206	47	253

Hardware

The breakdown of software titles by computer can be seen in Figure IV-3 and Tables IV-4 and IV-5. In both math and science, Apple commands a significant lead, having versions of 55% of all math and 78% of all science software titles.

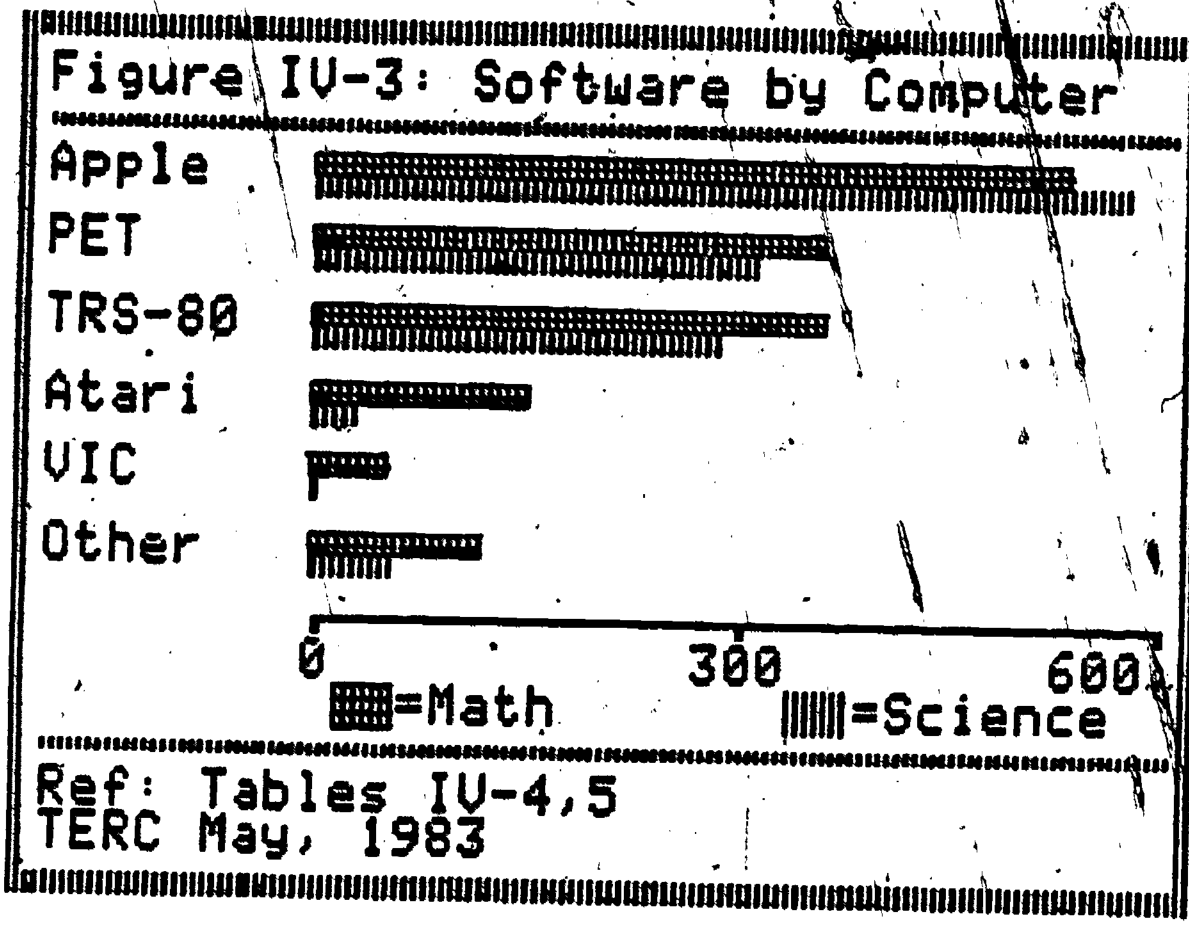


Figure IV-3 Software Titles by Computer

The PET and TRS-80 computers are each second with about 40% of science and 37% of math titles. There does not seem to be a significant shift in these percentages with grade level for either math or science.

These percentages exceed 100% because many titles are adapted to more than one machine. This is shown in Figure IV-4 and Table IV-6 which show that while 980 (61%) titles are only on one machine, 366 (23%) run on two and 355 (22%) run on three or more.

37

Table IV-4 Science Software by Computer and Grade

Computer	No. Titles	K	1	2	3	4	5	6	7	8	9	10	11	12	C
		GRADE LEVEL													
Apple	576	0	7	14	22	21	32	47	65	123	223	310	472	492	174
Atari	28	0	0	1	1	2	5	11	17	18	19	20	24	24	2
CBM	48	0	0	0	0	0	0	0	0	0	0	0	48	48	0
CP/M	2	0	0	0	0	0	0	0	0	1	1	1	1	2	1
Commodore 64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Exidy Sorcerer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IBM PC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KIM	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Mattell	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Monroe EC8800	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NEC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
North Star	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0
Ohio Scientific	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PET	311	0	5	9	15	14	19	30	34	61	72	120	251	272	62
SWTP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SYM	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
TI-99/4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TRS-80	287	0	0	0	9	11	13	26	57	89	139	163	243	256	100
TRS-80 Color Computer	3	0	0	0	0	0	0	1	1	1	2	2	3	3	0
Terak 8510	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
VIC	2	0	0	0	0	0	0	2	2	2	2	2	2	2	0

Table IV-5 Math Software by Computer and Grade

Computers	No. Titles	K	1	2	3	4	5	6	7	8	9	10	11	12	C
		GRADE LEVEL													
Apple	537	46	105	153	179	264	258	294	237	274	225	199	138	131	36
Atari	156	24	43	61	70	74	67	68	64	62	46	43	39	51	18
CBM	52	2	2	6	6	14	16	20	18	35	32	34	32	9	0
CP/M	3	0	1	1	1	1	1	3	3	3	3	3	3	3	0
Commodore 64	3	2	2	2	1	1	1	1	0	0	0	0	0	0	0
Exidy Sorcerer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IBM PC	14	0	0	0	0	0	0	1	1	1	1	1	1	14	13
KIM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mattell	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Monroe EC8800	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0
NEC	1	0	0	0	0	0	0	1	1	1	1	1	1	1	0
North Star	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ohio Scientific	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PET	363	29	67	90	123	197	207	242	173	186	126	104	61	48	14
SWTP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SYM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TI-99/4	20	5	8	9	13	9	7	10	6	6	0	0	0	0	0
TRS-80	364	35	63	94	125	179	193	236	210	208	159	122	77	88	12
TRS-80 Color Computer	23	5	9	11	17	16	14	12	8	7	5	4	2	2	1
Terak 8510	3	0	0	0	0	0	0	0	3	3	3	0	0	0	0
VIC	59	3	10	25	37	51	44	41	18	16	4	3	1	1	0

Figure IV-4 Software Titles by Number of Computers

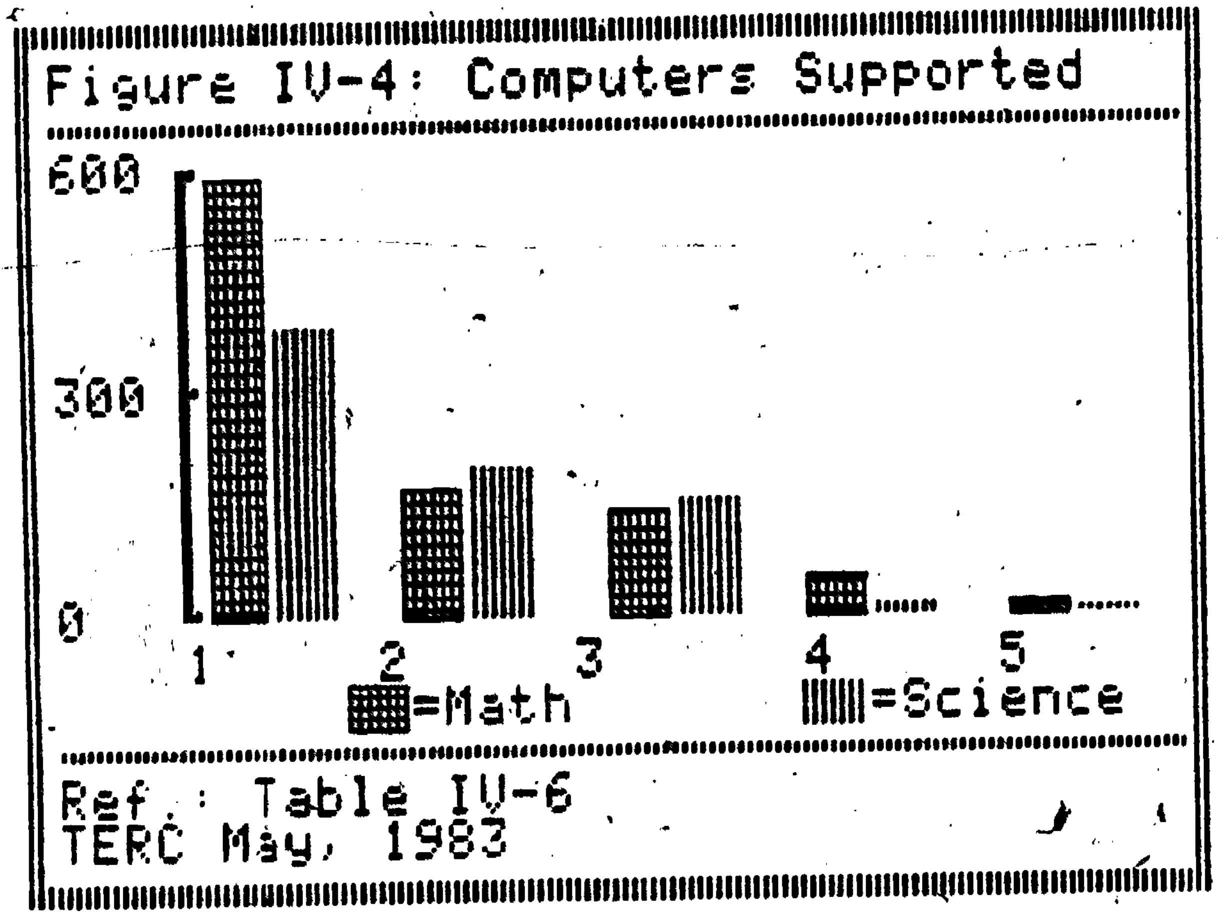


Table IV-6 Software Titles By Number of Computers

No. Computers	Science	Math	Total
No computers specified	0	9	9
1 Computer specified	384	596	980
2 Computers specified	197	169	366
3 Computers specified	150	139	289
4 Computers specified	8	45	53
5 Computers specified	0	13	13
** TOTAL **	739	971	1710

These statistics may, if anything, understate the commanding lead of the Apple. Many of the most imaginative software products, and almost all of the federally-funded software development projects, require an Apple.

This situation annoys some educators who feel the Apple is currently overpriced and technically out-of-date. On the other hand, Cromer (1983) argues that the dominance of a single machine is important because now, for the first time, software can be traded and distributed among very specialized users. According to Cromer, the large number of peripherals, easy repair and an expanding knowledge base for the Apple are most important in the thin-market advanced-level subjects and well worth the additional cost of the hardware.

C. Topic Coverage

Commercial software has non-uniform coverage of mathematics and science topics. There is almost no elementary science software, and many high school math and science topics have no supporting software. In this section, math and science topics will be reviewed by subject area in terms of the quantity and styles of software available and needed.

General Observations

The broad distribution of topic coverage is illustrated in Figure IV-5. This is based on the detailed analysis of available software topics by grade shown in Table IV-7 for science and Table IV-8 for math. While these data form the basis of the discussion throughout this section, some general trends emerge that are worth noting here:

- The heavy use of software in advanced science is reflected in the popularity of physics titles (28%) followed by chemistry (25%) and biology (18%).
- The elementary orientation of math is reflected in the preponderance of basic skills software (76%), especially in arithmetic (34% of all math titles).
- The converse of the elementary orientation of math software is the paucity of advanced topics like calculus (1.2%), analysis (1.7%), and trigonometry (1.2%).

43

Table IV-7 Science Topics By Grade Level

Topic	No. Titles	GRADE LEVEL 1	2	3	4	5	6	7	8	9	10	11	12	C
Chemistry -- All Topics	183	0	0	0	0	0	2	2	4	19	41	141	180	86
Chemistry (Acid-Base Chemistry)	18	0	0	0	0	0	0	0	0	1	5	16	18	8
Chemistry (Atomic Struct. and the Periodic Table)	31	0	0	0	0	0	0	0	0	0	3	26	31	18
Chemistry (Chemical Composition)	6	0	0	0	0	0	0	0	0	0	0	6	6	4
Chemistry (Compounds and Formulas)	22	0	0	0	0	0	0	0	0	2	3	15	22	10
Chemistry (Miscellaneous)	6	0	0	0	0	0	1	1	2	3	3	6	4	0
Chemistry (Nucleonics)	3	0	0	0	0	0	0	0	0	0	0	2	3	1
Chemistry (Reactions/Equilibria)	25	0	0	0	0	0	0	0	0	4	9	20	25	14
Chemistry (Organic Chemistry)	15	0	0	0	0	0	0	0	0	0	1	3	15	13
Chemistry (Solutions)	10	0	0	0	0	0	0	0	0	0	2	10	10	1
Chemistry (States of Matter)	20	0	0	0	0	0	0	0	1	5	6	18	20	3
Chemistry (Stoichiometry)	9	0	0	0	0	0	0	0	0	1	1	8	9	3
Chemistry (Multiple Topics)	7	0	0	0	0	0	0	0	0	0	1	3	7	5
Chemistry (No Topic)	9	0	0	0	0	0	1	1	1	3	6	7	8	4
Physics -- All Topics	210	0	0	0	1	4	6	5	9	26	74	195	206	66
Physics (Electricity)	52	0	0	0	0	1	1	0	0	0	25	48	51	28
Physics (Heat)	15	0	0	0	0	0	0	0	0	3	5	15	15	3
Physics (Light)	18	0	0	0	0	0	0	0	1	3	7	18	18	4
Physics (Mechanics)	59	0	0	0	0	0	0	0	0	3	12	56	59	16
Physics (Nuclear Physics)	11	0	0	0	0	0	1	1	1	4	5	11	11	2
Physics (Relativity)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Physics (Sound)	7	0	0	0	1	2	2	1	1	0	1	5	5	0
Physics (Waves)	24	0	0	0	0	0	0	1	1	4	6	20	24	8
Physics (Multiple Topics)	2	0	0	0	0	0	0	0	0	0	0	1	2	1
Physics (No Topic)	22	0	0	0	0	1	2	2	5	9	13	21	21	4
Astronomy	31	1	1	5	5	7	8	17	20	27	25	24	23	8
Basic Skills	34	0	2	2	5	7	9	16	32	29	30	28	28	0
Biology	133	3	4	7	6	15	23	28	45	99	117	115	105	36
Earth Science	36	0	3	7	7	5	6	22	25	29	29	22	21	1
Ecology/Environment	38	0	0	0	0	0	3	7	18	27	33	38	38	3
General Science	18	0	1	1	1	2	5	9	9	14	13	14	11	3
Geology	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Multiple Subjects -- Science	5	0	0	0	0	0	2	2	3	4	5	5	4	1
Natural History	6	1	2	4	5	3	1	0	0	0	0	0	0	0
Physical Sciences	22	1	1	2	1	2	2	1	18	18	18	17	1	0
Technology Education	3	1	1	1	0	1	2	1	1	1	1	1	1	0

BEST COPY AVAILABLE

Table IV-8 Math Topics by Grade Level

Topic	No.Titles	K	1	2	3	4	5	6	7	8	9	10	11	12	C
		GRADE LEVEL													
Basic Skills -- All topics	740	109	231	327	385	480	467	491	343	318	200	130	75	72	8
Basic Skills (Arithmetic)	330	48	141	221	254	253	229	214	97	88	46	25	20	19	5
Basic Skills (Coordinate Systems and Graphing)	21	1	1	3	6	8	11	12	11	12	13	9	5	5	0
Basic Skills (Currency)	17	0	2	6	14	17	15	10	2	2	1	0	0	0	0
Basic Skills (Decimals and Percents)	53	0	0	0	2	24	27	49	49	49	21	8	5	5	0
Basic Skills (Equations)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Basic Skills (Factoring)	27	0	0	0	3	10	10	21	21	20	16	17	7	7	0
Basic Skills (Fractions)	77	2	2	5	10	59	71	72	69	61	40	28	7	7	1
Basic Skills (Inequalities)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Basic Skills (Logic)	16	3	4	6	10	13	11	12	10	9	9	8	6	6	0
Basic Skills (Measurement)	21	0	0	1	2	13	16	18	16	15	14	10	8	7	1
Basic Skills (Number Sentences)	4	0	0	1	3	4	4	2	0	0	0	0	0	0	0
Basic Skills (Numeration)	61	35	40	31	21	18	16	17	13	11	11	6	3	3	0
Basic Skills (Ratio and Proportion)	4	0	0	0	0	2	3	4	4	4	2	1	0	0	0
Basic Skills (Telling Time and Distance)	24	9	16	20	17	11	6	4	3	3	2	1	1	1	0
Basic Skills (Word Problems)	22	0	1	3	13	16	16	16	11	9	4	4	3	2	0
Basic Skills (Estimation)	3	0	0	0	0	1	1	1	2	2	2	2	2	2	1
Basic Skills (Number Theory)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Basic Skills (Multiple Topics)	36	5	13	16	16	18	19	25	21	21	9	3	2	2	0
Basic Skills (No Topic)	8	5	6	7	6	6	5	4	3	3	2	1	1	1	0
Algebra	92	0	0	0	1	4	4	7	18	69	86	85	62	31	4
Analysis	17	0	0	0	0	0	0	1	1	2	4	12	17	17	5
Calculus	12	0	0	0	0	0	0	0	0	0	2	3	6	12	10
Consumer Math	21	0	0	2	3	12	12	18	17	15	11	10	8	8	0
General Math	5	0	0	0	0	0	1	3	4	5	5	5	5	5	0
Geometry	33	1	1	1	3	3	6	10	19	20	24	24	20	17	3
Multiple Subjects -- Mathematics	5	0	0	0	1	2	2	2	2	4	3	3	3	3	0
Statistics	31	0	0	0	0	0	0	3	5	7	10	16	17	31	15
Trigonometry	12	0	0	0	0	0	0	0	1	1	4	10	10	12	2

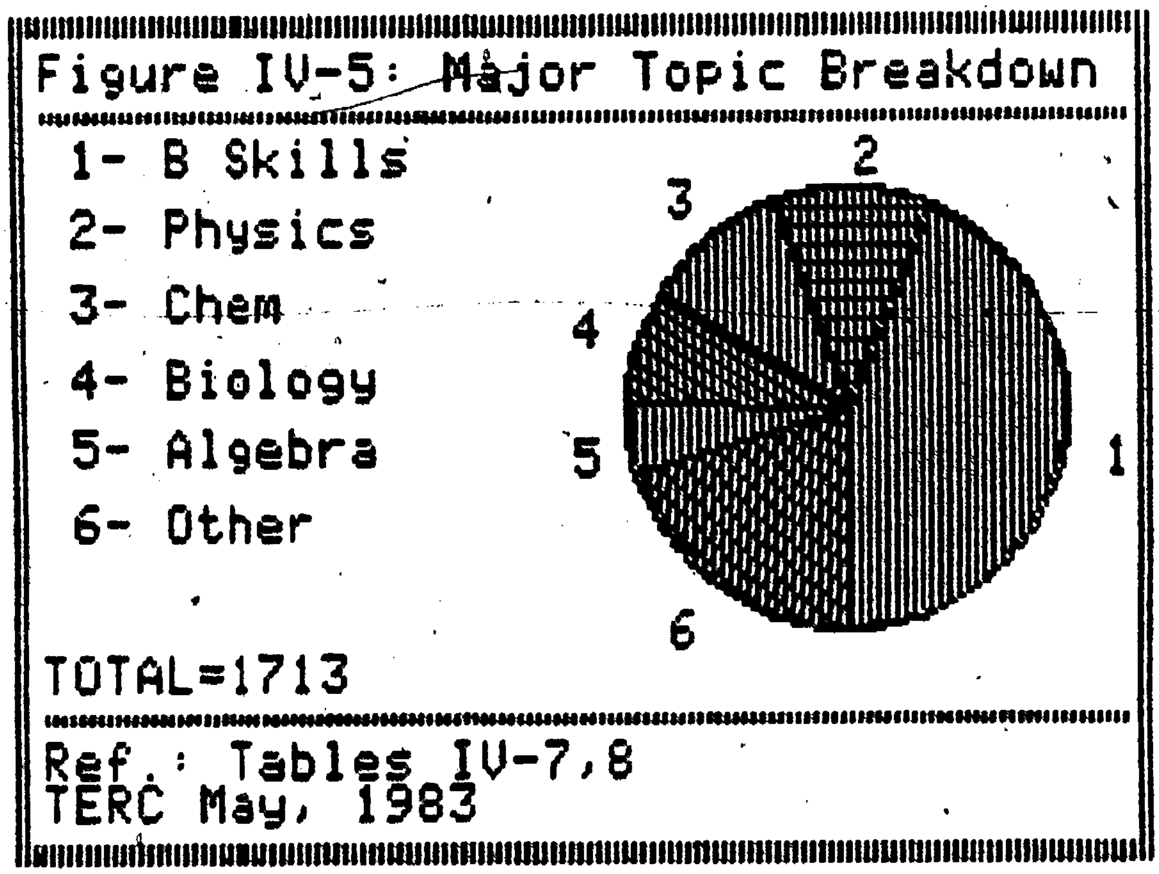


Figure IV-5 Major Topics of Software Titles
"B Skills" are basic math skills or arithmetic.

The teaching styles adopted by software are summarized in Figure IV-6 and detailed in Table IV-9 for science and Table IV-10 for math. Important generalizations that emerge are:

- Drill and practice is very common, especially in math where it dominates with 42% of titles (22% for science).
- Combining games, which are usually disguised drill, with the drill and practice category accounts for fully 67% of math but only 26% of science titles.
- Games and drills are more common in elementary grades than advanced grades. In math, they drop from 78% at the fourth grade to 62% at the eighth grade and 38% at twelfth grade.
- Tutorials are popular, especially for science topics (32%), but also for math (22%).
- Simulations are used in 28% of science titles - they represent the most frequently used style in eleventh and twelfth grades.

Table IV-9 Science Teaching Styles by Grade Level

Teaching Style	No. Titles	GRADE LEVEL K	1	2	3	4	5	6	7	8	9	10	11	12	C
Author Language	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Computational Support	59	0	0	0	0	0	0	3	6	8	28	42	56	55	9
Concept Demonstration	21	0	0	0	0	1	1	2	3	4	5	10	19	19	2
Data Base Retrieval	4	0	0	0	0	0	0	0	1	1	2	2	4	4	0
Educational Game	27	0	2	4	4	6	7	6	11	15	19	18	23	22	3
Language Processor	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0
Multiple Type	3	0	0	0	0	0	0	0	0	0	1	2	3	3	2
Rote Drill	163	0	0	0	0	1	1	4	14	19	28	43	130	156	42
Sample Long-Term Project	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Simulation	207	0	0	0	6	6	9	16	25	46	86	119	194	197	54
Skills Practice	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Teacher Aid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Test Generator	4	0	0	0	0	0	0	0	1	2	3	4	3	3	1
Text Processor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tutorial	235	0	5	11	19	18	29	39	51	91	130	157	175	164	80
Word Processor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Multiple Styles	6	0	0	0	0	0	0	0	0	0	0	6	6	6	5
No Style Specified	9	0	0	0	0	0	0	0	0	1	1	2	6	8	8
** TOTAL **	739	0	7	15	29	32	47	70	112	188	304	406	620	638	206

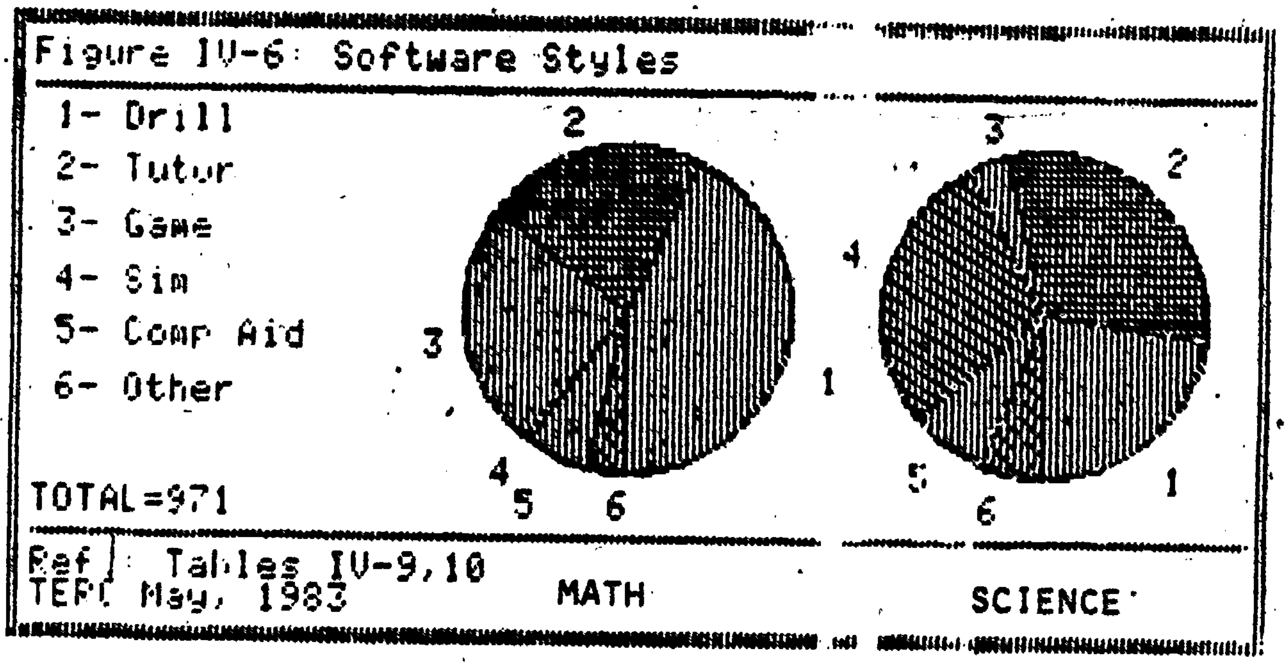
BEST COPY AVAILABLE

Table IV-10 Math Teaching Styles By Grade Level

Teaching Style	No. Titles	K	1	2	3	4	5	6	7	8	9	10	11	12	C
		Columns: GRADE LEVEL													
Author Language	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0
Computational Support	60	0	0	0	1	3	3	8	10	16	23	38	42	55	24
Concept Demonstration	17	1	1	1	0	4	5	6	6	8	8	10	6	8	4
Data Base Retrieval	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Educational Game	234	26	71	109	141	166	156	164	122	110	72	54	35	34	0
Language Processor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Multiple Type	5	1	1	1	1	3	3	4	4	3	0	0	0	0	0
Rote Drill	413	69	128	170	179	224	220	225	149	165	134	92	74	46	5
Sample Long-Term Project	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Simulation	7	0	1	1	2	6	6	6	4	4	2	2	2	2	0
Skills Practice	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Teacher Aid	6	1	2	2	3	3	4	4	5	5	1	1	1	1	0
Test Generator	6	0	0	4	4	5	5	6	2	2	2	2	1	1	0
Text Processor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tutorial	212	10	26	40	60	84	83	104	105	126	107	100	65	64	14
Word Processor	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0
No Style Specified	9	2	2	2	3	3	6	6	3	2	2	2	0	0	0
TOTAL	971	110	232	330	394	502	492	535	412	443	352	301	226	211	47

43

Figure IV-6 Software Teaching Styles



Certain styles of software are sometimes associated with particular topics, as can be seen from Tables IV-11 and IV-12. For instance, 90% of ecology/environment software are simulations, and 15 of the 17 analysis titles are computational aids.

The following descriptions distinguish the various possible educational styles of software:

Authoring Language: An authoring language is a piece of software that allows a teacher to write drill and practice or tutorial material without using a general programming language.

Computational Support (Comp Aid): A software tool primarily designed for a particular type or class of computations or computer function. Examples include graphing packages, function evaluators, and statistical packages.

Concept Demonstration (Demo): a piece of software that shows, explains, or illustrates a particular concept without requiring student interaction. These are often used as part of lecture demonstrations.

BEST COPY AVAILABLE

Topic	No.Titles	D&P	Sim	Game	Data	Demo	Tutr	Comp	Test	Mult	None
	PROGRAM STYLE										
Chemistry -- All Topics	171	85	37	2	1	4	39	7	1	1	5
Chemistry (Acid-Base Chemistry)	18	8	4	0	0	2	3	1	0	0	0
Chemistry (Atomic Struct. and the Periodic Table)	31	20	3	1	1	0	4	1	0	0	1
Chemistry (Chemical Composition)	6	0	5	0	0	0	0	0	0	0	1
Chemistry (Compounds and Formulas)	22	17	1	0	0	0	2	1	0	0	1
Chemistry (Miscellaneous)	6	1	2	0	0	0	3	0	0	0	0
Chemistry (Nucleonics)	3	2	1	0	0	0	0	0	0	0	0
Chemistry (Reactions/Equilibria)	25	8	8	1	0	0	8	0	0	0	0
Chemistry (Organic Chemistry)	15	3	1	0	0	0	10	0	0	0	1
Chemistry (Solutions)	10	8	0	0	0	0	1	1	0	0	0
Chemistry (States of Matter)	20	7	9	0	0	1	2	1	0	0	0
Chemistry (Stoichiometry)	9	7	0	0	0	0	0	1	0	0	1
Chemistry (Multiple Topics)	7	1	1	0	0	0	3	0	0	1	0
Chemistry (No Topic)	9	2	1	0	0	1	3	1	1	0	0
Physics -- All Topics	210	55	60	8	0	11	51	17	0	1	2
Physics (Electricity)	52	19	13	0	0	0	15	0	0	0	0
Physics (Heat)	15	5	2	0	0	0	4	4	0	0	0
Physics (Light)	18	8	4	1	0	2	2	1	0	0	0
Physics (Mechanics)	59	12	24	3	0	1	13	4	0	0	2
Physics (Nuclear Physics)	11	0	8	0	0	0	1	2	0	0	0
Physics (Relativity)	0	0	0	0	0	0	0	0	0	0	0
Physics (Sound)	7	4	0	0	0	0	2	1	0	0	0
Physics (Waves)	24	4	7	0	0	4	9	0	0	0	0
Physics (Multiple Topics)	2	0	2	0	0	0	0	0	0	0	0
Physics (No Topic)	22	3	0	4	0	4	5	5	0	1	0
Astronomy	31	2	9	2	1	2	11	4	0	0	0
Basic Skills	34	3	1	3	0	1	25	0	1	0	0
Biology	133	6	55	5	1	1	52	12	1	0	0
Earth Science	36	7	7	1	0	0	20	1	0	0	0
Ecology/Environment	38	0	34	0	1	1	1	1	0	0	0
General Science	18	3	2	5	0	0	6	2	0	0	0
Geology	0	0	0	0	0	0	0	0	0	0	0
Multiple Subjects -- Science	5	0	0	0	0	0	1	1	0	1	2
Natural History	6	0	0	0	0	0	6	0	0	0	0
Physical Sciences	22	0	0	0	0	0	21	0	0	0	0
Technology Education	3	0	1	0	0	0	2	0	0	0	0

BEST COPY AVAILABLE

Table IV-11 Science Topics By Software Style
See text for an explanation of the software style categories.

Topic	No.Titles	D&P	Sim	Game	Demo	Tutr	Comp	Aid	Auth	Test	Mult	Word	None
						PROGRAM STYLE							
Basic Skills -- All topics	740	343	3	223	5	139	5	4	1	5	4	1	7
Basic Skills (Arithmetic)	330	157	1	113	2	48	0	2	0	4	1	0	2
Basic Skills (Coordinate Systems and Graphing)	21	7	0	7	0	7	0	0	0	0	0	0	0
Basic Skills (Currency)	17	9	2	3	0	3	0	0	0	0	0	0	0
Basic Skills (Decimals and Percents)	53	22	0	12	0	17	0	0	0	0	1	0	1
Basic Skills (Equations)	0	0	0	0	0	0	0	0	0	0	0	0	0
Basic Skills (Factoring)	27	5	0	12	0	7	2	0	1	0	0	0	0
Basic Skills (Fractions)	77	37	0	15	3	20	0	0	0	1	0	0	0
Basic Skills (Inequalities)	0	0	0	0	0	0	0	0	0	0	0	0	0
Basic Skills (Logic)	16	0	0	16	0	0	0	0	0	0	0	0	0
Basic Skills (Measurement)	21	8	0	4	0	7	2	0	0	0	0	0	0
Basic Skills (Number Sentences)	4	1	0	2	0	1	0	0	0	0	0	0	0
Basic Skills (Numeration)	61	39	0	12	0	7	0	0	0	0	0	0	3
Basic Skills (Ratio and Proportion)	4	2	0	2	0	0	0	0	0	0	0	0	0
Basic Skills (Telling Time and Distance)	24	14	0	5	0	4	0	0	0	0	1	0	0
Basic Skills (Word Problems)	22	11	0	0	0	11	0	0	0	0	0	0	0
Basic Skills (Estimation)	3	1	0	2	0	0	0	0	0	0	0	0	0
Basic Skills (Number Theory)	0	0	0	0	0	0	0	0	0	0	0	0	0
Basic Skills (Multiple Topics)	36	18	0	11	0	3	0	2	0	0	1	1	0
Basic Skills (No Topic)	8	4	0	2	0	2	0	0	0	0	0	0	0
Algebra	92	41	0	6	3	31	9	1	0	0	0	0	0
Analysis	17	0	0	0	1	1	15	0	0	0	0	0	0
Calculus	12	1	0	0	3	4	4	0	0	0	0	0	0
Consumer Math	21	7	3	1	0	9	0	0	0	1	0	0	0
General Math	5	3	0	0	0	2	0	0	0	0	0	0	0
Geometry	33	10	0	1	2	16	3	0	0	0	0	0	1
Multiple Subjects -- Mathematics	5	0	0	0	0	2	1	1	0	0	1	0	0
Statistics	31	4	0	0	2	4	21	0	0	0	0	0	0
Trigonometry	12	3	1	2	1	4	1	0	0	0	0	0	0

Table IV-12 Math Topics by Software Style

Database Retrieval (Data): A software package primarily characterized by a large base of data that students can access to answer their own questions.

Educational Game (Game): Either an intrinsic or an extrinsic game which has scoring.

Multiple Type (Mult): A software package that has as a major component two or more of these categories.

Rote Drill (Drill or D&P): Software in which students are tested for comprehension of information through drill and practice. This category excludes games and software that presents new material which would be classified as a tutorial.

Simulation (Sim): A simulation of a mathematical or statistical situation including a simulated laboratory.

Teacher Aid (Aid): A software package, like an electronic gradebook, designed for teacher use but not student use. This excludes test generators.

Test Generator (Test): Software designed to generate tests for students, essentially drill and practice without answers.

Tutorial (Tutor): Software which presents new material to students. Tutorials can include software with substantial amounts of drill and practice that are based on the material presented in the software.

Chemistry

In chemistry, a wide range of topics is covered in simulated lab and drill-and-practice formats. Approximately twenty distinct topics are treated in each format, with an average of two titles for each topic. There are only a handful of tutorials and software tools. The simulated laboratories are almost all standard chemistry laboratories - titrations, gas law experiments, determination of equilibrium constants, calorimetry and the like. Hopefully, these simulations are being used to augment and not to replace traditional hands-on laboratory experiments. However, costs and over-reaction to safety concerns may force many schools to substitute lab simulations for the real thing.

There are four comprehensive elementary chemistry software packages suitable for high school students, each providing at least half a year's worth of work. Programs for Learning also covers a good deal of the general chemistry area and might be considered a fifth comprehensive package, although the titles are sold separately.

One unusual software package is the database Chemdata which permits students to explore the periodic table and look for patterns in the properties of various elements. The program can graph any one of nine characteristics of elements or groups of elements. Another interesting package, MOLEC, is a graphics tool which allows the visualization of atoms in three dimensions.

The vast majority of software available in chemistry is aimed at 11th (77%) and 12th grade students (98%). Only twenty packages (11%) are stated to be of use before the ninth grade. All of the software is in English, and none of it is reported to be particularly useful for students with special educational needs.

Approximately twenty major vendors offer chemistry software, but over two-thirds of the titles are from two publishers - Programs for Learning and Microphys. Programs for Learning software is primarily simulated laboratories written for the Apple, PET, and TRS-80. The Microphys programs are primarily drill and practice for the Apple, CBM, and PET. Overall, 84% of chemistry software will run on the Apple.

Physics

In physics, there is a broad variety of topics and software styles available from approximately thirty major publishers. However, the coverage is not uniform; some software ideas are repeated many times. There are at least six simulations of the Millikan oil-drop experiment; seven packages that teach geometrical optics by drill and practice; five tools used to calculate vector addition, five programs that give drill and practice in solving resistor circuits, seven simulations or games involving two-dimensional projectile motion, and five more simulating orbits. There are seven tutorials on waves, four nuclear scattering simulations, and five drill and practice packages covering various topics in thermodynamics.

Other available software includes several commercially published microcomputer-based laboratory experiments, two of which use Pasco's photogate to turn the computer into a precision timer. There is a range of software on other physics topics, including 52 in electricity and magnetism, two in quantum mechanics, numerous simulations in mechanics, a few in sound,

including Fourier analysis, and one relativistic simulation.

As in chemistry, almost all software is aimed at the 11th (93%) and 12th grade (98%) level. Six titles are available at the elementary level, none in Spanish, and none oriented toward special needs students.

Of the publishers that carry physics software, two-thirds of the titles are supplied by five companies: Microphys; Classic Software, which carries drill and practice for the PET; Merlan, which has simulations and tutorials for the Apple and PET, Cross Educational Software, which produces mostly tutorials for the Apple and EduTech, which carries simulations and other types of packages for the Apple. The number of titles is a somewhat misleading statistic, because some vendors, like CONDUIT with only a few titles, have produced significant and important software. CONDUIT's Introductory Mechanics, for instance, is unusual in that it encourages students to write their own programs (or modify supplied software) to solve dynamics problems.

Astronomy

There seem to be two dominant types of software available in astronomy: tutorials on various aspects of planetary and stellar properties, and observational aids, such as TellStar, that are used to calculate the position of the stars and planets. There are also a few games and simulations based on planetary motion.

Certainly there are many other topics within astronomy that could be made interesting and accessible to pre-college students. Databases of star information could be used to simulate radio or optical telescope observations. Students could be guided to find regularities in these data related to stellar evolution or the expansion of the universe. Simulations of planetary motion could be written that would allow one to experience the apparent motion of planets as observed from the earth or from other planets.

Biology

For the purposes of this report, we have combined biology, ecology, and issues relating to large-scale energy production and consumption. This was done in part because it is difficult to draw lines separating these various areas and in part because

there is so little software that it would not be worthwhile making fine distinctions.

Software for biology is dominated by tutorials and simulations. Also included are approximately twelve computational aids and a few other kinds of software. A large number of the simulations are in the general area of ecology and environmental science. Many of these are based on the original Huntington II simulations. The single most popular topic is genetics with fourteen titles. Birdbreed is a typical example. In this simulated genetics laboratory the student must deduce a rule of inheritance by simulating the breeding of birds and then observing the characteristics of the offspring.

One important package which could be utilized in biology and ecology is MicroDynamo a microcomputer version of a computer language which is widely used for simulations (Roberts, 1981). High school students could use this language to create their own simulations and explore many ecological systems. It is important to give students the power to create their own simulations so that they can then control the underlying assumptions. These assumptions exist in any model and often affect its performance in subtle but profound ways. Students can experience this best when they build and use their own models rather than use preprogrammed models with nonalterable assumptions.

One would think that such a popular and important topic as health would be heavily represented in commercial software. However we found only one example, You're the Doctor??, which simulates medical problems for the user to diagnose and treat.

There is one example of microcomputer-based biology laboratories: Experiments in Human Physiology, a set of ten experiments that involve measuring heart rate, breathing rate, response time and skin temperature.

Given the large numbers of biology students, it is surprising how little biology software exists--only 18% of science titles. This is one area where drill and practice could help students learn the many terms and properties. Databases could provide numerous opportunities for discovery learning. Computer-based laboratories could help students see important relations. Simulations could help make the molecular basis of life more comprehensible.

Earth Science and Geology

There is very little software in the earth science/geology area at this time. There are a series of tutorials by TYC Software, six games, and a few drill and practice packages on assorted topics. We were unable to locate any currently available software that treated topics in oceanography, plate tectonics, the earth's structure, ice ages, or navigation.

There are rich opportunities for new software in this area. One can imagine games based on prospecting or weather predictions. Resource maps might be both interesting and useful. Databases which include information about minerals and crystals and three-dimensional graphing utilities which show crystal structures are certainly possible. Finally, microcomputer-based laboratories for weather recording, monitoring scale ocean models, and measuring mineral properties could help make this subject more empirical and accessible to students.

General Science

We have included in general science a number of topics which are hard to classify, cover very elementary concepts, or cover more advanced ideas that are not in the province of any one particular discipline. Software in this category includes three drill and practice titles; five games, most based on metric conversion; tutorials on statistics, scientific thinking, approximations and estimations; seven programs to help graph data (either in line graphs or histograms), and four statistical packages. One unique package in this area is the Semantic Calculator, a program that keeps track of units and measurement errors while performing calculations using the four basic operations. Animals, an interesting heuristic program, allows one to teach the computer to make distinctions between animals (or anything else) on the basis of observations.

Mathematics

A large number of software packages are available in mathematics, with a preponderance of elementary topics in tutorial or drill and practice formats. Of the 971 titles analyzed, 89% were applicable to grade 8 or lower, and only 23% were useful at twelfth grade. Of the programs for elementary students, more than half covered addition, subtraction, multiplication, and division of integers.

About three-sevenths of all math software titles involve rote drill and practice, about one-fourth provide drill in the form of educational games, and another two-fifths are tutorial. Of the remaining 11%, about half are computational tools, almost all of which deal with graphing functions or analyzing graphs. There are seven simulations and no database retrieval programs.

Software costs vary immensely. A number of comprehensive packages exist such as:

- Math Sequences from Milliken at $450
- Math A for grades 1 to 3 from SRA at $525
- Math B for grades 4 to 6 from SRA at $640
- Micromath for grades 6 to 12 from Sheridan College in Ontario at $500

Large packages such as these tend to include classroom management (CMI) capabilities as well.

At the other end of the scale are the large number of single-concept skill packages and single educational games in the $10 to $20 range and many somewhat more sophisticated packages in the $30 to $50 range.

A few programs offer the potential to dramatically affect the learning of mathematics and change the mathematics curriculum. One example is Green Globs, developed by Sharon Dugdale and published by CONDUIT and Harper & Row. This program uses microcomputers to establish an exploratory learning environment. In Green Globs the student tries to write equations which which will plot curves to pass through thirteen randomly placed green patterns on the screen. The goal is to "pop" all the globs with as few graphs as possible. The computer will draw the graph of whatever equation the student enters. The student can see the results and make subsequent entries accordingly. The computer does not tell the student what to do or evaluate right or wrong. Rather, Green Globs provides a manipulative environement in which the stu-

dent can explore, see "what happens if I try this": and develop a conceptual model of functions and graphs. Discovery Learning in Trigonometry, also from CONDUIT, and Darts, part of a PLATO package from Control Data, also incorporate this learning approach. There is very little such software now available in mathematics; much more needs to be developed.

In spite of the large number of mathematics software titles, many important topic areas are not covered. Poorly represented topics include ratio and proportion, applied math (other than consumer math), probability, estimation, Boolean algebra, complex numbers, solid geometry, and analytic geometry. The swing away from "new math" shows up here, for there is little software addressing sets, groups, axiomatic algebra, or alternative number bases. Except at the elementary or remedial level, there are no comprehensive software packages that could be used throughout common courses such as algebra or geometry. Algebra, a course required for most students but understood by few, would warrant the cost of developing a substantial, comprehensive, integrated software package.

An Agenda For Action: Recommendations for the Eighties, a publication of the National Council of Teachers of Mathematics (NCTM, 1980), sets problem solving as the important central theme that should permeate all mathematics instruction. The microcomputer offers an immense, almost totally untapped, potential for facilitating this with its number-crunching, graphics, and database capabilities.

The computational power of microcomputers also offers the opportunity to use real-world data to combine math and science. A project funded by the National Science Foundation and the Apple Computer Corporation, recently completed at the North Carolina School of Science and Mathematics, developed courseware along these lines (Davis & Frothingham, 1981). One can imagine a simulation for population growth in which it is possible to change the parameters and study the consequences of these changes, and databases which allow testing real-world population data against these various growth models.

Areas of Need

The preceding topic-by-topic review of available software reveals a number of areas where needs exist for new mathematics and science software. These needs will be reviewed in terms of topics, styles, and the intended student populations.

There are a number of important topic areas in mathematics and science for which little or no educational software exists. There is almost no software treating science topics outside the usual biology/chemistry/physics curriculum that could be taught at the precollege level - such areas as psychology, physiology, health sciences, technical occupations, technology, beginning engineering concepts, earth sciences, and advanced science topics at the college/high school border. There seems to be a surprising shortage of software that treats advanced math topics, especially calculus, analysis, trigonometry, and analytic geometry. Software for geometry and algebra is spotty.

There is also a shortage of elementary school science software. One interesting idea in this area would be to use microcomputer software to revitalize some of the excellent science curricula developed during the 1960s. One could take advantage of the tremendous investment those projects represented and produce educationally sound, large-scale curricula at a fraction of what it would cost to start from scratch. The only effort of this sort is at the University of California at Irvine (Arons, Bork & Collea, 1981).

Software in the math/science areas is currently dominated by material in drill and practice and tutorial styles. This does not mean a lack of opportunities for new software in these styles, particularly since the quantity of titles tends to exaggerate the amount of good instructional material available. Nevertheless, it seems there are strong opportunities for software written in other styles.

Microcomputer-based laboratories (MBL), represented by only a handful of available products, is a style of software with particularly strong promise. MBL is of use with both gifted and disadvantaged students and has applications in all of the sciences and mathematics at all levels.

There is a great need for software written in a style that creates an environment for students to learn by exploration: such software as microworlds, simulations, and games. When this software takes the form of simulated laboratories, there is less need for simulations of traditional school laboratory experiments than there is for the simulation of situations that would otherwise not be practical or safe.

The bulk of the available software is geared to the average student in schools. There is a great need to direct software to special populations: students for whom English is not the native language, students with special learning needs, and learners at home.

Given the large number of Spanish-speaking students in the U.S. and the major efforts that have been made in bilingual education, it is disappointing that so little software is available in Spanish. The bulk of the available math and science software is elementary arithmetic drill and practice. Heavily text-oriented materials, such as drill and practice, require a substantial rewrite to be adapted to Spanish. However, many of the more imaginative software packages, in fact, have very little text and would be relatively easy to adapt to Spanish and other languages.

A significant amount of the available software might be useful to students with special needs. But before we can reliably use existing software with special needs students, either our understanding of what makes software useful to these students must improve or procedures must be instituted to qualify software empirically for special populations.

It is important to realize that special needs students can utilize not only rote drill and practice but also more imaginative software, as has been shown in landmark reseafch with the Logo language (Goldenberg, 1979).

There are opportunities that would improve microcomputer-based learning by students in math and science which do not involve writing new software but rather involve utilizing what already exists more effectively. Often teachers need curriculum material or teaching ideas to use some of the more powerful software packages. Any of the general-purpose computer languages can be used in many ways to augment math and science instruction. Traditional text publishers and publishers of supplemental educational materials are beginning to take advantage of these opportunities. An example is <u>muMath</u>, a utility that supports symbolic, algebraic, and calculus manipulations. The software exists; what the teachers need is help in seeing how this facility could be used in teaching.

One last need relates to the current explosion of low-cost computers for which there is little educational software. Much of the software reviewed could be run on computers in the $50-$300 range. Additional savings could be realized by networking, especially in MBL applications. Hardware choices will increasingly be made on the basis of the availability of software, so schools will be able to take advantage of low cost hardware only when a substantial body of software exists for these computers.

D. Evaluation

Quality

It is very difficult to comment on the quality of available software, since different users look for different properties in software. Many observers feel that software is of quite low quality. For instance, in one report, 4,000 titles were reviewed, and only 3-4% were found acceptable (Bracey, 1983). To some extent, these negative attitudes stem from a lack of fit between available software and an individual teacher's requirements and expectations. One person might reject all drill and practice software as inapplicable, while another might find the perfect remedial lesson among these rejects. As the large number of textbooks for most courses attest, there are great differences of opinion about language, scope, and student autonomy that color teachers' evaluations of software.

There is no doubt, however, that a substantial amount of available software is trivial, unimaginative, and does not use microcomputers well. This kind of software is often the easiest to write and fastest to enter the marketplace. There are encouraging signs that more quality software is entering the market. Grant-supported software is being completed and commercial investments in substantial software packages are making an appearance. There is a growing appreciation of the importance of making software "user-friendly" and an increasingly clear idea of how to accomplish this through layout, fonts, color, menus, on-screen prompts, help facilities, and good documentation. More advanced hardware and programming techniques make some software faster, more interactive, and far more powerful than earlier efforts.

In spite of this progress, one senses that software could be much better. The occasional truly unique software package like Rocky's Boots reminds us that much of the software is not creative or original. If only more creative minds were put to educational invention, perhaps many more important new packages could be generated and made available.

Reviews

Teachers and school personnel are clamoring for software reviews. These reviews are needed in order to arrive at purchase decisions for hardware as well as to help select appropriate software for instruction. In response to this demand there are an increasing number of published reviews of educational software. The Software Finder lists journal reviews of math and science software from 53 different journals. The Spring 1983 edition contains one or more citations for reviews of 206 math and 125 science software packages. Many of the published reviews are in computer-related journals and therefore may not reach a large number of teachers who do not read these journals. The science establishment in particular has been very slow to incorporate software reviews into its regular publications. In addition to journal reviews, there are two major national review operations underway. The RICE database, emanating from the Northwest Regional Educational Laboratory and available from Bibliographic Retrieval Services (BRS), contains reviews of over 1,400 software packages, with 338 on math topics and 184 on science topics. The Educational Products Information Exchange (EPIE) is using classroom teachers who have received training in courseware analysis to review microcomputer software. These reviews are available on a subscription basis.

There is one unique publication in the area of reviews, namely, The Digest of Software Reviews: Education, edited by Ann Lathrop. This is a compilation of available reviews for selected software.

Not all reviews are free from sources of bias. With the exception of EPIE, which has adopted the Consumer Union approach to purchasing software, most reviewers are dependent upon free software samples submitted for review. Reviews that appear in computer magazines can be closely tied to the publishers through advertising. This, combined with the natural enthusiasm of many reviewers for any software, might be the reason that the vast majority of reviews are favorable.

Review Methodology

A great deal of effort has been put into the methodology of software reviews, led by early work done at the Northwest Regional Educational Lab by Edwards, Holtznagel, and Marler (1978). This has

led to a proliferation of review forms and review techniques. For a summary and sample of these, see Jones and Vaughan (1983).

There are problems with some of these approaches to evaluation. First, there are so many different kinds of software that no one evaluation form is appropriate for all. More important, however, is that many of these forms tend to concentrate on objective details and avoid answering the two questions that teachers ask most: 1) What is the content of the package?, and 2) Is there any advantage in using this software over print or other media? Many teachers simply need a reasonably detailed description of what a particular software package does. With this information they can make a decision about whether the software is of value in their own particular teaching environment.

Review Costs

Reviews are expensive. It is important in many cases to review educational software in the classroom and to compile opinions of a number of teachers who have tried the software, curriculum experts who have observed it and subject matter consultants who have reviewed it for accuracy. Even if economies of scale are achieved by reviewing large numbers of software packages, this process could easily cost $1,000 or more per software package.

An interesting example of the kind of mistakes one can make in evaluating software without classroom trial can be seen with the package Geology Search. In this simulation, designed for use with a class consisting of groups of 5-7 students, important information appears on the computer's monitor for only a few seconds. Most reviewers criticized this format, but in fact it was deliberately employed because it fosters cooperation between students. Each student needs to take responsibility for recording part of the information or vital facts will be missed. The use of this device assures the involvement of every member of the group in the game automatically, without instructions from the teacher. This illustrates the risk involved if, for the sake of economy, classroom trials are omitted from the evaluation schema.

This high cost of review of educational software raises some very difficult questions. Can professional or commercial journals - the traditional sources of educational software reviews - afford this cost? Can schools or state agencies afford to pay many times the purchase price of a piece of software just to review it for potential purchase? The 1,713 math and science software packages could easily cost $2 million to review.

properly; at the current rate of production of 100 new packages a month, $100,000 a month would be required to stay on top of just the commercial software in math and science.

Electronic Review

The high cost of software evaluation tends to indicate that some method of sharing evaluative information would be of tremendous value ~~to the~~ educational community. We have been intrigued for some time about the use of public access databases to improve the flow of information about software. There should be some electronically accessible file of software descriptions and evaluations to which educators can contribute new evaluations and comment on existing ones. This would reduce the cost of reviewing at the same time that it would increase the relevancy of the reviews because teachers from the field would add comments on their personal experiences with the software. In addition, we envision such a service as a way to find local resource persons who have used the software and can be consulted about its usefulness. We made the database generated for this project - - the Electric Software Binder - available during the project period for access and comment to test these ideas.

Two questions are raised by the possibility of such a dial-up database of software information. First, who would use it, and secondly, would they contribute reviews to the database as well as retrieve information from it? We asked these questions to our sample of math and science teachers. Based on 119 responses to questions concerning a dial-up database, seventy-five percent indicated that they would use it; 51% definitely, and 24% if it were not too expensive. Only 15% indicated that they would not use a dial-up database. Interestingly, 75% of the respondents also indicated that they would contribute reviews to such a database. Thus, we believe that a low-cost alternative in math and science to printed educational software reviews would be the use of contributed reviews by classroom teachers maintained on an electronic database.

In contrast to the optimistic statistics generated by our questionnaires is the low utilization of the already existing RICE software data base maintained by the Northwest Regional Educational Laboratory on the Bibliographic Retrieval Service (BRS) system. BRS is not able to monitor use of this particular file and so use statistics are not available. However, in our contacts with teachers, we find almost no one who is aware of the file's existence and, with the exception of librarians, no one who ~~has used~~ it. This might also be due to the relative difficulty in devising search strategies on this database.

E. Information Dissemination

While magazines and journals are extremely important sources of information about software, teachers inform themselves in many other ways. Roughly one-third report finding out about software by word-of-mouth, presumably through professional contact with colleagues. Another one-third of respondents reported that catalogs are important; one-quarter find out about software at conferences; and one-fifth through direct mailings. One-eighth learned about software through vendors and computer stores, and an equal number through reviews in magazines. Only eight percent reported salespeople were a source of software information while 15% reported still other sources, such as inservice workshops, user groups, and regional centers.

Software vendors are making a major effort to inform teachers about the available software. Software advertising is a major source of income for the many "computers in education" magazines and journals. Software suppliers also play a major role in many of the "computers in education" conferences through displays, space rental, free workshops, and presentations. Software is generally not sold through commission sales and therefore sales people do not represent a significant source of software information dissemination. On the other hand, software vendors do offer free workshops and informational meetings for groups of potential purchasers.

In spite of all these dissemination activities, it is extremely difficult for school personnel to know about all the software that is currently available. We have numerous examples of school people suggesting the development of software which unknown to them, already existed and was on the market. In general, it seems that the dissemination effort for software is not yet reaching the average teacher, but rather computer specialists, audiovisual coordinators, and librarians.

One interesting dissemination development is the emergence of a number of catalog-based software distributors. These catalogs provide a low-cost dissemination mechanism for the small software developer. They also provide a number of important functions for the schools including one-stop shopping, uniform return and review policies, and a level of quality assurance. Examples of these catalog sales operations include J.L. Hammett Co., a traditional supplier to schools, which carries a comprehensive line of software, Cambridge Development Laboratories (CDL), which deals exclusively in science hardware and software including some of its own developments, and Educational Materials and Equipment Co. (EME) which has a broad

range of math and science software as well as other topics. In most cases, the software distributed through these catalogs is not exclusive; that is, it is available from multiple sources although usually at the same price.

F. Acquisition and Use.

School Use of Packaged Software

It seems that the use of topic-oriented packaged software is in its infancy in schools. Many schools are just getting their first computers and are putting their primary emphasis on programming. They have not allocated funds specifically for software and are only beginning to address the question of software acquisition.

Contributing to this situation is the difficulty expressed by teachers in using prepackaged software in their instruction. Almost every piece of software embodies an educational philosophy, a set of prerequisite skills, and a particular reading level - any one of which may make it difficult to fit into an existing curriculum. Thus, the extensive use of topic-oriented software requires either curriculum modifications or extensive searches for software that fits in.

Large Packages

These adaptation problems mitigate strongly against large integrated software packages. Seventy-nine percent of math and science instructors polled would prefer not to use large innovative curriculum packages but instead would rather use small, stand-alone units. If people could find high quality packages that fit the existing curriculum and had good management features, they might be more widely used. However, many teachers expressed unwillingness to risk large amounts of money - in some cases their entire software budget - on a single package that might not fit in and would force a curriculum modification that they would be unwilling to undertake.

We suspect that large software packages also represent an opportunity cost to teachers. A major commitment to a large package lessens the probability that there will either be the

Programming

The most common use of computers by math and science faculty is to teach BASIC and then to have students program in BASIC to explore and to solve problems drawn from topics in math and science. Three-quarters of respondents were using microcomputers to teach BASIC and two-thirds of these reported their computers were used extensively for this purpose.

It is interesting to note the overwhelming vote of confidence given to BASIC in this regard. Logo was reported in moderate use by one-fifth of our respondents; Pascal was reported by only 4% and used only in moderation; and all other languages were used by only 6% of respondents, with use ranked a low 2 by these users.

It is interesting to examine in more detail the significance of this finding in light of the fact that we were asking not about computer literacy but about math and science instruction. It would seem that, from the perspective of these disciplines, having students know BASIC is an important objective. This finding must be tempered by an understanding of the respondents. In many cases the people who are advocating the use of computers in the school have a mathematical or scientific bent. Since computer programming is widely perceived as the first step in a computer literacy curriculum, these same math and science teachers end up teaching programming. Thus, at least some of the interest in teaching programming is not directly related to math and science instruction per se.

Software-Free Uses

The literature abounds with examples of computer uses in math and science that do not require commercial software. One author (Whisnaut, 1982) has written simulations of expensive instruments. Even when schools have these instruments in the laboratory, Whisnaut feels that the software simulations allow students to use the real instruments more quickly and learn more in a shorter period of time, thereby making the very limited amount of expensive instrumentation accessible and meaningful to a larger number of students. Another teacher, a high school biology instructor, has developed a personalized system of instruction (PSI) environment where all kinds of instructional material can be utilized. As part of the course students write packages which are then utilized in the PSI course. Seventeen

funds or the interest in purchasing unique small packages that may have important educational applications.

Thus we find schools are hesitant to use large packages that simplify purchasing. But they also seem not to have undertaken the formidable job of reviewing and selecting significant numbers of small packages that might be of use in instruction.

One piece of evidence for this latter assertion comes from the responses to a question we asked participants concerning their favorite pieces of software. A surprising number of people did not answer this question and indicated that they did not have favorite software or software suppliers. The only software source mentioned a significant number of times was Minnesota Educational Computing Consortium (MECC), listed by slightly less than one-third of all respondents. The Milliken Math Sequences was listed by 10 respondents. All other software vendors and titles were mentioned less frequently. To some extent this can be attributed to the fragmentation of the market, but it is, at least in part, attributable to a lack of familiarity and knowledge of topic-oriented software packages on the part of teachers.

Use of Logo

One of the interesting omissions in the discussions with teachers concerning math software was the Logo language. Perhaps because our methodology tended to emphasize software packages instead of languages, but also perhaps because this accurately reflects teachers' perceptions, Logo was not widely mentioned in regard to math teaching. It is perceived as a language suitable for early introduction to children but justified in terms of computer literacy and not math instruction. This view was at variance with that of its advocates, who emphasize the geometrical richness of Turtle Graphics and the importance in math of the procedural thinking which the language encourages. Papert (1981) speaks of Logo as providing a "Mathland" and emphasizes the experiential learning math students gain through exploring this land by use of the language. His view does not seem to have gained wide currency at this time. Only six teachers reported Logo as one of their favorite software titles, and 21% of schools responding reported teaching some Logo but at a low frequency, below 2 on our scale of 0-5.

packages have been developed by students including drill and practice, simulation, games and tests (Tocci, 1981).

In another typical application, a collection of programs from students and collegues was used in a physics course. Software included a demonstration of a 4-stroke engine, a simulation of a Maxwell Boltzman distribution, an orbit visualizer, laboratory analyses, and oil drop experiment gains (Rowbotham, 1981).

There are many reasons why this kind of microcomputer use is attractive in the schools. It requires no software; it can be done on any computer, including the least expensive; it is a skill that is relatively easily transported from one machine to another; and, when done properly, it can be a powerful way to learn math and science.

Popularity of Styles

When commercial software is used to teach math and science topics, the most popular technique is drill and practice. Three-quarters of respondents report this to be an application of their computers of moderate frequency, averaging just above 3 on our 0-5 scale. We find drill and practice used in both math and science at all levels, but with higher frequency in mathematics and higher frequency at the elementary levels. These findings mirror the commercially available software which has a preponderance of drill and practice materials, particularly at the elementary level and in math. Drill and practice material is attractive to schools because it is widely available, inexpensive, easy to generate locally, and satisfies a perceived need in education: practice in basic skills.

The second most popular form of educational software is simulations, split almost equally between simulated laboratories and other kinds of simulations. Approximately half of all respondents reported using one or the other or both of these kinds of simulations. The interest in simulations was led by science teachers, representing the most important kind of software for high school science instructors. Among respondents who use simulations, the frequency of use was only moderate, ranking 2.5 on a scale of 0-5.

One of the most striking things about simulations was the broad-based enthusiasm that we uncovered for this style of programming. When we asked practitioners what advantages microcomputers had over other means of instruction, 18% volunteered some comment about simulations and the ability these

give students to change variables and explore the consequences of those changes. When asked what kind of software should be developed, the overwhelming favorite was good simulations with animated graphics and user-controlled variables. Science teachers, in particular, seem to have embraced the idea of using simulations as a means of bringing a controlled part of reality into the classroom and permitting students a quantitative as well as qualitative method of exploring topics which otherwise could not be addressed.

Educational games are another form of software which is widely accepted, reported by 70% of respondents, but used with low frequency (2 on the 0-5 scale). Although we did not ask our respondents this explicitly, it is our impression that the majority of these games depend on extrinsic motivation, using scoring and competition to enliven drill and practice. Thus, the high ranking given games is probably a carry-over from the high ranking given to drill and practice. It should be noted that non-educational games are frequently used, particularly at the elementary level, either as a reward for good performance or as part of familiarizing students with the use of the computer. It may have been the kind of use reported by some teachers in this category.

There has been a lot of controversy concerning the educational use of games. Because of commercial video games and the widespread selling of game software for home microcomputers, microcomputers have been stigmatized as merely "toys", capable only of games. In addition there is the deep-seated feeling that education is serious business and not a game. In light of this, the 70% acceptance of educational games as one role for microcomputers in math and science instruction is remarkable.

The final widespread use of microcomputers in math and science instruction involves their use as computational tools. About 60% of respondents reported using their computers in this way with moderate frequency (2.5 on a 0-5 scale). One of the most popular items in this category is the use of graphing utilities which can rapidly display plots of mathematical functions as the student changes the value of the given variables. This is an example of the perfect use of a microcomputer to do what is impossible by any other means and which brings a great deal of clarity to a topic which traditionally is hard to teach and difficult to learn.

Another type of tool application includes having students write programs for computational purposes such as the calculation of *pi* or a planet's orbit. It is our impression, which we did not confirm by direct questions, that the popularity of student programming leads to the moderately widespread use of the computer as a computational tool.

Equity Issues

Many observers are concerned that the educational use of computers will be inequitable and exacerbate imbalances already present in the U.S. society. Less affluent schools are less likely to purchase computers and have lower numbers of computers per student (Braun, 1980). More important, poorer schools also tend to be less able to provide in-service training for their teachers on computer use, less able to buy varied software, and less likely to allow their faculty to initiate software requests. As a result, poorer schools are more likely to use unimaginative software and software that explicitly instructs the student. This may have the result of leaving students with the impression that computers control them, whereas students from more upwardly mobile suburbs are learning to control computers. In a society that will make constant, multiple uses of computers, the sense of control and mastery one has over computers could become an important determinant in the careers and opportunities to which an individual can aspire.

It is important to realize that hard data about the number of computers and the quality of their use as a function of socioeconomic student background are very hard to obtain. However, a growing amount of anecdotal evidence seems to indicate that this is a very real problem (Center for Social Organization of Schools, 1983).

A related issue that has not received the same degree of concern is the amount of access to computers of students in precollege vocational and technical programs. In our review of software we found almost no commercially available software aimed at this population. Furthermore, we found very few examples of schools in these categories making extensive use of computer software. One exception to this trend is Minuteman Vocational and Technical School in Lexington, Massachusetts, which not only uses computers extensively but has developed its own directory of 200 microcomputer software titles that are used in support of its academic programs. It also has some specialized Computer-Aided Design/Compter-Aided Manufacturing (CAD/CAM) hardware and significant support from parents and faculty. (Miller, 1982).

V. THE SOFTWARE POTENTIAL

Software can lead to substantial improvement in math and science instruction. It can increase student learning, lead to greater achievement, improve teaching productivity, and in the future may reduce educational costs. Appropriate software by itself cannot redress the major problems facing math and science instruction: inadequate budgets, underprepared staff, and insufficient inservice and preservice training. The appropriate use of software requires additional funding, staff with a deep understanding of math and science teaching, and access to training in the effective use of microcomputers. When all these conditions are met, software can make a substantial contribution.

We have divided software into two categories: That which teaches material explicitly through tutorials, dialogues, or drill and practice; and that which provides opportunities for exploration--primarily through simulations, software tools, and programming languages. It is the second kind of software that seems to have the greatest potential for increasing student learning. On the other hand, material that is more closely tied to current classroom needs, that is, more explicitly pedagogical, has an important role in increasing student achievement in traditional courses. Both of these kinds of software offer substantial opportunities to improve teaching productivity while sometimes reducing cost. In the section which follows, we consider how software can lead to increased learning and greater achievement. The potential of software for improving teaching productivity and reducing instructional cost is also considered.

A. Software for Increased Learning

There are many indications that microcomputer software can increase the range and depth of topics that can be successfully covered in math and science courses. More importantly, this software has the potential to help students develop fundamental skills in problem solving and to use their own learning strengths in approaching a problem. The three major kinds of software which have the greatest potential for improving student learning are: 1) simulations; 2) software tools; and 3) programming languages. While there has been little educational research demonstrating the impact of these types of software on student learning, many educators have reported anecdotally on the potential impact of the software.

Simulations

Simulations are used in a number of different ways. In its purest form, a simulation is a model of some aspect of reality, and a student is encouraged to explore this model by interacting with it. Sometimes this kind of simulation is labeled a microworld. One way to use a model or a simulation is to introduce a goal and a scoring algorithm that determines how closely the student reaches the goal. It turns the simulation into a game and provides motivation for exploring different aspects of the simulation. This particular kind of computer game, which has been labeled an "intrinsic" game, should be distinguished from "extrinsic" drill and practice games where the main goal is simply to learn the material being presented. Simulations are intrinsic games because the material to be learned is essential to playing the game (Malone, 1980).

With some simulations, it is possible for the students to build their own simulation models, thereby gaining a much deeper understanding of the underlying model. Model building takes place at the boundary between mathematics and science because it forces students to think about the underlying ideas of science and state them in a way that can be explored mathematically. The whole question of model bias is side-stepped because the student creates the model and, in fact, evaluates its validity in terms of how accurately it reproduces reality.

One of the most interesting educational applications of model building has been the work of Roberts (1981), who has simulated the adaption of the modeling language Dynamo to microcomputers, and has developed teaching materials which are effective in teaching middle- and high-school students to generate their own dynamic models.

Educators caution that in order for simulations to be effective, some teacher guidance is required (Bork, 1978; Peters, 1983). In many simulations, the user explores the model by controlling a set of parameters and observing the effect of these parameters on the model. Bork points out that unless some guidance is given, many students will randomly set these variables and get absurd results. They will get bored with the whole process long before they have discovered the relationships that teachers consider interesting in the model. Bork suggests using guided exploration where default values for the parameters are provided, along with suggestions for study. Peters calls for a script as an alternative to textbooks which allows students and teachers together to learn from computers.

What effect do simulations have on student learning? The MIT Logo Group, in one of the few studies of the educational effectiveness of simulations, has shown that students who use microworlds that model the dynamic behavior of objects improve their performance in a traditional physics course (White, 1981).

In one of the best controlled studies of simulations, Anderson (1981) exposed students to a variant of the Huntington II program Pollute for brief 10-minute episodes. The goal of the study was to detect ancillary learning and attitudinal changes, and it was found that students who used the simulation learned the material better than students in the control group. Differences in scores were significantly higher for the simulation group, both immediately after the intervention and six months later. This study shows that simulations can effectively teach the content of science lessons. In a similar study, Hughes (1974) showed that students can make significant achievement gains from simulated physics laboratories.

Within the realm of mathematics, Engle and Payne (1981) report effective use of simulations that are developed by either teachers or students. They feel that models in particular facilitate the connection between the real world and the somewhat abstract world of mathematics (Darfler, 1978).

Based on the limited evidence that is available, Peters feels that simulations will play an important role in the educational revolution that will be caused by computers. The interactive nature of simulations is a particularly compelling feature for learning about math and science. According to Peters (1983): "What could be a more natural application in the sciences than to allow students to go beyond the confines of the laboratory or field investigation...and carry out as many simulated investigations as they like focusing throughout on the logical problems of experimental design [and] hypothesis testing?"

Simulated Laboratories

Simulated laboratories are a special kind of simulation. The defining characteristic of a simulated laboratory is that it is a simulation of material traditionally covered in educational laboratories. In all other respects, the software required for simulated laboratories is indistinguishable from that of other kinds of simulations. The most common kind of simulated laboratory adheres very closely to laboratories currently used in science instruction. Such simulations are sometimes used as a pre-lab, particularly in chemistry, to give students familiarity with the real laboratory which follows. This can reduce the amount of time students need in the laboratory, reduce chemical losses, and minimize expensive mistakes. This kind of software use is widely accepted in all disciplines, especially in chemistry. For example, Butler and Griffin (1979) used simulated laboratories on the PET as pre-labs and claim that this use simplifies laboratory experience.

Two concerns have been raised about the use of such simulated laboratories. First, the idea of using a pre-lab so that students get procedures right tends to emphasize the cookbook nature of a laboratory experience. This may be appropriate in certain labs where correct procedures and related skills are the primary objective of the laboratory. But pre-labs of this simulation type may work against the goals of laboratories where students are expected to learn through exploration and problem solving. The use of pre-labs to emphasize the correct procedures may lead to an increased emphasis on a cookbook approach to laboratories. However, Palmer (1981) claims just the opposite--that the software avoids the "mindless" collection of data syndrome because students who use the pre-lab are better able to understand what is going on and to subsequently gain from the real laboratory.

The second concern is that simulated laboratories may come to replace real laboratory experiences. Laboratories are expensive; they require special rooms and special resources. Recently, there has been a hypersensitivity to safety issues that has all but banned many classroom chemicals. If these forces result in the substitution of simulated laboratories for real laboratories throughout the pre-college curriculum, students will lose an incredibly rich and important part of the learning experience.

There is another class of simulated laboratories that represents a net addition to the repertory of science teaching because it permits a form of student experimentation on systems that would otherwise be inaccessible because of cost, time, scale, safety, or other reasons. One of the most popular simulated laboratories of this type involves genetic breeding. Breeding experiments that would take months or even years can be done in a single session at the computer. Other popular simulated laboratories in this category allow students to experiment with the motion of planets, electric power reactors, and the entire economy.

Cromer (1983) specializes in laboratory simulations. Under his direction at Northeastern University about 4,000 students in the past three and one-half years have done about 20% of their laboratory work in this mode. He has students actually measuring with rulers and stopwatches events that take place on the video screen. "By measuring the period with a stopwatch and the radius with a ruler, the student determines the mass of a laboratory-size star" (Cromer, 1983). Computer simulations offer an opportunity to introduce new experiments into a conventional laboratory, and for institutions that have no laboratory, a way of exposing students to some experimentation. The danger is that some schools may see the availability of computer simulation as an excuse for not providing experiences where the student can explore: this must be guarded against.

Microcomputer-Based Laboratories

One of the most promising ways of using the microcomputer is in a real--not simulated--laboratory for direct measurement and control. In these applications, which we label microcomputer-based laboratories or MBL, the computer is interfaced to the laboratory and can directly gather, analyze, and display results. While this type of application has not gained widespread acceptance, it has been proven useful with students from fourth grade through college (Engh and Ratzlaff, 1980). These authors state three advantages for this kind of approach: Fast feedback permits students to evaluate immediately and maximize the amount of experimentation done; the computer can actually save money by replacing expensive hardware; and the system is quite flexible because enhancements to MBL are largely a matter of software.

Tinker (1981) also emphasizes the fast feedback and the consequent increase in the amount of experimentation students can undertake. They state that the increased control over the experimental situation the students have in MBL-type applications makes it possible for students to have a greater degree of participation in the direction of their investigations.

One of the important aspects of MBL is that it makes relatively abstract ideas quite concrete. Students can measure heat flow, light level, and force, and relate these concepts to their own senses. The computer has the ability to quantify and extend their senses into a wider range of measurements and in time domains that are outside the students' sensory capabilities. In our initial observations, this process of extending one's senses was so immediate and obvious that students quickly began dealing with abstractions which, from a cognitive development perspective, would seem unlikely. Students can create and interpret graphs, they seem to be able to appreciate the significance of statistical variations in data, and they can learn to work with decimals and units much sooner than has been commonly assumed possible.

Presumably, the cognitive explanation for this is that the degree of abstraction of these constructs is in the eye of the beholder; if students can relate points on a graph to concrete events that they have performed and sensed, then the graph is not nearly as abstract an ideas as it was when presented in a mathematical context. Perhaps it is not that the students are able to grasp more abstract ideas earlier, but that some ideas that we thought were fairly abstract can be made more concrete through this approach. This hypothesis needs more investigation by educational researchers.

A number of scientists have developed software for MBL applications (Cromer, 1983; Ford, 1983; and Sparkes, 1981, 1982). The lack of associated laboratory interface required for these applications creates a problem in many teaching situations.

However, experiments that measure reaction time (HRM Software) and memory and cognition (CONDUIT) are available and do not require laboratory interfacing. Recently, a number of products which combine software laboratory equipment and guides to the experiments, thus greatly simplifying the hardware problem, have appeared on the market. (Sources include HRM Software, Cross, Cambridge Development Laboratory, EduTech, Pasco, Vernier Software, and Atari Corporation.)

We feel there is a great deal of educational potential in combining the data gathering and modeling capabilities of a microcomputer so that students could both gather data and generate models to explain these data. Properly done, the computer could handle all the drudgery of calculation, data gathering and storing, and graphing. The student would then be free to concentrate on the process of gathering and explaining data about the natural world. To the best of our knowledge, no one has attempted to combine MBL and modeling activities in this way at the pre-college level.

User Acceptance of Simulations and Lab Software

When we asked mathematics and science teachers how microcomputers can meet their needs, a large number described applications which were some form of simulation. Teachers want simulations with high quality graphics and user-controlled variables. These are perceived as being highly motivational, unique, capable of presenting relevant topics, and capable of integrating mathematics and science. They are also seen as being an excellent problem-solving tool with the ability to individualize instruction. Simulations were perceived by respondents to be important because they gave students immediate feedback in a highly interactive environment by quickly demonstrating the results of manipulating variables. When asked what advantages microcomputers had over other means of instruction, fully 18% of the participants mentioned the user control that could be obtained with simulations. When asked what software they would like to see developed, the overwhelming favorite was additional simulations with animated graphics and user-controlled variables.

We asked science teachers a number of questions related to simulated laboratories and microcomputer-based laboratories. Fifty-one out of 58 respondents felt that MBL applications were a good or excellent way of using microcomputers. Of the seven that did not respond positively to this application, five expressed concern about its costs. Fifty-eight out of 62 respondents felt that simulated laboratories were a good idea; four were concerned about the replacement of real laboratories by simulations.

In order to get some sense of how important these computer laboratory experiences were, we asked participants how much time they would have students spend on the various kinds of activities. The respondents tended to put equal emphasis on non-computer laboratories, computer-simulated laboratories, and computer-interfaced laboratories. These data indicate that science teachers are generally enthusiastic about simulated laboratories and MBL activities and would use them to enrich student learning experiences in the laboratory.

Software Tools

Student use of software tools for calculations, graphing, data analysis, and data retrieval should be an important part of mathematics and science education for both pragmatic and pedagogical reasons. The pragmatic reasons are that these tools are widely used in mathematics, science, and related occupations, and thus represent important material from a vocational and cultural viewpoint. More important, however, is the fact that these tools are pedagogically valuable because they enlarge the domain of content areas that students can explore and learn.

The use of computer tools for computation, of course, raises many of the questions that the calculator raised. What parts of the traditional mathematics curriculum are supplanted by this powerful tool? What parts can we afford to give up in favor of more sophisticated concepts and still not deprive students of important computational skills and underlying concepts? While few educators would overthrow all of arithmetic, many (e.g., Lipson and Fisher, 1982) are willing to let some of the more advanced arithmetic skills join square root extraction as the sort of skill that very few people need to know. Rather than withholding the powerful computational capability of the computer from students out of fear of the possible dependence on these tools, it is probably best to give students the computational tools, and allow them to determine when and how to use these tools. It is possible that initial contact with these tools stimulates students to learn the computational procedures that the computer is using.

There are a number of specialized computational aids that are also useful teaching devices. One is the Semantic Calculator which can be used like a four-function calculator that also permits units and error to be specified for each number used. With associated text material, students can learn the ideas of dimensional analysis and error analysis and also use the tool for conversions between systems of measurement (Schwartz, 1982).

Another novel computational use of the computer is for the generation of theorems and the exploration of axiomatic systems. Battista (1982) has implemented Hofstader's MIU formal system

(a simple system of axioms in which non-obvious theorems can be proven in much the way theorems are proven in the more complex system of Euclidian geometry). This kind of application could be used in a number of formal systems to give students a powerful theorem generator capability. Obvious applications are in algebra, geometry, and group theory.

Of particular interest throughout mathematics and science is the computer's ability to graph. There are literally dozens of software packages on graphing available commercially and in the public domain. This widespread availability of software no doubt reflects an acceptance of the importance of this particular application. The relationship between a function and its graph and the effect on the graph of various operations on functions are ideas that are very difficult to teach. With graphing software, students can master these concepts through their own exploration. The many different kinds of graphs (such as polar and three-dimensional) and the many different kinds of functional representations (such as parametric, differential, and complex) can all be used and represented in graphic forms for greater understanding and intuition.

A project undertaken at the North Carolina School for Science and Mathematics begins to exploit the power of analytical and graphing tools. In this project, students were introduced to the idea of least squares fit in mathematics. They were led through a series of explorations of applied mathematics on the technique for interpolation and the determination of functional relations with a set of tools that operate on data sets (Davis and Frothingham, 1981). Other material used in science courses takes advantage of the same set of software tools to help students discover functional relations in mathematics, physics, and chemistry.

Another software tool that is ideally suited for the computer and widely used in mathematics and science is the analytical computation tool for statistics and data analysis. This is of particular interest in the quantitative social sciences where a number of statistical packages are available that rival the capability of mainframe statistical packages in power and far surpass them in graphing ability and interactiveness. This kind of application is not widely used at the pre-college level because the traditional mode of instruction is to teach the underlying mathematics before teaching the statistics. There are indications that this order is not necessary. For instance, we have found that through the simple device of making some computer-based measurements of reaction time, students in sixth grade spontaneously develop ideas of average, standard deviation, and statistical significance (Tinker, 1981). It is possible that as statistical packages become more widely available to pre-college students, modifications will be made in the scope and sequence of math instruction so that students will learn statistical skills at earlier grade levels.

Programming

The extensive use of student programming is no doubt an endorsement on the part of practitioners of its effectiveness in improving science and, especially, math comprehension. There have been a number of studies of this carry-over effect, with conflicting results. Overton (1981), in an excellent review of the effectiveness of programming, cites a number of studies at many educational levels. The studies tend to indicate that student programming improves math achievement. It is interesting to note that many of these studies predate the microcomputer. With the introduction of the microcomputer, and its fast responsiveness, graphics, and ease of use, we would expect even greater gains in math achievement.

The Logo originators and advocates (Papert 1981, Feurzeig et al., 1969) feel that student programming in Logo has a large carry-over into mathematics and support their assertions with a considerable body of case studies and anecdotal information. A number of recent studies of children's use of Logo have had difficulty quantifying ths carry-over, and some reviewers (Pea and Kurland, 1983) feel there are cognitive reasons to doubt at least the broader assertions of the Logo advocates. Pea (1984) advocates that we step back from our attempts to pinpoint the precise carry-over effects of Logo. Instead, we should view programming as a "potential vehicle for promoting thinking and problem solving skills" (1984) and begin describing the range of effects that Logo can have in promoting these skills.

When Logo was first made commercially available, there was no logical place for it in the existing curriculum. Recently, schools have made modifications in their curricula so that Logo can be incorporated (usually into math classes). There are strong regional differences in the degree of acceptance of Logo, and there is not widespread acknowledgement of its role in mathematics. Nevertheless, there are strong indications that its use in schools is increasing dramatically. Logo software is being given to every school in California as the only educational software to accompany the free Apples that each school receives. Thus, we see a substantial nationwide implementation of a programming package which teaches some rather sophisticated mathematical concepts.

The real potential for student programming comes not from programmming in the abstract, but in putting students to work on content-related programming tasks. Through well-formulated exercises, students gain new insight on the relations of numbers and science ideas because of both the detailed and highly specific nature of the programming path and the results of the program they have written. The benefits of this approach can only accrue through math and science classes when educators believe that all students have the ability to program. As computers gain wider acceptance in schools and society, we will

be able to make the assumption that all children can program, beginning from a fairly early age. Once programming is introduced in the elementary schools on a wide-scale basis, math and science instruction are sure to benefit.

Implications for Learning

Software which provides opportunities for exploration through simulations, tools, and programming languages offers great potential for creating a more scientifically and mathematically literate citizenry. Because this class of software allows students to master process goals such as problem solving and critical thinking, it broadens the experiential base upon which both mathematics and scientific concepts are built and empowers students to compare theory and observations. This can result in increased science learning, both in the schools, through a new definition of scope and sequence, as well as outside the schools, in homes and other informal learning situations.

The achievement of process goals is usually an indirect consequence of using "learner-centered software" (simulations, tools, and programming language). Achieving these goals depends on not only having the right software, but also having the right learning environment. The software removes the necessity for many of the time-consuming and often mind-dulling activities that traditionally occupy a great deal of instructional time. More time is available for asking questions, for exploring, and for investigating students' own hypotheses. There is, of course, no guarantee that teachers will choose to take advantage of this opportunity. The point is that with appropriate software, increased opportunities for learning do exist.

A New Scope and Sequence

Microcomputers may have the greatest impact on education outside of schools. Many publishers are developing their educational products primarily for this huge market and only incidentally for the school market. Other software developers have a dual strategy: marketing the same product with different packaging to both home and school markets. At this time, most of the home-oriented educational software being produced is aimed at very young children and is generally math drill and practice. As powerful software becomes increasingly available to the out-of-school market, it is not unreasonable to suppose that some children will develop passions for simulations, microworlds, and programming.

It is important to realize that there are many hardware and software limitations that reduce the effectiveness of the current generation of software. The speed of operation, the size of available databases, the detail of graphics images, and animation are all areas which limit currently available software.

There is every reason to believe that new software and hardware technologies, some already usable except for the cost, will vastly improve the quality of simulations, tools, and programming languages. The enormous storage and graphics image capability of videodiscs is thought by some to add an extremely important new dimension to the interactiveness and attractiveness of software. Other observers feel that intelligent computer-assisted instruction will provide revolutionary breakthroughs in educational software that will lead to substantially increased learning of math and science.

B. Software for Greater Achievement

In this section we will consider explicitly pedagogical software that is designed to improve student achievement in existing curriculum units. Most software in this category uses either tutorial or drill and practice instructional techniques.

Software of this sort has a number of characteristics that can make it an effective adjunct to instruction. The graphics and animation of modern microcomputers with color, motion, and user interaction can be important both to keep student interest and to illustrate ideas and concepts that may be hard to describe in words. Many observers report that the patience and non-judgmental character of the computer tutor are important attributes, especially for students who have had bad educational experiences or who have experienced difficulties in learning.

Computer-Assisted Instruction

Many studies have shown that the computer as tutor is as good as other forms of instruction and in some cases is superior to traditional instruction (Overton, 1981, Kulik et al., 1983). Over the years, a plethora of comparison studies on computer-assisted instruction (CAI) vs. traditional instruction have been conducted, with many of the early studies using mainframes as a vehicle for introducing CAI. Some of the studies pit "traditional instruction" (variously defined as teacher lecture, workbook activities, independent study with no teacher guidance, or videotapes of teacher lectures) against CAI (again, variously defined and usually not clearly described). Other studies determine the effectiveness of CAI as a supplement to traditional instruction.

Outcome measures often consist of cognitive gains as demonstrated on: 1) standardized achievemet tests such as the Math Diagnostic Achievement Tests, California Achievement Test, Iowa Test of Basic Skills, or Differential Aptitude Test; 2) teacher-made tests; 3) researcher-made tests; or 4) grades in courses or on final exams. Attitudinal outcomes are frequently examined as an adjunct to learning outcomes. Attitudes about school, computers, math or science courses, self as a computer user, and enjoyment of computerized lessons are variables commonly assessed.

Most of the reviews of studies on the effectiveness of CAI conclude that at the secondary school level, CAI is an effective instructional method and can produce substantial gains in achievement when used either by itself or as a supplement to traditional instruction. The typical study shows a small, statistically significant advantage for CAI. Many of the studies show that the learning gains associated with CAI are more substantial with low-achieving students. While most studies show modest gains in achievement for students using CAI, many show no difference between teaching methods.

Note that in most reviews of outcomes studies, there is no clear identification of the subject matter being taught through CAI. One exception is Overton's review (1981), which showed that the effects of CAI in math are basically the same as the overall achievement effects discussed above.

Attitudinal outcomes associated with CAI have generally proved very positive (Thomas, 1979). In general, students using CAI express more liking for their materials than do students not using CAI. Bukoski and Korotin (1975) report that students showed increased motivation and interest in mathematics after exposure to CAI. In an interesting study of Mexican-American junior high students studying remedial mathematics, it was found that students thought the computer was more fair than the teacher because it kept promises and was not seen as arbitrary (Hess and Tenezakis, 1973).

Another consistent outcome is that CAI can lead to substantial instructional time-savings with no sacrifice in student learning. Comparisons of computer instruction versus traditional instruction often show 30 to 50% time savings for those using computers ((Overton, 1981; Clark, 1983).

Recently, research on CAI has been seriously questioned (Clark, 1983). A major methodological flaw plagues almost all of the CAI outcomes studies: There is no control of the instructional methods being used when comparisons are made of CAI and traditional instruction. Many different methods of instruction can be employed and emphasized through CAI, and the same is the case with "traditional" instruction. In the typical comparison

study, a teacher is asked to present a given topic using whatever methods she/he chooses, and whatever amount of time she/he feels is necessary to cover the material. This "condition" is compared with an often undescribed CAI treatment of the topic. If either method proves statistically superior, we do not know why: Was it the pacing, interactivity, personalization, interest level, feedback, reinforcement, use of examples, or some combination of these? In most studies, none of these factors is either controlled or examined.

These flaws have led one researcher to conclude: "Thus far there are no definitive research investigations in this area" (Feurzeig, 1981). Some have even gone so far as to suggest that no more media comparison studies should be conducted because these studies are neither methodologically sound nor do they ask the right questions. With respect to methodology, Clark (1983) and Hanley (1984) convincingly argue that any new studies should control for such factors as: 1) content of material presented via CAI versus traditional instruction; 2) teaching methods employed (pacing, feedback, etc.); 3) instructor presenting the material. But Clark contends that if these factors were controlled, there would be no differences between CAI and traditional instruction! His view is that the computer--or any medium--is only the vehicle for transmitting instructional messages and has no impact on the effectiveness of instruction.

Like Clark, Hanley (1984) also believes that it is inappropriate to measure the global effectiveness of CAI. He particularly argues against using traditional empirical methods to examine the relative effectiveness of CAI against other instructional methods. Instead, Hanley sees value in using the case-study approach to examine the features of CAI and the procedures used to implement CAI that seem to contribute to positive outcomes as perceived by users. We should step back from "black box" studies that identify only the magnitude of effects, and instead, according to Hanley, begin identifying "all the essential parameters describing the successful use of CAI."

One promising avenue for improving students' achievement is through the use of well-designed interactive tutorials. While there has been little research on the ways in which these tutorials are used or on their effectiveness, the characteristics of this type of software--namely interactivity and branching--suggest that it could be extremely valuable.

Bork (1981) has been designing and testing such software in both classrooms and public libraries. For example, Bork's module on statistical inference asks students to determine how many jumping grasshoppers are in a large field. The student is told that it is not possible to directly count all of the grasshoppers and is asked to suggest some alternative strategies

(estimation and sampling). Through a series of questions, the student is led to count the grasshoppers in one square of the screen, then determine the total area of the field and get an estimate of the total number. To examine the effectiveness and usefulness of this interactive tutorial, Bork and his colleagues are videotaping groups using the material, and analyzing participants' conversations, emotions, and the way they use the computer. This research--which starts by examining actual usage of the software--may prove much more useful to educators than the media comparison studies discussed above.

Computer-Managed Instruction

Another role for microcomputers in improving student achievement comes in its ability to manage instruction, usually called computer-managed instruction, or CMI. In this case, the computer is not used to instruct but rather to monitor a student's progress and to present alternative instructional materials to students. Approaches to personalized instruction, sometimes called the Keller Plan, PSI, self-paced study, or mastery learning, often have a heavy administrative overhead. Students study different materials at their own rate and take one of a number of equivalent tests to see if they have mastered the material. Some schools that have implemented such a system do find that a microcomputer significantly reduces the administrative overhead and can improve the quality of the course by helping identify weak instructional units and poor test items.

Micros can be used to generate tests by random item selection and by producing randomized data for numerical test items. A computer can, of course, also be used to grade tests, both those that it makes itself and standardized tests. Inexpensive optical and mark-sense reader attachments are available for microcomputers to speed this process.

Microcomputer-based CMI of the sort described above does not seem to be widely-implemented for math and science instruction at this time. The only commercially available software that support this application are a few test generator programs of very limited capacity. Our survey of users located only one site using CMI. The one respondent was extremely enthusiastic about this application and felt that it was one of the most important applications of microcomputers in education.

C. Software For Improved Teaching Productivity and Reduced Costs

Computers have much potential to improve learning and increase teaching productivity--but what is the cost? As is the case with many educational interventions it is difficult to assess the costs and weigh them against the benefits that will accrue to students. Although there is certainly some potential for reducing instructional costs through the use of micro-computers, computers often add to instructional costs in the short run.

School administrators are increasingly aware of the initial costs of hardware and software. However, there are also many unanticipated maintenance expenses that represent substantial continuing costs of the computer system (Feurzeig, Horwitz, and Nickerson, 1981). As changes are made in hardware and operating systems, software often requires updating to run on the new system. The costs of software maintenance can easily exceed the original purchase price of the software.

Another major cost is teacher training. We have no good estimates of the costs involved in preparing teachers to use computers in the classroom. As noted above, math and science teachers tend to be early adopters of computerized instruction and are in many cases self-taught programmers. But using the computer for educational purposes requires more than a rudimentary knowledge of programming--it requires a whole new approach to teaching math and science. Teachers need to know how to identify and select appropriate software, how to use the software creatively, and how to integrate it into the regular curriculum (or how to change the regular curriculum to take full advantage of the software). Learning these skills clearly requires a great deal of inservice training and the cost of this training usually must be assumed by the school system itself. Ultimately, this expense will be reduced as preservice programs equip new teachers with solid backgrounds in the instructional use of computers. In the meantime, school systems should be prepared to make a substantial investment in teacher training.

It has been suggested that the costs of computerized instruction may be offset by a reduced need for teachers. The threat of computers taking over jobs was taken very seriously by teacher unions during the 1960s and 1970s, and union opposition was one of the major barriers to the increased use of educational technology (Blaschke and Sweeney, 1977). The reality is that computers simply cannot replace teachers and are unlikely to result in the overall reduction of teaching staff. Even the best software can accomplish only a small number of the instructional functions that teachers perform. As Lipson and Fisher (1982) have pointed out, "the small modern computer is very stupid compared to a teacher."

While computers cannot replace teachers, they can certainly make teachers' work more efficient. Cost savings can be achieved through computer-managed instruction (CMI) and computerized lab situations by reducing the amount of clerical work needed to monitor student work, assign units, and calculate grades. These tasks are often time-consuming and tedious, but can be accomplished very efficiently by a computer. Teachers, in turn, can devote more time to actual instruction.

In addition to taking over routine tasks, the computer has the potential to make the diagnostic aspect of teaching more efficient. Good instruction means that a teacher must know the learning strengths and weaknesses of each student--and this can be an overwhelming demand. Computers can simplify this task by pinpointing problem areas for individual students. However, there have been few attempts by software developers in math and science to incorporate such diagnosis. Given the lack of appropriate software, it is not surprising that teachers have made little use of computers as a diagnostic tool. A few projects are underway to determine the value of diagnostic programs for special education teachers (Malouf, 1984), but little has been done in the areas of science and math.

Along with increasing teacher productivity, computers have the potential to increase learner productivity. Much of the literature on computer-assisted instruction (CAI) that was discussed above shows that materials can be learned in a shorter period of time when it is presented via computer. Simulations can lead to similar time savings: A review of 32 studies of computer simulations--mostly in math and science--showed that students learned material up to 50% faster when the material was presented on the computer. While increased learning efficiency produces no real dollar savings (since students are required to attend school for a certain number of hours each day), it does mean that instructional time is freed for other purposes such as advanced study or investigation of additional topics. In this way, student productivity can be substantially increased.

One interesting and hidden benefit of computerized instruction may be improved school attendance. Researchers have shown that high school and community college students who used CAI math programs had higher attendance and lower attrition rates. In the case of the community college, the dropout rate in basic math went from 60% when traditional instruction was used to only 20% when CAI was introduced. These figures should be interpreted with caution, because they may be attributable to a novelty effect that will diminish as students become more familiar with computers. Yet the findings on student attendance were impressive enough to lead Braun (1980) to the conclusion that using computers to aid instruction "can result in a substantial gain in use of the tax dollar to education."

One way in which learner productivity can be enhanced through computerized instruction is by coordinating school and home uses of the computer. As more families acquire home computers, it will become possible to reinforce lessons introduced at school with homework on the computer. Technical Education Research Center's recent work with middle school students shows that by the end of 1983, between 35 and 50% of these students had access to home computers. Given these figures, it already may be possible for teachers to assign optional work for students to complete at home. The amount of time that students have to work on a limited number of classroom computers could be optimized by allocating a larger share of class computer time to those who do not have home computers and a smaller portion of time to those who can do the work on their home computers.

At this point, the costs of computer instruction may seem high in relation to the benefits that have been demonstrated. Yet the cost of hardware is rapidly declining, making cost figures from recent studies already obsolete. It has been estimated that during the next five years, the cost of hardware will decrease by a factor of two and at the same time this hardware will become more powerful and flexible (Braun, 1983). Technological developments in both hardware and software hold definite promise for increased learning at decreased cost.

VI. RECOMMENDATIONS

While microcomputers can play a significant role in improving mathematics and science education, they will not do so unless we make a sustained, conscious effort to make the best use of this potential. Teachers must be trained in the effective use of microcomputers, software must be developed which utilizes the unique capabilities of microcomputers, materials and facilities for microcomputer-based learning must be implemented in the classroom. Also, avenues for dissemination of information about hardware and software must be expanded, curriculum must be redesigned, and basic research on the use of microcomputers in schools must be supported. The following sections contain recommendations for accomplishing these essential tasks.

A. Teacher Training and Support

Placing microcomputers in mathematics and science classrooms will not have any significant effect on instruction without strong teacher training and support. Teachers who lack the theoretical and practical expertise to use computers can hardly be expected to integrate microcomputers effectively into their math and science curricula. There is a need to build confidence in teachers and administrators and to give them a firm foundation in the practical use of microcomputers without overwhelming them with technical details.

However, this sort of familiarization is only the first step required in microcomputer training. The promise of microcomputers will only be realized when there are substantial changes in the scope and sequence of instruction itself. This requires teachers to alter the topics they cover to include more sophisticated, more quantitative, and more varied content. Finally, the best use of microcomputers often demands a particular style of instruction--guided exploration and self-discovery with the aid of software tools and simulations--which is not popular and places extra demands on teachers. The training required to upgrade teachers' mathematics and science competency and pedagogy represents a major undertaking but is a necessary concomitant to the effective use of microcomputers. This is a task which needs a combination of carefully coordinated local, regional, and federal resources, and which must be implemented in gradual stages if it is to be effectively accomplished.

For economic reasons, the bulk of the inservice teacher training will have to be done at the local level, delivered by school personnel and supplemented by consultants and university

faculty. While it is realistic to expect these local resource people to fill much of the need to familiarize teachers with microcomputers, a broad base of expertise does not exist at this time to undertake the much larger job of instruction in mathematics and science content and pedagogy. This latter job should be undertaken by well-trained scientists and mathematicians, skilled in teaching, and knowledgeable about microcomputer software and hardware. Because this combination of qualifications is rare, the challenge is to develop alternative models for teacher training. Possibilities include the development of teacher training materials, training the trainers, establishing regional resource centers, and exploring the use of new technologies in teacher training.

Beyond the local level are regional and state resources. Many communities are finding it efficient to set up collaboratives to fulfill certain educational needs of teachers. Some of these collaboratives have computer consultants, training staff, and training facilities where group classes can be held. We highly recommend their use. Unfortunately, such facilities and expertise are expensive and are probably out of reach to many smaller school systems.

In addition to minimal staff training on an introductory level, ongoing training must be provided. It is to be expected that microcomputer technology and uses will continue to change at a rapid rate and school systems would be wise to build in-house capabilities in the area of microcomputers. Special funding might be sought to set up such resources, but local school budgets should eventually support this function. Staff will be necessary to monitor new applications of software and to train teachers in the operation of such software.

But it is not just teachers who need to keep conversant with software and hardware developments. Administrators and curriculum planners will play pivotal roles in the integration of microcomputers into math and science classrooms and laboratories. It will be their responsibility to obtain hands-on experience with microcomputers and to apply new research and educational applications to their schools.

While all school systems are not necessarily playing a catch-up game with inservice training, another area where maximum effort should be exerted is in bringing preservice microcomputer courses to all colleges and universities involved in teacher education. Some schools are now requiring such courses for education majors and some states are requiring this for certification. We highly recommend such actions.

All teachers are hard pressed for time. Teaching is a particularly demanding occupation and ways must be found to give teachers sufficient release time for microcomputer training during the school year or to provide summer institutes. Many teachers and administrators contacted in this study mentioned the

summer institutes formerly sponsored by the National Science Foundation as being of high quality and particularly worthwhile. We recommend reinstituting and expanding these programs.

Teacher training materials are also urgently needed. Materials used to teach programmers, office personnel, or computer scientists are not appropriate to the needs of teachers. While there are certain computer operations, routine maintenance, and even programming skills which are needed, we find that these are relatively easy to both teach and to learn. The difficult area of knowledge is how to use this flexible creative tool in the classroom. What skills are best taught with computers? How does one plan a lesson incorporating computer-assisted instruction? What type of hardware and software does a teacher need to use a microcomputer in a physics laboratory? What are the particular ways that microcomputers have been found to be useful thus far and how might they be utilized in the future? Good teacher training materials that deal with these issues are almost nonexistent and efforts should be made to develop and disseminate such materials.

We should also look to the new technology itself to provide new ways of training. One of the limiting constraints in providing training for teachers is the availability of skilled trainers. It will take time to train the trainers, but we could increase the exposure of existing trainers through the medium of educational and cable television, through the use of videotape and videodisc presentations, and through the use of well-planned, self-paced software with accompanying documentation. Many businesses use this type of technology in their training programs and much could be learned from their experiences and then applied to the design of teacher training courses.

A significant number of math and science teachers attend national and regional conferences such as the ones sponsored by the National Science Teachers Association and the National Council of Teachers of Mathematics. Training workshops on the use of microcomputers are held at such conferences and could be expanded. In addition, many vendors set up display booths, and actually give mini-workshops on the use of their products. Teachers whom we surveyed regard these vendor displays and the opportunity to talk with vendors as a valuable source of information. We recommend that these conferences be expanded and perhaps subsidized in order to capitalize on vendors' experience and their commitment to improving education.

Finally, there is an even more problematic area in math and science teacher training that we have not touched upon. The microcomputer is only a tool. Like any tool it needs an experienced, skillful craftsperson to utilize it. The fields of science and mathematics have undergone tremendous advances in the last few decades and due to numerous factors, math and science teachers who are masters of their craft are in short supply.

Radical action must be taken to improve this situation or the true potential of microcomputers in math and science education will never be realized.

We must emphasize that teachers well-trained and confident in their abilities to use microcomputers are the key to success in this area. We must be generous in our support of all teacher training programs. Excellent facilities, modern hardware, useful software, and skilled staff must be provided for teacher training. And they must be provided for all school systems--rural and urban, wealthy and poor--throughout the United States. We recommend vigilance on the part of state and federal authorities as well as financial support to see what inequities are avoided in this process.

B. Software Development

The research upon which this report is based has shown that while there are a substantial number of software titles, the topic coverage is not uniform, and the quality of the software is quite variable. While it was not feasible for us to carefully evaluate all of the titles in math and science as part of this project, it is clear that some of the software is poorly designed and a large amount of it has limited applicability. Software that takes full advantage of the medium and uses sound pedagogical approaches is rare and difficult to develop.

The fact that the software market is flooded with inferior products is no reason to summarily dismiss the medium. It is very easy to develop and bring to market low quality software, whereas it takes a concentration of resources and originality to get high quality software into teachers' hands. We feel that there is a substantial maturation period during which educators, software developers, distributors, and teachers learn to use the medium to enhance math and science learning. As the technology changes, and the hardware situation in schools changes, we need to devote sustained and substantial resources to software development so that we can effect this learning.

It is important to realize that at present the best software almost invariably has its roots in substantial federal funding. This is hardly surprising, given the high costs of software development. It means that much of the best software is coming to fruition as a result of prior national investments in development. Following this logic, it may be that the current relative lack of major federal projects supporting educational software in math and science will lead to a dearth of important new developments in the next five years.

There is an urgent need to reverse this situation by providing long-term, substantial funding for software development in math and science. Enormous opportunities exist in the subject

areas described in the body of this report, particularly for developing software that promotes problem-solving skills and provides direct experiences with mathematical and scientific phenomena. This software development can be encouraged in a number of ways, both through direct government grants for software development, and indirectly, through policies and procedures that will encourage the development of high quality software. The following recommendations detail how high quality software can be encouraged.

Direct Support. There is no substitute for direct, sustained financial support. We need to bring together interdisciplinary teams of programmers, scientists, educational psychologists, and designers to concentrate on the development of effective software. In order to attract the best people, the funding needs to be adequate to be able to offer competitive salaries, and to be sufficiently long-term to allow professionals to make commitments in this area.

Some would argue that this kind of federal funding is inconsistent with the free market approach to educational software. However, we have not seen examples of industry willingness to spend the kind of money that is necessary for substantial innovations and development. Private enterprise can profit from government investment in software development. Procedures have already been developed and are currently under refinement at the National Science Foundation that create a partnership between private enterprise, government, and software developers--a partnership that is fair and provides incentives for all parties to get good software into the hands of teachers.

Information Assistance. There will always be important segments of the educational software market which are too small to warrant the allocation of major resources from either government or private sources. For instance, one can imagine a powerful software package which could teach the elements of symmetry and group theory to eighth graders through innovative use of graphics, software tools, and tutorials. However, at least in the short run, such a product is unlikely to have a very wide market. It may be exactly what some teachers want but appeals to only a small number of teachers.

High quality software is needed in many topic areas. As outlined in the body of this report, certain math and science topics are not well covered by software, particularly software that takes full advantage of the technical strengths of microcomputers. We recommend the development of microcomputer-based software tools and microworlds that students can use for exploration and problem solving.

Microcomputer Enrichment of Innovative Curricula. One of the major barriers to the widespread use of software in education is fitting it into the curriculum. Software developers should take increased cognizance of this fact, and where possible, design software that meshes well with existing high quality curricula. Particularly rich sources of excellent curricula that could be enriched with appropriate software are the major curriculum projects of the '60s and '70s. For example, good software would both enhance and improve the dissemination of the NIE-supported Comprehensive School Mathematics Program (CSMP) curriculum. The Project Physics curriculum or the Biological Sciences Curriculum Study (BSSC) are also excellent candidates for software enhancement.

Software for Special Populations. Very little software has been developed for special populations, including limited English-speaking, physically disabled, and learning disabled students. The absence of this software means that special populations are not able to benefit from the educational advantages offered by computers. We recommend that more resources be devoted to developing and disseminating software for special populations.

C. Classroom Implementation

The availability of good educational software does not guarantee its widespread use in classrooms. Schools and teachers need assistance in incorporating software into the curriculum. In some cases, where innovative software offers new learning opportunities, this may require a major revamping of the entire math or science curriculum.

It is surprising how little educational software is in use in math and science classrooms. Educational software can undoubtedly make strong contributions and should be implemented widely. The following steps would assist in this implementation effort.

Increase School Acquisition of Hardware and Software for Math and Science Instruction. Most school microcomputer implementation seems to be oriented towards computer language instruction. One feature of this kind of implementation is that it requires little or no software. As a result, as schools begin moving computers into math and science instruction, they find that they have no money to purchase the necessary software.

Substantial budgets should be available in order to increase acquisition of both hardware and software. When hardware is donated by industry or other private sources, both the schools and the donors should be aware that the computer is of limited use in math and science without appropriate software.

Forge Links to the Home. Schools can greatly increase the quantity of accessible hardware and software by collaborating with parents and community groups on hardware and software acquisition and loans. Parents are very eager to buy both hardware and software for their children's education. This interest can be harnessed and utilized by establishing joint school/parent purchases of hardware and software, by establishing a lending library of educational software, by loaning computers, and by establishing computer clubs. Qualified parents can, in certain circumstances, be brought in to augment the instruction and to develop needed software.

Use Students' Programming Abilities. In the future, math and science instructors will be increasingly able to assume that students know how to program. This represents an important new educational resource that can be built upon with the appropriate curriculum materials. With a knowledge of programming, students can explore situations that are more complex and realistic such as mathematical problems in calculus and differential equations. There is thus a need to develop math and science software that builds on students' knowledge of programming.

Integrate Software into the Curriculum. In order to assist teachers in selecting appropriate software, there is an urgent need for comprehensive software descriptions and evaluations, organized around the curriculum content of math and science courses. For instance, an Algebra I teacher should be able to find a few dozen references to software that would help students learn how to graph linear equations. This kind of software description and evaluation would make it easier to integrate software into the existing curriculum.

Eliminate Software Piracy. Budget shortages and shortcomings in planning may increase the motivation to make illegal copies of software. Every effort must be made to discourage this practice. It undermines morality and is in the long run counterproductive because it discourages entrepreneurial risk-taking in the development of new software.

D. Research

Far too little educational research has been done on the effectiveness of software or identifying the characteristics of software that lead to its effectiveness. The most exciting aspect of computer technology in math and science instruction is that it can permit new curriculum and new approaches to math and science. However, large-scale changes need to be supported by an active research effort. The following kinds of research are particularly important at this time.

New Technologies. The technology underlying the microcomputer revolution is still very much in evolution. Research into computer applications that might be feasible in five, ten, and even twenty years should be pursued actively. This research would be a great planning aid for educators.

Beyond Drill and Practice. A great deal of research on computer effectiveness has been based on drill and practice software, or has not made clear distinctions between styles of software. There is a great need at this time to either substantiate or debunk the claims enthusiasts have made for educational software in styles other than drill and practice. Work is needed in measuring skills and problem-solving techniques and determining the impact of computerized instruction on these skills. Finally, research is needed on the characteristics of educational software that make it successful--graphics, locus of control, interactivity, type of feedback--or any other characteristics that may be important.

Curriculum Research. Appropriate microcomputer software can permit large-scale changes in both the math and science curriculum. Schools that are willing to undertake curriculum experiments in this area should be supported with additional resources of all kinds to do careful research on the effectiveness of these new curricula. In particular, there are a number of computational and laboratory applications of computers in math and science at the postsecondary level that might reasonably be brought into the pre-college curriculum. These include stepwise solutions of dynamical systems, the use of symbolic algebraic manipulators, microcomputer-based laboratory data acquisition analysis, curve-fitting, vector and matrix algebra, Boolean algebra, and statistics. Applied curriculum research is urgently needed to establish effective and appropriate ways to accomplish the intergration of those topics into the curriculum.

Research on Equity. There is a great need to monitor the extent to which we give students equal access, both in terms of quality and quantity, to microcomputers on a nationwide basis. There is a particular need to study the kinds of software and educational environments that are appropriate for special needs students. The appropriateness of software to special populations needs to be evaluated. In many cases, good software will work with special needs students, but teachers and curriculum planners need to know this and need guidance on how to utilize this software in each student's Individualized Educational Plan.

Cognitive Foundations. Research on the cognitive foundations of math and science learning and the relationship of this learning to the microcomputer's use needs to be expanded. The properties of microcomputer software, such as fast feedback, data transformation, and active control, offer the opportunity to increase our understanding of how children learn science while at the same time increasing our knowledge of the properties that make software successful.

E. Software and Information Dissemination

One of the greatest problems reported by educators is their inability to select appropriate software. To correct this situation, the amount of information about software and its use in the math and science curricula must be vastly increased and communicated to teachers in a number of ways:

Microcomputer Resource Centers. Microcomputer resource centers can provide a very important multiservice role. Resource centers, staffed by knowledgeable professionals and containing representative software and hardware and a good library, can be the site for both educational workshops and individual study. They can also serve as an important resource for software developers and researchers by providing ready access to teachers and students.

Software Reviews. There is a great need for more reviews of math and science software, especially comparative reviews and reviews based on classroom experience. It appears that many teachers do not read specialized computer magazines. We recommend that math and science journals that are widely read by teachers should publish more software evaluations.

Information Exchange. While databases of software evaluations do exist in electronic form, they are used by only a few educators. This is probably because the information on the database is seldom more current than what can be obtained in print and because in each case it is generated by a single-information provider. A more appropriate use of electronic communications would be to establish a mechanism that would permit decentralized input as well as output of computer software information. A single database could be used to store information about individual school software holdings, research citations, and user-contributed evaluations and information. The database on which this report is based has been structured so that it can be used in such a way.

Information for Small Developers. The information needs of small software suppliers should be met so that they can disseminate their products. An interesting model for this is the federally-supported Market Linkage project which supplies this kind of information for developers of computer aids for handicapped students.

Alternative Software Distribution Channels. The dominant mechanisms for distribution of software at this time involve the commercial sale of protected software or the sharing among professionals of public domain software. Alternatives to these two modes have been tried and need further encouragement. These alternatives provide low-cost ways of getting reasonably well-documented software into teachers' hands, and usually involve collecting, testing, and documenting teacher-developed materials. Project Seripham has done this for chemistry software, the Atari Program Exchange has done it for Atari software, CONDUIT for science software, and MECC for a broad range of topics. Each of these efforts has provided an important source of software that otherwise would not be accessible to teachers. Efforts such as these should be expanded and replicated.

Use of Generic Software Tools. One of the most promising approaches to educational computers is to make extensive use of a few general-purpose tools such as graphing, modeling, and data acquisition utilities. By using these general tools in math and science instruction, students gain an appreciation for the way computers are used in the larger world. A number of problems relating to software acquisition and local dissemination are simplified. In this case, the problem is not disseminating the software, but rather disseminating ideas on how general-purpose software tools can be used in teaching.

F. Conclusions

While microcomputer software holds great promise, its use in mathematics and science instruction in schools is in its infancy. BASIC programming and drill and practice are the most widespread current uses of the technolgoy. Only modest, scattered instructional improvements can be expected by extrapolating current trends.

However, with a major commitment of public and private resources at all levels, substantial instructional improvements are possible. To realize these improvements, there must be major efforts to enhance teacher training, to develop better software, and for teachers to acquire and use appropriate software. In addition, there is a need for research on computer-related learning, development of new curricula, increased software evaluation, better dissemination of software information, and response to the equity issues raised by microcomputer use.

BIBLIOGRAPHY

Abramson, T., M. Weiner, S. Malkin, and J. Howell. Evaluation of the New York City computer assisted instruction program in elementry arithmetic (Research Report 70-16). New York: Research and Evaluation Unit, Division of Teacher Education, City University of New York, 1970.

Anderson, Ronald E., Daniel L. Klassen, Thomas P. Hansen, and David C. Johnson. "The Affective and Cognitive Effects of Microcomputer Based Science Instruction." Journal of Educational Technology Systems, 1980-1981, Vol. 9, No. 4, pp. 329-355.

Arons, Arnold. "Achieving wider scientific literacy." Daedalus, Spring, 1983, Vol. 112, No. 2, pp. 91-122.

____________, Alfred Bork, and Francis Collea, "Science Literacy in the Public Library - Batteries and Bulbs." NECC Proceedings, June 17-19, 1981, Denton, TX.

Aspden, P., An Investigation Into the Effect of Prior Laboratory Experience on the Learning of Practical Skills. Milton Keynes, England: The Open University, 1973.

____________. "Where Will Computer-Assisted Instruction (CAI) Be in 1990?" Educational Technology, April, 1978, Vol. 18, pp. 60-63.

Atkinson, Richard C. and Joseph I. Lipson. "Instructional Technologies of the Future." Paper delivered at the 88th Annual Convention, American Psychological Association, Montreal, Canada, September 3, 1980.

Ausubel, D.P. "Education For Rational Thinking: A Critique." In Anton E. Law (ed.), The Psychology of Teaching for Thinking and Creativity. Columbus, Ohio: The Ohio State University, 1979.

____________, J.D. Novak, and H. Hanesian. Educational Psychology: A Cognitive View (Second Edition). New York: Holt, Rinehart and Winston, 1978.

Baltz, B.L. Computer graphics as an aid to teaching mathematics (Doctoral dissertation, The Ohio State University, 1977). Dissertation Abstracts International, 1977, 38A, p. 679.

Barclay, William L., III. "Buggy." Mathematics Teaching, September, 1980, No. 92, pp. 10-12.

Bardige, Art. "The Problem-Solving Revolution." Classroom Computer News, March, 1983, Vol. 3, No. 4, pp. 44-46.

Barrow, Gordon M. "Computer-Based Studies for Physical Chemistry." Journal of Chemical Education, October, 1980, Vol. 57, No. 10, pp. 697-702.

Battista, Michael T. "Formal Axiomatic Systems and Computer-Generated Theorems." Mathematics Teacher, March, 1982, Vol. 75, No. 3, pp. 215-220.

Becker, Henry Jay. Microcomputers in the Classroom - Dreams and Realities. Report No. 319, Center for Social Organization of Schools. Baltimore, MD: The Johns Hopkins University, January, 1982.

Becker, Larry. "The Timex/Sinclair As An Intelligent Lab Station Terminal." Hands On!, Summer, 1983, Vol. 6, No. 2, pp. 1, 11.

Bell, F.H. "Computer Literacy, school mathematics and problem solving: Three for the price of one." AEDS Journal, 1979, Vol. 12, pp. 163-170.

Bell, Trudy E. "Learning To Win." Personal Computing, April, 1983, pp. 134-139.

Bitter, G.G. and W. Slauchert. The effect of computer applications on achievement in a college introductory calculus course. Paper presented at the annual meeting of the American Educational Research Association, New York, February, 1971. (ERIC Document Reproduction Service No. ED 047 963).

Blaschke, Charles, & Sweeny, J. Implementing cost-effective educational technology: Some reflections. Educational Technology, January, 1977.

Bork, Alfred. "Computers as an aid to increasing physical intuition." American Journal of Physics, August, 1978, Vol. 46, No. 8, pp. 796-800.

____________. "In Defense of Computer-Assisted Learning. An interview with Peter Kelman." Classroom Computer News, March, 1983, Vol. 3, No. 4, pp. 17-23.

____________. "Observation and Inference - A Computer Based Learning Module." Paper presented to NECC Conference, June 6-8, 1983, Baltimore, MD.

_____________ and H. Peckham. "Computer Applications in Mechanics." Reprinted in A. Bork, Learning With Computers, Bedford, MA: Digital Press, 1981.

_____________, Stephen Franklin, Ruth Von Blum; David Trowbridge, and Barry L. Kurtz. "Science Literacy in the Public Library." NECC Proceedings, June 17-19, 1981, Denton, TX.

Boyd, A.L. "Computer aided mathematics instruction for low-achieving students" (Doctoral dissertation, University of Illinois at Urbana-Champaign, 1972). Dissertation Abstracts International, 1973, 34A, p. 553.

Bracey, Gerald. "Computer Assisted Instruction: What the Research Says and Doesn't Say, Can and Can't Say About Its Effectiveness." Presented to the 2nd Virginia Educational Computer Conference, Richmond, VA, September, 1982.

_____________. "Computers in Education: What the Research Shows." Electronic Learning, November/December, 1982, Vol. 2, No. 3, pp. 51-54.

_____________. Issues and problems in devising a research agenda for special education and technology. Paper presented at the Special Education R & D Symposium, Special Education Programs, U.S. Department of Education, June, 1984.

Braun, Ludwig. "Computers in Learning Environments - An Imperative for the 1980s." Byte, July, 1980, Vol. 5, No. 7, p. 6.

_____________. "Report on Educational Technology." Paper submitted to the National Science Board Commission on Precollege Instruction in Mathematics, Science and Technology, June 27, 1983. Old Westbury, NY: New York Institute of Technology, 1983.

Bringuier, J.C. Conversations With Piaget. Chicago: University of Chicago Press, 1980.

Broderick, W.R. "Off-Line Computer-Aided Learning Project." London: Borough of Havering Social Science Research Council, April, 1974. (ERIC Document Reproduction Service No. ED 129 280).

Brown, J.S. and R.R. Burton. "Diagnostic models for procedural bugs in basic mathematical skills." Cognitive Science, 1978, Vol. 2, pp. 155-192.

_____________ and K. VanLehn. "Repair Theory: a generative theory of bugs in procedural skills." Cognitive Science, 1980, Vol. 4, pp. 379-426.

_____________, R.R. Burton, and A.G. Bell. "SOPHIE - a step toward creating a reactive learning environment." Journal of Man-Machine Studies, 1975, Vol. 7, No. 5, pp. 675-696.

__________, R.R. Burton, and C. Hausmann. "Representing and using procedural bugs for educational purposes." Proceedings of the 1977 Annual Conference of the Association for Computing Machinery, pp. 247-255. October 16-19, 1977, Seattle, WA.

__________, A. Collins, and G. Harris. "Artificial intelligence and learning strategies." In Harry O'Neil (ed.), Learning Strategies. New York: Academic Press, 1978.

Bukoski, W., & Korotkin, A. Computing activities in secondary education, 1975. (ERIC Document Reproduction Service No. ED 112935).

Burton, C.R. and W.A. Marosz. "The construction of uses of CATIA, a computerized mathematics testbank." Two Year College Mathematics Journal, 1977, Vol. 8, pp. 212-216.

Butler, William M., and Henry C. Griffin. "Simulations in the General Chemistry Laboratory with Microcomputers." Journal of Chemical Education, August, 1979, Vol. 56, No. 8, pp. 543-545.

Camp, John S. and Gary Marchionini. "Computer Enhanced Mathematics." Viewpoints in Teaching and Learning, Spring, 1981, Vol. 57, No. 2, pp. 1-12.

Carmony, Lowell, Robert J. McGlinn, Ann Miller Millman, and Jerry P. Becker. Problem Solving in Mathematics with Pascal. Murphysboro Apple Festival and Problem Solving in Mathematics with Pascal: Teachers' Manual. Murphysboro Apple Festival. Carbondale, IL: Southern Illinois University, Department of Computer Science, n.d.

Case, R. "Intellectual Development and Instruction: A Neo-Piagetian View." In Anton E. Lawson (ed.), The Psychology of Teaching for Thinking and Creativity. Columbus, Ohio: The Ohio State University, 1979.

Center for Social Organization of Schools. "How Much Are Micros Actually Used?" School Uses of Microcomputers: Reports from a National Survey, June, 1983, Vol. 1, No. 2, pp. 1-8. Baltimore, MD: , The Johns Hopkins University.

Chonacky, Norman. "Microcomputer data management in an introductory physics laboratory." American Journal of Physics, February, 1982, Vol. 50, No. 2, pp. 170-176.

Clark, R.E. Reconsidering research on learning from media. Review of Educational Research, 1983, 53 (4), 445-459.

Clement, J. "Common Preconceptions and Misconceptions as an Important Source of Difficulty in Physics Courses." Working Paper, Department of Physics and Astronomy, University of Massachusetts, July, 1979.

Cohen, H.D., D.F. Hillman, and R.M. Agne. American Journal of Physics, 1978, Vol. 46, p. 1026.

Collins, A. and A.L. Stevens. Goals and strategies of interactive teachers. Report No. 4345, Cambridge, MA: Bolt, Beranek and Newman, Inc., 1980.

Conference Board of the Mathematical Sciences, National Advisory Committee on Mathematical Education. Overview and analysis of school mathematics grades K-12. Reston, VA: NCTM Press, 1975.

Cranford, H.R. "A study of the effects of computer-assisted instruction in mathematics on the achievement and attitude of pupils in grades five and six in a rural setting" (Doctoral dissertation, University of Southern Mississippi, 1976). Dissertation Abstracts International, 1976, 37A, p. 5660.

Crawford, A. "A pilot study of computer-assisted instruction and practice in seventh grade remedial mathematics." California Journal of Educational Research, 1970, 21, pp. 170-181.

Cromer, Alan. "Present Applications and Future Outlooks for Computers in Physics." Paper presented to the New York State Section of the American Association of Physics Teachers, Spring, 1983, State University College at Oneonta, NY.

__________. "Stable Solutions Using the Euler Approximation." American Journal of Physics, 1981, Vol. 49, p. 455.

Darfler, W. "Discussion." In D.C. Johnson and J.D. Tinsley (eds.), Information and mathematics in secondary schools. Amsterdam: North Holland Publishing Co., 1978.

Davis, Steve and Phyllis S. Frothingham. "Computing at a New Public High School for Gifted Students." onComputing, Spring, 1981, pp. 72-75.

Duckworth, E. "The Dilemma of 'Applying Piaget.'" Harvard Educational Review, 1979, Vol. 49, No. 3, pp. 297-312.

Dugdale, Sharon. "Green Globs: a Microcomputer Application for Graphing of Equations." Mathematics Teacher, March, 1982, Vol. 75, No. 3, pp. 208-214.

__________. "There's A Green Glob In Your Classroom...And Teaching Mathematics Will Never Be The Same." Classroom Computer News, March, 1983, Vol. 3, No. 4, pp. 40-43.

__________, and D. Kibbey. Elementary mathematics with PLATO. Urbana, Ill.: University of Illinois, 1977. (ERIC Document Reproduction Service No. ED 163 992).

Edwards, J., et al. "How Effective is CAI? A Review of the Research." Educational Leadership, November, 1975, Vol. 33, pp. 147-153.

Eisenkraft, A. "Microcomputers in a Physics Curriculum." The Journal of Computers in Mathematics and Science Teaching, Winter, 1981, Vol. 1, No. 2, p. 9.

Ellis, Lynda B.M., John R. Raines, and Nina Hakanson. "Health Education Using Microcomputers. II: One Year in the Clinic." Preventive Medicine, 1982, Vol. 11, pp. 212-224.

Encyclopedia of Associations. 17th Edition, 1983. Gale Research Co., Book Tower, Detroit, MI 48226.

Engel, Charles W. and H.E. Payne, Jr. "Computer Simulation in Secondary School Mathematics." Viewpoints in Teaching and Learning, Spring, 1981, Vol. 57, No. 2, pp. 114-128.

Engh, Steve and K.L. Ratzlaff. "The Use of Computer-Assisted Instruction in the Teaching Laboratory." Journal of Chemical Education, November, 1980, Vol. 57, No. 11, pp. 815-818.

ERIC - Educational Resources Information Center. National Institute of Education, U.S. Dept. of Education, Washington, DC 20208. (202) 254-7934.

Eschenburg, W.F. "Computerized prescription mathematics in support of group instruction" (Doctoral dissertation, Wayne State University, 1976). Dissertation Abstracts International, 1976, 37A, p. 7007.

Farringer, L. Dwight. "The VIC-20 For Laboratory Interfacing." Hands On!, Summer, 1983, Vol. 6, No. 2, pp. 7-10.

Faughn, J. and K. Kuhn. "Computer Tutorial Programs in Physics." The Physics Teacher, December, 1979, Vol. 17, p. 598.

Feurzeig, W. "Computer systems for teaching complex concepts." BBN Report No. 1742. Cambridge, MA: Bolt, Beranek and Newman, Inc., March, 1969.

__________, E.P. Goldenberg, R. Mattlae, and R. Stachel. "Demonstration of mixed-initiative CAI capabilities in SSA training applications." BBN Report No. 4043. Cambridge, MA: Bolt, Beranek and Newman, Inc., June, 1979.

__________, P. Horwitz, and R.S. Nickerson. "Microcomputers in Education." BBN Report No. 4798. Cambridge, MA: Bolt, Beranek and Newman, Inc., October, 1981.

__________, S.A. Papert, M. Bloom, R. Grant, and C. Solomon. Programming Languages as a Conceptual Framework for Teaching Mathematics, BBN Report No. 1889. Cambridge, MA: Bolt, Beranek and Newman, Inc., 1969.

Flake, J.L. "The use of intefactive computer simulations for sensitizing mathematics methods students to questioning behaviors" (Doctoral dissertation, University of Illinois at Urbana-Champaign, 1973). Dissertation Abstracts International, 1973, 34A, p. 7623.

Ford, Bruce. "Pixels and Pendulums." Classroom Computer News, March, 1983, Vol. 3, p. 47.

Foster, T.E. "The effect of computer programming experiences on student problem solving behaviors in eighth grade mathematics" (Doctoral dissertation, The University of Wisconsin, 1972). Dissertation Abstracts International, 1972, 33A, p. 4239.

Frandsen, Henry. "Trigonometry - Mathematical Curricular Areas and Instructional Computing." Viewpoints in Teaching and Learning, Spring, 1981, Vol. 57, No. 2, pp. 82-95.

Friend, Jamesine E. Domain Referenced Adaptive Testing. Cognitive and Instructional Sciences Series CIS-10. Palo Alto, CA: Xerox Palo Alto Research Center, Cognitive and Instructional Sciences Group, April, 1981.

Fuhrer, S. "A comparison of a computer-assisted testing procedure and standardized testing as predictors of success in community college technical mathematics" (Doctoral dissertation, New York University, 1973). Dissertation Abstracts International, 1973, 34A, p. 3036.

Gagne, R.M. The Conditions of Learning, (Third Edition). New York: Holt, Rinehart and Winston, 1977.

Gallagher, James Joseph and Robert E. Yager. "Science Educators' Perceptions of Problems Facing Science Education: A Report of Five Surveys." Journal of Research in Science Teaching, June, 1981, Vol. 18, No. 6, pp. 505-514.

Gold, Robert S. and David F. Duncan. "Computers and Health Education." The Journal of School Health, November, 1980, Vol. 50, No. 9, pp. 503-505.

Goldenberg, Paul. Special technology for special children. Baltimore, MD: University Park Press, 1979.

Gould, Laura and William Finzer. "A Study of TRIP: A Computer System for Animating Time-Rate-Distance Problems." International Journal of Man-Machine Studies, 1982, Vol. 17, pp. 119-126.

Griffiths, D.H., American Journal of Physics, 1976, Vol. 44 p. 81.

Hanley, T. Macro-research on technology; Micro-research on education. Paper presented at the Special Education R & D Symposium, Special Education Programs, U.S. Department of Education, June, 1984.

Hargan, C., N. Hibbits, and R. Seidel. D.C. Secondary Schools Project for Adopting Computer-Aided Education. (HumRRO Final Report 78-11), Human Resources Research Organization, Alexandria, Virginia, August, 1978.

Hart, M. "Using computers to understand mathematics." Mathematics Teaching, 1977, Vol. 80, pp. 40-42.

Hatfield, L. L. "Computers in mathematics instruction." Studies in Mathematics, 1969, Vol. 19, pp. 129-152.

Haven, Robert. School Microware Directory. Dresden, ME: Dresden Associates. Semiannual.

____________. School Microware REviews. Dresden, ME: Dresden Associates, P.O. Box 246. Three issues/year.

____________. Ed. The Software Finder. Dresden, Maine: Dresden Associates and TEchnical Education Research Centers, Inc. 1983.

Haven, R. "Laboratory program for computer assisted learning (Project LOCAL)." In D. Post (ed.), The use of computers in secondary school mathematics. Newburyport, Ma.: ENTELEK, 1970.

Henry, Loren L. "Computers and Middle School Mathematics." Viewpoints in Teaching and Learning, Spring, 1981, Vol. 57, No. 2, pp. 46-54.

Hess, R., & Tenezakis, M. The computer as a socializing agent: Some socioaffective outcomes of CAI. AV Communication Review, 1973, 21 (3), 311-325.

Hoffman, I. "Effect of computer application on generalization skills and achievement in a second year algebra course" (Doctoral dissertation, University of Denver, 1971). Dissertation Abstracts International, 1971, 32B, pp. 2856-2857.

Hofstadter, D.R. Godel, Escher, Bach: An Eternal Golden Braid. New York: Basic Books, 1979.

Holoien, M. "Calculus and computing: a comparative study of the effectiveness of computer programming" (Doctoral dissertation, University of Minnesota, 1970). Dissertation Abstracts International, 1971, 31A, p. 4490.

110

Hughes, D. "A Study of the Use of Computer-Simulated Experiments in the Physics Classroom." Journal of Computer-Based Instruction, 1974, Vol. 1, pp. 1-6.

Hundhausen, Joan Rohrer. "Advanced Ordinary Differential Equations in an Engineering Environment: A Computer-Assisted Approach." The American Mathematical Monthly, October, 1980, Vol. 87, No. 8, pp. 662-669.

Jamison, D., P. Suppes, and S. Wells. "The Effectiveness of Alternative Instructional Media: A Survey." Review of Educational Research, 1974, Vol. 44, pp. 1-67.

Johnson, D.C. Preliminary reports I and II: Computer assisted mathematics project. Minneapolis-St. Paul: University of Minnesota, 1966.

Johnson, R.E. "The effect of activity oriented lessons on the achievement and attitudes of seventh grade students in mathematics" (Doctoral dissertation, University of Minnesota, 1970). Dissertation Abstracts International, 1971, 32A, p. 305.

Jones, Nancy Baker, and Larry Vaughan. Evaluation of Educational Software: A Guide to Guides. Chelmsford, MA: Northeast Regional Exchange, Inc., 1983.

Kantowski, Mary Grace. "The Microcomputer and Instruction in Geometry." Viewpoints in Teaching and Learning, Spring, 1981, Vol. 57, No. 2, pp. 71-81.

Karian, Z.A. "The use of computers in simulations." Paper presented at the annual meeting of the American Educational Research Association, New York, April, 1977. (ERIC Document Reproduction Service No. Ed 165 706).

Karplus, Robert. "Teaching For the Development of Reasoning." In Anton E. Lawson (ed.), The Psychology of Teaching for Thinking and Creativity. Columbus, Ohio: The Ohio State University, 1979.

Kay, Alan C. "Microelectronics and the personal computer." Scientific American, September, 1977, pp. 231-244.

Kepner, Henry S., Jr., (ed.). Computers in the Classroom. Washington, D.C.: NEA Publishing, 1982.

Klingsporn Paul E. "Response to 'Comment on 'Fraunhofer diffraction patterns from apertures illuminated with nonparallel light.''" American Journal of Physics, Vol. 48, No. 4, pp. 314-316.

Koethe, W. "Computers in the math curriculum." In J.B. Edwards, A.S. Ellis, et. al. (eds.), Computer Application in Instruction. Hanover, N.H.: Houghton Mifflin, 1978.

Kraus, William H. "Using a Computer Game to Reinforce Skills in Addition Basic Facts in Second Grade." Journal for Research in Mathematics Education, March, 1981, Vol. 12, No. 2, pp. 152-155.

Krull, S.M. "An analysis of the effects of learning to program a computer on student math performance and attitude toward school" (Doctoral dissertation, Saint Louis University, 1979). Dissertation Abstracts International, 1979, 11A, p. 5711.

Kulik, James A., Robert L. Bangert, and George W. Williams. "Effects of Computer-Based Teaching on Secondary School Students." Journal of Educational Psychology, February, 1983, Vol. 75, No. 1, pp. 19-26.

Lathrop, Ann. "San Mateo County Office of Education and Computer-Using Educators: SOFTSWAP." The Computing Teacher, April, 1983, Vol. 10, No. 8, pp. 20-28.

__________, Ed. The Digest of Software Reviews: Education. Fresno, California: School and Home Courseware, Inc. Quarterly.

Lawson, Anton E. and C.A. Lawson, "A Theory of Teaching For Conceptual Understanding, Rational Thought, and Creativity." In Anton E. Lawson (ed.), The Psychology of Teaching For Thinking and Creativity. Columbus, Ohio: The Ohio State University, 1979.

Layman, J.W. "Crisis in physics teaching: A report to the American Association of Physics Teachers Crisis committee." State University of New York at Stony Brook, Stony Brook, NY, 1983.

Levin, Henry and Russell Rumberger. "The Educational Implications of High Technology". Technology Review, August/September, 1983, Vol. 86, No. 6, pp. 18-21.

Lewin, Ann White. "Down With Green Lambs: A discussion of games, software and suggestions for developments in the field." Theory Into Practice, Fall, 1983

__________. "Microcomputers in Informal Learning Centers." Accepted for publication by Journal of Educational Technology Systems, Fall, 1983, Vol. 12, No. 2.

Licklider, J.C.R. "Impact of information technology on education in science and technology." In D.K. Deringer and A.K. Molnar (eds.), Technology in Science Education: The Next Ten Years: Perspectives and Recommendations, Washington, D.C.: National Science Foundation, July, 1979, pp. 1-10.

Lipson, Joseph I. and Kathleen M. Fisher. Public Education and Electronic Technologies. Madison, WI: Wisconsin Center for Education Research, The University of Wisconsin, June, 1982.

Lowry, Pat G. "An Irrational Calculation of Pi." Creative Computing, December, 1981, pp. 238-239.

Luehrmann, Arthur. "Computer Literacy." EDUCOM Bulletin, Winter, 1981, pp. 14-15, 24.

________________. "Slicing Through Spaghetti Code." The Computing Teacher, April, 1983, Vol. 10, No. 8, pp. 9-15.

McInerney, Michael and Kenneth Williams. "Interfacing the Atari Microcomputer in the Science Laboratory." The Journal of Computers in Mathematics and Science Teaching, Spring, 1983, Vol. 2, No. 3, pp. 21-25

Malone, T.W. What Makes Things Fun to Learn? A Study of Intrinsically Motivating Computer Games. Palo Alto, CA: Xerox Palo Alto Research Center, 1980.

Malouf, D. Research on the effectiveness of microcomputers in special education. Project abstract, Office of Special Education Programs, U.S. Department of Education, 1984.

Marsh, David P. "Applications of microcomputers in the teaching of physics - 6502 software." American Journal of Physics, November, 1980, Vol. 48, No. 11, pp. 933-937.

Mason, R.E.A. and R. Seidel, Individualization Project of the Ontario Institute for Studies in Education. Consulting Report, December 12, 1978. Toronto: Ontario Institute for Studies in Education, 1978.

Matz, M. "Towards a process model of high school algebra errors." In D.H. Sleeman and J.S. Brown (eds.), Intelligent Tutoring Systems. London: Academic Press, 1981.

_______. "The acceptance of C.A.I. in the educational field and its probable use in elementary and secondary education in the next decade." Paper presented at the Third International Learning Technologies Congress and Exposition, Washington, D.C., February 25-27, 1980.

Melmed, Arthur S. "Information Technology for Education: An Agenda for the Eighties." T.H.E. Journal, November, 1980, Vol. 7, No. 6, p. 46.

113

Miller, Inabeth. Microcomputer Directory: Applications in Educational Settings, 2nd ed. Cambridge: Gutman Library, Harvard Graduate School of Education, 1982.

Miller, J.L. Natural Learning. Unpublished Ph.D. Thesis. Cambridge, Massachusetts: Harvard University, 1980.

Milner, S.D. "The effects of teaching computer programming on performance in mathematics" (Doctoral dissertation, University of Pittsburgh, 1972). Dissertation Abstracts International, 1972, 33A, p. 4183).

Modisett, D.M. "Effects of computer-assisted instruction on achievement in remedial secondary mathematical computation" (Doctoral dissertation, Fordham University, 1980). Dissertation Abstracts International, 1980, 11A, p. 5770)

Moore, John W. "Instructional Computer Programs - Chemistry." and "SERAPHIM Catalog". Prepared by Project SERAPHIM, NSF Development in Science Education, May-June 1983. (Moore is at Department of Chemistry, Eastern Michigan University, Ypsilanti, MI)

Morgan, C. and W. Richardson. "Computer-Assisted Instruction Program." Board of Education of Montgomery County, Rockville, Maryland, 1974.

Naiman, Adeline. "It's time for the next step." Computer Update, January/February 1983, pp. 10-11.

National Council of Teachers of Mathematics, An Agenda for Action - Recommendations for School Mathematics of the 1980s. Reston, VA: NCTM Press, 1980.

National Science Board, Commission on Precollege Education in Mathematics, Science, and Technology. "The Mathematical Sciences Curriculum K-12: What Is Still Fundamental and What Is Not?" Washington, D.C., October 4, 1982.

National Science Board, Commission on Precollege Education in Mathematics, Science, and Technology. "Today's problems, tomorrow's crises." Washington, D.C., 1982.

Novak, J.D. "Implications for Teaching of Research on Learning." In Mary Budd Rowe (ed.), What Reseach Says to the Science Teacher, Volume 2. Washington, D.C.: National Science Teachers Association, 1979.

__________. A Theory of Education. Ithaca, New York: Cornell University Press, 1977.

Office of Technology Assessment, Congress of the United States, Computer Technology in Medical Education and Assessment (Background Report), Washington, D.C.: U.S. Government Printing House, September, 1979.

Olds, Henry F., Jr., Judah L. Schwartz, and Nancy A. Willie. People and Computers: Who Teaches Whom? Newton, Ma: Education Development Center, September, 1980.

Olivo, Richard F. "Microcomputers as Laboratory Instruments." NECC Proceedings, pp. 81-95. June 17-19, 1981, Denton, TX.

Overton, Victoria. "Research in Instructional Computing and Mathematics Education." Viewpoints in Teaching and Learning, Spring, 1981, Vol. 57, No. 2, pp. 23-36.

Pachter, S.N. "A computer assisted tutorial module for teaching the factoring of second degree polynomials to regents level ninth year mathematics students" (Doctoral dissertation, Columbia University Teachers College, 1979). Dissertation Abstracts International, 1979, 4A, p. 1843.

Palmer, Glenn E. "Computer Applications in the Freshman Laboratory." Journal of Chemical Education, December, 1981, Vol. 58, No. 12, p. 995.

Papert, Seymour. Mindstorms: Children, Computing and Powerful Ideas. New York: Basic Books, 1981.

_______________. "Teaching Children to be Mathematicians Versus Teaching About Mathematics." Logo Memo No. 4. Cambridge, MA: Logo Group, Laboratory for Computer Science, Massachusetts Institute of Technology, July, 1971.

Parrot, Wilbur. "Math Education: A Bibliography, Parts I and II." Classroom Computer News, September/October 1981, Vol. 2, No. 1, pp. 10-11 and November/December, 1981, Vol. 2, No. 3, pp. 8-9.

Pascual-Leone, J. "On Learning and Development, Piagetian Style: Reply to Lefebvre-Pinard." 1 Canadian Psychological Review. 1976, Vol. 17, pp. 270-289.

Pea, R. D., & Kurland, D. M. Learning LOGO programming and the development of planning skills (Tech. Rep. No. 16). New York: Center for Children and Technology, 1983.

__________. Symbol Systems and thinking skills: Logo in context. Paper presented at LOGO 84 conference, Massachusetts Institute of Technology, 1984.

Peters, Hal. "The Future of Computers in Science Education: An Imaginary Dialogue." Paper prepared for the SAMSON Project, January, 1983. See the appendix of this report.

Piele, Donald T. "Spirolaterals." Creative Computing, March, 1982, Vol. 8, No. 3, pp. 114-122.

Polya, G. How to Solve It: A New Aspect of Mathematical Method. Second edition, Garden City, New York: Doubleday/Anchor, 1957.

Powers, Michael H. "A Computer Analogy for Illustrating Entropy Concepts." The Journal of Computers in Mathematical and Science Teaching, Winter, 1982, Vol. 2, No. 2, pp. 19-21.

Rafert, Bruce and R.C. Nicklin. "Microcomputers in the Laboratory." American Journal of Physics, February, 1982, Vol. 50, No. 2, pp. 108-113.

Ragosta, Marjorie, Paul Holland and Dean Jamison. "Computer-assisted instruction and compensatory education: The ETS/LAUSD study." Final Report. Princeton, N.J.: Educational Testing Service, 1981.

Ralston, Anthony. "Computer Science, Mathematics, and the Undergraduate Curricula in Both." American Mathematical Monthly, August-September, 1981, Vol. 88, No. 7, pp. 472-485.

________________. Report to the National Science Board Commission on Precollege Education in Mathematics, Science, and Technology, May 24, 1983. Washington, D.C.: National Science Foundation, 1983.

RICE - Resources in Computer Education. Northwest Regional Educational Laboratory, 300 S.W. 6th Avenue, Portland, OR 97204. (503) 248-6800.

Roark, Anne C. "Mathematics: Fundamental Changes Ahead." The Chronicle of Higher Education, Vol. 16, No. 12, May 15, 1978, p. 3. [RFT: H.O. Pollak quote in this article]

Roberts, Nancy. "Introducing Computer Simulation into the High School: An Applied Mathematics Curriculum." Mathematics Teacher, November, 1981, Vol. 74, No. 8, pp. 647-652.

Robitaille, D.F. "The effect of computer utilization on the achievement and attitudes of ninth-grade mathematics students." Journal for Research in Mathematics Education, 1977, Vol. 8, pp. 26-32.

Romero, S.R. "The effectiveness of computer assisted instruction

in mathemtics at the middle school" (Doctoral dissertation, University of the Pacific, 1979). Dissertation Abstracts International, 1979, 2A, p. 577.

Ronan, F.D. "Study of the effectiveness of a computer when used as a teaching and learning tool in high school mathematics" (Doctoral dissertation, The University of Michigan, 1971). Dissertation Abstracts International, 1971, 32A, pp. 1264-1265.

Rosen, A. "A Computer-Oriented Course in High School Physics." The Science Teacher, June, 1975, Vol. 42, p. 18.

Rowbotham, Neil. "Using a microcomputer in science teaching." The School Science Review, September, 1981, Vol. 63, No. 222, pp. 70-77.

Sauer, G. "Playing the quantum shuffling game with programmable pocket calculators and microcomputers." Physics Education, 1981, Vol. 16, pp. 108-111.

Schwartz, Judah L. "The Semantic Calculator." Classroom Computer News, March-April 1982, Vol. 2, No. 4, pp. 22-24.

Schoolman, Harold M. and Lionel M. Bernstein. "Computer Use in Diagnosis, Prognosis, and Therapy." Science, May, 1978, Vol. 200, pp. 926-931.

Shapira, Gary and Joseph B. Justice, Jr. "Computer-Assisted Instruction in General Chemistry." Journal of Chemical Education, September, 1981, Vol. 58, No. 9, p. 694.

Shaw, C.N. "Effects of three instructional strategies on achievement in a remedial arithmetic program" (Doctoral dissertation, The Florida State University, 1968). Dissertation Abstracts International, 1968, 29A, pp. 1479-1480.

Sheingold, Karen. "Issues related to the implementation of computer technology in schools: a cross-sectional study." Children's Electronic Laboratory, Bank Street College of Education, Memo No. 1, Preliminary Report to the National Institute of Education.

Shymansky, James A. and Bill W. Aldridge. "The teacher crisis in secondary school science and mathematics." Educational Leadership, November, 1982, Vol. 40, No. 2, pp. 61-62.

Slattow, G., Demonstration of the PLATO IV computer based education system, Final Report, January 1,1972 - June 30, 1972. Urbana, Ill.: University of Illinois, 1977. (ERIC Document Reproduction Service No. ED 158 767).

Sloyer, Clifford and Lynn H. Smith. "Applied Mathematics via Student-Created Computer Graphics." The Journal of Computers

in Mathematics and Science Teaching, Spring, 1983, Vol. 2, No. 3, pp. 17-20.

Sparkes, Bob. "Microcomputers in physics." Physics Education, 1981, Vol. 16, pp. 145-151.

___________. "Microcomputers in science teaching." The School Science Review, March, 1982, Vol. 63, No. 224, pp. 442-452.

Spero, S.W. "Computers and calculators: a strategy for combined usage in trigonometry." Paper presented at the Conference on Computers in the Undergraduate Curricula, East Lansing, Michigan, June, 1977. (ERIC Document Reproduction Service No. ED 156 208).

SPIF - School Practices Information File. Education Service Group, 1200 Route 7, Latham, NY 12110. (518) 783-1161.

Stinson, Stephen. "Technician program uses advanced instruments." Chemical and Engineering News, March 16, 1981, Vol. 59, No. 11, pp. 23-24.

Thier, H.D. "Mathematics and science education in the secondary school." Lawrence Hall of Science, University of California, Berkeley, 1983.

Thomas, C.W. and W.H. Churchill. "Computer aided problem solving (oriented toward mathematics)." School Science and Mathematics, 1977, Vol. 77, pp. 291-293.

Thomas, David B. "The Effectiveness of Computer-Assisted Instruction in Secondary Schools." AEDS Journal, Spring 1979, Vol. 12, No. 3, pp. 103-116.

Thomas, William E. "Science-Based Simulation Development: An Example in Physics." The Journal of Computers in Mathematics and Science Teaching, Spring, 1983, Vol. 2, No. 3, pp. 10-16.

Thompson, G.E. "A comparative study of the effectiveness of four types of feedback in a CAI unit on achievement in mathematics of elementary education majors" (Doctoral disserttion, The Ohio State University, 1977). Dissertation Abstracts International, 1977, 38A, pp. 2634-2635.

Tinker, Robert F. "Idea Exchange: Revolutionizing the Curriculum." Hands On!, Summer, 1983, Vol. 6, No. 2, p. 21.

___________. "Low Cost Science Experiments Using Micros." Hands On!, Spring, 1982, p. 8.

___________. "Microcomputers in the Teaching Lab." The Physics Teacher, February, 1981, Vol. 19, pp. 94-105.

___________. "New Dimensions in Math and Science Software."

Classroom Computer News, March, 1983, Vol. 3, p. 36.

_______________ and Gene A. Stringer. "Microcomputers: Applications to Physics Teaching." The Physics Teacher, October, 1978, Vol. 16, pp. 436-445.

Tocci, Salvatore. "The Microcomputer/Biology 'Interface'." The Science Teacher, May, 1981, Vol. 48, No. 5, pp. 60-62.

Toothacker, W.S. "A critical look at introductory laboratory instruction." American Journal of Physics, June, 1983, Vol. 51, No. 6, pp. 516-520.

Trowbridge, David and Alfred Bork. "Computer-Based Learning Modules for Early Adolescence." Proceedings of the 3d World Conference on Computers in Education, Lausanne, Switzerland.

U.S. House of Representatives, Committee on Science and Technology, 96th Congress. Information technology in education. Washington, D.C., April 2-3, 1980.

Uttal, William R., Miriam Rogers, Ramelle Hieronymous, and Timothy Pasich. "Genertive C.A.I. in analytic geometry." Ann Arbor, Mi.: University of Michigan, November, 1969.

VanLehn, Kurt. Bugs are not Enough: Empirical Studies of Bugs, Impasses and Repairs in Procedural Skills. Palo Alto, CA: Xerox Palo Alto Research Center, Cognitive and Instructional Sciences Series CIS-11, March 1981.

Vargas, J.S. Behavioral Psychology for Teachers. New York: Harper and Row, 1977.

Vinsonhaler, J.F. and R.K. Bass. "A Summary of Ten Major Studies on CAI Drill and Practice." Educational Technology, July, 1972, Vol. 12, pp. 29-32.

Wagner, William J. "Circles, Polar Graphs, and a Computer - Some Unexpected Results." Mathematics Teacher, April, 1982, Vol. 75, No. 4, pp. 323-328.

Washburn, R.M. "CEMP - A computer enriched mathematics program" (Doctoral dissertation, Cornell University, 1969). Dissertation Abstracts International, 1969, 30A, p. 5179.

Watt, Daniel H. "A Comparison of the Problem Solving Styles of Two Students Learning Logo: a Computer Language for Children." Creative Computing, December, 1979, pp. 86-90.

_______________. Final Report of the Brookline LOGO Project, Part III. Cambridge, MA: Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1979.

Weiner, M., S. Malkia, C. Helm, and J. Howell. An evaluation of the 1968-1969 New York City computer assisted instruction project in elementary arithmetic. New York: Office of Instructional Research and Program Evaluation, Division of Teacher Education, City University of New York, 1970.

Weir, Sylvia. Final Report of the Brookline LOGO Project, Part II. Cambridge, MA: Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1979.

Wells, S. The impact of varying levels of computer-assisted instruction on the academic performance of disadvantaged students. Research bulletin. Princeton, N.J.: Educational Testing Service, 1974. (ERIC Document Reproduction Service No. ED 157 899).

Weichers, G. "Programming as an educational tool." International Journal of Mathematical Education in Science and Technology, 1974, Vol. 5, pp. 699-703.

Weiss, D.J. (ed.). Proceedings of the 1977 Computerized Adaptive Testing Conference. Held at the University of Minnesota, July, 1977. Minneapolis, MN: University of Minnesota, Department of Psychology, 1978.

Whisnant, David M. "Instrument Simulations for the General Chemistry Laboratory." The Journal of Computers in Mathematics and Science Teaching, Winter, 1982, Vol. 2, No. 2, pp. 8-10.

White, Barbara Y. Designing Computer Games to Facilitate Learning, Cambridge, Mass.: Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Februry, 1981.

Wilkins, P.W. "The effects of computer assisted classroom management on the achievement and attitudes of eighthgrade mathematics students" (Doctoral dissertation, Arizona State University, 1975). Dissertation Abstracts International, 1975, 36A, p. 3379.

Woodruff, Sharon, (ed.), Microcomputers in Education: Resource Handbook. Cambridge, MA: Technical Education Research Centers, Inc., 1982.

Wright, E.B. "Investigation of selected decision-making processes for aspects of a computer-assisted and mastery learning model in basic mathematics" (Doctoral dissertation, The Pennsylvania State University, 1977). Dissertation Abstracts International, 1977, 38A, p. 2634.

Yager, R.E. "The crisis in biology education." American Biology Teacher, 1982, Vol. 44, No. 6, pp. 328-336.

__________, Sarah E. Klein, and Donald W. McCurdy. "Science Activities Are Central to Science Education in the Elementary School: An NSTA Position Statement." Science and Children, October, 1981, Vol. 19, No. 2, pp. 42-43.

APPENDIX A

MATH AND SCIENCE SOFTWARE VENDORS

Math and Science Software Vendors

3R Software
P.O. Box 3115
Jamaica, NY 11431

A.U. Software
P.O. Box 597
Colleyville, TX 76034

Abbott Educational Software
334 Westwood Avenue
E. Longmeadow, MA 01028

Academic Software
22 East Quackenbush Ave.
Dumont, NJ 07628

Academy Software
P.O. Box 9403
San Rafael, CA 94912

Acorn Software Products
634 North Carolina Ave. S.E.
Washington, DC 20003

Active Systems, Inc.
Box A-187
Hanover, NH 03755

Addison Wesley Publishing Co.
South Street
Reading, MA 01867

AdminAid MicroSoftware
886 Bransford Court
Fairfield, CA 94533

Adrian Vance Productions
Box 49210
Los Angeles, CA 90049

Aquarius Publishers, Inc.
P.O. Box 128
Indian Rocks Beach, Fl 33535

Atari Program Exchange
155 Moffet Park Drive B-1
Sunnyvale, CA 94086

Athena Software
727 Swarthmore Drive
Newark, DE 19711

Avant-Garde Creations
P.O. Box 30160
Eugene, OR 97403

Basics & Beyond
Box 10
Amawalk, NY 10501

Behavioral Engineering
230 Mount Hermon Rd., Suite 207
Scotts Valley, CA 95066

Bertamax, Inc.
101 Nickerson, Suite 550
Seattle, WA 98109

Bipacs
33 West Walnut Street
Long Beach, NY 11561

BLS/Random House, Inc.
400 Hahn Road
Westminster, MD 21157

Borg-Warner Educational Systems
600 West University Drive
Arlington Hts., IL 60004

BrainBank, Inc.
220 Fifth Avenue, Dept. A
New York, NY 10001

Cactus Software
1442 North McAllister
Tempe, AZ 85281

Cambridge Development Labs
100 5th Avenue
Waltham, MA 02154

Classic Software Productions
7566 John Avenue
St. Louis, MO 63129

Math and Science Software Vendors

Color Software Services
P.O. Box 1708
Greenville, TX 75401

Concept Educational Software
P.O. Box 6184
Allentown, PA 18001

Colmaldor
P.O. Box 356, Postal Station O
Toronto, Ontario M4A 2N9,
CANADA

Conduit
P.O. Box 388
Iowa City, IA 52244

Comm*Data Computer House
P.O. Box 325
Milford, MI 48042

Cove View Press
Box 810
Arcata, CA 95521

Compress
P.O. Box 102
Wentworth, NH 03282

Cow Bay Computing
Box 515
Manhasset, NY 11030

Compu-Tations
P.O. Box 502
Troy, MI 48099

Creative Publications
P.O. Box 10328
Palo Alto, CA 94303

Computer Courseware Services
300 York Avenue
St. Paul, MN 55101

Cross Educational Software
P.O. Box 1536
Ruston, LA 71720

Computer Island
227 Hampton Green
Staten Island, NY 10312

Curriculum Applications
P.O. Box 264
Arlington, MA 02174

Computer Learning Center for Children
1775 East Tropicana Avenue
Liberace Plaza #8
Las Vegas, NV 89109

Computer Resources Inc.-CRI
Route 4
Barrington, NH 03825

Cybernetic Information Systems
Box 9032 Upper Union
Schenectady, NY 12309

Compuware
15 Center Road
Randolph, NJ 07869

Data Processing Consultants
2405 San Pedro, N.E.
Albuquerque, NM 87110

ComQuest
221 East Camelback, St. 1
Phoenix, AZ 85012

Datatech Software Systems
19312 East Eldorado Drive
Aurora, CO 80013

Math and Science Software Vendors

Davidson & Associates
6090 Groveoak Place #14-C
Rancho Palos Verdes, CA 90274

Demi Software
6 Lee Road
Medfield, MA 02052

Desert Sound
16268 Main Street
Hesperia, CA 92345

Developmental Learning Materials
One DLM Park
Allen, TX 75002

Dickens Data Systems
478 Engle Drive
Tucker, GA 30084

Digipac Computer Consulting
907 River Street East
Prince Albert, Sasktchwn, S6V OB3
CANADA

Earthware Computer Services
P.O. Box 30039,
Eugene, OR 97403

Edco
P.O. Box 30846
Orlando, Fl 32863

Edu-Soft
4639 Spruce Street
Philadelphia, PA 19139

Edu-Ware Services, Inc.
P.O. Box 22222
Agoura, CA 91301

Educational Activities, Inc.
P.O. Box 392
Freeport, NY 11520

Educational Computer Systems
136 Fairbanks Road
Oak Ridge, TN 37830

Educational Computing
3144 Valentino Court
Oakton, VA 22124

Educational Courseware
3 Nappa Lane, Dept. GT
Westport, CT 06880

Educational Materials & Equipment Co.
P.O. Box 17
Pelham, NY 10803

Eduational Micro Systems
P.O. Box 471
Chester, NJ 07930

Educational Software
P.O. Box 746
Mc Minnville, OR 97128

Educational Software & Design
P.O. Box 2801
Flagstaff, AZ 86003

Educational Software Marketing
1035 Outer Park Drive, Suite 309
Springfield, IL 62704

Edupro
P.O. Box 51346
Palo ALto, CA 94303

EduTech, Inc.
634 Commonwealth Avenue
Newton Centre, MA 02159

Edutek Corporation
P.O. Box 11354
Palo Alto, CA 94306

Encyclopedia Brittanica Educational Corp.
425 N. Michigan Avenue
Chicago, Il 60611

Fullmer Associates
1132 Via Jose
San Jose, CA 95120

Math and Science Software Vendors

Generic Software
P.O. Box 790
Marquette, MI 49855

George Earl
1302 S. General McMullen
San Antonio, TX 78237

H.E.L. Labratories
95A Halls Croft
Freehold, NJ 07728

Hartley Courseware, Inc.
P.O. Box 431
Dimondale, MI 48821

Hayden Software Company
600 Suffolk Street
Lowell, MA 01853

High Technology Software Products
P.O. Box 60406
Oklahoma City, OK 73146

Hugh Ward
P.O. Box 3412
DeLand, FL 32720

Human Relations Media, Dept. S
175 Tompkins Avenue
Pleasantville, NY 10570

Ideatech
P.O. Box 62451
Sunnyvale, CA 94088

Indian Head Software
1002 Indian Head Drive
Snow Hill,NC 28580

Information Unlimited Software
2401 Marinship Way
Salsalito, CA 94965

Instant Software
Peterborough, NH 03458

International Software Marketing
120 E. Washington Street - 421
Syracuse, NY 13202

ITC Technologies Corporation
7100 Boulevard East #2J
Guttenberg, NJ 07093

J & S Software
140 Reid Avenue
Port Washington, NY 11050

J.B. Hirsch
225 Duke Ellington Blvd. #14A
New York, NY 10025

James P. Birk
Department of Chemistry, Arizona State Univ.
Tempe, AZ 85287

Jensen Software
1440 Rockway
Lakewood, OH 44107

JMH Software of Minnesota, Inc.
4850 Wellington Lane
Minneapolis, MN 55442

Johnson Software
1200 Dale Avenue
Moutain View, CA 94040

Krell Software
130 Stoney Brook Road
Stoney Brook, NY 11790

L.I.F.E. Software Ltd.
c/o Richvale Telecommunications
10610 Bayview Plaza, Unit 18
Richmond Hill, Ontario L4C 3N8
CANADA

Lane Robbins Computer Programs
RD #3 Box 365
Cortelyou Lane
Sommerset, NJ 08873

Math and Science Software Vendors

Lawrence Hall of Science
Math/Computer Education Project
University of CA
Berkeley, CA 94720

The Learning Company
4370 Alpine Road
Portola Valley, CA 94025

Learning Well
200 S. Service Road
Roslyn Heights, NY 11577

Little Bee Educational Programs
P.O. Box 262
Massillon, OH 44648

Little Genius Ltd.
Albany House, Suite 504
324 Regent Street
London, Ontario W1R 5AA
CANADA

M-R Information Systems
P.O. Box 73
Wayne, NJ 17470

MASBO Cooperative Corp.
99 School Street
Weston, MA 02193

Math City/Mathware
919 14th Street
Hermosa Beach, CA 92054

Math Software
1233 Blackthorn Place
Deerfield, IL 60015

MCE, Inc.
157 S. Kalamazoo Hall, Suite 250
Kalamazoo, MI 49007

McGraw-Hill - EDL
1221 Avenue of the Americas
New York, NY 10020

McGraw-Hill/Gregg
1221 Avenue of the Americas
New York, NY 10020

Med Systems Software
P.O. Box 3558
Chapel Hill, NC 27514

Mega-Byte Systems
66 Church Street
Ellenville, NY 12428

Mentor Software, Inc.
P.O. Box 791
Anoka, MN 55303

Mercer Systems, Inc.
87 Scooter Lane
Hicksville, NY 11801

Merlan Scientific
P.O. Box 25
Depew, NY 14043

Meta-Designed Software
P.O. Box 136
Haddonfield, NJ 08033

Micro Learningware
P.O. Box 2134
North Mankato, MN 56001

Micro Music
Musitronics Inc. Distributor
555 Park Drive - P.O. Box 441
Owatonna, MN 55060

Micro Power & Light
12820 Hillcrest Road # 224
Dallas, TX 75230

Math and Science Software Vendors

Micro-Ed, Inc.
P.O. Box 24156
Minneapolis, MN 55424

Microcomputer Software Systems
4716 Lakewood Drive
Metairie, LA 70002

Microcomputer Workshops
103 Puritan Drive
Port Chester, NY 10573

MicroGrams, Inc.
P.O. Box 2146
Loves Park, IL 61130

Microphys
2048 Ford Street
Brooklyn, NY 11229

Milliken Publishing Co.
1100 Research Blvd.
St. Louis, MO 63132

Minnesota Educational Computing
Consortium
2520 Broadway Drive
St. Paul, MN 55113

Modtec
4144 N. Via Villas
Tucson, AZ 85719

Monument Computer Services
Village Data Center
P.O. Box 603
Joshua Tree, CA 92252

MUSE Software
347 No. Charles Street
Baltimore, MD 21201

National Software Marketing
4701 McKinley Street
Hollywood, FL 33021

NCCCD-National Coordinating Center
for Curriculum Development
State University of New York
Stony Brook, NY 11794

Orange Cherry Media
7 Delano Drive
Bedford Hills, NY 10507

Professional Computer Systems
318A Lincoln Court
Bloomingdale, IL 60108

Programs for Learning, Inc.
P.O. Box 954
New Milford, CT 06776

Project COMAL
Commack Public Schools
Hauppage Road
Commack, NY 11725

Project LOCAL Software
c/o Dresden Associates
P.O. Box 246
Dresden, ME 04342

Quality Educational Designs
2924 N.E. Stanton
Portland, OR 97212

Random House
201 East 50th Street
New York, NY 10022

Readers Digest Services
Educational Division
Pleasantville, NY 10022

Redcomp Services
624 West Chenango Road
Castle Creek, NY 13744

Reston Publishing Co.,Inc.
11480 Sunset Hills Road
Reston, VA 22090

Math and Science Software Vendors

Right On Programs
P.O. Box 977
Hunington, NY 11743

Robert Baker Jr.
5845 Topp Court
Carmichael, CA 95608

Robert Davis and Associates
3355 Lenox Road
Atlanta, GA 30304

Sandpiper Software
P.O. Box 336
Maynard, MA 01754

School & Home Courseware
Suite C
1341 Bulldog Lane
Fresno, CA 93710

Science Research Associates
155 North Wacker Drive
Chicago, IL 60606

Society for Visual Education
1345 Diversey Parkway
Chicago, IL 60614

Software Industries
902 Pinecrest
Richardson, TX 75080

Solartek
P.O. Box 298
Guilderland,NY 12084

SouthWest EdPsych Services
P.O. Box 1870
Phoenix, AZ 85001

Sterling Swift Publishing Co.
1600 Fortview Road
Austin, TX 78704

Storybooks of the Future
P.O. Box 4447
Santa Clara, CA 95054

Sunburst Communications
39 Washington Ave. Room VF414
Pleasantville, NY 10570

T.H.E.S.I.S.
P.O. Box 147
Garden City, MI 48135

T.I.E.S.
1925 West County Rd. B2
St. Paul, MN 55113

Taylormade Software
P.O. Box 5574
Lincoln, NE 68505

Teacher's Pet
c/o Glenn Fisher
1517 Holly Street
Berkeley, CA 94703

The Teaching Assistant
22 Steward Drive
Hunington Station, NY 11746

Teaching Tools Microcomputer Service
P.O. Box 50065
Palo Alto, CA 94303

Teck Associates
P.O. Box 8732
White Bear Lake, MN 55110

TYC Software
40 Stuyvesant Manor
Geneseo, NY 14454

Tycom Associates
68 Velma Avenue
Pittsfield, MA 01201

Vernier Software
2920 S.W. 89th Street
Portland, OR 97225

Versa Computing, Inc.
3541 Old Conejo Road #104
Newbury Park, CA 91320

APPENDIX B

THE FUTURE OF COMPUTERS IN SCIENCE EDUCATION:

AN IMAGINARY DIALOGUE

by Harold G. Peters

The Future of Computers in Science Education: An Imaginary Dialogue

Harold J. Peters

In the following, an anonymous questioner, QU, engages Hal Peters, HP, in an imaginary dialogue concerning the future role of computers in education, with special emphasis on science education. In the course of the dialogue, criteria for new science curricula in a computer age come to light...

QU: What's all the excitement about computers in education?

HP: The computer is revolutionizing the way we handle information. Since the stock in trade of education is information, the computer promises to revolutionize education.

QU: But every new wave of technology is hailed as a savior of education. Look at television; it has completely transformed communications technology, and yet what has it done for education?

HP: I have two reactions to that. First, I would argue that television has had an enormous effect on education -- it just doesn't happen to have occurred in the traditional structure of the educational institutions. Second, and more important, I think most thoughtful observers will agree that the computer has tremendously greater potential to affect educational practice than television ever had.

QU: Why is that? What is so special about the computer?

HP: The single most distinguishing characteristic of the computer is its interactiveness. With so much of what we do in education -- and this certainly holds for many of the real failures of institutionally-based educational television -- the student just sits passively, hoping at best to be entertained. With the computer, on the other hand, when it is used well in education the student is almost continuously active: responding to questions, making inquiries, or perhaps creating new program code of his own.

QU: You spoke of a revolution. What are we moving from? What are we moving toward?

HP: In Thomas Kuhn's The Structure of Scientific Revolutions, he speaks of science moving from one paradigm to another, where a paradigm signifies the characteristic mode of attack for scientists of a given era. The paradigm for education over the last couple hundred years at least has featured the classroom, the teacher, and the textbook, where the teacher has been the chief actor, and the textbook has provided the script.

QU: So you would say the textbook is really the characteristic feature of today's educational paradigm?

HP: Yes, and I think that to a large degree the computer will become the defining characteristic of the next educational paradigm as the revolution unfolds.

QU: What of the teacher and the classroom? Won't the computer affect them too?

HP: Of course. Some are predicting that the classroom, that is, the traditional educational institutions, will disappear, with education becoming a home-bound, almost entirely individualized pursuit. I think this is extremely doubtful. The social contributions of the classroom are too important to cast aside.

QU: You're speaking of the social skills that students learn in school?

HP: Not just that. There's a great deal to be gained, too, from the interactions among different students as they are collectively trying to understand new intellectual concepts. And not incidentally, some of the most exciting new research in educational computing concerns what goes on in small group learning with a computer involved.

QU: Does this mean that the teacher disappears from the new paradigm?

HP: Not at all, in my estimation. In the old paradigm, we saw the teacher taking the lead role in the educational process, following the script provided by the textbook. In the new, I think we will more frequently find the teacher joining with students as a learner, using the computer as a tool to actively explore the subject matter.

QU: Okay, so the computer doesn't replace the teacher. Does it replace the textbook? Does it then provide the script in the new paradigm? Or is there no script?

HP: I think we in education have learned from our experiences with so-called "discovery" learning that unguided learning, without a script, is at best much too inefficient, and at worst fails completely. So it seems to me there must still be a script as part of the new paradigm, and I won't be surprised if in most cases it is provided by a textbook.

QU: Now you have really lost me. A while ago I thought you were saying that as part of the revolution, computers would be replacing textbooks.

HP: You could well have inferred that from what I said, but that is not precisely what I meant. Let me try approaching it a little differently. In the present educational paradigm, the overwhelming emphasis is on the acquisition of facts (or at best, concepts). Textbooks are reservoirs of facts and thus provide the script by dictating which are the facts to be acquired. In the new paradigm, the focus will shift from content to process. The emphasis will be on acquiring learning skills, per se, with the computer as an essential tool to be used by the learner. So in this sense the computer does replace the textbook as the defining characteristic of the new paradigm.

QU: There's still the matter of the script.

HP: That's right. But in keeping with the paradigm shift, the new scripts must focus on process instead of content; they must guide learners -- students and teachers together -- through activities, frequently utilizing the computer, that sharpen the learners' skills at learning.

QU: It sounds like we need a whole new generation of textbooks.

HP: Precisely. The frequent cry is that we desperately need educational software. But I can cite instance after instance of excellent software whose educational potential is virtually untouched for lack of appropriate scripts.

QU: Can you cite a particular example that would illustrate?

HP: One software package that virtually everyone familiar with the new personal computers is aware of is Visicalc. Here is software that was created for a non-educational purpose, of

course, but electronic spreadsheets of this sort have great educational potential. With the appropriate templates and scripts, they could be used very effectively in support of laboratory calculational needs and for that matter, in support of general purpose problem solving.

QU: Is there an example that is closer to education in its original purpose?

HP: One that comes to mind is ARBPLOT, which is a large set of programs designed to support a calculus lecturer in the classroom. Here an excellent script already exists, but it is directed to the teacher. While this package is becoming very popular with calculus teachers, I think it could be even more popular with their students, if the appropriate script were provided, detailing exercise after exercise that applied these general-purpose support programs to specific topics in the calculus course.

QU: Let's return to one feature of your new educational paradigm that intrigues me: the role of the teacher. Why do you portray the teacher perpetually in the role of a learner? Won't future teachers be trained before entering the classroom? How can they teach if they are not trained?

HP: Those are all good questions, but I must digress a bit in order to answer. The central, most salient feature of our modern society is change. Assuming that the purpose of formal education is to prepare young people for a productive life in society, then it immediately follows that a primary function of education must be to equip its graduates for dealing with change. This in turn implies that the graduate will be capable of continuing his education, learning anew to cope with his environment as that environment continues to change.

QU: And it implies that teachers, too, must keep on learning?

HP: That's right, especially in the sciences, where the pace of change is so rapid.

QU: It just occurred to me that the computer has a curious role here. Isn't it a major contributor to many of the changes that are now in process?

HP: It certainly is. So as educators we must address the computer as one of the important phenomena for our students to be introduced to. But because it is such a powerful

device for the manipulation of information, the computer promises to be an important part of the solution.

QU: It seems that for science education, the significance of the computer is heightened even further.

HP: Yes, I think so. Science educators espouse a dual purpose: to prepare all graduates for knowlegeable participation in a high-tech society, and to help prepare a select few for careers in science. Since computers are such a prominent feature of the new technology, it is imperative that all who want to fully participate be computer literate. And computers have become an especially important tool for scientists, so for the science career-bound student, computer literacy is doubly imperative.

QU: But isn't it also true that many of the topics of science, as well as its processes, fit naturally with those techniques that are emerging as successful instructional applications of the computer?

HP: In fact, it is in the sciences that the most promising educational applications of computing have thus far been demonstrated.

QU: Well, we've all heard of drill and practice, tutorials, simulations and games, etc. Is that the sort of application you are talking about?

HP: That's it, but only in part. In the context of what we were discussing earlier, each of these types of educational computer usage needs to be viewed in a little different light. Most of these types of application have only scratched the surface in terms of how effective they could be.

QU: Why don't you get more specific. I have seen computer-based drills that help students learn the symbols for the chemical elements. What more can be said about something as straightforward as that?

HP: There are at least two major points to be made here. First, such drills as you refer to are almost always handled in a non-optimal, non-individualized manner. While there is still much to be researched concerning optimization, some important principles have evolved from years of experience at Stanford and Illinois (PLATO). Using these ideas, drill

authors could even today be producing drills that would be much more efficient for student learning.

The second, and perhaps more crucial point relates to our earlier discussion of the need for formal education to lay the basis for lifelong learning. Much more valuable to the student than the facts that can be mastered through use of a specific drill would be the skill for constructing the student's own future drills whenever the occasion might demand some facility with a given body of factual information.

QU: It sounds like we have a clearly defined software need here: drill routines that incorporate the best that is known regarding optimization techniques, and that allow students to provide their own lists of facts.

HP: Right, but we should say "learners" rather than "students", both to imply that learning will hopefully continue beyond one's student days, and -- once again -- that teachers will be joining together with students in the learning process. So in the best classroom uses of computer-based drills, we might well see students and teachers deciding jointly what are some of the factual data that should be drilled.

QU: Let's move to another of the more common educational computing applications, the simulation. This seems to have been a very popular application in the sciences.

HP: Of course! What could be a more natural application in the sciences than to allow students to go beyond the confines of the laboratory or field investigation -- which may be inaccessible, too costly, dangerous, or time-consuming anyway -- and to carry out as many simulated investigations as they may like, focusing throughout on the logical problems of experimental design, hypothesis testing, etc?

QU: That sounded a bit sarcastic.

HP: Well, the problem is that it just doesn't work out that way most of the time. As beautifully executed as many simulations are, the typical student simply doesn't know how to make good use of them without guidance.

QU: It sounds like we are talking about the need for scripts again. Can you give us any illustrations here?

HP: One of the best illustrations I know is a genetics simulation called CATLAB. It allows students to produce litter after litter of kittens, with coats of many colors, displayed brilliantly on the computer. As a bare computer program, it would probably entertain many students for a while, but it would be a rare student who learned much genetics from it.

Thanks to a conscientious author, however, a very complete script -- in the form of printed guides for the student and teacher -- is provided. Following these guides, student and teacher together can learn a great deal about the inheritance of coat color in cats, and thereby about Mendelian genetics. At another level, they will be learning a great deal about the methodology, the process of science.

QU: And at still another level, if students perform some of the investigations as a group, they will undoubtedly learn something about the sociology of science, as they collectively wrestle with which hypotheses to accept and reject.

HP: There's a great deal more to be said about the use of simulations in science education, but let me bring up just one point that touches on another important area of need. Practicing scientists in the course of their activities frequently rely on the computer in constructing models of natural phenomena they are studying. Students can profitably be introduced to this model-building activity, both to become acquainted with some more of the methodology of science, and to gain facility with an approach that can serve them well in problem-solving situations of many kinds.

QU: Can you be a little more specific? How will facility with computer-based modeling help a person in solving problems?

HP: Consider a homely little analogy. Many succesful problem-solvers find it helpful to make sketches of their problem, showing some of the relationships among the parts of the problem. Just this act of visualizing the structure of the problem can often begin to suggest a solution.

I envision future problem solvers routinely building computer models of the problems they are trying to solve. Here again, just the act of systematizing the problem to get it into the computer may well begin to suggest a solution. But beyond this, once the problem is modeled, the person trying to forge a solution can try all sorts of "what if?" types of scenarios, exploring tentative solutions and their consequences.

QU: I can see the utility now, but where is the educational need?

HP: We must give students practice at using the computer to help them think about problems. They need to become facile at formulating problems in ways that are compatible with the computer's capabilities.

QU: But...

HP: You are correct in stopping me there. I have it exactly backwards! And that gets us more directly to what is the need. We as computer specialists need to come up with software systems that are compatible with the nature of problems that people are trying to solve.

QU: That sounds like a pretty ambitious agenda. Can you bring it back down to earth at all?

HP: Sure. There are two very nice examples of the kind of thing I am driving at that fortunately already exist. One is LOGO, Seymour Papert's brainchild, which among other things allows students to <u>modularize</u> problems into small manageable chunks that can be handled with small, self-contained <u>procedures</u>. A second example is the version of DYNAMO -- Forrester's simulation language -- that is now available on small computers. And happily DYNAMO comes with the kind of script for educational use that we were talking about earlier.

QU: So the need here is for more LOGOs and more DYNAMOs?

HP: Yes, and for many more scripts that lead learners through more varieties of problems to which these languages can be applied.

QU: Are we ready to move on from the topic of simulations, then?

HP: Not quite. There's still one more point to be made that has very broad implications across the science curricula. We can get at this point by considering one of the common simulations of projectile motion that have students fire simulated missiles at simulated targets.

QU: You mean the type where I can specify a starting velocity for the projectile, as well as an angle of inclination, and then follow its trajectory to where it strikes?

HP: Yes, that's the type. And the point I want to make is that simulations like this can be used for at least two very different purposes. The most obvious one is the one we already touched on earlier, that is, students can use the simulation to become a lot more familiar with a phenomenon that is rather difficult to study closely in its natural form. They can become very familiar with how starting velocity and angle of inclination qualitatively affect the trajectory before they ever try to tackle the abstract form of these relationships, as embodied in mathematical equations. The second purpose is just the other side of that coin: they can use the simulation to study the mathematical equation itself, watching carefully just how the trajectory changes when different parameters in the equation are varied by different amounts.

QU: Once more I see the utility of what you are describing, and once more I must ask: Where is the educational need?

HP: We are back to what we mentioned much earlier: the need for new textbooks, for entire new curricula across the sciences. In one or the other of the two ways I was just describing, simulations -- supplemented by printed or on-line scripts -- can be effectively applied to just about every major topic that is or ought to be addressed in the school science curricula. We need whole new science textbook series that invoke the computer at virtually every turn. I have emphasized simulations at this point because of their great instructive potential, but there are of course many other types of applications that should be called upon, too. Some of these we touched upon earlier.

QU: Throughout our conversation, you have discussed implications and uses of the computer that seem applicable across all the different sciences. Do you feel your observations are also applicable at all grade levels?

HP: To some degree, yes. I believe, however, that one of the special problems in elementary level science education is that teachers at that level are in general woefully prepared in the sciences: they don't know the subject matter and in most cases they are not aware of the best pedagogy.

QU: Do you see any hope from the computer?

HP: If we can come up with the new science curricula as we have been discussing, and if these employ the computer well to become effective <u>self-instructional</u> materials for joint use by students and teachers, then I think the answer is self-evident. The teachers will be learning the subject matter, and at the same time will be exposed to some of the best pedagogy.

QU: Do you envision the new materials being used in preservice training for teachers, as well as in the elementary classrooms themselves?

HP: The preservice use is the most critical, because this is how we are most likely to bring the new educational paradigm into being.

QU: It seems to me that producing these new curricula will be enormously expensive. Can you give us any idea of the likely costs?

HP: Let's go back to ARBPLOT to try to provide one benchmark. I have implied that the software here could provide the basis for a new calculus course. Conservatively, there are two to three man-years of investment in the development of that package as it now stands, let's say an investment of $100,000. It would probably take that much more to produce the new textbook.

QU: So we are looking at something in the neighborhood of a quarter million dollars for one new course. By the time that development is complete don't we run the danger that new hardware developments will render the course obsolete?

HP: That is always a danger, of course. To avoid that, we need to be constantly vigilant that we are designing the new materials in hardware-independent ways.

QU: I guess that could lead into another whole dialogue of its own.

HP: That's right.

References

Brown, A.R. and Harris, M. ARBPLOT. CONDUIT: Iowa City, 1982.

Kinnear, J. CATLAB. CONDUIT: Iowa City, 1982.

Kuhn, T.S. The Structure of Scientific Revolutions. The University of Chicago Press: Chicago, 1970.

Papert, S. Mindstorms: Children, Computers, and Powerful Ideas. Basic Books: New York, 1980.

Pugh-Roberts Associates. Micro-DYNAMO. Addison-Wesley: Reading, MA, 1983.

Software Arts, Inc. Visicalc. Personal Software, Inc.: Sunnyvale, CA, 1981.

APPENDIX C

SOURCES OF SOFTWARE INFORMATION

SOURCES OF SOFTWARE INFORMATION

SOFTWARE REVIEWS

This listing is limited to publications whose primary purpose is to provide review and evaluation of software. In addition to these sources, many computer-related periodicals reserve a section in each issue for evaluating new software products. See the Periodicals listings for computer magazines and newsletters and note that those with an asterisk are also sources for courseware review.

Courseware Report Card. 150 West Carob Street, Compton, CA 90220. 5 issues/year. In-depth description and evaluation. Average 20-25 reviews per issue.

The Digest of Software Reviews: Education c/o School and Home Courseware, Inc., 1341 Bulldog Lane, Suite C, Fresno, CA 93710. 4 issues/year. Abstracts reviews of 50 educational software packages in each issue. Assigns grade level, ERIC descriptors, Sears and Library of Congress subject heading.

Dvorak's Software Review. 704 Solano Avenue, Albany, CA 94706. 8 issues/year. Averages 2-5 reviews per issue. North Star software.

EPIE and Consumers Union Micro-Courseware PRO/FILE and Evaluation. EPIE Institute, P.O. Box 620, Stony Brook, NY 11790. Basic packages for software and hardware evaluation are updated monthly with 2-4 page evaluations sent with MICROGRAM Newsletter.

The Journal of Courseware Review. Apple Educational Foundation, 20525 Mariani Blvd., Cupertino, CA 95014. In-depth evaluations. Contains photo of screens from the programs. Average about 21 per issue, 2-4 pages in length. Apple software.

MicroSIFT News. MicroSIFT, Northwest Regional Educational Laboratory, 300 S.W. Sixth Avenue, Portland, OR 97204. Analysis of field-tested software. Average 12 per isue, one-half page in length.

Pipeline. CONDUIT, University of Iowa, Box 388, Iowa City, IA 52244. 3 issues/year. Primarily college level. Average 8 reviews, 1 or 2 pages in length.

80 Software Critique. P.O. Box 134, Waukegan, IL 60085. 4 issues/year. Offers rating scale 1 100.

Software Review, Microform Review. 520 Riverside Avenue, Westport, CT 06880. 4 issues/year. Average 2-6 reviews, 6-10 pages in length.

BEST COPY AVAILABLE

Other Sources of Software Review and Information: On-Line Databases

Various search services currently available provide descriptive and evaluative information about educational software. Some of these databases also serve as a mechanism for user communication.

Bibliographic Retrieval Services (BRS)
Education Service Group
1200 Route 7
Latham, NY
(518) 783-1161

Among BRS' more than 50 files are several of particular use in locating software information: School Practices Information File (SPIF) lists over 1,500 educational software descriptions including the MARCK and MicroSIFT catalogs; Resources in Computer Education (RICE) contains over 2,000 courseware descriptions with Northwest Regional Educational Laboratory (NWREL) evaluations for approximately 10% of the packages, and DISC, which indexes 17 computer and educational computing periodicals for software reviews as well as articles of general interest.

DIALOG Information Services
3460 Hillview Avenue
Palo Alto, CA 94304
(800) 277-1927

Over 150 databases can be searched on DIALOG including the Microcomputer Index which indexes 23 educational and personal computing magazines fully and provides partial coverage of 10 more business and educational journals for reviews of software and peripherals as well as reports and articles. The Index is also available as a quarterly publication. The Internal Software Database (ISD) lists descriptive information provided by vendors on software available for micros and minis. For a search fee a user can have ISD conduct the search for her/him by writing or calling OneStopSoftShop, 1520 South College, Fort Collins, CO 80524, (303) 482-5000 or (800) 525-5995. Both the Microcomputer Index and ISD can be searched on DIALOG's Knowledge Index, the nighttime and weekend service.

BEST COPY AVAILABLE

The Electric Software Finder
Technical Education Research Centers
1696 Massachusetts Avenue
Cambridge, MA 02138
(617) 547-3890

This free service provides descriptive information on over 1,700 K-12 math and science software packages drawn from Dresden Associates' The Software Finder and from original research funded by the U.S. Department of Education.

PC Telemart
PC Clearinghouse Inc. Publishers
11781 Lee Jackson Highway
Fairfax, VA 22033
Available at retailers

This on-line shopping service lists over 20,000 software applications. Approximately 8% of these are in the education field. A count of the 7th edition of the PC Clearinghouse Directory found 112 math listings and over 80 science packages.

Sofsearch
Sofsearch International Inc.
P.O. Box 5276
San Antonio, TX 78201
(512) 340-8735
(800) 531-5955

For a yearly fee, subscribers receive up to 5 search reports, updated quarterly. Individual, one-time searches are also available on a single-fee basis. Sofsearch provides coverage of all kinds of software including packages for business and general applications.

Software Search
Software News
5 Kane Industrial Drive
Hudson, MA 01749
(617) 562-9308

The software database of Software News is accessible on a one-time basis and covers a range of applications packages.

Software Directories

Below is an annotated listing of the major directories of educational software. Some directories are machine-specific; others provide general listings for a variety of educational levels.

GENERAL

Appleseed. Software Publications, 6 South Street, Milford, NH 03055. Lists software for Apple, TRS-80, Atari and Pet microcomputers, including some educational packages. Free.

Huntington Computing Catalog. P.O. Box 787, Corcoran, CA 93212. Programs primarily for the Apple and some for Atari, PET and TRS-80. New educational catalog out soon.

Index to Computer-Based Learning. 1981 Edition. Instructional Media Laboratory, University of Wisconsin, P.O. Box 413, Milwaukee, WI 53201. Lists over 4800 computer-based learning programs. Each is cross-indexed by source, programming language, central processor type, and programming category.

Instant Software. Peterborough, NH 03458. Free direct mail catalog offered by publishers of Microcomputing (formerly Kilobaud).

Instructor's 1982-1983 Computer Directory for Schools. Attention: Elsa Silander, P.O. Box 6099, Duluth, MN 55806. Guide to the selection of microcomputers, peripherals, books, resources, journals, and free materials, as well as instructional and administrative software. Distributed to secondary and elementary school principals and available to others for $19.95. Currently being updated.

K-12 Micro Media. P.O. Box 17, Valley Cottage, NY 10989. Lists and describes 250 programs from 50 vendors for the Apple, TRS-80, PET and Atari computers.

Marck, Inc. 280 Linden Drive, Branford, CT 06082.

BEST COPY AVAILABLE

Programs listed by computer (Atari, Apple, PET, and TRS-80), subject areas and publisher.

Microcomputers Corporation Catalog. P.O. Box 191, Rye, NY 10580. Hundreds of programs listed by type, including educational. Also lists computer accessories.

Opportunities for Learning, Inc.. Dept. L-4, 8950 Lurline Avenue, Chatsworth, CA 91311. Lists educational software, elementary through college for Apple, PET, Atari, and TRS-80 computers. Listings by subject, and within subject areas, by grade. Free.

PC Clearinghouse Directory. PC Clearinghouse, Inc., Publishers. 11781 Lee Jackson Highway, Fairfax, VA 22033. Eight percent of the 21,000 packages listed are educational with approximately 112 math and 87 science packages included. PC Telemart will soon be available at retailers as an on-line shopping service.

Queue. 5 Chapel Hill Drive, Fairfield, CT 06432. Catalogs educational software available for Apple, Atari, PET, and TRS-80 microcomputers. Covers 40 educational software publishers. Programs grouped by subject and grade.

Reference Manual for Instructional Users of Microcomputers. JEM Research, Discovery Park, University of Victoria, P.O. Box 1700, Victoria, British Columbia V8W 2Y2, CANADA. Indexes over 100 educational software titles referenced according to subject and grade level. Descriptors and evaluations of over 200 representative programs included. An update with approximately 50 reviews appeared in Summer, 1982.

Scholastic Microcomputer Instructional Materials. 904 Sylvan Avenue, Englewood Cliffs, NJ 07632. Free catalog lists books about computing, computer accessories, and educational software for Apple, PET, TRS-80, TI-99/4 and Atari microcomputers.

The Software Catalog: Microcomputers. Elsevier Science Publishing Co., Inc., 52 Vanderbilt Avenue, New York, NY 10017. Two catalogs and two supplements each year are produced from the International Software Database. The Spring 1983 catalog has approximately 750 educational applications entries. It is the successor to the International Software Directory.

The Software Directory. Software Central, P.O. Box 30424, Lincoln, NE 68503. Guide to programs including a category of educational software. Educational list is subdivided into programs for various microcomputer systems. Programs are briefly described.

The Software Finder. Dresden Associates and Technical Education Research Centers, Inc., 8 Eliot Street, Cambridge, MA 02138. Comprehensive guide to educational software arranged by subject and indexed by machine and title. Description gives type of program, grade level, system, language and minimum hardware requirements, as well as price, brief annotation, and review citations.

Swift's Educational Software Directory. Sterling Swift Publishing Company, 7901 South H-35 Austin, TX 78744. Contains information on educational software publishers; provides contact information.

VanLoves Educators Handbook and Software Directory for Microcomputers. Vital Information, Inc., 7899 Mastin Drive, Overland Park, KS 66204. Indexes by subject and grade level educational and administrative software. Directory has overall ratings of 1 to 3 stars assigned by the editors. Articles and bibliographies on special topics are also included.

APPLE

The Apple Software Directory. Volume 3, Education. WIDL Video, 5245 West Diversey Avenue, Chicago, IL 60639. Lists educational software available from over 400 vendors, described briefly and cross-referenced by subject matter. WIDL Video also publishes an Apple resource directory listing hardware, boards, and accessories; and the Apple II Blue Book, which includes all of this information in one volume.

Starbeks Software Directory. 11990 Dorsett Road, St. Louis, MO 63043. Describes over 1,000 programs for the Apple microcomputer. Educational software is divided by subject and includes programs for Grades K-12.

Swift's Educational Software Directory. Apple II Edition. Sterling Swift Publishing Co., 7901 South IH-35, Austin, TX 78744. 1983-84 edition has software annotations arranged by subject and indexed by title and publisher.

BEST COPY AVAILABLE

TRS 80

Radio Shack TRS-80 Educational Software Sourcebook. From Radio Shack stores. Describes programs for all TRS-80 microcomputers. Descriptions are grouped in eleven subject categories and are indexed by title, user level, and instructional technique.

APPLE AND ATARI

MECC Instructional Computing Catalog. Minnesota Educational Computing Consortium, 2520 Broadway Drive, St. Paul, MN 55113. Catalog of Apple and Atari software MECC has developed. Programs are available through MECC.

ATARI

Atari Program Exchange. Atari, Inc. P.O. Box 427, Sunnyvale, CA 94086. Large catalog of user-developed software for Atari.

IBM

IBM Software/Hardware Directory. Sapana Micro Software, Pittsburg, KS 66762. Published twice a year with monthly updates. Includes software product listings and indexes product reviews and book reviews of IBM products in most magazines.

COMMODORE

Commodore Software Encyclopedia. Commodore Business Machines, System Marketing Group, 681 Moore Road, King of Prussia, PA 19406. Software for the PET 2000, 4000, 8000 and PET/CBM systems listed for seven areas, including education.

HEWLETT-PACKARD

Hewlett-Packard Series 80 Software Catalog. Series 80 Users' Library, 1010 N.E. Circle Road, Corvallis, OR 97330. Reference guide to programs available for Series 80 Personal Computers including both contributed programs and applications packages

TEXAS INSTRUMENTS

Texas Instruments Home Computer Program Library, 1982. From Texas Instruments dealers. Lists programs created by Texas Instruments and by independent software publishers. Categories include a section of educational listings.

BEST COPY AVAILABLE

Periodicals

Over 200 periodicals now exist in the computer industry. The following is an extensive list of those journals and newsletters that are useful to educators.

AEDS Bulletin
Association for Educational
Data Systems
1201 16th Street, N.W.
Washington, D.C. 20036
Quarterly

AEDS Monitor
Association for Educational
Data Systems
1201 16th Street, N.W.
Washington, D.C. 20036
Bi-monthly

The Apple Journal of
Courseware Review
Apple Computer, Inc.
10260 Bandley Drive
Cupertino, CA 95014
Published irregularly

Arithmetic Teacher
National Council of
Teachers of Mathematics
1906 Association Drive
Reston, VA 22091
9 issues/year
(Includes membership)

Atari Computer Enthusiasts
3662 Vine Maple Drive
Eugene, OR 94705
10 issues/year

BYTE
70 Main Street
Peterborough, NH 03458
Monthly

Calculators/Computers
Dymax
P.O. Box 27
Menlo Park, CA 94025
7 issues/year

Chicatrug News (TRS-80)
Chicago TRS-80 Users Group
203 N. Wabash, Room 1510
Chicago, IL 60601
10 issues/year

Chime
Clearinghouse of
Information on Micro-
computers in Education
Oklahoma State University
108 Gundersen
Stillwater, OK 74708
Monthly

CUE Newsletter
Computer-Using Educators
Box 18547
San Jose, CA 95158
Bi-monthly
(Includes membership)

Classroom Computer News
Intentional Educations, Inc.
341 Mt. Auburn Street
Watertown, MA 02172
Bi-monthly

Closing the Gap
(Computers and the
handicapped)
Rt. 2, Box 68
Henderson, MN 56044
Bi monthly

153

BEST COPY AVAILABLE

Compute!
Box 5406
Greensboro, NC 27403
Monthly

Computer
IEEE Computer Society
10662 Las Vaqueras Circle
Los Alamitos, CA 90720

Computer Graphics World
714 Stockton
San Francisco, CA 94133
Monthly

Computer Update
Boston Computer Society
3 Center Plaza
Boston, MA 02108
Bi-monthly

Computers and Education
Pergamon Press, Inc.
Maxwell House, Fairview Park
Elmsford, NY 10523
Quarterly

Computers and Electronics
(formerly Popular
Electronics)
Ziff-Davis Publishing Co
One Park Avenue
New York, NY 10016
Monthly

Computers and People
Berkeley Enterprises
815 Washington Street
Newtonville, MA 02160
Bi-monthly

Computers and the Humanities
Pergamon Press, Inc.
Maxwell House, Fairview Park
Elmsford, NY 10523

Bi-monthly

The Computing Teacher
Department of Computer and
Information Science
University of Oregon
Eugene, OR 97403
9 issues/year
(Includes membership in
International Council
for Computers in
Education)

Creative Computing
Box 789-M
Morristown, NJ 07690
Monthly

Dr. Dobb's Journal
People's Computer Co.
Box E
1263 El Camino Real
Menlo Park, CA 94025
10 issues/year

EDU
Educational Products
Group
Digital Equipment Corp.
Mail Slot MR03-2/E11
2 Iron Way, P.O. Box 1003
Marlboro, MA 01752

ETA Newsletter
Pouch F
Juneau, AK 99811
10 issues/year

Educational Computer
Magazine
Box 535
Cupertino, CA 95015
Bi-monthly

Educational Technology
140 Sylvan Avenue

155

Englewood Cliffs, NJ 07632
Monthly

BEST COPY AVAILABLE

80 Micro
Wayne Green Publications
80 Pine Street
Peterborough, NH 03458
Monthly

80 U.S.
3838 South Warner St.
Tacoma, WA 98409
Monthly

Electronic Education
1311 Executive Center Dr.
Suite 220
Tallahassee, FL 32301
10 issues/year

Electronic Learning
902 Sylvan Avenue
Englewood Cliffs, NJ 07632
Monthly

Hands On! Newsletter
Technical Education
Research Centers, Inc.
1696 Massachusetts Avenue
Cambridge, MA 02138

High Technology
Technology Publishing Co.
38 Commercial Wharf
Boston, MA 02110

InfoWorld
375 Cochituate Road
Box 880
Framingham, MA 01701
Weekly, 51 issues/year

Instructional Innovator
Association of Educational

Communications/Technology
1126 16th Street, N.W.
Washington, D.C. 20036
8 issues/year

Instructor
7 Bank Street
Dansville, NY 14437
9 issues/year

Interface: the Computer Education Quarterly
Mitchell Publishing Co.
915 River Street
Santa Cruz, CA 95060
Quarterly

Interface Age
16704 Marquardt Ave.
Cerritos, CA 90701
Monthly

Journal of Computer-Based Instruction
ADCIS International Headquarters
Computer Center
Western Washington University
Bellingham, WA 98225
Quarterly

Journal of Computers in Mathematics and Science Teaching
P.O. Box 4455
Austin, TX 78765
Quarterly

Journal of Educational Technology Systems
Baywood Publishing Co.
120 Marine Street
Box D
Farmingdale, NY 11735
Quarterly

BEST COPY AVAILABLE

Kilobaud Microcomputing
See Microcomputing

The Logo and Educational Computing Journal
1320 Stony Brook Rd.
Suite 219
Stony Brook, NY 11790

MACUL Journal
Michigan Association for Computer Users in Learning
Wayne County, ISD
33500 Van Born Road
Wayne, MI 48184
3 issues/year
(Includes membership. No longer includes reviews)

Mathematics Teacher
National Council of Teachers of Mathematics
1906 Association Drive
Reston, VA 22091
9 issues/year
(Includes membership)

Media and Methods
1511 Walnut Street
Philadelphia, PA 19102
Monthly

Microcomputers in Education
Queue
5 Chapel Hill Drive
Fairfield, CT 06432
Monthly

Microcomputing
Wayne Green Publications
80 Pine Street
Peterborough, NH 03458
Monthly

MicroInk

[illegible]
34 Chelmsford St.
P. O. Box 6502
Chelmsford, MA 01824
Monthly

Micro Media Review
Box 425
Ridgefield, CT 06877
10 issues/year

Midnite Software Gazette
Central Illinois PET
Users Group
635 Maple Court
Mt. Zion, IL 62549
Bi-monthly

Nibble
Box 325
Lincoln, MA 01773
8 issues/year

Output
Technical Publishing
666 Fifth Avenue
New York, NY 10019

Peelings II
P. O. Box 188
Las Cruces, NM 88004
9 issues/year

Personal Computer Age
10057 Commercial Avenue
Tujunga, CA 91042
Monthly

Personal Computing
50 Essex Street
Rochelle Park, NJ 07662
Monthly

Popular Computing
70 Main Street
Peterborough, NH 03458

Monthly

The Rainbow
5803 Timber Ridge Drive
Prospect, KY 40059
Monthly

SIGCUE Bulletin
Association for Computing Machinery
Special Interest Group on Computer Uses in Education
1133 Avenue of the Americas
New York, NY 10036
Quarterly

School Microcomputer Bulletin
Learning Publications, Inc.
Box 1326
Holmes Beach, FL 33509

Simulation and Games
275 South Beverly Drive
Beverly Hills, CA 90212
Quarterly

Softside
Box 68
Milford, NH 03055
Monthly

Softalk Magazine
11021 Magnolia Boulevard
North Hollywood, CA 91601
Free for first year to new Apple owners
Monthly

Software Digest
7620 Little River Turnpike
Suite 414
Annandale, VA 22003
Weekly

BEST COPY AVAILABLE

Software Exchange
P.O. Box 55056
Valencia, CA 91355

Spreadsheet
InterCalc
Box 254
Scarsdale, NY 10583
Bi-monthly

T.H.E. Journal
Technical Horizons
in Education
P.O. Box 992
Acton, MA 01720
8 issues/year

TRS-80 Users Journal
P.O. Box 7112
Tacoma, WA 98407
Bi-monthly

Teaching and Computers
Scholastic, Inc.
902 Sylvan Avenue
Englewood Cliffs, NJ 07632
8 issues/year

Users: The MECC
Instructional Computing
Newsletter
2520 North Broadway Drive
St. Paul, MN 55113
Monthly

Window
469 Pleasant Street
Watertown, MA 02172
5 issues/year

Indexed Periodicals

The following is a list of journals and newsletters whose software reviews have been indexed by the Digest of Software Reviews: Education (DSR), The Software Finder (SF), and the databases Microcomputer Index (MI) on DIALOG and DISC on Bibliographic Retrieval Service. The depth of indexing varies for each source as does the period of coverage. The addresses for many of these periodicals may be found by consulting the appropriate sections of the appendix or by checking DRS, SF, MI, or DISC themselves.

The Digest of Software Reviews: Education. c/o School and Home Courseware, Inc. 1341 Bulldog Lane, Suite C, Fresno, CA 93710.

The Software Finder. Technical Education Research Centers, 8 Eliot Street, Cambridge, MA 02138.

Bibliographic Retrieval Services. Education Service Group, 1200 Route 7, Latham, NY 12110. (518) 783-1161

DIALOG Information Services. 3460 Hillview Avenue, Palo Alto, CA 94304. (800) 227-1927.

Access: Microcomputers in Libraries - DISC, MI
ANTIC - MI
The Apple Journal of Courseware Review - DSR
Apple Orchard - MI
Arithmetic Techer - DRS
Atari Computer Enthusiasts - DRS
The Book Report - DSR, SF
Booklist - DRS
Business Week - MI (partial coverage)
BYTE - DISC, DSR, MI, SF
Boston Computer Update - SF
California Library Media Consortium - DSR
Call A.P.P.L.E. - MI
CHIME - DSR
CIMSE - SF
Chicatrug News (TRS) - DSR
CUE Newsletter - DSR, SF
Classroom Computer News - DSR, MI, SF
Closing the Gap - DSR, SF
Compute! - DISC, DSR, MI, SF
The Computing Teacher - DSR, MI, SF
Creative Computing - DISC, DRS, MI, SF

Courseware Report Card - DSR, SF
Curriculum Produce Review - SF
Curriculum Review - DSR, SF
DATAMATION - DISC
DataBase - DISC
Desktop Computing - DISC, MT
Dr. Dobb's Journal - DISC, MT
Educational Computer - DSR, MT, SF
Educational Computing - SF
Educational Technology - DSR, MT (partial coverage), SF
80 Microcomputing - DSR, MT, SF
80 - U.S. Journal - DSR, MT (partial coverage)
Electronic Education - DSR, SF
Electronic Learning - DSR, MT, SF
EPIE Report - DSR
EPIE Report #98/99 SF
EPIE Micro-Courseware PRO/Files - DRS
ETA Newsletter - DSR
FECR News - SF
InfoAge - SF
InfoWorld - DSR, MT, SF
Instructional Innovator - MT (partial coverage)
Instructor - DSR
Interface Age - DISC, MT, SF
JEM Research - DSR, SF
Journal of Computers in Mathematics and Science Teaching - DSR, MT, SF
The Journal of Courseware Review - DSR, SF
Journal of Learning Disabilities - DSR
Kilobaud Microcomputing - SF (see Microcomputing)
Learning - DSR
Mathematics Teacher - DSR, SF
Media and Methods - DSR, MT (partial coverage), SF
(MECC) Users Newsletter - DSR, SF
Micro: The 6502/6809 Journal MT, SF
The Micro - DISC
Microcomputing - DISC, DSR, MT
Microcomputers in Education - DSR
Micro-Scope - DSR
Monitor - DSR
Micro Media Review - DSR
MicroSIFT Reviews - DSR, SF
MACUL Journal - DRS, SF
Midnite Software Gazette - DRS
Microsystems - MT
MUMPS Users Group Quarterly - MT (partial coverage)
Nibble - DSR, MT
Nation's Business - MT (partial coverage)
onComputing - see Popular Computing
Online - DISC
Online Review - DISC
Personal Computer Age - DISC, DSR
PC: The Independent Guide to the IBM Personal Computer - DISC, MT
Peelings II - DSR, MT

Personal Computing - DISC, DSR, MI, SF
The Physics Teacher - SF
Popular Computing - DISC, DSR, MI, SF
Purser's Atari Magazine - DSR (no longer published)
Purser's Magazine - SF (no longer published)
Popular Electronics - MI (partial coverage)
The Rainbow - DSR
Radio-Electronics - MI (partial coverage)
The Reading Teacher - DSR
The S-Eighty - SF
Simulation and Games - SF
School Microware Reviews - DSR, itself (no longer published)
S.E.C.T.O.R. Project - DRS
Softside - DSR, MI
Software Review - DSR
Softalk - DSR, MI (partial coverage)
School Library Journal - DSR
Small Business Computers - MI
Sync - MI
Teaching and Computers - DSR
T.H.E. Journal - MI

BEST COPY AVAILABLE

Distributors of Mathematics and Science Software

Biology Media. 918 Parker Street, Berkeley, CA 94710 (415) 549-3210. Distributes self-instructional modules, over 60 pieces of microcomputer courseware, and other media-based products for biology education.

Cambridge Development Laboratory. 36 Pleasant Street, Watertown, MA 02172 (617) 890-8076. Develops as well as distributes simulations, tutorials, and lab tools for math and science curricula. CDL researches areas of hardware and software compatibility and develops products not likely to be undertaken by other vendors.

Creative Publications. 3977 East Bayshore Rd., P.O. Box 10328, Palo Alto, CA 94303; (415) 968-3977. Approximately 30 software programs on most topics are distributed as well as books, accessories, graphics, and manipulatives.

Educational Materials and Equipment Co. P.O. Box 17, Pelham, NY 10803. (914) 576-1121. EME specializes in science software and other media and actively seeks to commercially publish new science programs, especially in the areas of physics, chemistry, energy, and environmental studies.

Educational Software Consultants, Inc. P.O. Box 30862, Orlando, FL 32862. (305) 851-5119. Distributes courseware for Apple, Atari, TRS-80, PET, and TI but does not identify initial developers. Includes foreign languages and business subjects.

Gamco Industries, Inc. Box 1911C45, Big Spring, TX 79720-024. (915) 267-6327. Over 400 K-12 programs for the Apple, PET, Commodore 64, Atari, Vic-20, and TRS-80 are available as well as books, accesories, and audio-visual materials. Only MECC-developed programs are identified.

J.L. Hammet Co., Inc.. Microcomputer Division, Hammett Place, Box 545, Braintree, MA 02184. In Massachusetts, (800) 972-5056; outside Massachusetts, (800) 225-5467. Also (617) 848-1000. Lists over 250 software programs in all categories for Apple, TRS-80, CBM/PET. Also features books and supplies for the computerized classroom.

Instant Software. Wayne Green Inc., Peterborough, NH 03458 (800) 343-0728 Some educational programs are listed as well as those for business and game applications.

K-12 Micromedia. P.O. Box 17, Valley Cottage, NY 10989 (201) 391-7555. Distributes the programs of over 100 software publishers with over 60 math and approximately 20 science programs listed.

Marck. 280 Linden Avenue, Branford, CT 06405 The catalog lists only sample-tested products covering a range of high school and college topics with original publisher clearly identified. Products sold are for Apple II, Atari, PET and TRS-80 computers.

Opportunities for Learning, Inc.. 8950 Lurline Ave., Dept. I44, Chatsworth, CA 91311. (213) 341-2535. Over 100 math and science programs are distributed as well as courseware for other topics including administrative uses.

Queue, Inc.. 5 Chapel Hill Drive, Fairfield, CT 06437 (203) 335-0908. Also (800) 232-2224. Features high school and college-level software arranged by subjects including life skills. The detailed annotations identify developer/proprietary vendor of software titles.

Scholastic, Inc.. 904 Sylvan Avenue, Englewood Cliffs, NJ 07632 The 1982/83 catalog lists over 350 programs in math, science, language arts, social studies, foreign languages, music, and computer literacy for the Apple, PET, TRS-80, TI 99/4A and Atari.

Sterling Swift Publishing Company 1600 Fortview Road, Austin, TX 78704. (512) 444-7570. Sterling Swift has learning management courseware in elementary math as well as chemical engineering and Spanish language programs.

Sunburst Communications. Room VT16, 39 Washington Ave., Pleasantville, NY 10570. (800) 431-1934. For New York state or Canada, call collect (914) 769-5030. Sunburst has developed programs as well as distributing those produced by other vendors. It claims to have the largest selection of courseware developed by Minnesota Educational Computing Consortium (MECC).

T.H.E.S.I.S.. P.O. Box 147, Garden City, MI 48135. (800)

354-0550. Distributes Apple and Atari programs in several topics including math and science. Speech synthesizers may also be purchased

<u>Texas Instruments</u> Local TI dealer. Applications Program hotline: (800) 858-4565. In Texas: (800) 692-4279. The catalog lists programs developed by independent software producers as well as those programs developed by TI with full name, address and phone number of those producers provided.