Aspects of microcomputers in mathematical education

This item was submitted to Loughborough University’s Institutional Repository by the/an author.

Additional Information:

- Master’s Dissertation submitted in partial fulfilment of the requirements for the award of the degree of M.Sc. in Mathematical Education of the Loughborough University of Technology.

Metadata Record: https://dspace.lboro.ac.uk/2134/10398

Publisher: © Keith Curry

Please cite the published version.
Aspects of Microcomputers in Mathematical Education.

by

Keith Curry

A Master's Dissertation submitted in partial fulfilment of the requirements for the award of the degree of MSc in Mathematical Education of Loughborough University of Technology, January 1988.

Supervisor: Dr. D. R. Green

by Keith Curry 1988
Acknowledgements:

I am grateful to the following for their assistance -
Dr. David Green for his confidence in me and many helpful suggestions,
Peter Armstrong for all the time and effort he has devoted to organising the course,
and my wife Gill for her forbearance.

I declare that this dissertation is entirely my own work.
Abstract

This work covers a number of aspects of the use of computers in Mathematics. They range from uses of software through programming for pupils to practical considerations for the classroom.

Chapter One looks briefly at the aims of education in general and Mathematics in particular. There is further consideration given to how computers can help with especial reference to the aims of HMI. Chapter Two poses arguments in favour of teaching programming skills.

In the third chapter there is a brief discussion about which language would be suitable for teaching purposes. Having suggested Logo, there is a brief history and description of this language before considering some criticisms. The final part of chapter three deals with the practicalities of use in the classroom and chapter four looks more generally at the uses of software in the classroom.

The final chapter considers the way ahead for computers in the classroom. There is especial reference to financial and temporal needs. The two appendices give more details regarding the personal experiences used in chapter three in particular.
Contents:

Chapter One: What is Mathematical Education and how can computers assist.

Chapter Two: Arguments in favour of teaching programming skills.

Chapter Three: Which language to teach and Details about Logo.

Chapter Four: Teachers practical considerations with software.

Chapter Five: The way ahead.

Bibliography.

Appendix 1: Details about Menzies High School and its' organisation.

Appendix 2: Details of Classes introduced to Logo and other software.
Chapter One

Before considering the uses of computers in mathematical education, we should consider, at least briefly, the purposes of education in general and mathematics in particular.

Gabriel Chanan (1977) argues that in a democratic society our aim should be to educate for democracy, although this does seem a little too broad in its intentions. Stenhouse (1975) declares that "teachers must deal in public knowledge", while Musgrave (1985) claims that education must involve preservation of culture, personal development - including powers of logic, aesthetic appreciation and preparation for employment or leisure. Pring (1978) refers to pupils having a "survival-kit" on leaving school. All of these ideas should be taken into consideration when designing a school curriculum. In fact the present one and the new National Curriculum seem to owe more to Hirst's (1970) seven forms of knowledge rather than any educational ideals. Although much consideration is needed to clarify our intentions in education, we must also have a pragmatic view of the circumstances. Mathematics is laid down as an essential area to be covered but why?

It can be argued that education in school is a preparation for life afterward, mainly in employment (Jenkins and Shipman 1978). Many pupils might feel that the topics taught in mathematics are biased towards the requirements...
of higher education, in many respects they are correct. Because we only select a small fraction of mathematical topics, the content is less important than the necessity for children to develop analytical and intuitive skills. The following are some reasons for teaching mathematics:

1) To pass on the knowledge possessed by society. There is such a large amount here that only a selection may be chosen - hence the many different syllabi for subjects such as history and geography and the differences between traditional and modern mathematics syllabi. Hirst in particular divides knowledge into seven different areas including mathematics and formal logic. He argues that it is important to appreciate the differences between his categories as well as to develop the thought processes involved in each area. Pring argues that mathematics has "social utility," e.g. understanding number is important for anyone living in our society.

2) Gabriel Chanan argues that we must aim to educate for democracy. To further this idea, pupils must understand the principles and different types of voting mechanisms as well as the statistical information produced by the news media and political parties. Students must learn to be critical of data, to be aware of what it does mean as well as what it is intended to mean. Statistical work on data analysis becomes an important area of the curriculum, and computers can be used for "number-crunching" as well as data presentation.

3) Cockcroft includes in a list of foundation topics the
ability to count, tell the time, to be able to use money and timetables as well as to estimate and approximate. It is debatable whether one day can be passed without having to make use of some if not all of these ideas.

4) Mathematics is used as a service subject. Many subject disciplines require varying levels of mathematical awareness. These range from the closely related physics to more intuitive uses in economics and straightforward uses such as measurement in CDT. 5) When people apply for jobs, many firms require qualifications in mathematics as an essential. Employers regard the ability to pass a particular exam as indicating suitable intelligence. Thus they, and parents, expect mathematics to be taught to provide a qualified and flexible workforce. 6) Our technology depends upon the basic use of mathematics. Whether we consider how firms can make a profit and thus how much to pay their employees or what pieces of machinery are used or made, development work is undertaken by suitably qualified people who know enough mathematics for their particular post and have sufficient confidence to use their knowledge. Therefore mathematics should be taught to produce the next generation of scientists, technicians and teachers etc.

This list of six reasons is by no means comprehensive, but it does contain some of the most important reasons for teaching mathematics. Computer technology has developed rapidly and now it is a vital component of research methodology, as well as in everyday life. Familiarity
with the use of computers will help every pupil while in school for educational purposes and once they have left.

Cockcroft and others divide mathematical understanding into separate categories. He uses four: Facts, Skills, Conceptual Frameworks and General Strategies. Although they may be linked in various ways they have separate areas of meaning. For example facts include conventions of notation, e.g. 56 represents 5 tens and 6 units, and pieces of arbitrary information such as eight pints make one gallon - a fact required in a recent Proficiency in Arithmetic mental paper from WMEB. Skills include making use of number facts, while conceptual structures include the recipes for using skills. General strategies assist the choice of skills or knowledge to be used. Therefore, how can we use computers to aid the mathematical development of our pupils?

The mathematical requirements of an adult are many and varied according to Cockcroft. There are hardly any pieces of mathematics which everyone uses, and this makes developing a syllabus problematic. Mathematical ideas are used frequently in other areas of the curriculum both at school and in further education - so some topics must be taught to cover these possibilities. Teachers need to develop the mathematical skills and understanding of their pupils in order to prepare them for adult life. Since many employers prefer to train their workforce in situ, the content of mathematics is less important than the development of analytical abilities i.e. flexibility of
mind is required.

Fitzgerald (1986), in a study to investigate the uses of technology in employment, found that systems tended to dominate, but that some familiarity and confidence with keyboards would be beneficial. In general programming languages were not required by employers, although an ability to adapt packages was an advantage for some users. The use of spreadsheets and graphics packages were also amongst the more common uses of technology. The broad base of technology means that the mathematical abilities required will be correspondingly wide. Some skills such as arithmetic ability were found to be less in use, others such as algebra may be required in the adaptation of packages. There was found to be less need for substitution and the act of visually looking up references in files. Computer filing systems are a source of concern to many people, hence the Data Protection Act, but for some purposes, such as medical records and cross-referencing, the computer can cut down on the time factor involved considerably. Fitzgerald found that the meaning of numerical data had become more important and, although this tended to be heavily context-based, noting, observing and analysing trends. The age of employees seemed to have little effect on their enthusiasm for new technology. In his summary Fitzgerald noted that it created motivation for some employees whilst also boosting their confidence. This was partly because there was less emphasis on traditional pen and paper methods of solution and therefore workers were encouraged and allowed some
element of free will. Companies preferred to have flexible and adaptable employees who were capable of learning new methods of operation.

If the use of computers in the world of employment can increase motivation and boost morale, can we also use computers in education to obtain a similar effect?

In Mathematics 11-16 (1984) HMI list some aims of mathematics teaching. They claim that these aims are essential but not necessarily the only ones to be considered. Although they do not always specify the use of computers, many of the intentions can be achieved by their use. I would like to consider each of these aims separately.

1) "Mathematics as an essential element of communication." Many teachers will agree that pupils soon lose interest in their work if it appears boring and repetitive. Doing pages of arithmetic or algebra soon depresses the interest of even the most enthusiastic of children. If the context of the work is put into perspective, then pupils are more likely to be able to observe relationships and interpret results. Mathematics may be used to "...describe, illustrate, interpret and to explain" (HMI 1985), and learners need to see a reason for the symbolic notation used according to Wright (1987) He goes on to say that since a computer is "neutral" it can remove some of the inhibitions in thinking aloud. Furthermore the speed of
feedback can encourage the user to make predictions. As HMI put it "...the main reason for teaching mathematics is its importance in the analysis and communication of ideas....the mere manipulation of numerical or algebraic symbols is of secondary importance." Ideas do not solely belong to mathematics and a computer can be used to develop concepts in other areas given the confidence, knowledge and ability of the user.

2) "Mathematics as a powerful tool." HMI claim that skills such as measuring, constructing and arithmetic are not necessarily important aim in themselves. They only become vital when involved in meaningful activities. Cockcroft suggests that much work in mathematics lessons lacks context for pupils. Wright refers to spreadsheets, databases, graphics packages as being essentially normal tools to be used in mathematics lessons.

If we accept that mathematical ideas occur in many circumstances, pupils require some familiarity with the skills needed to analyse, interpret and solve problems. The computer with its storage capacity can make use of data normally too tedious to be coped with easily. For example looking at population trends in both geography and history becomes more straightforward.

3) "Appreciation of relationships." Wright claims that the computer can take care of the "donkey-work" and leave the student free to explore a conceptual framework. HMI believe that "...in very simple terms mathematics is about
relationships." By working with Logo for instance, a student can produce startling patterns with ease and yet also be required to understand rotations, angle, shapes and their various links. HMI are concerned that pupils fail to comprehend relationships because there is an over-emphasis on learning the details, such as expanding pairs of brackets, and yet such details are comparatively unimportant in the analysis of problems.

4) "Awareness of the fascination of mathematics." I am not convinced that most pupils could find mathematics fascinating, although I do agree that it is a worthwhile objective. The many books of puzzles available in bookstores give an indication of the interest it is possible to generate. Pupils do enjoy using a computer, arcade-type games apart, but we must beware of losing their interest by too much emphasis on drill and practice routines and these hardly employ the facilities of a modern microcomputer. Wright believes that playing some games can aid in the development of mathematical concepts, whilst maintaining the children's interest. Do children become fascinated by the mathematics, the computer or a combination of the two? Does it matter so long as the child is benefitting from the experience?

5) "Imagination, initiative and flexibility in mathematics." Many children in the past have been taught a "recipe" for solving problems. All the data given in the problem will usually be employed, and to obtain the one correct solution the question may be couched in abstruse
language. It is hardly surprising that children find it difficult to acquire the quoted characteristics. By allowing children to use open-ended software and encouraging them to direct the machine, we can help them to take the initiative in developing their mathematical awareness. This can lead to pupils programming computers themselves and the results will depend on the imagination of the user. Employers require flexibility of mind in their workforce and by providing children with a varied environment, including computers, we can provide the opportunity to participate in their own learning program.

6) "Working in a systematic way." To aid pupils in their efforts to work systematically, it is too easy to present them with ready-made algorithms and methods. This opposes what the previous aim is all about. In their attempts to solve problems children may need to adopt different modes of thinking. For instance the IF(condition) THEN(action) type may be applied in computer programming, but not exclusively, whilst a more freewheeling approach could employ the use of suitable computer packages. Although using software or programming are possible approaches to problem-solving, they are not the only ones and not necessarily suited to every child. Reasons for this include lack of suitable packages and operating systems, as well as the difficulty in teaching young learners how to program - in a later chapter I discuss the use of Logo as a programming language for young learners and low-attainers. I feel that it is important that we create situations where children can gain experience of computer
usage in order to decide for themselves how best they solve problems. The selection of computer packages to achieve these ends is crucial, as is the programming language to be taught.

7) "Working co-operatively." I have observed that my pupils discuss their mathematical (and other) work frequently. A number of classes seem well capable of not only dividing themselves into groups but sharing out the work required in a mature way. Even when I have a group using our network, so that they can have a computer each, they rarely stay by themselves for any length of time. They move around, chat about one another's programs and frequently sit down in twos and threes to work on problems. Wright suggests that there is considerable interaction with small groups using a computer. He goes on to say that a whole class may co-operate on solving a problem with the teacher merely acting as keyboard operator. I worry about such groupings, it is too easy for shy, quiet children to be overlooked in such large groups. Although the possibility of being able to try different ideas produces significant gains in motivation, some children prefer and need to work in smaller groups. The teacher's role in this is crucial. It is essential that he knows how each child works while still providing the opportunity for gaining experience in disliked groupings.

8) "In-depth study of mathematics." I referred earlier to Cockcroft's comment that mathematics lessons tend to be
about nothing. HMI aver that to most pupils mathematics appears to be "extremely fragmented." An in-depth study of some mathematical topic can help to create a link between different areas of the subject and provide an opportunity for pupils to perceive the subject as a complete unit rather than piecemeal. Software is suggested by HMI as one area to be considered. By writing their own work, pupils can gain insights into how a topic may be linked with other mathematical work as well as providing pupils with the opportunity to explain to others, via a program, how a topic may be understood. I believe that work can be developed using the computer as a database. As well as writing educational programs, pupils may also program machines to solve problems, display information they have collected and to play mathematical games. The only restrictions are availability of funds, suitable software and a usable computer language.

9) "Pupils' confidence in their mathematical abilities." Cockcoft refers to the Bath Study which was commissioned to investigate the mathematical requirements of adult life. It was discovered that many adults not only lack confidence in their mathematical ability, but in some cases avoided using it at all costs and refused to be interviewed. Mathematics is regarded as being difficult by pupils but also an essential area of the curriculum. HMI believe that, although pupils should be extended in mathematics, this should not be so far as for them to experience failure. Pupils need to believe in their own ability and to achieve success. For many the seeds of
failure are sown by an inability to do arithmetic, even though calculators are ready to hand. The use of computers can create a learning environment which can boost confidence. Pupils will make mistakes but they can acquire the confidence to learn from them. In various software packages pupils play numerical guessing games, where they are given clues such as high or low, hot or cold to indicate the whereabouts of the number to be discovered. Children very rarely guess the number first time and they proceed to adopt various strategies to locate it. The computer does not criticise, does not write comments on work and does not litter pages with red crosses! By boosting the confidence of users, computer work can enable children to go on and explore situations rather than solely seeking a single result.

Hooper and Toye (1975) believe that microcomputers can give children firsthand experience of realistic situations. Too many problems posed in mathematical textbooks have rather unreal data and situations. Kelman et al (1983) refer to the difficulties in teaching problem-solving and claim that this has tended to lead to various "recipes" which use only one method. Traditional mathematics instruction has rarely provided opportunities for children to make their own discoveries and use their own methods. Textbooks and teachers present concepts and develop specific solutions. Many children have been faced with repetitive story-type problems which bear little resemblance to reality. The use of computers in lessons allows users to explore a problem situation. The process
of arriving at a solution is more important than the eventual answer. There is not necessarily only one way to solve problems, there may be many suitable techniques, ideas and tools which can be implemented. Kelman et al believe that people who have learnt problem-solving skills will have flexible minds - one of HMI's aims. The use of computers in school can provide the opportunity for children to test their own hypotheses and solutions.

Kelly (1984), Kelman et al and Hooper & Toye all believe that since computers can be used to process data they can relieve some of the boredom that results from number-crunching problems. In the past mathematics questions have frequently resulted in "nice" answers, such as integers, and even with the availability of calculators, the style has not altered much. Hooper and Toye claim that the use of a computer can give pupils the opportunity to both explore and practise numerical approaches to problem-solving. Kelly refers to the by-passing of some mathematical limitations such as arithmetic and the provision of access to "real" problems. Using actual data not only gives children the context of the problem but it enables them to acquire what Cockcroft refers to as an "at-homeness" with numbers. There is more to numeracy than basic computation skills. Understanding when they can be used as well as estimation are other components.

Simulations of real-life can give children access to situations which would normally be prohibited. Users are
allowed to alter various factors and see the effects of such changes. When operating with such systems, children are actively encouraged to ask themselves "...what happens if...?" A facility to rerun an experiment or idea gives the user more control over his learning. Discovery learning has its detractors who believe that children find it too difficult to abstract from a mass of detail the required concepts. Kelman et al, amongst others, believe that the use of computers can speed up the learning process by focussing on specific details. A facility to rerun an experiment or idea gives the user control over his learning. Thus, by providing a learning situation of this kind, users are free to explore different avenues in their quest for solutions. The outlay for a package, which could be used by many pupils, over a long time could make economic sense especially if the experiment is difficult or expensive to set up.

Hooper and Toye warn of the inbuilt assumptions made by the programmers of simulation packages. To transfer a situation on to a computer program a programmer requires to change it into algebraic and numeric notation. In doing this he/she must omit some factors which could affect the situation but are not programmable. Kelman et al warn that children may learn the strategy for a computer simulation rather than the actual circumstances, therefore children's approaches to simulations must be noted. They believe that users need to be open-minded and say that any policy of decision-making tends to lapse into wild guessing rather than analysis. Any deductions
made have more to do with the program than the reality from which the problem originated.

A solution to this is to teach children mathematical modelling, since simulations are models themselves. But what is mathematical modelling? McClone (1976) states that modelling is a means of describing real-life situations with numbers and symbols etc., so that a better grasp of events can lead to an improved understanding. This may help us with predictions for the future, or it may be a one-off solution to a particular problem. Therefore modelling can be described as a scaling-down of reality into a form anyone can understand. It may be argued that teaching modelling skills is not necessary just to assist children's understanding of computer simulations. I believe that modelling represents an interpretation of some of the National Criteria for GCSE.

Examinations have for many years been an unfair test of children's ability in a subject. Many children who have achieved "success" at O-level Mathematics imagine they will gain corresponding success at A-level. A large number of teachers will be able to quote examples of pupils who, having obtained high grades at 16+, flounder terribly at A-level. Although there are undoubtedly many reasons for this, we must include an examination which tends to be more a test of memory and comprehension than of mathematical ability and teachers who, for many reasons both good and bad, teach rigidly to the syllabus. GCSE seeks to test children's understanding of mathematics and
its processes. Instead of five hours - less in some cases - in which to demonstrate their ability, students will now have the opportunity via practical and investigational work to show more of the qualities which make up mathematical competence.

In the National Criteria list of Assessment Objectives, 3.16 and 3.17 specifically refer to methods of examining which cannot be carried out by time-limiting examinations. Although oral responses to questions (3.16) may be evaluated by official tests of mental arithmetic, discussion of mathematical ideas and mental calculations certainly would have their place in a modelling exercise and/or computer work. Objective 3.17 is a requirement for students to "carry out practical, investigational work and undertake extended pieces of work". HMI aim 1.10 refers to this.

In the Assessment Objectives, apart from the two already mentioned, all could be applied to practical work and each is a necessity for a modelling activity. They range from the recall and application of mathematics in everyday situations to the applications of combinations of mathematical skills involved in problem-solving.

The use of computers in modelling is obvious. Not only can computers store vast quantities of data, but they can display the information graphically, calculations and allow interaction. This allows the student the time and the opportunity to explore any given situation while the
computer carries out the donkey-work.

Modelling skills, therefore, should be taught as part of the mathematical learning process. Not only to assist in achieving the aims of the National Criteria, but also in understanding the underlying assumptions behind computer simulations in mathematics and the rest of the school curriculum. A computer can help in the storage and retrieval of data in order that students may be assisted in the problem-solving process.

Before beginning to use computers in the classroom, we must also consider the possible misuses. Cockcroft refers to activities which lack purpose being taught in maths lessons. Terry (1984) believes that the use of computers can promote pupil-directed inquiry and therefore aid the development of children’s understanding of mathematical concepts. He believes that we must consider what are our educational goals and then see how a computer may be employed to assist us in achieving them. Kelly warns of activities which lack purpose e.g. using a computer to merely keep children occupied or happy. I suspect many teachers permit pupils to use a computer as a reward for good behaviour or work. This seems to overlook the gains to be achieved by a more positive approach to computer work.

The use of computers do not guarantee educational progress. Wragg (1984) has written about the current educational policy to ensure that there is a computer in
every school. He goes on to pose three questions: (i) Are teachers favourably disposed to the use of microcomputers in the classroom? (ii) Will teachers be properly trained for their use? Here we must include programming as a necessary skill. (iii) Will there be enough good software? I will go further here, how will teachers be able to find out about software to preview it and consider the financial outlay? Wragg concludes with a final question: Will anyone inquire how it (the computer) is actually used?

Summary

Since the range of mathematics topics required by school leavers is so broad, the content of a mathematics syllabus is less important than the ability of pupils to develop flexible minds. In Mathematics 11-16 HMI list ten aims which they believe are essential to maths teaching. The versatility of a computer, allied to good software, enthusiastic and suitably trained teachers and pupils with programming skills, make these aims achievable. Some writers believe that using a computer can give pupils more direct experience of real-life problems since the drudgery of doing vast quantities of arithmetic can be removed. Simulations can help to illustrate problems and then pupils can experiment and analyse factors and their effects. The problem with such packages is that programmers, out of necessity, tend to simplify reality. I believe that the skills of mathematical modelling should also be taught. Children may then understand some of the
underlying assumptions behind mathematical formulae in general and simulations in particular. Modelling is a useful skill in itself and can be considered as essential to the area of assessment under the National Criteria objectives 3.16 and 3.17. The introduction of computers must be carefully thought out. Our aims and objectives should be decided upon before we analyse in which areas and how the use of a computer can aid in the development of mathematical awareness.
Chapter Two

Hartley (1975) defines programming as "an attempt to make a computer do what you want it to do in the language it permits you to use". By teaching programming skills to pupils we give them another weapon in their problem-solving armoury and we encourage them to think logically. With reference to the previous chapter, if we are to initiate investigative and practical work, then a good environment and a variety of equipment is vital. I include here a classroom with a variety of places to work and essential equipment such as rulers, tape measures, calculators and computers i.e. a mathematics laboratory.

Software packages can produce passive children who simply react to the changing picture without any thought of the reality. To maintain the interest of children, not only in mathematical work but also computers, we must allow them some control over their learning. If schools direct children along pre-designated learning schedules as in CAL (Computer Assisted Learning — see next chapter for more details), then pupils will find computers as boring and uninteresting as mathematics often is to them now. All the work referred to in the previous chapter will require software which can stimulate children but allow them some element of free will. A graph-plotting package would be a suitable tool to employ when analysing data collected for some practical situation, but a package will not necessarily be sufficiently versatile, because of the programmer's bias, to cater for every minute detail.
Teaching programming skills can enable children to use computers to solve problems of their own devising, using their own methods.

Kelman et al (1983) argue that programming is problem-solving. It is necessary to analyse a problem, break it down into smaller pieces and then do the calculations and these are skills used in programming. Hartley claims that programming turns the analysis and solution of mathematical problems, and the logic involved, into a craft. He goes on to say that this can allow children to learn from their mistakes, provided they have the confidence, and to develop, test, and rework their theories. By altering small details of a program, children can observe any differences. We must beware of merely presenting them with program listings from magazines, books or worksheets. Because children do not "own" these programs they tend to lose any initial interest in them and this may be seen in frequent typing errors. Programming provides a reason to talk and think about problems while also noting the different styles of thinking of others. These ideas can then be "stored" for future use since they resemble something of an analytical tool-kit. This may be an interpretation of what Pring (1976) describes as pupils' survival kit.

In chapter one I referred to Lawton (1983), who comments on the arguments of Schwab, Phenix, Broudy and Hirst. They argue that knowledge can be divided into separate categories. Since computer technology at the times of
their writings was primitive and non-existent in schools, it is unlikely to have come into their spheres of thinking. Hirst includes in his seven categories mathematics and formal logic. Traditional Euclidean geometry has been taught to pupils in order to aid the development of their abilities in logical thinking. Wardle (1983) claims that this never really worked for the majority of pupils. Circle theorems, for example, do not normally occur within the experience of most children. They inevitably find difficulties in understanding the context of the problem. This produces a corresponding fall-off of interest, and further evidence for children that mathematics is a dull, boring, esoteric and difficult subject. If we teach programming then, as Hartley claims, this will aid in the development of reasoning skills, although he also says that this assumes an algorithmic approach to solving problems is valid. I agree with this point since we expect children to observe and describe patterns, for instance, in a mathematical way.

Polya (1944) divides mathematics into two areas. One he describes as "mathematics in the making .... appears as an experimental, inductive science". The other he refers to as the "rigorous science of Euclid". By using a computer, a child has control over a situation and the step-by-step approach necessary for programming requires him to think clearly and logically and present carefully written lines of program. Thus instead of dull, dry circle geometry questions which are supposed to aid in the development of logical thought, a child gains access to a more
stimulating environment. The thought processes are still inherent and a child gains in motivation and interest. Wardle claims that many children show an amazing amount of perseverance and determination in their efforts to solve a problem - far more in fact than in doing normal work. The argument of logical thinking achieves HMI's aim 1.7 (1985) and enters into one of Hirst's categories of knowledge.

According to Lawton, Schwab is less specific in curriculum terms than Hirst, in that he argues that there are only three categories of knowledge and they are: Investigative, Appreciative and Decisive. It may be argued that they constitute mathematical awareness. By programming a computer, a pupil illustrates his understanding of a particular mathematical concept. If we encourage pupils to produce programs plus documentation as part of their GCSE coursework, then they will have to demonstrate their problem-solving abilities and understanding. This however presupposes that pupils will have access to an easily-used language. Most microcomputers are programmable in Basic, which Bork (1984) describes as the "junkfood of modern computing". The next chapter discusses some of the requirements of a language for use by pupils.

Algebra is in many senses the language of mathematics. Having collected data, theorised, tested etc., we may wish to generalise our findings into an easily remembered and usable form. Children may understand the concept involved, but find the notation used difficult to grasp. Tall (1983) gives as an example knowing that \( x + y = 10, \)
where \( x \) and \( y \) may be length and width. Children find this reasonably easy to grasp. If, however, we alter the information and we have to find \( y \) knowing \( x \), this becomes a different problem altogether. If the numbers involved are "easy" - whole numbers - then success is high, but if the numbers used are decimals or fractions, the success rate drops considerably. It could be argued that this is due to lack of confidence with such fractions and decimals, and I would agree that this is true, but by being unable to "see" that if \( x + y = 10 \) then \( x = 10 - y \), children are also missing the point algebraically.

Algebra is so often dull and boring, questions are repetitive and lack any context for children, for example see Longman Mathematics 3 (1970) ch.11 in particular. Investigative work can encourage children to use letters for numbers in their efforts to generalize ideas. By learning to program children have to know how to manipulate terms in algebra. A big advantage is the sensible use of variable names instead of merely initial letters. Wardle claims that children, therefore, see an immediate use of algebraic notation and have less difficulty in its implementation.

Apart from the availability, Basic can be quite useful in aiding children in developing algebraic ideas. It is quite easy to write programs and produce numerical results. One of the drawbacks of Basic is the difficulty in producing interesting graphical effects from a limited knowledge. Using some languages it is easy to obtain
results quickly and easily and they look good. Tall, amongst others, believes that children need a play stage of learning to program where they have fun before progressing towards more applicable programs.

There are differences between written and computer algebraic notation and we must be aware of these when teaching. Many pupils have difficulties with the slanting solidus in fractions. I suspect that this is partly the fault of publishers, printers and typesetters who often decide to cut costs at the expense of pupils' confusion. In computing there is no ambiguity with \( \frac{1}{x+y} \), but is it \((\frac{1}{x})y\) or \(\frac{1}{xy}\) in written algebra? I have deliberately placed them on one line to maintain the illustration of ambiguity. We can get around the problem by encouraging the use of brackets to ensure correct order of calculation. A change in the conventions on notations may also help us to ease any confusion in the minds of pupils. Other differences include index notation and some symbols such as multiplication. A very important problem is the notation used to have a counter. For instance, in Basic and some other languages, \(A = A + 1\) means increase the value of \(A\) by one and store it in \(A\). In formal algebra this has no meaning at all. Differences in notation are comparatively minor problems which will be overcome with the extra motivation and interest of the pupils involved.
Summary

A computer is a mathematical tool. We can use commercial packages to aid investigatory work but not every situation can be taken into account by them. A facility for programming a computer would be useful for pupils to be able to carry out some tasks.

Hirst, Schwab and others divide knowledge into separate categories. In his seven areas, Hirst includes mathematics and formal logic. The ability to program - for other subjects not just mathematics - implies an ability to work and think logically. Traditional geometry was thought of in the past as aiding the development of logical thought. This idea has never really worked for the majority of pupils. Writing programs would help pupils to order their thoughts. The step-by-step approach necessary for programming requires both clarity of thinking and precise instruction. Polya refers to mathematics being divided into two areas. The first he describes as the "rigorous science of Euclid", presenting mathematics as both systematic and deductive, the second is an "experimental and an inductive science".

Algebraic work for children causes many problems. Perhaps the notation necessary, the lack of context, or the convoluted language used in written problems are causes of this. Programming a computer requires the ability to understand and handle variables. The facility for referring to variables by name will help in this area, but
we must be careful to take note of some specialised computer notation which has a different meaning when seen in conventional written algebra.

If we are to introduce programming skills into the mathematics curriculum, there will be problems. The choice of a suitable language I will discuss in the next chapter but how is this work to be fitted into an already crowded curriculum? There is a danger also that we may appear to be forcing children to use and program a computer. We would be merely providing the opportunity for using another tool while solving problems and hopefully allowing children their own choice of methods.
Chapter Three

With the current availability of high-technology and possible further developments, I believe it is essential that all pupils should learn some programming skills. A number of textbooks have introduced the idea of flow charts but they tend not to be popular with children. Perhaps the tasks lack any practical application or they are too difficult or too easy to set down in such a fashion. Programmers do not necessarily adopt conventional ways of illustrating their programs. If a child is going to produce a flow chart as an aid to solving a problem then I believe that this needs to be taken a step further and applied to a computer.

The difficulty arises as to which language should be used. Virtually all microcomputers available to schools have Basic as the standard language. In research, Maxwell (1984) tested a class of intelligent ten and eleven year-olds by teaching them a limited subset of Basic, sufficient for use in guessing games and low-resolution graphics. Only 20% of the class were able to cope with programming and when the idea of loops was introduced, this decreased considerably to only 5%. The nature of the language makes it difficult to write programs which do interesting things. Thus a pupil could easily become frustrated from lack of success. The process of coding the instructions necessary for a program was getting in the way of actual programming and therefore not supporting the whole process. Thus pupils were not able to discover
easily whether or not the idea behind the program would work. Debugging, Maxwell discovered, was another major source of frustration. In work with a group of sixteen year-olds I have given them programs to be typed into the computer. The idea then being that the students investigate what the program does. All of the pupils had some familiarity with the computers and most had a working knowledge of Basic. The sessions did not go well. Many pupils found the task of typing in a program, already written for them, to be very difficult. Errors abounded and, with the tedious way of editing the programs, frustration and loss of interest crept in.

In the conventional classroom a child rarely thinks about how a mistake occurred. The whole notion of making an error in mathematics is highly emotive. Those children used to failing in the subject, for whatever reason, will merely dismiss any further error, even on a computer, as just another one in a continuing saga. On the other hand, pupils who regularly achieve success in the classroom can get upset at even a thought of failure. The idea of debugging is crucial to the solution of problems and especially so in programming. The child's thinking is made concrete in a procedure, therefore it is vital to able to "see" a program operate, to be able to iron out any wrinkles in programming or syntax. If the faulty program produces an interesting result, the fear of failure and its accompanying frustrations can be lessened. Trying to debug in Basic is hard enough for comparative experts.
If we therefore discard Basic as a serious language for beginners, what language could be used? Comal is required for the MEG GCSE in Computer Studies, but only for one small item. In fact it is easier to convert programs into Basic for debugging than to leave them in Comal. Therefore because it would be so easy to continue in Basic, there will be less incentive to use and learn Comal. If this language is proposed for one small item why is it not required as the standard language? Perhaps if we consider languages used commercially there may be a suitable one that is versatile, easy to learn and powerful for all pupils to be taught. A requirement might be that a program can be dealt with in smaller procedures which can then be applied simply and quickly in other programs. Pascal falls into this latter category, but the initial stages of learning it are very daunting for beginners. Users need to declare variables at the beginning and this assumes that beginners are able to organise themselves in this way and understand the algebraic concepts. This is difficult for many children as APU surveys indicate. Fellgett (1987) recommends Algol 68 although this is a high level, powerful language. He claims that it is possible to begin by learning an easy language and then graduate to a more difficult and versatile language. He argues that it is time-wasting to have to learn another language after becoming proficient in one, and that users need to learn the "freedom of a new language". This can be summarised as learning programming rather than learning a language. There is some merit to this argument since
Algol is consistent and the ideal language has yet, if ever, to be developed. After considering the available languages I would recommend Logo.

Logo owes a great deal to Piaget for his research into the stages of learning. Seymour Papert worked with Piaget before moving to Massachusetts Institute of Technology (MIT). After some work on Artificial Intelligence, using the language Lisp, the language Logo was developed. Originally there was no graphics, and this facility was added as an afterthought. The idea of using graphics is probably the most common way of entry into Logo. At its simplest level Logo is an interactive graphics system where the user can type a command and immediately see any response either on the screen of a VDU or on the floor by means of a "turtle." On a screen a turtle is shown as a small triangle. It is also possible to have a mechanical device, which moves around on the floor, following instructions from the user. A simple command such as FORWARD 60 will make the turtle move sixty units forward, this applies to both screen and floor turtle. RIGHT 30 makes the turtle turn thirty degrees to the right, LEFT has a corresponding effect. Thus a user can see immediately the results of any instruction. If the system permitted only straightforward commands of this nature, then users, especially children would not find it at all interesting or useful. It is possible to define new commands by programming the computer. For instance:

TO SQUARE
FD 100 RT 90 FD 100 RT 90 FD 100 RT 90 FD 100
will produce a square of side 100 turtle units, starting wherever the turtle happens to be. Note that forward may be abbreviated to FD, although this may not be the abbreviation used in every version, and that all these instructions are given on one line although this is not essential. The REPEAT command can shorten still further the amount of typing required. The computer now only requires the command SQUARE for the program to be executed and used in more complicated programs. The language can be used as a tool for creating and editing both words and pictures. Children can type in stories and design their own illustrations to be placed anywhere in the text far more simply than using Basic and a wordprocessor. A further use of the language is in calculations, such as are required in algebraic work, and list processing. Again these facilities can be used in conjunction with those already mentioned.

The above describes what Logo is about, however briefly. It may be thought of as a collection of pedagogical principles i.e. how children learn. Papert (1980) claims that, as a child, he was able to handle gear-wheels and, because he was able to relate his mathematics work to his early experimenting, many concepts came easily to him. He has tried to introduce something of this philosophy of learning into Logo. He argues that to learn a foreign language such as French then an excellent way of doing this is to live in France for a time. Therefore to learn mathematics we require an equivalent "Mathland". Logo was
developed as a language to assist learning. Not only does it encourage good programming habits, such as dividing problems up into smaller more easily programmable pieces, but it allows users to gain access to ways of thinking and talking about mathematical ideas. In my own work with Logo many of the pupils see interesting programs developed by their classmates. They talk and take notes and adapt these programs for their own use. Being able to see the result of commands and programs, either on the screen or drawn by a floor turtle, children have something concrete on which to base their understanding and further development. From a teacher's point of view, observing children's results and the methods used, gives an indication of how they think. This is invaluable when dealing with mathematical work away from the computer.

It is possible to obtain limited versions of Logo which only contain the popular graphics. These are quite restrictive by comparison. Variables are not so easily manipulated, there is no colour and there is certainly no text facility. Logo in its full version is a way of helping children think with confidence that mistakes can easily be corrected and form an important part of the learning process. An example of this is Susan, a girl in 3G05, who has a poor attendance record and a succession of low results in mathematics since the first year. She has recently learnt to alter the colour of both background and drawing colour. The patterns produced, with the corresponding happiness she displays, convince me that Logo is working for her. Working from a textbook, she has
shown more motivation and reasoning power than at any time that I have been teaching her.

But what other reasons are there for learning Logo? Abelson (1984) claims that it makes programming accessible to young children, without them getting confused by complicated results. By simply defining a new procedure for the computer, children are developing programming skills and logic. If we interpret programming as teaching the computer, then, because Logo is both simple and powerful, it is claimed that children will not only develop programming skills, but they will also be able to develop their understanding of mathematical concepts.

There are obvious mathematical benefits to be gained from children using Logo. They can develop a feeling for number, listed by Cockcroft as a requirement for numeracy, by comparing the sides of shapes and looking at the same lengths at different angles. When I originally illustrated some Logo commands to my pupils, I started with FORWARD. But pressing Return at this stage produces the response from the computer: "NOT ENOUGH INPUTS TO FORWARD." I am not sure how to explain this error message in the context of Papert's philosophy but children seem to comprehend the idea that the computer needs to be told how far forward. Similar logic can be used when discussing other commands such as REPEAT, and variables as inputs to produce squares with any length of side.
As an initial task, I set a problem to 3G05 to discover how big the display screen is. Anthony, who is not good at arithmetic, was quickly able to move the turtle around the screen and then add up the numbers used in the forward commands. Obviously one example does not constitute a proof, but, by having the problem in context and an interest in what he was doing, Anthony was able to use his arithmetic and mental skills to solve the problem. Also involved in the same problem was the idea of estimation when he was trying to get the turtle to join its original line after it had wrapped around the screen. The next problem I set was to draw a square. Again most of the pupils were not able to recall a vital piece of data ie 90 degrees is a right-angle. By trial and error, erasing incorrect guesses they soon rediscovered the answer. Many of the children wanted to produce a circle - I would not tell them - and, with much experimenting with the commands, they completed this the first task they had set themselves. In doing so they learnt a lot about angles, polygons and scale since if the side is too long the diagram wraps around the screen and appears split up.

It has been claimed by Tall (1983) about programming in general, and Papert (1980) on Logo in particular, that the formal language of mathematics appears to have a clear purpose. Many children find algebra difficult to understand, APU results suggest that many conventions are only understood in a confused way. Learning programming skills can create the situation where a pupil learns to handle variables. For example Geoffrey in 3G05 wanted to
draw a square of any side he wished - a preliminary to another task he had set himself. The introduction of the variable name "SIDE" helped him produce the program he initially required. By using MAKE he was then able to increase the value of SIDE and produce further squares without any more input. I claim that by handling the variables and seeing the results, he is acquiring more understanding of algebraic concepts than solving many linear equations from worksheet or textbook.

To look at Logo as an aid in learning specific mathematical ideas is too restrictive. Any syllabus is merely a selection from a vast amount of mathematical ideas. The requirements of further education and industry include flexibility of thinking and logical thought (Fitzgerald 1986). By learning mathematics through Logo, children are in a dynamic learning process. A pupil requires a good base from which to learn further skills, the use of Logo drawings can act as a link with classroom work. The Chelsea Study (1980) and SESM (1984) indicated that, within their selected areas of testing, Algebra, Fractions, Ratios etc., many children adapted their own methods of solution and did not use what may be described as standard methods i.e. those taught by teachers. Logo is not only a serious programming language, Paul an A-level mathematician who markets his own software is very interested in it, but because its basic commands, known as Primitives, have simple visible effects, children have control over their own learning situation. This creates a successful learning environment with the consequence of
increased motivation. In fact users are encouraged to develop their own problem-solving styles by discussing their work with both teachers and classmates.

Many adults as well as children lack confidence in their mathematical ability, (Cockcroft 1982). Pupils require positive feedback at an early stage in order to overcome what Papert refers to as "Mathphobia." In Computer Assisted Learning (CAI) a programmer, remote from the actual user in the classroom, has broken down a situation into what he/she considers to be manageable pieces. The computer may give a quick response of right or wrong, but it cannot indicate where any error may have occurred. By acting out the behaviour of the turtle, a child has something tangible to work with. In fact working in a group, one person may be blindfolded and asked to follow instructions. The writing and modifying of programs allows users to explore not only the result but also the process by which it was achieved. By the very nature of its program structure, Logo makes it easy to debug any program. The idea of debugging according to Papert is at the very heart of mathematical thinking but children need the confidence to believe that mistakes are inevitable but that they can learn from them. Eric, in 3ST4, was observed trying to load the Smile "Next 17" programs. He did the same thing at least five times without any success, then gave up and started disturbing his neighbour. I would argue that he was making mistakes but that he was unable to analyse and learn from them. Because procedures in Logo can easily be combined, it is
comparatively straightforward to isolate any error. Since each part of a program may be tested separately. Even mistakes in programming can produce interesting results and these can be stored for future implementation in other situations.

Papert claims that Logo explicitly encourages users to develop what he refers to as "top-down" thinking, i.e. analysing a problem and then breaking it down into smaller more manageable pieces. This may be an eventual outcome but children need to develop a "toolkit" of procedures first. Weir (1986) suggests that if someone is unaware of the steps to be used in a problem-solving process, then they are likely to have difficulty in incorporating the suggestions of others, especially teachers, into that process. The acceptance of help tends to take the form of requests for specific pieces of information but not the detailed explanation that normally accompanies it! Pupils need to become familiar with the language and its uses before broadening their skills and adopting a top-down approach to the analysis of problems. Programming in Logo produces a collection of tools of thought such as hierarchical structure, procedural organisation and recursion. This is similar to the processes involved in mathematical modelling when not only has the problem to be analysed but some mathematical skills are required to produce a solution and language skills are needed to communicate the results.
Logo does have its detractors. Children do appear to be playing when working with Logo, but a good way of learning is to engage in guided messing about. Children do this naturally but are gradually brainwashed out of it by schools. Logo has the potential to continue the play and learn situation but it may be necessary to contrive the messing about to make it more appropriate. Discovery learning seeks to encourage children to be happy in what they are doing and therefore to increase the motivation to learn. Parents and teachers are naturally concerned about the academic progress of their children. Some are fearful that discovery learning will lead to sloppiness of thought and a corresponding loss of academic rigour. I disagree with this on two counts. First, by acquiring programming skills children learn to think logically and to put their ideas in order. Logo has an easy way of structuring programs, illustrating easily how a complete package is built up. Programmers need to be rigorous and Logo presents the opportunity for pupils to develop their own styles which will be flexible enough to employ on any computer in whatever language. Second, it may be possible to produce strongly academic children, but what have they learnt and how can it be applied? As Cabell (1926) puts it "Do you acquire all knowledge first and hope for understanding later?" The small sample of mathematical ideas taken for an examination syllabus is comparatively unimportant to most pupils, what is vital is learning the ability to think in a mathematical way. The idea that happy children do not progress as well as they could seems totally ridiculous. If we are enjoying what we are doing
then surely this will provide the motivation required to investigate further. This is especially true in mathematics which is frequently regarded with horror by adults and children alike. Therefore the promotion of enjoyment and the boosting of confidence which can result when using Logo provide reasons to develop its use.

In research Wellington (1986) aimed to test some of the claims made about Logo. In his survey he found that motivation was something of a problem. The pupils tested seemed to have little inclination to explore and the idea of open-ended discovery learning did not seem to exist. Many of the pupils seemed to ask "What next?" "What else are we supposed to do?" "So what?" Wellington's answer was to introduce worksheets. Although this seems to oppose the philosophy behind Logo, I am inclined to adopt whatever measures are available to me to get my pupils started. Wellington himself admits that the previous educational experiences of the pupils did not lend themselves to computer users working imaginatively. The development of GCSE and the coursework requirements of practical and investigational work could be blended with Logo programming to improve the understanding of many pupils. I will explain later how I introduced Logo to my classes. I have not used worksheets, although I have posed a starting problem, and I have found little problem with motivation. In fact the supply teacher has been taught Logo programming by various members of 3G05. Wellington also points out that a number of children did not see the point of the activity, I am reminded of
Cockroft's comment that children rarely see the point of a mathematics lesson. In the testing, five of the pupils were unable to read or write. Out of these, two were quite happy using Logo while the remainder thought it a pointless exercise. Depending on the pupils I might claim this to be a success! The three who were unhappy are likely to have had many failures in the past and could therefore have insulated themselves from success. Adrian in 5(6) is something of a problem, like many in the group his ability would normally place him in a higher set. His attitude and poor attendance have prevented this. I have shown him Logo commands as part of a detention. He enjoyed this. When we were able to use the network he was able to demonstrate, without showing off, his abilities to his cronies and this was something he enjoyed. Interest and motivation are subjective because the circumstances of environment, subject content and teacher-pupil-class relationship will vary. One of Papert's claims is that top-down problem-solving skills are developed, Wellington's findings seem to dispute this. As already indicated, before top-down thinking can be implemented, children must be sufficiently imaginative and they must have an adequate base from which to work. Referring to skills required in mathematical modelling, the ability to note similarities in what appear to be different circumstances is essential. For example consider the half-life of radioactive isotopes and the way drugs are absorbed into the body. Logo encourages the use of procedures to build up patterns but if a child does not know how to produce a square how can they be expected to
separate a house into a rectangle topped by a triangle? Trial and error methods of solution are quite acceptable and can easily be attempted using Logo. Top-down is only one way of thinking about a problem. Many adults as well as children prefer a free-wheeling, inspirational approach with some trial and error added. Logo provides a learning situation where many different styles may be attempted. Papert believes that the approaches and skills acquired using Logo will be transferred across the whole curriculum, Wellington found no evidence to support this although he does report that the teachers found their pupils to have developed more social skills and the ability to work in groups more easily. I would agree here that this is not necessarily due totally to the use of Logo.

Thorne (1987) indicates that there is conflicting evidence regarding any improvements made by children using Logo. He goes on to say that this is especially true of low-attainers. In my work I find quite the opposite. The low-attainers, in particular 3G05 who have had an extended period working with Logo, have improved considerably. Their ability to use the SMP 11-16 Green series of books has certainly developed. With other groups it is difficult to note any specific mathematical improvements, perhaps this is a problem in this type of assessment. McShane and Simon (1987) suggest that discovery learning requires guidance. They maintain it is difficult to pick out the required concepts from a huge mass of detail. An analogy here is attempting to pick out from a box of
jigsaw pieces one that will fit immediately, the more pieces that have already been placed the easier it becomes. They state that active discovery learning needs the addition of feedback and a child is not able to supply this for itself. In research they tested three sets of ten year-olds using a game to assist problem-solving. The first group were allowed unrestricted practice on the game. The second and third groups were taught a winning strategy with group three only being given the reasons behind it. The third group obtained the best results using a similar game to the test. McShane and Simon contend that a structured environment assists children's learning and they conclude that Logo should be taught. Thorne goes on to say that there were certainly signs of improvement from children, teachers reported back to say that there were gains in confidence, language development and ability to co-operate in group work.

I hesitate to criticise the findings of researchers, but how can the findings of Wellington be compared with those of Hughes, Macleod and Potts (1987) who report many successes with socially disadvantaged children in Scotland. Weir in America has also had many successes using Logo with autistic children and disturbed children. There are different ways of solving problems and children will inevitably find their own methods.

Logo may not achieve the claims of Papert when used in the classroom. There are likely to be varying circumstances, in my own case the irregularity of access to the
computers. Logo can provide a learning situation for children so that they can develop more confidence when faced with mathematical ideas. Certain concepts will inevitably be used such as angles and numbers. Observation of the resulting patterns can provide the opportunity and encouragement for noticing similar things both inside and outside the mathematics lesson.

Blamires (1987) argues that Logo could be discarded because many of its facilities are slow and unwieldy in comparison to specialised software. His analogy is that of a dinosaur. Because the technology has been developed comparatively recently, databases, spreadsheets and wordprocessors etc. are more easily accessible. There is some validity in this argument but, by making programming skills easy to acquire, Logo makes the computer a more dynamic part of the learning process and it can still be used alongside the faster more efficient specialised hardware and software. In fact there is work being done at MIT to produce an updated version of Logo called Boxer which uses some of these newer ideas.

Mathematics is often regarded in isolation from other subjects in schools, yet it is a service subject. With technology work going on in schools, Logo can be used to program the movement of machines and gadgets. Developing the idea of mathematics across the curriculum gives children the opportunity to see the subject from the outside as well as the inside and this is one of Hirst’s ideas in his forms of knowledge argument.
I first became aware of Logo about five years ago after seeing a TV documentary and hearing my wife enthuse over Papert's writing in "Mindstorms". Until recently, however, I have not had access to the language either for myself or any pupils. The member of staff in charge of computing has acquired it for me and he has made it available on both stand alone computers and the network. There is a problem - we have no documentation and, like different versions of Basic, there are slight variations from one implementation to another.

My department are fortunate enough to have a supply teacher based with us, and, while he teaches some of my classes, I am able to withdraw some of the pupils for small group work on two computers. I opted to select groups of three pupils. They were chosen so that there was already a working relationship and an even spread of boys and girls. The initial session went well although I felt that three at this stage was too many. The pupils concerned now tend to spend most of their lunchtimes programming in both Logo and Basic. From the same class I selected only four pupils for the next session, some of them caused problems in the classroom. They thoroughly enjoyed using Logo and were able to produce a program that generated a very effective circle within forty minutes of experimenting. Since then I have been able to negotiate a weekly session on the network of computers for the whole group. This seems to have both advantages and disadvantages. I have only given very basic graphics
commands viz: FORWARD, BACK, LEFT, RIGHT and an initial starting problem of drawing first a square then a rectangle. The children have learnt very much by themselves, I have shown them some abbreviations (FD, BK etc.) also the REPEAT command and how to program and edit most of this work has come at the request of the children. Recently to vary the work we have looked at colour. I have introduced Logo to three other classes 2ST1, 3ST4 and 5(6) and, although they have not had so much opportunity, they certainly show a great deal of enthusiasm.

Important questions need to be considered at this stage, such as "What are the children getting out of the work"? "Are they missing out on learning in mathematics which will harm their examination prospects"? If we are to believe the claims made about the transference of skills to other areas of the curriculum, how are we to measure any gains and should we take the claim seriously?

The first question is a perennial one, especially when applied to investigatory work in any case, and the others are possibly unanswerable without resorting to an in-depth statistical analysis. This may seem rather flippant but such questions should not be ignored. Let me respond in more detail.

Investigatory work is considered by Cockcroft, HMI and the examination boards to be so important that it is considered to be an essential part of GCSE coursework
assessment. This kind of work allows children to think in
different ways and in many different directions from a
given starting scenario. Students get involved in
analysis and using the mathematical skills they have
learnt. Traditional problem-solving has tended to suggest
specific methods and the use of all the data to produce
the answer. Many pupils regard this as using a "recipe".
The analysis of a problem requires the ability to take out
the relevant pieces of information from a mass of data and
use them. Children used to the recipe method of solution
may lack the imagination and flexibility of mind
necessary. A Logo environment provides an opportunity to
test some of their speculations as well as encouraging
them to be imaginative in their ideas and thinking.

Children who are labelled low-attainers are not always
lacking in intelligence. Many of the pupils in 5(6) fall
into this category. I would claim that it is frequently
"attitude" that puts them into such lowly sets. By this I
mean their overall behaviour in class and their approach
to working give teachers the wrong idea about their
abilities. In my experience with such groups, working
with computers in general and Logo in particular provides
interest and motivation. The examination results of 5(6)
were almost as good as those in 5(3) on the same exam
paper. With autistic, low-ability and disturbed children,
Weir has achieved many good results. In my experience
with Susan et al. I have also.

Do children of above average ability benefit from exposure
to Logo without detracting from examination studies? To parents and teachers this is obviously an important question. I can find no evidence of research to answer this, but I can make some suggestions after observing ZST1. The booklets of the SMP 11-16 series rarely tax any of them, exercise books tend to be a succession of correct answers. Most of the questions seem to require no more than a single solution. I vary the diet with puzzles, projects, investigations, practicals and computer work with and without Logo. Their ability to solve problems has always been good but the presentation of solutions, methods and reports has been poor. Since starting Logo, reports have improved dramatically, written evidence is clearer and more attempt has been made to justify conclusions and observations. I have no statistical evidence, but I believe that this indicates a positive benefit from working with Logo.

If it is agreed that Logo be introduced to a school, what should be the role of the teachers and how can it be blended into an already crowded curriculum? Hughes et al. believe that the teachers' role is crucial. Organisation of the classroom and pupils is vital to the success of a lesson. They maintain that teachers must combine the skills of boosting confidence whilst allowing pupils the freedom to experiment. Hoyles and Sutherland (1985) believe that at the beginning teachers should not try to direct pupils since exploration removes the emphasis from the result.
Teachers need to acquire programming skills themselves and to undertake the activities of their pupils. I have obtained a list of primitives available on our Logo but without the documentation to go with it. In a sense I have felt both the joy of success of discovering results from my own experimenting and the frustration involved when I have not been able to guess the correct syntax for a primitive. It may be more successful to restrict the list of primitives for some pupils - by having too much to work with they may not be able to abstract sufficient correct data. To assist in debugging and checking programs, pupils could use volunteers to act the role of turtle and build into a program the TRACE command which indicates which line of the program is being executed, perhaps also building in a time delay using WAIT, can give a further opportunity for error analysis. I have used lessons for discussion purposes to ascertain what my pupils know and to remind them of how the commands work. They have listed all the commands they have learnt and then used their own words to describe syntax and use. Other pupils have looked at the list of primitives and have tried to find out their uses and this is not easy! To encourage children to discuss their work it has been suggested that teachers declare to their classes "If you need help ask at least two friends before me!" This can serve to make pupils discuss things rather than fall into the trap of "learner helplessness" quite common with low-attaining pupils.

The giving of assistance to children is crucial to their
development. Teachers need to ask themselves questions such as: "When should a child be left alone? When should help be given, how much and in what form?" Maxwell suggests replying "Try it and see," or "What do you think?" These questions will apply to every pupil but with a different emphasis for each. It is imperative that teachers know the limitations and psychological make-up of individual pupils. Instead of children asking "Sir will this work?" they need to be encouraged to try it and see.

It is naive to think that just by writing a few Logo programs the attitudes and abilities of both pupils and teachers will improve instantly. Success may be measured in small amounts such as in Weir's work with mentally and physically handicapped children. The short attention span shown by many low-attaining pupils can be seen in the changing of targets and in Logo without corresponding behaviour problems. Blamires believes that for some purposes the altering of goals may be necessary, but children need to be encouraged to return to their original task also.

Schools need to think about their policy when introducing Logo. Some have reacted by trying to accommodate it into their syllabus and devising worksheets to ensure that it is covered. An inevitable consequence of this method is children regarding Logo as just another piece of mathematics. With reference to the APU surveys on children's attitude to mathematics, pupils may come to regard Logo as "another bit of maths work." This will
cause children to lose out on all the benefits to be gained from using Logo.

Summary

If we are to assume that programming skills should be taught as part of a mathematics course, then we must decide on a suitable language. Basic is discredited by experts as encouraging bad habits, other languages such as Pascal are too daunting for beginners. The language Logo is described as a computer-based learning environment. It is structured so that programs may be built up quickly and easily, both editing and debugging are straightforward. Criticisms of it are not conclusive. Research in both the USA and this country give conflicting results. The crucial relationships are that of teacher-pupil-class, since they are closely involved in the learning process. Implementation of Logo in the classroom requires time, expertise and good organisation.
Rushby (1984) refers to four areas of Computer Assisted Learning (CAL). They are:
Instructional - Breaking down the knowledge and skills to be learnt into suitable learning units.
Revelatory - This is described as learning by discovery, using Logo is a good example.
Conjectural - Giving assistance in ordering and testing ideas. Simulations help in this area.
Emancipatory - The reduction of the workload by the use of graphics programs for instance.

Therefore when considering the implementation of software, we need to consider what are the educational purposes for which we intend to use it, how are we to use it and where do we obtain it. I would like to consider these areas separately.

1) What are the purposes behind our software?

Stenhouse (1975) refers to ideas in programmed instruction as developed from Skinner's Behaviourist theories of learning. Before a scheme of work can be written there are two main areas for consideration, viz. what are the pupils expected to learn and what skills and ideas do they possess already. A learning unit can then be designed to ensure that pupils acquire the intended skills and concepts. Computer packages can be designed to implement these processes. Tasks are broken down into
instructional units and these are displayed for the student to attempt. The pupil's response is then evaluated and analysed, within the confines of the program, and the next learning unit for an individual student is selected. Packages of varying degrees of sophistication can be written to implement this sort of idea, and therefore pupils' responses will determine the learning path that they will follow. The advantage of this kind of package is that teaching content and the pace of learning will be suited to individual students. It is further claimed that since the computer package evaluates rather than criticises, this will aid in the development and maintenance of pupils' confidence.

This type of approach has a number of faults. Stenhouse claims that the learner will tend to be rather passively involved in the process i.e. his responses will be to the program without necessarily any thought given to ideas from external agencies. Any package of this nature will have built into it the bias of the designer. Included here are the basic structure of the proposed learning process and the evaluation of responses. There are two further weaknesses listed by Stenhouse, the first relates to the "restricted opportunity for the transfer of learning," and the second refers to motivational aspects. I would include a further criticism at this stage. Because pupils get little direct feedback while using the package, they are unlikely to discover what parts of their analysis and method used for a task were valid. An important aspect of Logo, for instance, is the facility
for seeing the effects of commands and any readjustments immediately. Programmed instruction tends not to give such direct feedback. In terms of motivation, children will tend to ask what the point of the work is supposed to be - and this can lead to boredom with subsequent problems in using other packages.

The use of language involved in both documentation and screen display can prove ambiguous for learners. Research by Otterburn and Nicholson (1976) illustrates that many pupils do not understand many supposed "basic" mathematical words. I would summarise these criticisms of programmed learning packages as being the bias of the designer and the lack of both positive and negative feedback.

Kelly (1984) avers that instruction packages are produced for linearity of learning, but pupils do not necessarily conform to this sort of pattern. He goes on to say that individuals using these methods can develop solitary, antisocial attitudes in learning situations. Both Cockcroft and HMI believe that pupils should discuss their work with one another and teachers. Education goes beyond the mere acquisition of certain skills and knowledge, it is a process which is moral, social and emotional as well.

A fault of textbook based schemes of work is the rigidity that they engender in a school mathematics policy. Instead of teaching mathematics, the scheme is taught - SMP11-16 is an example of this. There is some excellent
material, but the course can be followed too slavishly. Similar problems arise when using an instruction package on a computer. It is too easy to follow the design of the package, no matter how flexible it may be, and not allow diversions as and when they occur. It is debatable whether or not mathematics is learnt in neat packages. Software designed to teach new skills to pupils needs to be investigated carefully to see how if at all it can be used.

The above refers to educational purposes which specifically apply to packages and their use and implementation. Other uses of software tend to relate more to the specific requirements of teachers and departments. In previous chapters I have referred to the computer as tool for data analysis and presentation etc., and as a learning environment. Therefore, assuming that the package to be used has been selected, we now need to consider how the hardware may be employed in the classroom to achieve our objectives.

2) How do we use the computers?

Classroom management by the teacher is crucial, in fact I would suggest the most important factor in any lesson. As with most ideas, some teachers can make use of software that others would not even consider and produce a good, successful lesson. The material selected and the attitude of teacher are important considerations.
There are further restrictions concerning the availability of computers, the class size and the classroom itself. All of these factors in combination require different qualities in a teacher.

Let me consider the circumstances of having one machine in the classroom with disc drive and printer. If the computer is to be used as a tool, it should be available to all children at any time during a lesson. The size of a class involved could produce queueing and the attendant discipline problems, so perhaps some form of rota could be worked out. This may not be suitable since children will need to use the computer at different stages of their work and for different reasons. Shy, quiet children who lack confidence in themselves and their abilities can easily lose their place in a queue. Dividing a class into groups can cut down on the number of children likely to be waiting, but this may also prevent some children gaining hands-on experience in these circumstances. There are too many variables for which to account and I refer back to the teacher-class-material relationship being the vital element. I feel that there is no easy solution to this sort of problem except the obvious one of providing more computers. If we wish to use the computer as a database, wordprocessor, tool for solving problems then the size of most classes means that more than one computer to a room is a necessity.

I will discuss particular items of software later in the chapter, so let us suppose we have a package to use an
adventure game, e.g. "The Last Adventure", "Flowers of Crystal" or "Xor." Many junior schools are able to timetable the computer for the use of each class, and this may be once, twice or even three times a week. The teacher, having introduced the package to a class must then ensure a fair and equitable use of computer time for everyone. If we assume a school day of five and a half hours and a class of twenty-five, this gives approximately thirteen minutes for each pupil during the day and it also assumes that there will be no interruptions. This is hardly worthwhile, but a division into pairs may work better and even this gives only twenty-five minutes for each pair. Problems arise when a group overrun their time allowed, what should the teacher do? Ruthless adherence to a schedule can encourage users to concentrate but it does not necessarily allow time for ideas to germinate.

In a comprehensive school the longest lesson is likely to be no more than ninety minutes. How can a package be introduced, discussed and then used in that time? Those pupils not able to use the computer in the first session require sophisticated note-taking abilities to note all the details necessary for working with the package. It may be possible for the teacher to act as the operator while the rest of the group attempt solutions to the problems posed in the package. The teacher has to be careful about any hints or clues given, since pupils then do not necessarily develop their own styles and ideas, and also ensuring that all the class are involved in the
lesson. Tall (1983) maintains that this form of three-way dialogue is difficult to employ successfully.

Using a network of computers has advantages and disadvantages. It is a great benefit that pupils are able to work on their own or as part of a group, at their own pace with comparatively little time pressure. However the teacher needs to be more flexible and aware of the progress of each child and/or group. The time to give advice and the method employed is vital to the development of each child. But by having a network of computers, the availability of the system is likely to decrease due to the pressure from other subject areas whose teachers may also want to use it.

3) Where do we obtain the software?

It is comparatively easy to preview textbooks, most publishers will send inspection copies for perusal. With computer packages, bearing in mind the likely expenditure, it is not usually possible to preview the material in this way since a disc may be copied. Yet a class teacher needs to check on program suitability and documentation to decide if it can be used. The versatility of software is a further consideration. Many packages, which advertise problem-solving as integral components, only require one solution and once this has been obtained will pupils get any further use out of it? A danger here is that early users of the package pass on details of the solution to others. Pupils will do this in any case, but it detracts
from the value of the package if only a single solution is required. This also furthers the myth that there is only one solution to mathematical problems.

Playing games on computers can have surprising results. Adventure games are quite popular in books, magazines, and program form. Users progress via a series of problems and contests towards some goal. Many of these, for instance "Arthur" and "The Last Adventure", have too much text displayed on the screen giving rise to confusion, boredom, and frustration. I do not intend to write a detailed critique of these packages but "Arthur" has only one solution and it is too easy to return to the beginning for no apparent reason. Pupils do lose interest rapidly after some unsuccessful attempts. "The Last Adventure" contains fifteen problems of the sort which belong to algebra textbooks of the 1950s. Where "The Last Adventure" does score is the extra package available with which pupils can design their own adventure game problems. This seems to give them more purpose, and Trott (1987) refers to a great deal of mathematical activity from the pupil designers.

Continuing with the theme of game-playing, a number of my pupils have invented their own games and competitions to be used with some programs which were not originally designed for these purposes. On Microsmile "The First Thirty" disc three there is a program called "Snooker." This program displays a facsimile of a snooker table. Users are invited to select an angle, measured from the upward vertical anticlockwise, and speed to be able to pot
a black ball. I have observed a number of pupils who had seemed to rely on hit-and-miss tactics, until one of them suggested an idea for a game. This has caught on, with discussions on rules paramount. At one stage someone discovered that a particular angle-speed combinations frequently potted the ball and now a rule to nominate the pocket has been introduced for these more expert players. By playing the game pupils are acquiring a feel and instinct for angle while also gaining insight into the concept of tactics. The crucial part of their motivation is that this game belongs to them, it has not been suggested or modified by me.

Until recently software has been produced by freelance programming experts who show little understanding of the needs of pupils and teachers. Poor documentation and screen displays have done little to commend packages from these sources. Instructions need to be clear and brief, since many people do not bother to read large amounts of text on the screen. Some teachers produce their own software but this is not necessarily an efficient use of their time and effort, even though programs from these sources would have the benefit of being written for specific circumstances. Commercial computer packages have a bright, glossy presentation no matter their educational value. Can teachers duplicate this? Do users have the right to expect this sort of presentation? My answer to the first question is yes but how long would it take to produce a bug free package? The second question is difficult to answer since pupil motivation is involved.
However because commercial packages are bright, musical and cheerful any package which is not may appear to be dull and boring by comparison.

Shuard and Rothery (1985) amongst others have written much concerning the use of English in mathematical work. Many children find it difficult to understand what is required as a response from a piece of writing. This applies to textbooks as well as examination questions. Clear, precise documentation for teachers and correspondingly unambiguous instructions for pupils are necessary parts of a computer package. It is too easy for relevant information to be swamped by a mass of extraneous detail. Consider the game "Master" on Microsmile "The First Thirty" disc two. The computer selects a random number comprising three digits and the player has to discover what they are. The user makes guesses and receives feedback in the form of a black square to indicate a correct digit in the right place and a white square to indicate a correct digit but in the wrong place. No clues are given as to the correct order. If children are expected to read and understand what appear to be fairly easy instructions, why do so many of the pupils I have observed misread them? The answer to this question is too complex to be answered in detail. The influence of television is large but children tend to watch it passively allowing details to pass over them. When faced with a computer screen they do not "see" all the details and instructions. I have observed with classes watching mathematics programmes on TV that they miss details of the
text and instructions. I have remedied this by giving a short test on observations at the end of each program. Questions have included ones on the mathematical content as well as on observation and details of the actors and production team. Many children started taking notes about each programme and this served to offset the original problem of "lack of attention." This kind of problem can be avoided if there is limited access to computers and the teacher acts merely as operator, but it would not include the success of the game developed from the snooker program.

Drill and Practice routines offer a graded series of questions to the user who passes on to the next level when a pass rate is achieved. Some are described as a modern and exciting way of learning multiplication tables or spelling, they are described as the ultimate weapon of the "back-to-basics" lobby according to Chandler. This type of package assumes that mathematical skills are learnt like physical ones through continued practice. A further problem with this sort of routine is again the lack of feedback both positive and negative. Pupils do not find out from a wrong answer which parts of it were correctly done. Errors could be due to carelessness or more fundamental with regards to the concept involved. If pupils do not get involved in error analysis then failing to progress to the next level could lead them to believe less in their own ability. Also, after an initial burst of enthusiasm when using a computer, pupils' interest will wane not only for mathematics but also the use of
computers. Chandler refers to the computer as being a "tool of awesome potency" and the use of such routines making it possible to take a "giant step backwards to the nineteenth century". Research carried out in the USA prior to 1984, referred to by Chandler, claims that 80% of software packages were for tutorial-type applications. He goes on to claim that a similar percentage would be a reasonable approximation for uses in this country although no survey had been carried out.

To design software for use in the classroom, we must analyse what computers can do well. They can store and process data and display it in various ways, within limits they can draw pictures and make noises. A programmer needs to be aware of the possibilities and limitations of computer usage. We must consider the problem areas of the curriculum and decide whether a computer package could assist learning in these areas. Teachers' considerations here are concerned with economics. How does the time involved in producing software tailor-made for a group of pupils match the expense, slightly less direct applicability and probably better-presented commercially-produced packages. In fact will the use of a computer actually help in the learning process?

Summary

When choosing software, our first consideration should be what are the educational purposes for which we intend to use it? We may wish to use it as part of mathematical
work i.e. as a tool or as an instructional program. The idea that programs can be developed to teach concepts in mathematics owes much to Skinner's Behaviourist theories of learning. Essentially a concept is broken down into learning frames and these are displayed for users to work on. Answers are analysed within the program to determine the next frame for that user. Each pupil has the opportunity, which will vary according to the flexibiltiy of design of the package, to progress at his own rate along a learning path relevant to his ability. This sort of approach does not give sufficient feedback to children. They do not find out enough about their mistakes in order to learn from them, or to repeat correct parts. Much emphasis is also placed on the learning process by the package designers, they decide how a concept may be learnt and this does not necessarily apply to all users.

Management of computer use in the classroom is the prime concern of the teacher. Obvious factors, such as environment, availability and number of machines, affect the organisation of a lesson. Networking has its advantages over stand-alone computers in giving more children computer experience at any one time. This situation does call for different skills from the teacher and can also lead to timetable pressure from other departments. Fairness for all users is part of the teacher-class relationship and perhaps positive discrimination in favour of girls will improve some of their attitudes to mathematics and computers.
Computer software requires reviewing before use inside the classroom, and this is not always possible. Teachers may have to rely on word of mouth and magazine reviews to find out about software. Game-playing can enhance the learning process, especially if the children don't realise what the mathematical content is! Some adventure games, although interesting at first, tend to further the idea that there is only one solution to mathematics problems. Children learn better when they are participating. Given the opportunity, with some packages they can invent their own situations and rules. This maintains motivation and develops mathematical awareness. The presentation of software written by teachers or commercially produced is vital to its success. Good documentation and clear wording make the success of a program more likely. Teachers writing their own programs have problems with time and presentation. How long can they afford to spend on development and does the presentation match that of commercial material?

Drill and Practice routines rely on the assumption that mathematical skills are learned like physical skills through practice. Lack of feedback and the onset of boredom contribute to the paucity of the educational content.

When designing software for the classroom we need to be aware of the limitations as well as the advantages of computers. Perhaps teachers can identify areas of the curriculum which could be taught better using a computer
and then work with programmers to produce more relevant software.
In the preceding chapters I have referred to the many uses of computers in the classroom, with some reference to particular items of software. The outlay in this area is comparatively small against that of expenditure on computer hardware. Problems exist for schools and LEAs regarding the new generation of computers. Over 80% of educational establishments have the BBC model B computer. This ceased production recently and has been superceded by the Master series. Many schools are already committed to BBCs and the corresponding software, therefore careful consideration must be given to any updating of the system. Replacing Model Bs with Masters can be achieved on a long term basis and therefore financial outlay can be spread over a number of years. There are some problems regarding the compatibility of software for the new and old systems. For instance, there is difficulty in loading "The Last Adventure" onto a BBC Master and the menu of Microsmile "The Next 17" comes out as gobbledygook, although there does not seem to be any problem with the program operation. To convert programs is time-consuming for all concerned but may end up being a necessity. The new Amstrad PCs, with large memory, printer, good quality word processor and built-in disc drive, are also IBM compatible. Since IBM compatibility is regarded as something of a benchmark for commercial software, and allied to this the Amstrads are competitively priced, should schools be looking in this direction? They could be an interesting purchase. Research Machines have the
Nimbus, Commodore have the Amiga, and both are fast and powerful with pull-down menus, windows and mice. In computer literature much is written about 8-bit, 16-bit and 32-bit systems, but where will this lead and what can and should schools do about it?

Earlier this year an extra £19M was allocated to the education budget for Technology. This works out to be approximately one computer unit per school and one adviser per Authority. Hartley advocates that the economics of buying hardware need to be altered if we are to have modern up to date systems in schools. Therefore to answer questions regarding renewal of existing hardware, we must consider the purposes for which it is intended to be used. Any system purchased by schools needs to have plenty of software available for it. This obviously applies to all departments. This material should include spreadsheets and databases from commercially-orientated software as well as programs specifically applied to education. If a network of computers is considered to be essential, this can restrict the choice to be made. In fact this updating only maintains the present situation which is rather unsatisfactory. On average in secondary schools there is one microcomputer for every sixty pupils. It is also possible that even this figure is inflated since, for example, in my own school there are three computers taken up by administration work. Megarry (1987) believes that to get really worthwhile benefits from computers in education then this ratio needs to be improved to at least one computer for every ten secondary pupils and twenty
primary school pupils. This means increasing the number of systems in schools by a factor of six!

It is usually possible to obtain an educational discount and this will obviously vary. The more expensive the system the larger seems to be the discount. LEAs, by standardising machines within an Authority, could order in far greater bulk than any individual school and thus obtain a substantial saving and possibly a better system. For example Robinson (1987) refers to the pricing of the Macintosh-Plus. This retails for £1,995 in shops but for schools ordering up to five this drops to £1,095 and decreases still further to £895 for orders of eighteen or more. It is unlikely that a school could afford this many but a LEA could. There are further complications involved. The consequences of schools opting out of Local Authority control or having control of their own budgets have yet to be revealed.

What should be the role of the Adviser in the choice of hardware and software? A mathematics Adviser's primary responsibility is to that subject, although this is changing, but the purchase of hardware for schools has wider implications. A computer expert is required who can coordinate the needs of all the schools in an Authority so that bulk purchases may be made. This means that an Adviser on Technology needs to know about possible developments in hardware and software for all purposes. The purchase of software is a problem area. I have already referred to the difficulty in previewing packages,
understandably software houses are reluctant to issue inspection copies. Reading reviews is not enough, teachers as well as pupils need to have hands-on experience to assess possibilities. Even then some expert help from someone familiar with a package would help teachers decide on its suitability. Jones and Green (1987) refer to the high value that the teachers sampled placed on in-service training. This included special mention regarding both seeing and using software. An Adviser in Computing cannot concentrate in only one area of the curriculum, there is a good argument for someone to be responsible for technology as suggested by the DES.

The attitude of teachers to computers in the classroom is crucial to success or otherwise. If we acknowledge that some awareness of computer usage is an essential part of the curriculum, then not only do we need to consider the content of the curriculum in order to accommodate it but also how teachers will be trained for its implementation. For a number of reasons many teachers of mathematics have transferred into computing. This has had the effect of creating a mathematical bias in the subject and leaving gaps in mathematics departments which are difficult to fill. Mathematics teachers need to acknowledge that some of their pupils are likely to be better at programming computers than they are themselves. It is possible that some teachers fail to make use of computers to avoid the embarrassment that this may cause.

Much money has already been spent, and will need to be
spent in the future, on getting computers into schools. Most computers, however, are set aside for computer studies or information technology. Tall (1987) refers to a study carried out early in 1986 which estimates that nearly 60% of computers are installed in one room and only 5% were specifically set aside for mathematics. This works out to be approximately one hour's access per year per pupil. Teachers in this study kept diaries and some of their comments give reasons for the non-use of computers: a probationer lacked confidence, we may ask what experience was provided while training whereas a more experienced teacher explained that computer use did not fit in with exam preparation.

Depending on the level of priority we place on making children able to work with and on computers, what Tall describes as "computerate", we must look at the mathematics syllabi in detail and consider changing some elements. For instance it is required at A-level to be able to divide polynomial expressions by others. Students find this difficult since they are unlikely to use the method with numbers because of using calculators. If we analyse the things that computers do well should this lead us to ideas of curriculum change in mathematics and a domino effect in other subjects?

A comparatively easy way of teaching is exposition by the teacher. The class can be "controlled" if all are supposed to be listening and then doing the same piece of work by identical methods. Unfortunately not everyone who
is listening will learn. Investigatory and practical work usually means groups of children working on different topics with all the attendant noise of discussion and apparent chaos in the classroom. Many teachers feel out of their depth in this kind of situation, a noisy classroom being equated with poor discipline hence a bad teacher. Colleges of Education need to convince students that this is not necessarily so. Using computers should be a common occurrence in mathematics lessons and not necessarily one for each pupil. Mathematics is learnt best by pupils discovering methods by themselves in a relaxed and social atmosphere. Furthermore, practising teachers need to be convinced of the benefits to be gained by their pupils when using computers.

Primary school children are likely to be familiar with computers. With parents having more choice in selecting schools for their children, computer experience may be disjointed. Close liaison between secondary schools and their nominal feeder schools can do something to alleviate this problem. This could involve teaching children to program in Logo at an early age but it also means pupils allowed more access on a regular basis in comprehensive schools. It is possible that teachers in junior schools do not possess the knowledge or lack the confidence to cater for this. Visiting teachers from comprehensive schools can assist in this area given suitable timetabling. To help in the transfer of pupils from junior to secondary, always a problem area, relationships can be established by pastoral and departmental work with
pupils in junior schools. The links can be maintained after transfer by teacher exchanges. This can serve the dual purposes of developing mathematical awareness, with and without computers, and provide children with the opportunity of forging relationships with new teachers.

With the current availability of low-priced microcomputers, parents are willing to buy them for the home. Gray, Hartmann and Murdock (1987) believe that, at home, computers are most frequently used for playing games only. This is not necessarily a bad idea but many are trivial and others merely exercises in dexterity. Children regard these games as practice for the "real thing" in arcades, and this neglects the real power of the micro. One of the problems in home computing is the difficulty in acquiring sufficient skill in programming. Not only is Basic a poor language to learn but computer manuals are difficult to comprehend. Telford (1987) refers to problems encountered by teachers attempting to understand manuals and lists the reading ages required to comprehend some. If teachers get into difficulties how are many parents and children going to cope?

The role of parents involved in education is an area for much discussion, especially in mathematics. The subject is regarded by most people as being an essential part of the school curriculum but it is also regarded as being one of the most difficult. Papert has coined the word "Mathphobia" to describe a morbid fear of mathematics and much of this fear is caused by parental attitudes. This
applies no matter the mathematical ability of the parents. These attitudes have their effect when educational programs are chosen by the unequal assistance given to sons and daughters and parental control when working. The choice of programs available to parents is at the level of soap-operas according to Kelman et al, and these include "modern and exciting ways" of practising multiplication tables and spelling. The problem here is that everyone who has some basic competence in mathematical skills believes that they know how to teach them. Programs are written/designer by computing experts not mathematical educators. The relative cheapness of a computer may be offset by the lack of a suitable screen. If a home computer has to be set up using the family TV this can create difficulties and therefore negative attitudes towards computer usage. Disc-drives are far more useful for loading and saving programs, but many parents do not invest in them and buy cassette systems instead - these may actually be built into the computer itself. They are tedious to use for long programs and are also liable to have minor problems, such as sensitivity, to hinder use. Compatibility, or lack of it, with school computers makes it more difficult for children to develop programs at home and work on them at school and vice versa. Although this has been something of a catalogue of problems which do not necessarily apply to all families, they do exist and can create a dampening of enthusiasm within children.

There is great potential in this area, parents can work with schools. Perhaps recommendations can be given as to
suitably compatible systems. If musical instruments can be loaned out then computers could be borrowed by children. Although there may be problems involved here, it is worth investigating if computers can be purchased under education budgets thus offering sizable discounts. Is it also possible to make Logo easily and cheaply available on computers, as with Atari, since programming skills are easily learnt in this language.

Conclusions

I do not consider that using computers is an essential part of mathematics education. It is perfectly possible to develop mathematical awareness in students without their aid although this does depend on the attitudes and abilities of individual teachers. In a survey reported by Jones and Green only 12% of the departments sampled used computers more than once a week. This figure will be decreased further when individual teachers are taken into consideration. It is reasonable to suppose, therefore, that many good teachers are not involved in using computers in the classroom.

Computer technology abounds in modern society, ranging from programmable washing machines to cash dispensers at banks. Pupils need to be aware of the influences, uses and abuses of computers. In Fitzgerald's survey on the occurrence of technology in industry, he refers to employees being confident about its use. Government initiatives, such as City Technology Colleges (CTC), give
an indication of the importance of computers and technology. Therefore if computers are considered to be so vital a part of education then let us integrate them into the system.

There are many problems in trying to integrate computers completely into the curriculum. I would like to single out three in particular, although there are links between them. They are, with no order intended, time, money and attitudes, and will now be considered separately.

1) Time

If so few teachers do not use computers in the classroom, we must ask why not? Is it insufficient access in school or lack of knowledge on the part of the teacher? The problem of access may be solved by finance. The second may be solved by providing opportunities for teachers to gain the necessary skills of programming and classroom management. Teachers need time to gain expertise and discover the learning potential of pupils using computers. They also need access to and time to review software with someone who knows the package well. It is easy to overlook the potential of programs with only a cursory glance through the documentation. A plea here for more easily understood manuals and documentation. We also need to consider the time necessary to aid pupils' learning about computers, do we need to change the mathematics syllabus again, so soon after GCSE, to accommodate the things that computers do well? I think not. It can be
argued that too much time is spent teaching the present syllabus, especially to average and below average ability pupils in years four and five. I, myself, have taught my present fifth form nothing they had not already covered in the first three years in comprehensive and possibly junior school as well. Given that we start a computer education programme in junior schools, will there be sufficient time to cover any current mathematics syllabus. Pupils' extra motivation, in having more control over their learning, may increase the speed of their development. They need to regard the computer as another piece of equipment and to become as familiar with it as they are with rulers.

Some current initiatives, such as TVEI, Information Technology, while having some value, also take up much of the computer time available. Their good points could be absorbed into the rest of the curriculum or extra finance could be provided for more resources. The government's proposed CTCs could drain the human resources of the state system and if schools are to have extra control over their finances this could lead to less buying power for LEAs. It is difficult to judge what sort of effect these changes will have.

The proposed National Curriculum, with testing in mathematics at ages seven, eleven, fourteen and GCSE at sixteen, could have a great effect on the way mathematics will be taught. There is a danger that teachers will too easily fall back to tried and trusted methods of teaching to a syllabus rather than concentrating on the development
of mathematical awareness. The potential of computers may be lost in a morass of Drill and Practice programs which rely on the assumption that mathematical skills can be learnt like physical ones through lots of practice.

2) Money

One of the reasons why computers are not used more often in the classroom is insufficient access to them. Many schools place their computers in one room and this limits the time available for teachers to use them since only one group, of say thirty pupils, will be using the room at any time. The current emphasis placed on Information Technology restricts access by other curriculum areas. A solution would be to equip classrooms with at least six computer systems for use during lessons. If a cheap unit such as an Amstrad were bought, this would be approximately £2500 per classroom and to outfit four rooms in this way would cost £10 000 per school. To concentrate these systems in the mathematics department may seem slightly biased, genuine cross-curricula work could be initiated and the value of mathematics as a service subject would be easier to see for pupils. They would then have the opportunity to become more familiar and confident with the use of computers.

We need software to assist our work in mathematics and other curriculum areas. I have already referred to the time required to seek out and investigate software. Finance for such items is difficult to justify at present.
Software costing £20 is the price of four or five textbooks or three or four resource books. To assist in changing the attitudes of teachers, grants for the purchase of software would help. Perhaps teachers could discuss with programmers the requirements of educational software, a link which could produce cheaper and more applicable packages.

Aside from the time required by courses for teachers supply cover would be needed for those members of staff on a course. Occasional one day courses would be sufficient for teachers to acquire the necessary programming and classroom management skills.

3) Attitudes.

Possibly the greatest resource in education is the teaching force. Little or nothing could be achieved without their good-will, hard work and all-round enthusiasm. Therefore, if we wish to make students aware of the uses and abuses of computers, teachers need also need to be convinced of the benefits to be gained from having them in the classroom. The advent of GCSE has meant a change in policy for many teachers, they are encouraging group, practical and investigatory work rather than the ubiquitous exposition by the teacher type of lesson. It may be argued that we need to settle down and evaluate the effects of change, but I disagree. We have the opportunity to make mathematics more fun and exciting and to allow pupils some element of control over their
learning.

The attitude of parents can have a great bearing on their children's outlook. It has taken many years to convince parents about the benefits of calculators and still many refuse to allow their children, of whatever ability, to use them. Parental help with computers could assist the progress of children but an authoritarian attitude could equally dampen enthusiasm. Parents, like teachers, need to see the benefits of computers in terms of exam results. Schools could invite them in to see and use computers so that they can get a feel for the excitement and enjoyment of successfully solving a problem, or just drawing a picture using computer graphics. There is a need for good communications with parents.

LEAs are restricted in the amount of finance available for education so it is difficult to judge what would happen in some Authorities given financial assistance for developments in the field of computer education. In terms of finance, the government is the only body which can truly provide the necessary time and equipment.

Papert refers to the "QWERTY" phenomenon. By this he means the retention, on electronic keyboards, of a lettering system which was developed to overcome problems in mechanical typewriters. No such problem exists for computers but the keyboard design is retained. I have seen examples of keyboards which are set out alphabetically, I wonder at the effectiveness of these.
Users have commented that once they get used to the idea, they are quite easy to operate. Having seen pupils operate keyboards and find great difficulty in finding the correct keys, I sense some of the frustration that builds up. Even intelligent pupils who know the alphabet find the conventional keyboard difficult to operate initially. Is it feasible to carry out some research in this area, bearing in mind the success of Concept Keyboards and limited function units used by low-ability students?

I will conclude by saying that the present state of affairs is not good. The financial constraints mean that computers will not be available for a significant amount of time to most pupils. However, reading Papert in "Mindstorms" gives me a feeling of optimism for what children can achieve given the opportunity.

Summary

In planning for the future, we must consider what kind of computers will be used. The current generation of BBC model Bs is no longer being produced and the replacement system has its drawbacks. To achieve worthwhile gains from computers, the ratio of computers in comprehensives needs to improved to one for every ten pupils at least and one for every twenty in junior schools.

The role of Advisers in Mathematics, Computing and Information Technology is crucial for developing the use of computers and assisting teachers in their
implementation. Development work needs to begin in primary schools by teaching programming skills via Logo.

Computers at home are a possible source of assistance but lack of disc-drives and an overly authoritarian attitude from some parents can dampen enthusiasm and this may be transferred to a school situation. A high reading age is required to understand some manuals. Too much jargon can lead to incomprehensibility and frustration.

To conclude I list three areas which need consideration before we can further the use of computers in school. Time is required to show teachers the possibilities and to integrate their use into the curriculum. Money is needed to finance extra training and to provide both hardware and software. Attitudes from the government, some teachers and parents need to change. Although the present situation is rather a mess, there is the optimism engendered by Papert's ideas of what children can achieve given the opportunity.
Bibliography


Hoyles C., Sutherland R. and Evans J. "..and then she had the idea of dividing by 360", TES, 1/3/85, p52.


Leith G.O.M. undated. Education Review: Occasional Publication 1


Appendix 1: A description of Menzies High School and its Organisation.

The school is situated in a large expanse of council houses and it is the old West Bromwich Grammar School. It is designated eight form entry, with approximately 1300 pupils on roll including sixth form, but this frequently works out as nine forms because of its popularity with parents in the authority. In fact a first year intake of approximately 240 will contain about 140 from the nominal feeder schools with the remainder coming into the school on appeal.

The school is on a large site with departments allocated to various blocks. The pastoral system is at present based on four houses, although we are soon to change to a year system. Each House (Greek(G), Olympian(O), Spartan(S), Trojan(T)) has a Head, a man, and an Assistant, a woman.

The mathematics department comprises six full-time teachers with one Head of House, who has about three-quarters of a timetable, one teacher who does some computer studies with three-quarters mathematics. In addition there is a Deputy Head, the Head of careers and another computer studies teacher and between them they account for 25-30 periods a week. We operate the SMP11-16 Mathematics scheme with its booklets in years one and two for all children and the textbook series (yellow, blue green) for the majority of pupils from year three onward.
We are allocated five thirty-five minute periods per week, although this is under review due to a proposed change in the school day.

For timetabling purposes years one, two and three are split into parallel halves based on the Houses Greek/Olympian and Spartan/Trojan. We set in year one after the first half term so that pupils with learning difficulties can get more attention in a small group. In year four the pupils are set into ten groups based on a combination of their abilities in maths and english. With two separate groups of five sets, the two groups either side of the border, sets five and six, tend to be peculiar, disjointed sets.

There is one computer room, situated in the mathematics block, with a fifteen station network of BBC model Bs run by a Winchester hard-disc file server. In years one and two a course in Information Technology is taught to all pupils and fourth and fifth form courses include Computer Studies, Information Technology, Keyboard Skills and Personal and Social Education. There is no A-level Computer Studies offered although there is an option in General Studies and a one year GCSE Computer Studies course. This leaves only four periods a week for any other work in the computer room, although some time may be negotiated.

The mathematics department has no computer for their own use, although it is possible to borrow two from other
areas in the block. There are a number of other departments which have their own computers and make use of them extensively.
Appendix 2: A brief description of my classes. This includes explanatory notes to explain the notation used in the work.

Out of a total of 240 pupils in year 5, we would expect between thirty and forty equivalent O-level passes and a similar number of pupils who would not be entered for any exam. The MEG(SMP) exam offers four papers to be taken in pairings of 1/2, 2/3, 3/4. Papers 1/2 are the easiest and equivalent to Foundation level GCSE now. Most of set1 will take papers 3/4 and the remainder, along with set2 and some from set3, will take papers 2/3. The rest of set3 and sets 4, 5 and 6 will take papers 1/2. Most of sets 7 and 8 will take the RSA arithmetic papers.

Classes plus explanation of notation

2ST1 is the top set in year two of the Spartan/Trojan half year. There are approximately 33 in the group. They follow the booklets of SMP11-16 with supplementary work on puzzles, practicals, investigations and computers.

2G04 is the bottom set in the Greek/Olympian half year. There are 23 in the group and I have them for two periods a week. A number of the pupils are withdrawn for extra english lessons and they have two lessons with an extra teacher from the Pupils Support Service. This class follows the booklets slowly.

3ST4 is the fourth set out of five in the Spartan/Trojan
half year. There are twenty in the group following the Green series of books from SMP. I have negotiated one lesson per week in the computer room on fairly permanent basis.

3GO5 is the bottom set in the Greek/Olympian half and there are 21 in the group. They are attempting to follow the Green series books although some find this rather difficult. Some pupils are withdrawn from lessons and I have negotiated five lessons per fortnight in the computer room.

5(6) is set6 out of ten in year 5. They are the top group of the lower band with 27 in the group. They have shown little interest in most work from the syllabus but enjoy doing puzzles and investigations. They have no regular slot in the computer room but we use it when it is available on an ad hoc arrangement.

L8 is an A-level group comprising 5 students all with grade C at O-level. We occasionally borrow two computers to illustrate mathematical work. Two of the group are very interested in computing and one has his own system at home.

In addition there are lunchtime sessions for any pupil interested and a number of pupils attend even though I don’t teach them. Many work on Logo and others work on their Computer Studies projects.