West Sussex Institute of Higher Education

Bognor Regis College · Upper Bognor Road · Bognor Regis · Sussex · PO21 1HR
Bognor Regis (0243) 865581

MATHEMATICS EDUCATION CENTRE

LEARNING MATHEMATICS WITH MICROCOMPUTERS

Adrian Oldknow

LEARNING MATHEMATICS WITH MICROCOMPUTERS

Adrian Oldknow   West Sussex Institute of Higher Education

1. As A Teaching Aid.

In the report of the Cockcroft Committee "Mathematics Counts" there is a frequently quoted recommendation concerning the desirable range of learning activites:

243 Mathematics teaching at all levels should include opportunities for
- *exposition by the teacher;*
- *discussion between teacher and pupils and between pupils themselves;*
- *appropriate practical work;*
- *consolidation and practice of fundamental skills and routines;*
- *problem solving, including the application of mathematics to everyday situations;*
- *investigational work.*

Recent papers by Anita Straker (seconded from Adviser for Mathematics, Wiltshire C.C. to lead the MEP Primary Project team) and Hugh Burkhardt (Professor of Mathematical Education at the Shell Centre, Nottingham University) point to software for mathematics teaching that has been developed at both Primary and Secondary level and to styles of use that enable the mathematics teacher to employ a microcomputer to act as an aid in each of the types of learning activity referred to above.

The Association of Teachers of Mathematics (ATM) have published a valuable guide: "Some lessons in mathematics with a microcomputer" which gives examples of the use of specific programs in the classroom in a variety of learning activities - and which is accompanied by versions of the programs for particular microcomputers. They also publish a useful pamphlet: "Working notes on microcomputers in mathematical education".

The Microelectronics Education Programme (MEP) publish an in-service course "Micros in the Mathematics Classroom" prepared by the Shell Centre in collaboration with the ITMA project and will shortly be publishing much of the Primary mathematics software developed by Anita Straker.

The Mathematics in Education and Industry (MEI) project have recently published a collection of programs in connection with their A-level course that provide a very good range of classroom aids for the teacher of sixth-form mathematics. Other projects, such as the Schools Mathematics Project (SMP), are preparing programs to accompany their other classroom materials.

Software packages for the teaching and learning of mathematics continue to be produced by national projects, regional centres, LEAs, subject associations and individuals - this variety of sources of production does much to ensure the promise of a wealth of material from which the interested teacher of mathematics may choose. However the effectiveness of their role in the learning of mathematics depends as much upon the imaginative development of classroom use by the individual teacher as the original design of the programs. In the hands of a skilled teacher even a poorly designed program may prove an effective aid.

A criterion often put forward as important in the educational evaluation of a piece of software is: "could the activity be equally as well effected without the use of a computer?". While this makes good sense it has been observed that many teachers who do extend their range of activites to include discussion and investigation when aided by pieces of suitable software (but which could be sustained without any use of a computer) would not have done so if left with conventional teaching aids. Thus the very need to plan out the use of a particular program in the classroom can bring about some of the changes of approach that Cockcroft seeks.

2. As A Problem Solving Tool.

Undoubtedly the computer is a very powerful and versatile tool in the hands of a mathematician engaged in problem-solving. It is arguable that it is now at least as important for a practising mathematician to be able to program a (micro)computer as, say, to be able to use a calculator or a set of drawing instruments. In this context alone there is an onus on the teacher of mathematics to provide opportunities to illustrate this form of mathematical activity. There are numbers of books, journals and articles which provide valuable source material on this aspect of computer usage for the teacher. The following example appears in the book "Learning Mathematics with Micros" by Oldknow and Smith, published by Ellis Horwood, and is discussed further in "What if the Model Doesn't Work" by Oldknow in Teaching Mathematics and Its Applications Vol.1, No.3.

The bookcase problem:

A long bookcase 1ft wide is to be passed down a long corridor of rectangular cross-section 4ft wide and 7ft tall. What is the maximum height of such a bookcase that can just be passed down the corridor (by tilting)?

Suppose the case is tilted at angle A to the horizontal and its maximum height is h ft. Then from Fig.1 we have that:

$$h \sin A + \cos A = 7$$
$$h \cos A + \sin A = 4$$

These are two non-linear simultaneous equations in h and A. It is possible, with some determined algebraic and trigonometric manipulation to eliminate either h or A between them. This yields either:

$$\sin A.(4 - \sin A) = \cos A.(7 - \cos A)$$

or

$$h^4 - 67 h^2 + 112 h - 64 = 0$$

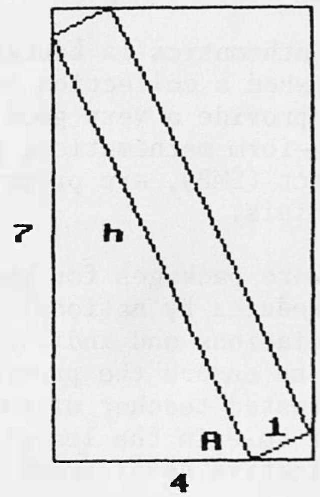

Fig 1

In either case we are left with equations that cannot be solved by simple means and so would be forced into numerical methods. With the computer we can easily explore a solution directly from the original pair of equations (which only required an elementary knowledge of trigonometry to derive). First rearrange each of them to give h as a function of A:

h = (7 - cos A)/sin A

h = (4 - sin A)/cos A

We must obviously avoid using A as 0 or 90 degrees as that would involve division by zero and, of course, we must remember that BASIC uses radians for its sines and cosines.

A simple program to tabulate the values of the functions is:

```
10 FOR A = 5 TO 85 STEP 5
20   AR = A*3.14159/180
30   S = SIN(AR) : C = COS(AR)
40   H1 = (4-S)/C : H2 = (7-C)/S
50   PRINT A,H1,H2
60 NEXT A
```

Which gives the following output:

5	3.92779074	68.8859984
10	3.88537951	34.6401404
15	3.87315552	23.3138913
20	3.89274073	17.7191674
25	3.94720371	14.4189152
30	4.0414513	12.2679582
35	4.18288982	10.775987
40	4.38252795	9.69831943
45	4.65685182	8.89950018
50	5.03113804	8.29875579
55	5.54563357	7.84521823
60	6.26794048	7.50555645
65	7.32028533	7.25734006
70	8.94771584	7.08527581
75	11.7227151	6.97898503
80	17.3636851	6.93165957
85	34.4643084	6.93924971

From which it is clear to see that the values of the functions are closest when A = 65 giving a value for h of about 7.3 ft. It is also quite easy to imagine the shapes of the graphs of the two functions. If greater accuracy is required than we just have to alter the limits of the A loop in line 10 e.g.:

```
10 FOR A = 60 TO 70 STEP .5
```

to 'zoom in' on the appropriate part of the domain of A. If the computer has high resolution graphics then it is very easy to insert a line to draw the graphs on the screen. Suppose the computer has a command PLOT X,Y to put a point on the high-resolution screen then we just need to find, empirically, a suitable enlargement factor E to scale the graph to fit the screen. For many of the current micros the following additions are suitable:

```
5  E = 2.5
50   PLOT E*A , E*H1
55   PLOT E*A , E*H2
```

The output from such a program can often be printed on a suitable printer which is how Fig.2 was obtained.

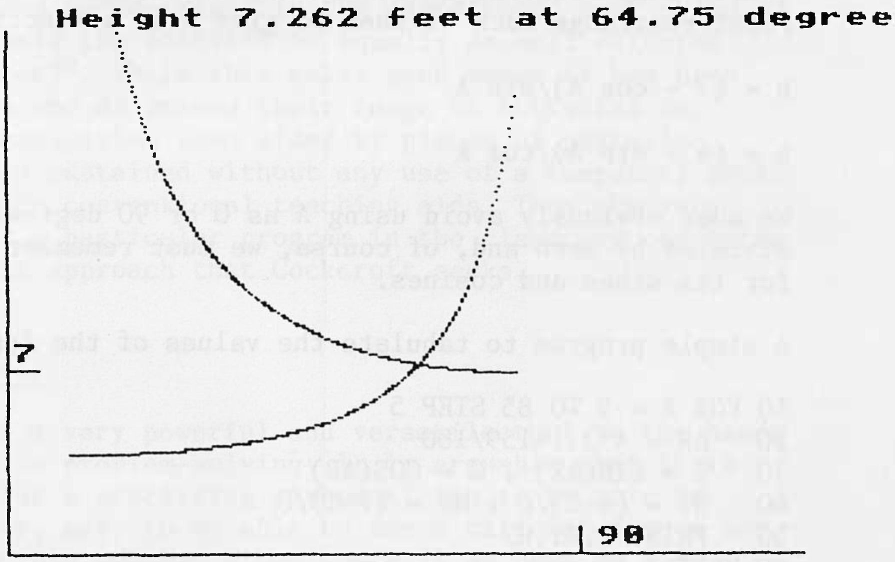

Fig 2

3. The value of short programs.

If a teacher of mathematics is able to control a computer to tackle problems in the way illustrated above then he or she also has the necessary skills to be able to exploit the power of the micro as an aid to mathematics learning inside and outside the classroom. One of the most powerful of such uses makes the micro into a 'test-bed' to facilitate exploration of mathematical behaviour. To illustrate the point there follow three simple examples from different branches and levels of mathematics all of which exploit the power of the micro to draw pictures.

The programs are written in BASIC but they are not language dependent and could easily be written in most other languages. Although BASIC has a reasonably standard subset which most manufacturers agree on there is, unfortunately, no standardisation on high-resolution graphics commands. In addition to the command PLOT X,Y ,which I assumed earlier, most micros have a command to draw a line from the last point visited to the point (X,Y) and I shall assume a command LINE X,Y that does this.

3.1 A simple graph plotter.

Here the problem is to map a part of the co-ordinate plane onto the micro's display screen (or part of it) - this is known 'in the trade' as a window/viewport mapping.

Suppose (X,Y) are the co-ordinates of a point in the plane and (XS,YS) are to be the corresponding screen co-ordinates and suppose we wish to map the part of the plane with X between XL and XH and Y between YL and YH onto the part of the screen with XS between 0 and SW and YS between 0 and SH, as in Fig 3.

We can use simple ratios to derive the relationships as follows:

XS/SW = (X-XL)/(XH-XL)

YS/SH = (Y-YL)/(YH-YL)

A simple program to plot points on the function Y = FNA(X) is:

```
10  DEF FNA(X) = EXP(-X*X)
20  INPUT "XLOW,XHIGH"; XL,XH
30  INPUT "YLOW,YHIGH"; YL,YH
40  XR = XH - XL : YR = YH - YL
50  SW = 320 : SH = 200
60  FOR XS = 0 TO SW
70    X = XL + XS*XR/SW
80    Y = FNA(X)
90    YS= SH*(Y-YL)/YR
100   PLOT XS,YS
110 NEXT XS
```

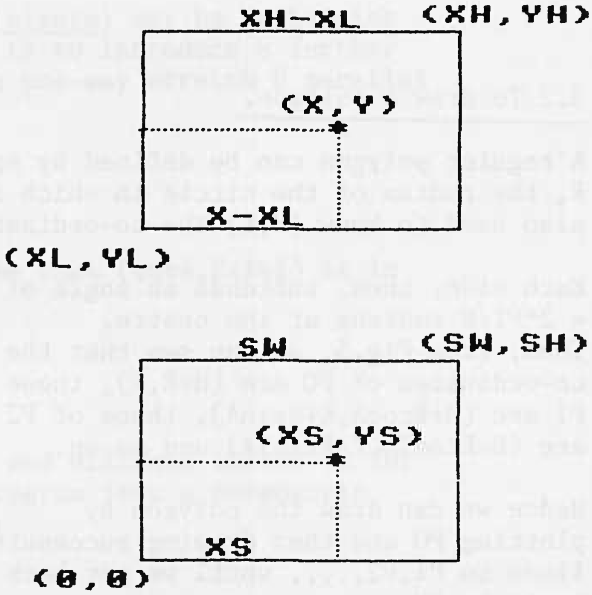

Fig 3

To explore different parts of the graph we can run the program putting in different values for XL,XH,YL and YH and to explore different functions we just need to change the function definition in line 10. Of course it might be nice to have properly scaled and labelled axes, error trapping etc. and there are plenty of commercially available graph plotting programs that do just that. However there is a satisfaction in knowing that it is simple to produce such a program and, of course, there is much more scope for applying one's mathematics in deriving such a program than in loading someone else's!

The graph could be made continuous by changing line 100 to

```
100   LINE XS,YS
```

However we must make certain that it starts correctly by plotting the first point e.g.:

```
55 PLOT 0, SH*(FNA(XL) - YL)/YR
```

A sample output is shown in Fig 4.

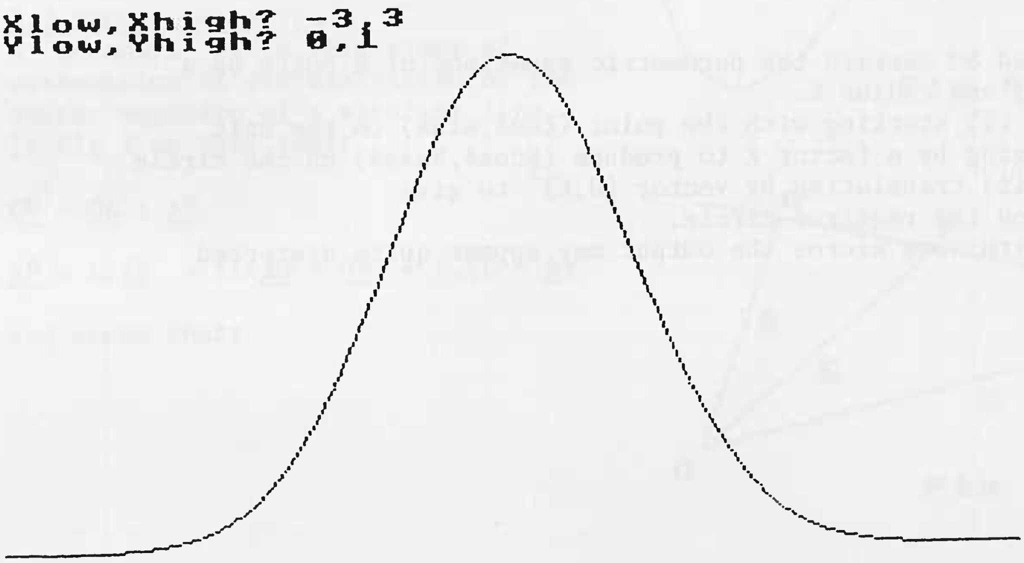

Fig 4

3.2 To draw a polygon.

A regular polygon can be defined by specifying N, the number of sides, and R, the radius of the circle in which it lies. To fix it on the screen we also need to know (H,K) the co-ordinates of its centre.

Each side, then, subtends an angle of A = 2*PI/N radians at the centre. Thus, from Fig.5, we can see that the co-ordinates of P0 are (H+R,K), those of P1 are (H+RcosA,K+RsinA), those of P2 are (H+Rcos2A,K+Rsin2A) and so on.

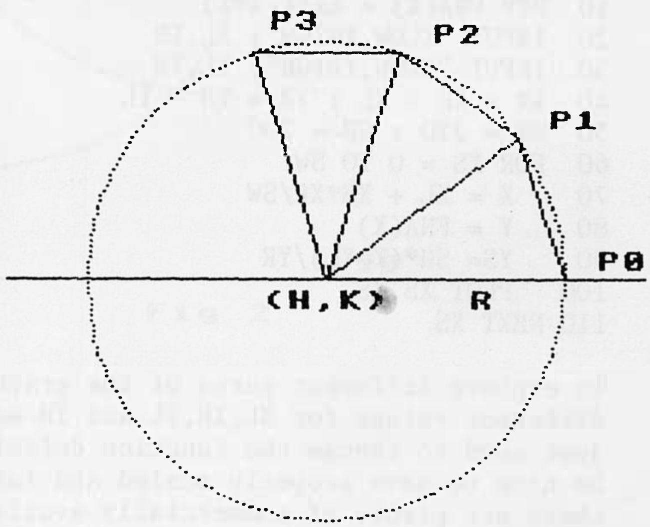

Fig 5

Hence we can draw the polygon by plotting P0 and then drawing successive lines to P1,P2,..., until we get back again to P0.

A simple program to do this is:

```
10  INPUT N
20  A = 6.283/N
30  H = 160 : K = 100
40  R = 75
50  PLOT H+R,K
60  FOR I = 1 TO N
70     X = H + R*COS(I*A)
80     Y = K + R*SIN(I*A)
90     LINE X,Y
100 NEXT I
```

For large values of N this will approach a circle. In fact for values of N greater than 40 the accuracy does not improve but the time taken for display increases and so a 40-point polygon has become the 'industry standard' for a circle in this sort of resolution
Many pleasing effects can be had by putting the body of the program (lines 50 to 100) within another loop. Thus, for example, we can have nested polygons by the following changes:

```
40  FOR R = 20 TO 80 STEP 10
110 NEXT R
```

In fact lines 70 and 80 contain the parametric equations of a point on a circle centre (H,K) and radius R.
We could think of (i) starting with the point (cosA,sinA) on the unit circle, (ii) enlarging by a factor R to produce (RcosA,RsinA) on the circle of radius R and (iii) translating by vector $(H,K)^T$ to give (H+RcosA,K+RsinA) on the required circle.
On some displays with some micros the output may appear quite distorted

because the little picture drawing elements (pixels) may be rectangles rather than squares. A possible cure for this is to introduce a further transformation between (i) and (ii) to apply a one-way stretch E parallel to one of the axes. Suitable amendments are:

```
35  E = 0.8
80    Y = K + E*R*SIN(I*A)
```

From a consideration of the geometry we can see that (cosA,EsinA) is in fact a point on the ellipse:

$$x^2 + (y/E)^2 = 1$$

So, for the price of a polygon we get circles and ellipses thrown in for free In fact we can easily change the whole program into a parametric graph-plotter by the following simple changes:

```
10  DEF FNX(T) = COS(T)
15  DEF FNY(T) = SIN(2*T)
17  N = 40
50  PLOT H+R*FNX(0),K+R*FNY(0)
65    T = I*A
70    X = H + R*FNX(T)
80    Y = K + R*FNY(T)
```

This draws a curve defined by N points of the parametric form: x=f(t) y=g(t) for values of t between 0 and 2*PI , shown in Fig 6.

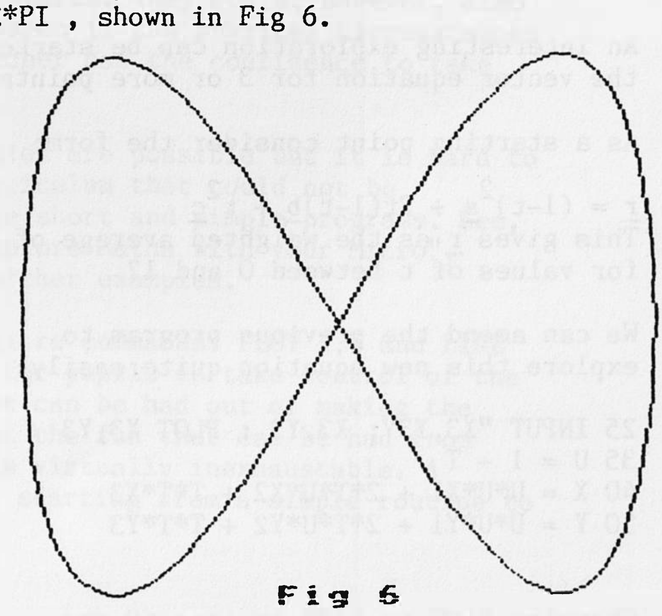

Fig 6

3.3 Vector geometry.

A 'standard' sixth-form piece of mathematics is the derivation of the vector equation of a straight line. In Fig 7 we have that:

$\underline{OP} = \underline{OA} + \underline{AP}$

$\underline{AP} = t.\underline{AB} = t.(\underline{AO} + \underline{OB}) = t.(\underline{b} - \underline{a})$

and hence that:

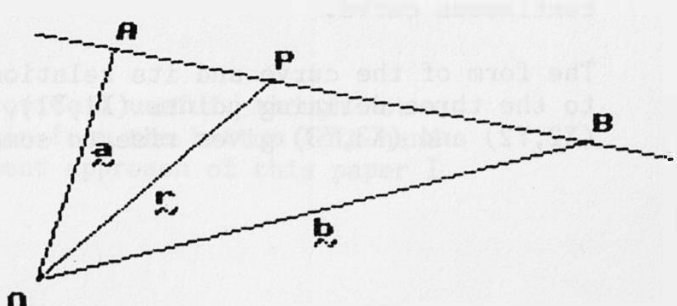

Fig 7

$\underline{r} = \underline{a} + t.(\underline{b} - \underline{a})$

$\underline{r} = (1-t).\underline{a} + t.\underline{b}$

To make this symbolism 'come alive' we can easily embed the ideas into a program.

```
10 INPUT "X1,Y1"; X1,Y1 : PLOT X1,Y1
20 INPUT "X2,Y2"; X2,Y2 : PLOT X2,Y2
30 INPUT T
40 X = (1-T)*X1 + T*X2
50 Y = (1-T)*Y1 + T*Y2
60 PLOT X,Y
70 GOTO 30
```

{NOTE: Line 70 is put here deliberately to keep the 'BASIC is beastly' brigade happy}

We can easily see that for values of T between 0 and 1 we have points on the line joining (X1,Y1) to (X2,Y2) and so it easy to produce a line-drawing program by placing lines 40-60 inside a T loop:

```
30 FOR T = 0 TO 1 STEP 1/128
70 NEXT T
```

An interesting exploration can be started by considering generalisations to the vector equation for 3 or more points.

As a starting point consider the form:

$\underline{r} = (1-t)^2\underline{a} + 2t(1-t)\underline{b} + t^2\underline{c}$

This gives $\underline{r}$ as the weighted average of 3 vectors. What is the locus of P for values of t between 0 and 1?

We can amend the previous program to explore this new equation quite easily:

```
25 INPUT "X3,Y3"; X3,Y3 : PLOT X3,Y3
35 U = 1 - T
40 X = U*U*X1 + 2*T*U*X2 + T*T*X3
50 Y = U*U*Y1 + 2*T*U*Y2 + T*T*Y3
```

Changing PLOT to LINE in line 60 and adjusting the starting point will give a continuous curve.

The form of the curve and its relations to the three defining points (X1,Y1), (X2,Y2) and (X3,Y3) gives rise to some

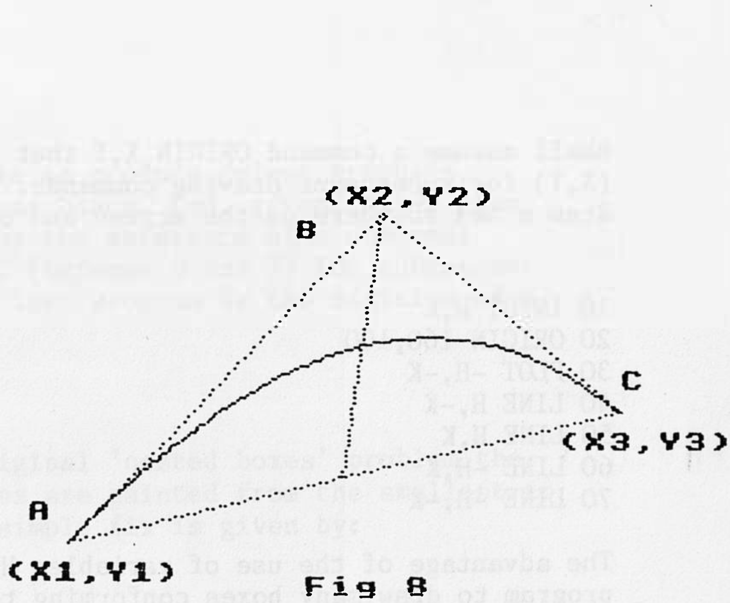

Fig 8

interesting speculation. Try varying the position of (X2,Y2) to see how the curve deforms.

From Fig 8 it looks as if the curve is tangent to $\underline{AB}$ at A - what happens if we try to differentiate the vector equation of the curve with respect to t?

$d/dt(\underline{r}) = -2(1-t).\underline{a} + (2-4t).\underline{b} + 2t.\underline{c}$

Thus, when t=0, we have:

$d/dt(\underline{r}) = -2.\underline{a} + 2.\underline{b} = 2.(\underline{b} - \underline{a}) = 2.\underline{AB}$

Try a similar argument for the tangent at C (X3,Y3). What about the point where $t=\frac{1}{2}$? Can you extend the idea to 4 or more points?

In fact this is far from just a framework for a good classroom discussion about mathematics - it is the heart of new 'real-world' application. These are called Bezier curves and were developed by P.Bezier of Renault cars for use in interactive computer-aided design (CAD).

Short programs, such as those just described, can easily be developed by a teacher away from the classroom (on site or at home) just as an OHP slide or a spirit-duplicator master might be prepared. They could, however, also be developed with a class as part of a lesson if the physical circumstances and resources permitted it and if the teacher has the confidence to take the associated risks.

In this short paper only a very few examples are possible but it is hard to think of any part of the mathematical curriculum that could not be enlivened and enriched by the use of quite short and simple programs. See, for example Professor David Johnson's "Explore Maths with your Micro", Heinemann 1983 and Oldknow & Smith for further examples.

Perhaps more significantly, though, the extra commands: PLOT X,Y and LINE X,Y are just about all that is necessary for pupils to take control of the computer. There is a limit to the fun that can be had out of making the machine print a table of square roots, but the fun that can be had from drawing pictures (preferably in colour) is virtually inexhaustable. A simple example is given by drawing boxes, starting from a simple routine to draw a square.

```
10 PLOT 100,100
20 LINE 200,100
30 LINE 200,200
40 LINE 100,200
50 LINE 100,100
```

Many micros with high-resolution graphics provide a useful, though not strictly necessary, command to move the origin from the bottom left-hand corner. In keeping with the machine-independent approach of this paper I shall assume a command ORIGIN X,Y that moves the origin to the screen point (X,Y) for subsequent drawing commands. Thus we can develop a routine to draw a box anywhere on the screen and of any dimension.

```
10 INPUT H,K
20 ORIGIN 160,100
30 PLOT -H,-K
40 LINE H,-K
50 LINE H,K
60 LINE -H,K
70 LINE -H,-K
```

The advantage of the use of variables H,K is that we can easily adapt the program to draw many boxes conforming to some relationship between H and K. Thus a simple generalisation is provided by:

```
10 FOR H = 5 TO 100 STEP 5
15   K = H/2
80 NEXT H
```

Which draws lots of boxes all of similar shape. But a rather more pleasing result can be obtained by drawing lots of boxes all of the same area.

```
15   K = 100/H
```

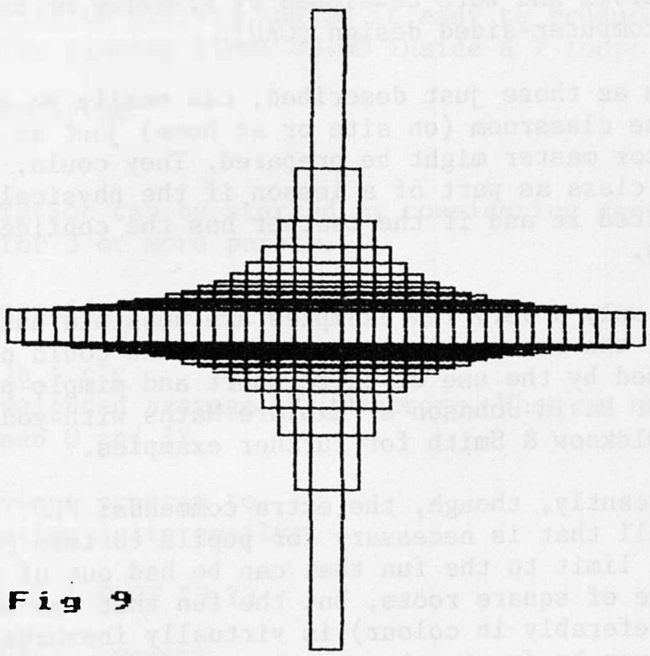

Fig 9

The repertoire can be extended in many ways but the addition of colour and the filling in of solid shapes are desirable extensions. The box program can easily be extended to produce a filled-in shape.

```
30 FOR Y = -K TO K
40   PLOT -H,Y
40   LINE H,Y
60 NEXT Y
70
```

Of course some micros do have commands to fill shapes such as rectangles or triangles. Again many machines are able to produce colour graphics, usually in the so-called Teletext shades: black, red, green, blue, cyan, yellow, magenta and white. If we suppose the existence of a command: COLOUR C - which selects colour code C (between 0 and 7) for subsequent graphics work then we can liven up the last program by the addition of a line to select random colours.

```
17   COLOUR INT(8*RND(1))
```

If this technique is applied to the original 'nested boxes' problem the program will need debugging as the boxes are painted from the smallest to the largest rather than vice versa. A simple fix is given by:

```
10 FOR H = 100 TO 5 STEP -5
```

Programs such as these and many others, such as those connected with 'curve-stitching' can prove particularly stimulating for the application of co-ordinates, negative numbers, algebra etc. even in pupils whose attainment in mathematics is otherwise quite low.

A pleasant and creative project which synthesises these ideas is to create sets of large printing characters in different fonts and styles.

4 Home computers.

In the majority of the literature on the relationships between computers and mathematical education the emphasis is on teaching and classroom practice. However 'home computers' are now in the same price range as bikes and, as we know, many pupils now have access to a computer. We can safely predict a rapid and dramatic rise in the numbers of pupils with such access. With the current generation of microcomputers using BASIC interpreters and needing co-ordinates, trig, random numbers, algebra etc. to perform their best tricks we have an extremely powerful 'maths. talking' phenomenon at our disposal. In the short term, at least, we have an opportunity to catch and develop pupils interest (particularly through the medium of graphics) and to make co-ordinates, trig, randomness, algebra, radians etc. all become tools of the trade with direct and obvious application. For the first time they actually enable pupils to apply mathematics in circumstances of their own devising.

Thus teachers may well be seen to have an added professional responsibility to encourage pupils in the pursuit of applications through which the need for some mathematical technique or other may well be prompted. This is a far more exciting and far-reaching use for a teacher's talents than writing sterile programs that provide no more stimulus than a conventional worksheet in an attempt to teach unmotivated parts of mathematics.

This may well involve teachers in spending extra time and energy outside the usual confines of the printed syllabus and examination schedule. In fact Papert has some very provoking things to say about our 'institutional' way of handling mathematics learning in "Mindstorms", Harvester Press, 1981 - but those who have had experience of working with children and computers in an informal atmosphere, akin to his 'Samba clubs', are left in no doubt about the quality of that experience for all concerned.

Two small personal experiences serve to illustrate this point. In the first an ex-student of mine was approached by one of his second year pupils who wanted to know whether his class could be taught matrices soon. He was trying to write a tank-battle program and he wanted to make objects move on the screen in the way they do on the up-to-date slot-machines. His elder brother, who was in the fourth year suggested that the kind of stuff, matrices ,that they were doing in maths might be of some help.

In the second I had to keep some sixth-formers amused during a slack period during an Easter conference. I did a short session on how to represent 3D objects in the computer and to display them in perspective and suggested some follow-up activities that they might try for themselves. However one lad was far more interested in a problem of his own to which he wanted to apply the 3D techniques we had been using. He wanted to be able to display the trajectory of a projectile as a curve in space shown in perspective. We beavered away at the equations of motion in 3D and after a while got some pleasing results. (Actually I suspect I was more pleased than he was) As he left he thanked me and said: "That was just what I wanted - with that trick I can put some realism into the computer golf game that I'm writing".

Again we must not underestimate the fascination that games have for children (of all ages). Such techniques as defining graphics characters to make up a plane or bomb, making it fly across or down the screen, detecting if a collision occurs, generating events randomly and adding sound effects can all involve much mathematical activity if only we are not too snobbish to admit (and encourage?) them.

5 Implications for teachers and training.

The DES and MEP staged a joint curriculum conference at Pendley Manor in Hertfordshire during May in which invited groups from three areas of the curriculum - geography, science and mathematics - were commissioned to produce papers on the future developments of their subject in the light of microelectronic developments. The mathematics paper is shortly to be published in the educational press and so I shall be sparing in taking quotations from it. Included in the introduction is:

1.3 We consider that mathematical programming should be a staple part of mathematics courses in the future. Just as a calculator has to be at hand if arithmetic is to be taught to best advantage so a computer will be needed if algebra, geometry, statistics and other branches of mathematics are to be taught to best advantage.

It then goes on to support the explicit study of algorithms within the mathematics curriculum and to confirm the value of short programs in the encouragement of learning mathematics. However, for all the strength of the arguments in favour of the style of computer use described in this paper, it would be quite wrong to underestimate the human problems in bringing about such a change. The key problem is in taking the first step - having the courage and finding the time to experiment with using the computer as a personal problem-solving tool at one's own mathematical level (consenting adults in private). If this does, as generally appears to be the case, convince a teacher of the value of the computer in extending one' enjoyment and learning of mathematics then the transfer of the implications of this experience to the learning of mathematics by pupils is far less of a problem.

Thus the active involvement of teachers in workshops and courses in which, through such experiences in problem-solving, confidence is gained in mathematical programming is seen as a major objective in the future professional training of mathematics teachers. To bring this about requires considerable investment in in-service training. Those involved with the pre-service training (B.Ed. and PGCE) of mathematics teachers need opportunities for such personal development no less than teachers working in school. To quote the concluding remarks of the Pendley Manor report:

"But if change on a substantial scale is to be encouraged this will require: ... substantial in-service training. In-service training should be provided by the co-ordinated use of all the available agencies. The type of training requires careful consideration. It will be clear from the discussion above that we do not give high priority to relatively advanced courses in programming, although there is certainly a place for them and, perhaps, at least one teacher in each school should have received such a course at some time. We attach high priority to the provision of lower level courses in mathematical programming and the use of computers, closely related to current courses, which will enable teachers to use machines in their day to day lessons with confidence - and which will help them to encourage their pupils to do the same."

Since this paper was first drawn up a further and very significant report has been published. In the DES discussion paper "Microcomputers and Mathematics in Schools" Trevor Fletcher HMI, until recently Staff Inspector for Mathematics and a member of the Cockcroft committee, provides many examples of some of the more exciting developments currently to be seen in schools and highlights many of the challenges which must be overcome if the potential of the microcomputer as a stimulus to mathematics learning is to be fully realised.

To finish, where I began, with Cockcroft it is worth noting that the report gives only modest reference to the kind of power that the micro may have for generating an environment for mathematics learning that we have not seen before. This must be seen as a cruel act of time (the report went to press before many of the developments that support these claims had a chance to make an impression) rather than a lack of vision on the part of the members. However there is a danger that Cockcroft may now become seen as providing 'chapter and verse' and that somehow any development that does not receive its positive support is to be treated with suspicion. Life, and technology, goes on after Cockcroft and we must hope that those in a position to influence events are sufficiently enlightened to recognise the power of the computer in encouraging the learning of mathematics.

REFERENCES:

Association of Teachers of Mathematics (ed. John Warwick)
Working Notes on Microcomputers in Mathematics
ATM 1982

Association of Teachers of Mathematics
Some Lessons in Mathematics with a Micro
ATM 1983

Burkhardt, Prof. Hugh
Microelectronics and the mathematics curriculum
DES/MEP Invited paper 1982 (unpublished)

DES/Microelectronics Education Programme
Report of the mathematics group (Pendley Manor)
MEP unpublished 1983

Fletcher, Dr.T.J.
Microcomputers and Mathematics in Schools
Department of Education & Science 1983

Great Britaiin
Mathematics Counts - Report of the committe of enquiry under the chairmanship of Prof. W. Cockcroft
HMSO 1982

Johnson, Prof. David
Explore Mathematics with your Micro
Heinemann 1983

Mathematics in Education and Industry (P.Couzens)
Programs for Mathematical Computing - 1
MEI 1983

Microelectronics Education Programme
Micros in the Mathematics Classroom
Shell Centre,Nottingham/ITMA 1982

Oldknow, Adrian
What if the Model Doesn't Work?
in Teaching Mathematics and its Applications
Vol.1, No.3, 1983

Oldknow, Adrian & Smith, Derek
Learning Mathematics with Micros
Ellis Horwood 1983

Papert, Seymour
Mindstorms
Harvester 1981

Straker, Anita
Microcomputers and Primary Mathematics
1982 (unpublished)

Listing of programs in BBC Basic

```
LIST
   10 FOR A = 5 TO 85 STEP 5
   20  AR = A*3.14159/180
   30  S = SIN(AR) : C = COS(AR)
   40  H1 = (4-S)/C : H2 = (7-C)/S
   50  PRINT;A;TAB(10);H1;TAB(25);H2
   60 NEXT A
```

```
LIST
    1 MODE 4
    5 E = 10
   10 FOR A = 5 TO 85
   20  AR = A*3.14159/180
   30  S = SIN(AR) : C = COS(AR)
   40  H1 = (4-S)/C : H2 = (7-C)/S
   50  PLOT 69,E*A,E*H1
   55  PLOT 69,E*A,E*H2
   60 NEXT A
```

```
LIST
   10 DEF FNA(X) = EXP(-X*X)
   20 INPUT "XLOW,XHIGH: " XL,XH
   30 INPUT "YLOW,YHIGH: " YL,YH
   40 XR = XH - XL : YR = YH - YL
   45 MODE 4
   50 SW = 1279 : SH = 1023
   60 FOR XS = 0 TO SW STEP 4
   70  X = XL + XS*XR/SW
   80  Y = FNA(X)
   90  YS= SH*(Y-YL)/YR
  100  PLOT 69,XS,YS
  110 NEXT XS
```

```
LIST
   10 DEF FNA(X) = EXP(-X*X)
   20 INPUT "XLOW,XHIGH: " XL,XH
   30 INPUT "YLOW,YHIGH: " YL,YH
   40 XR = XH - XL : YR = YH - YL
   45 MODE 4
   50 SW = 1279 : SH = 1023
   55 MOVE 0,SH*(FNA(XL)-YL)/YR
   60 FOR XS = 0 TO SW STEP 4
   70  X = XL + XS*XR/SW
   80  Y = FNA(X)
   90  YS= SH*(Y-YL)/YR
  100  DRAW XS,YS
  110 NEXT XS
```

```
LIST
    5 MODE 4
   10 INPUT N
   20 A = 6.283/N
   30 H = 640 : K = 512
   40 R = 400
   50 MOVE H+R,K
   60 FOR I = 1 TO N
   70  X = H + R*COS(I*A)
   80  Y = K + R*SIN(I*A)
   90  DRAW X,Y
  100 NEXT I
```

```
LIST
    5 MODE 4
   10 DEF FNX(T) = COS(T)
   15 DEF FNY(T) = SIN(2*T)
   17 N = 40
   20 A = 6.283/N
   30 H = 640 : K = 512
   40 R = 400
   50 MOVE H+R*FNX(0),K+R*FNY(0)
   60 FOR I = 1 TO N
   65  T = I*A
   70  X = H + R*FNX(T)
   80  Y = K + R*FNY(T)
   90  DRAW X,Y
  100 NEXT I
```

```
LIST
    5 MODE 4
   10 INPUT "X1,Y1: " X1,Y1
   11 PLOT 69,X1,Y1
   20 INPUT "X2,Y2: " X2,Y2
   22 PLOT 69,X2,Y2
   30 INPUT T
   40 X = (1-T)*X1 + T*X2
   50 Y = (1-T)*Y1 + T*Y2
   60 PLOT 69,X,Y
   70 GOTO 30
```

```
LIST
    5 MODE 4
   10 INPUT "X1,Y1: " X1,Y1
   11 PLOT 69,X1,Y1
   20 INPUT "X2,Y2: " X2,Y2
   22 PLOT 69,X2,Y2
   25 INPUT "X3,Y3: " X3,Y3
   26 PLOT 69,X3,Y3
   30 FOR T = 0 TO 1 STEP 1/128
   35  U = 1 - T
   40  X = U*U*X1 + 2*U*T*X2 + T*T*X3
   50  Y = U*U*Y1 + 2*U*T*Y2 + T*T*Y3
   60  PLOT 69,X,Y
   70 NEXT T
```

```
LIST
    5 MODE 1
   10 MOVE 400,400
   20 DRAW 600,400
   30 DRAW 600,600
   40 DRAW 400,600
   50 DRAW 400,400
```

```
LIST
    5 MODE 1
   10 FOR H = 20 TO 400 STEP 20
   15  K = 8000/H
   20  VDU 29,640;512;
   30  MOVE -H,-K
   40  DRAW  H,-K
   50  DRAW  H, K
   60  DRAW -H, K
   70  DRAW -H,-K
   80 NEXT H
```