TECHNOLOGY AND MATHEMATICS EDUCATION

Computer Environments for the Learning of Mathematics

David Tall, Warwick

Computer software for the *learning* of mathematics, as distinct from software for *doing* mathematics, needs to be designed to take account of the cognitive growth of the learner which may differ significantly from the logical structure of the formal subject. It is therefore of value to begin by considering cognitive aspects relevant to the use of computer technology before the main task of focusing on computer environments and their role in the learning of mathematics.

The growth of (mathematical) knowledge

The human brain is remarkable in its ability to store and retrieve complex information, but it is correspondingly limited in the quantity of independent pieces of data that may be manipulated in conscious short-term memory. To minimise the effects of these limitations, one method is to 'chunk' the data by using an appropriate representation which is more easily manipulable. For instance, standard decimal notation is a compact method of representing numerical quantities of any size, with corresponding routines for manipulation; algebraic notation can be used to formulate and manipulate certain types of data for problem-solving; graphical representations are appropriate for other tasks, such as representation of complex data in a single gestalt.

Traditional mathematics often consists in performing algorithms using these representations, minimising the cognitive strain by routinising the procedures so that they become automatic and require less conscious memory to perform. A more subtle transformation also occurs in which the symbols used to evoke a mathematical process begin to take on a life of their own as mental objects, so that processes become encapsulated as objects. Thus counting using the number words gives the numeric symbols a related meaning as numbers, the process of addition becomes the concept of sum, repeated addition becomes product, and so on. This long-term cognitive process in which procedures are routinised to become more compressed and

Published in ***Didactics of Mathematics as a Scientific Discipline***, (eds. R. Biehler, R. W. Scholz, R. Sträßer & B. Winkelmann), Kluwer Academic Publishers: Dordrecht, 1993, pp. 189–199.

then encapsulated as mathematical objects in their own right is referred to by Piaget and subsequent authors as *vertical* growth, in contrast to the *horizontal* growth of relationships between different representations.

Both vertical and horizontal growth impose difficulties on the individual. Vertical growth requires ample time for familiarisation with the given process to enable it to be interiorised and also for the cognitive re-organisation necessary during encapsulation of process as object. Horizontal growth requires the simultaneous grasping of two or more different representations and the links between them, which is likely to place cognitive strain on short-term memory resources.

These difficulties may be alleviated in various ways by using a computer environment to provide support. Software may be designed to carry out some of the processes, leaving the learner to concentrate on others chosen to be the selected focus of attention. The sequence of learning in vertical growth may be modified by providing environments which allow the study of higher level concepts in an intuitive form before or at the same time as they are constructed through encapsulation. Horizontal linkages between different representations may be programmed so that the individual operates on one representation and can see the consequences of this act in other linked representations. Moreover, because the computer can be programmed to respond in a pre-ordained manner, it can provide an environment in which the learner can explore the consequences of selected actions to predict and test theories under construction.

The computer tas a predictable environment for learning

Skemp (1979, page 163) makes a valuable distinction between different modes of building and testing conceptual structures (table 1).

	Reality Construction	
Mode	Reality Building	Reality testing
1	from our encounters with actuality: *experience*	against experience of events in actuality: *experiment*
2	from the reality of others: *communication*	comparison with the reality of others: *discussion*
3	from within, by formation of higher order concepts, by extrapolation, imagination, intuition: *creativity*	comparison with one's own existing knowledge and beliefs: *internal consitencey*

Table 1 : Modes of building and testing personal constructions

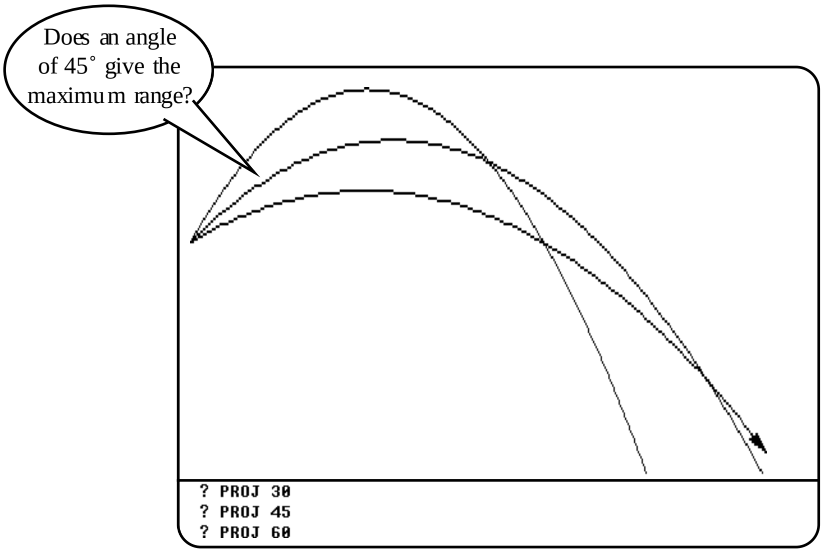
The introduction of computer technology brings a new refinement to this theory. Whereas mode 1 is seen as the individual acting on and experimenting with materials that are largely passive, a computer environment can be designed to *re*-act to the actions of the individual in a predictable way. This new form of interaction extends Skemp's theory to four modes (Tall 1989) where building and testing environments are :

- *Inanimate* : the stimuli come from objects in actuality which the individual may also be able to manipulate,
- *Cybernetic* : the stimuli come from systems which are set up to react according to pre-ordained rules,
- *Interpersonal* : the stimuli come from other people,
- *Personal* : the stimuli are from the individual's own cognitive structure.

The new cybernetic mode of building and testing concepts affords rich possibilities for the learning of mathematics.

Microworlds

The term *microworld* was originally used by Papert to describe 'a computer-based interactive learning environment where the pre-requisites are built into the system and where learners can become active, constructing architects of their own learning' (Papert, 1980, p.117) Initially the term 'microworld' was used specifically for programming environments (often in the computer language Logo). For instance, the program *Newton* (Pratt, 1988) is a microworld designed so that turtles move according to Newton's laws, allowing investigations of a variety of topics including motion under a central force. In figure 1 the student has designed an experiment to model an object being projected from a point above a plane to investigate the angle that gives the maximum range; it turns out differently from the expected 45˚.



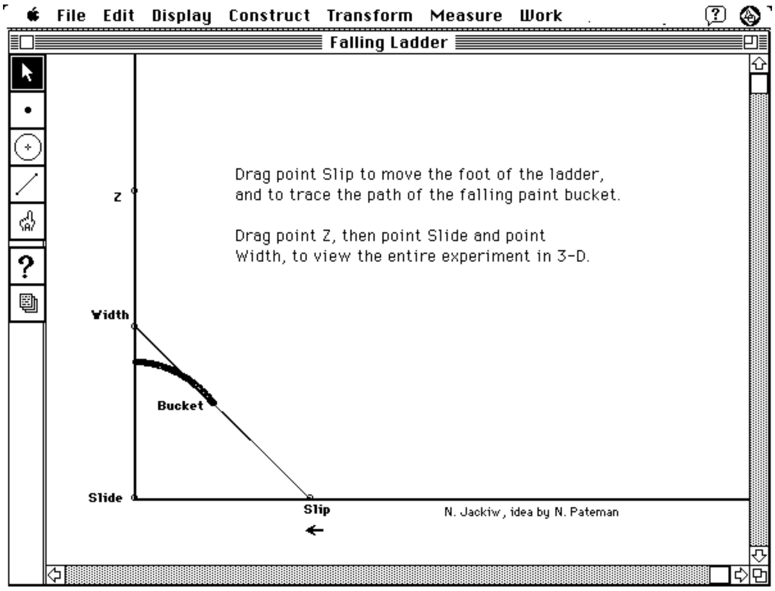
```
? PROJ 30
? PROJ 45
? PROJ 60
```

Figure 1 : Investigating the range of a projectile for different angles of projection

Such an environment provides facilities to construct ways of formulating and testing conjectures. In the early stages Papert considered such environments to encourage what he termed ‘Piagetian learning’, or ‘learning without a curriculum’, or ‘learning without being taught’ (Papert, 1980, p. 7). Children are often highly creative within such environments, but ‘powerful ideas’, particularly vertical growth of concepts, do not readily occur spontaneously and long-term curriculum objectives require external guidance and support.

Environments for enactive and visual manipulation

More sophisticated computer environments have been designed in recent years which take advantage of flexible computer interfaces. Geometric software such as *Cabri Geometre* or *The Geometer’s Sketchpad* allows figures to be drawn with specific relationships defined, such as a given point must always lie at the midpoint of a given line-segment, or be constrained to lie on a given circle. Then the figure may be pulled around enactively retaining all the defined constrains to investigate possible consequent relationships. Figure 2 shows a model of a bucket on a ladder set against a vertical wall and sketches the path taken by the bucket as the ladder slides.



Drag point Slip to move the foot of the ladder, and to trace the path of the falling paint bucket.

Drag point Z, then point Slide and point Width, to view the entire experiment in 3-D.

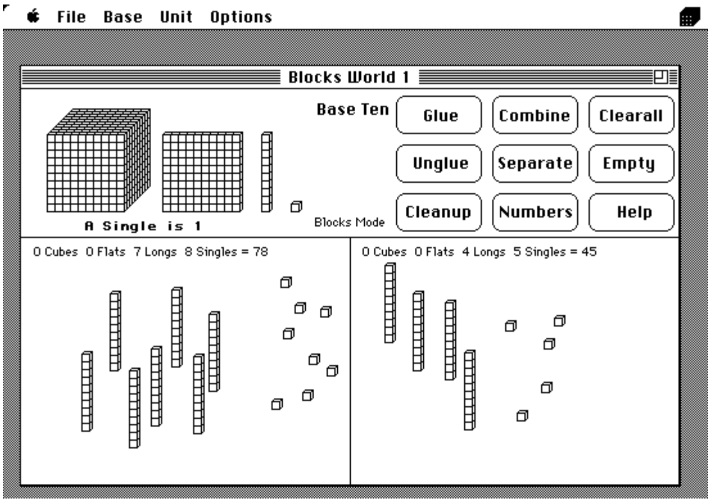
N. Jackiw, idea by N. Pateman

Figure 2 : The locus of a point on a line of fixed length constrained with its endpoints on two perpendicular lines

Such software may be used to gain enactive visual support in conjecturing and testing geometric theorems, enabling students to take an active part in the construction of their own knowledge, though once again, the formal proof structure of geometry will need separate consideration.

Multiple linked representations

Computer environments can be set up to link different representations of the same concept. The *Blocks Microworld* of Thompson (1992) is designed to link screen representations of Dienes' multibase blocks to numerical representations (figure 3). In the top right of the window are representations of different units in base ten, comprising a single, long (10 singles in a line), flat (10 longs in a square), and block (10 flats to make a larger cube). As the user selects one of these and pulls a copy to the lower part of the screen to build up collections of blocks, the corresponding numerical display is simultaneously updated. If the blocks in the figure representing 78 and 45 are combined by removing the vertical separator between them, the resulting collection of 11 longs and 13 singles can be re-organised by the learner to give 1 flat, 2 longs and 3 singles (123).



Base Ten

A Single is 1

Blocks Mode

0 Cubes 0 Flats 7 Longs 8 Singles = 78

0 Cubes 0 Flats 4 Longs 5 Singles = 45

Figure 3 : Manipulating Dienes' blocks on-screen, with the corresponding linked numerical representation

This environment may be used to give a direct link between physical experience and the formal symbolic notation, allowing children to explore their own algorithms for as well as giving meaning to the formal routines for addition and subtraction.

The principle of selective construction

What has been exemplified in all the environments so far described is the way in which the software can be programmed to carry out internal algorithms, leaving the learner free to explore other aspects. This can occur in horizontal growth of knowledge where the learner builds links between different representations, but it is even more powerful in vertical growth. Whereas a traditional development would almost always require the learner to become familiar with a given process and routinise it before beginning to consider the consequences, computer environments may carry out the processes and allow the user to explore the resultant concepts either before, after, or at the same time as, the processes. This ability to reorganise the curriculum to allow the learner to focus on one aspect of cognitive growth whilst the computer carries out others, I term the *principle of selective construction*.

In carrying out such a principle, it is important to consider the concept imagery that it may generate in the learner and the type of insight that such interaction may bring. Tall & Winkelmann (1988) described three different kinds of insight:

> external, analogue, specific.

External insight occurs when the user has no idea what is going on inside the software, but has knowledge which allows him or her to check that the results are sensible, *analogue* insight occurs when the user has an idea of type of algorithm in use, and *specific* insight is when the user is fully aware of how the software is programmed.

Specific insight into computer software is rarely possible or even desirable for the majority of computer users, but it is helpful for the student to have at least external insight or, preferably, analogue insight. The concept image of a cybernetic system constructed in the mind of the user is likely to be idiosyncratic and a teacher has a fundamental role to play through guidance and discussion. Tall (1989) describes the combination of a human teacher as guide and mentor using a computer environment for teaching, pupil exploration, and discussion as the *Enhanced Socratic Mode* of teaching and learning. It combines the *interpersonal* interactions between student and teacher, the *cybernetic* interactions with the computer environment to give an independent source of consistent evidence, and the *personal* constructions of the learner in building and relating together the different parts of the knowledge structure.

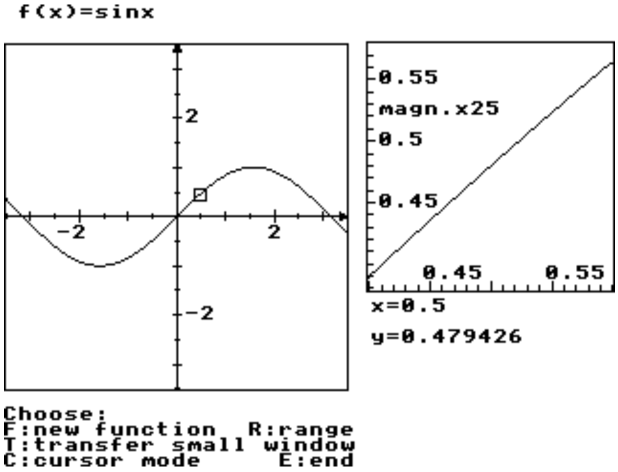
Generic Organisers

Ausubel et al (1968) defined an *advance organiser*, as

> ‘Introductory material presented in advance of, and at a higher level of generality, inclusiveness, and abstraction than the learning task itself, and explicitly related both to existing ideas in cognitive structure and to the learning task itself ... i.e. bridging the gap between what the learner already knows and what he need to know to learn the material more expeditiously.’

Such a principle requires that the learner *already has the appropriate higher level cognitive structure* available to him or her. In situations where this may be missing, in particular when moving on to more abstract ideas in a topic for the first time, a different kind of organising principle will be necessary. To complement the notion of an advanced organiser, a *generic organiser* is defined to be an environment (or microworld) which enables the learner to manipulate *examples* and (if possible) *non-examples* of a specific mathematical concept or a related system of concepts. (Tall, 1989). The intention is to help the learner gain experiences which will provide a cognitive structure on which the learner may reflect to build the more abstract concepts. I believe the availability of non-examples to be of great importance, particularly with higher order concepts such as convergence, continuity or differentiability, where the concept definition is so intricate that students often have difficulty dealing with when it fails to hold.

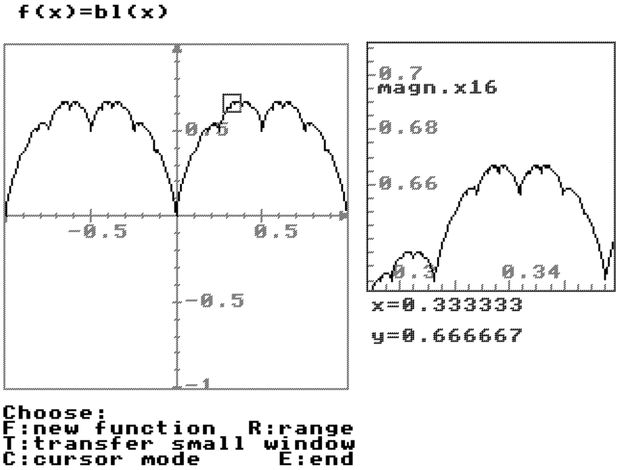
A simple instance of a generic organiser embodying both examples and non-examples is the *Magnify* program from *Graphic Calculus* (Tall *et al*, 1990) designed to allow the user to magnify any part of the graph of a specified function (figure 4).



```
x=0.5
y=0.479426
```

Figure 4 : A ‘locally straight’ graph

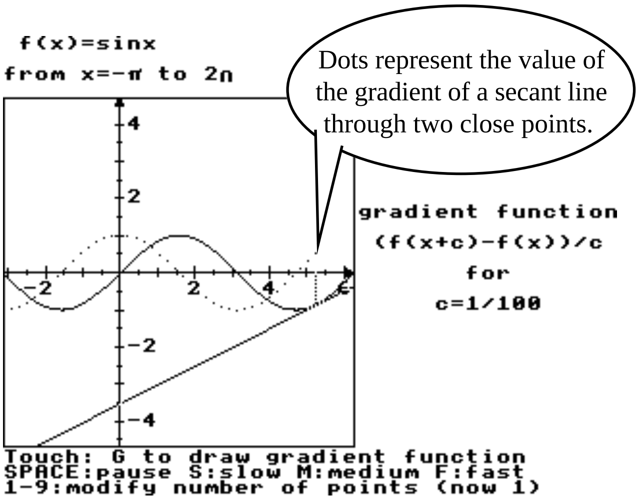
Tiny parts of certain graphs under high magnification eventually look virtually straight and this provides an anchoring concept for the notion of differentiability. Non-examples in the program are furnished by graphs which have corners, or are very wrinkled that they never look straight, providing anchoring concepts for non-differentiability.



```
x=0.333333
y=0.666667
```

Figure 5 : A highly wrinkled graph

The gradient of a ‘locally straight’ graph may now be seen graphically by following the eye along the curve, or a piece of software may be designed which traces the gradient as a line through two close points on the graph moves along in steps (figure 6).



```
f(x)=sinx
from x=-π to 2π
gradient function
(f(x+c)-f(x))/c
for
c=1/100
Touch: G to draw gradient function
SPACE:pause S:slow M:medium F:fast
1-9:modify number of points (now 1)
```

Dots represent the value of the gradient of a secant line through two close points.

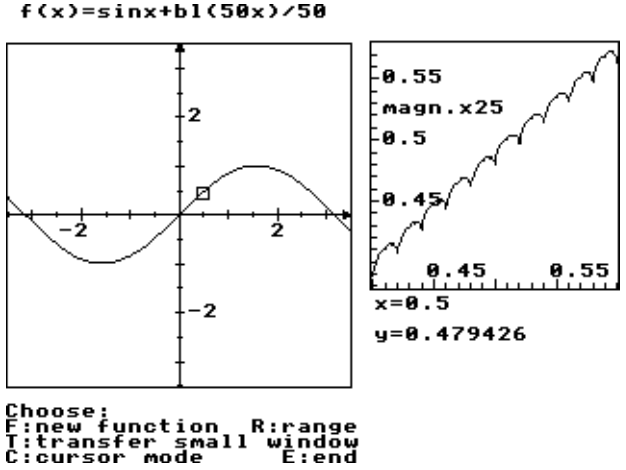
Figure 6 : Drawing the gradient (dotted) of sin*x*

In this way, a student with some experience of the shape of trigonometric curves will be able to conjecture that the derivative (gradient) of sin*x* is cos*x* from the shape of the dotted gradient, even though the manipulation of trigonometric formulae and the formal notion of limit is at present beyond his or her capacity.

Generic difficulties

Given the human capacity for patterning, and the fact that the computer model of a mathematical concept is bound to differ from the concept in some respects, we should be on the lookout for abstraction of inappropriate parts of the model. Visual illusions in interpreting graphs have been documented by Goldenberg (1988) and by Linn & Nachmias, (1987). In the latter case, one third of the pupils observing a cooling curve of a liquid on a computer VDU interpreted the pixellated image of the graph as truly representing what happened to the liquid – constant for a time, then suddenly dropping a little (to the next pixel level down).

Working with older students, the inadequacy of the representation may prove to be an advantage. It can be source of discussion that the jagged pixellated imagery does not represent the true conceptualisation in the mind, encouraging the student to make personal mental constructs of a more platonic form of the theory. For isnatnce, free play with a gradient-drawing program may lead the student to think that all reasonable looking graphs are differentiable, but this view may be challenged by being confronted by figure 7. This graph looks very similar to that in figure 4, but under high magnification, the wrinkles produced by the tiny added blancmange become apparent. Simple visualisation at a fixed scale is therefore inadequate: two graphs may *seem* to be similar at one level, yet at a deeper level, one is differentiable everywhere and the other nowhere. In this way the generic organiser reveals itself as only a step along the path of cognitive growth. The student progressing to more formal study has the opportunity to develop flexible concept imagery showing the necessity for more subtle symbolic representation of the mathematics, whilst the student who is only using the calculus in its applications has at least an intuitive appreciation of the possible theoretical difficulties.



f(x)=sinx+bl(50x)/50

x=0.5

y=0.479426

Figure 7 : A graph with a difference

Reflections

In considering the way in which computer environments can be used in the learning of mathematics, we see the possibility of providing cybernetic environments which react in a predictable manner to help the learner build and test their own mental constructions. The computer can carry out internal procedures, allowing the learner to focus on other facets of importance in the cognitive growth of mathematical knowledge. This can help develop a concept image of higher order concepts in a different sequence from the traditional method of routinisation and encapsulation. It must be noted that the mental objects may not have the same structure as is given by traditional learning sequence and that such exploration may give gestalts which do not link directly to the sequence of definitions and logical deductions in the formal theory. However, insights are possible for students who might not attain such a level in a traditional approach while those who are able to move to higher levels may have more appropriate concept imagery available to give a more rounded mental picture of the theory. The software described in this chapter invariably needs to be embedded in a wider conceptual context where the powerful ideas are made the explicit focus of attention. This is usually provided by prepared materials or by the teacher as mentor, although a solution has long been sought in which the computer itself can play the guiding role in a more intelligent manner. Meanwhile, nteractive video is beginning to provide flexible environments in which the study guide offers the student deeper levels of information as required, with interactive animated graphics and flexible computer environments of the type described in this chapter. As technology grows more sophisticated, such developments are likely to play an increasing role in the learning of mathematics.

References

Ausubel, D. P., Novak, J. D. & Hanesian, H., 1968: *Educational Psychology: A Cognitive View* (2nd Edition 1978), Holt, Rinehart & Winston.

Cabri Géomètre 1987: IMAG, BP 53X, Université de Grenoble.

Goldenberg, P., 1988: 'Mathematics, Metaphors and Human Factors: Mathematical, Technical and Pedagogical Challenges in the Educational Use of Graphical Representations of Functions', *Journal of Mathematical Behaviour*, 7, 2, 135-173.

Linn, M. C. & Nachmias, R., 1987: 'Evaluations of Science Laboratory Data: The Role of Computer-Presented Information', *Journal of Research in Science Teaching* 24, 5, 491—506.

Papert, S., 1980: *Mindstorms*, Harvester Press, Brighton.

Pratt, D., 1988: 'Taking a dive with Newton', *Micromath*, 4, 1, 33–35.

Skemp, R. R., 1979: *Intelligence, Learning and Action*, Wiley.

Tall, D. O., 1989: 'Concept Images, Generic Organizers, Computers & Curriculum Change', *For the Learning of Mathematics,* 9, 3, 37–42.

Tall, D. O. & Winkelmann, B., 1988: 'Hidden algorithms in the drawing of discontinuous functions', *Bulletin of the I.M.A.*, 24, 111-115.

Tall, D. O., Blokland, P. and Kok, D., 1990: *A Graphic Approach to the Calculus*, Sunburst, Pleasantville, USA, also published in German as *Graphix* by CoMet Verlag, Duisburg, and in French as *Graphe*, by Nathan, Paris.

The Geometer's Sketchpad, 1992: Visual Geometry Project, Key Curriculum Press.

Thompson, P., 1992: *Blocks Microworld*, University of California, San Diego.