Recent developments in the use of the computer to visualize and symbolize calculus concepts

David Tall

Mathematics Education Research Centre
University of Warwick
Coventry CV4 7AL, U.K.

In recent years the growing availability of personal computers with high resolution computer graphics has offered the possibility of enhancing the visual mode of thinking about mathematics in general and the calculus in particular. Interactive computer software can be used to give insight to students, teachers and professional mathematicians in a manner that could not have been imagined a decade ago. Nevertheless, the technology brings with it the challenge to re-assess what is important in the curriculum and this is proving to be the more difficult task for professional mathematicians with a rich experience of pre-computer technology. This paper therefore begins by considering different ways in which we process information and focuses on the need to complement sequential/deductive thinking with a global grasp of interrelationships. It then concentrates on different aspects of the calculus from numerical, symbolic and graphical viewpoints and places recent developments in visualization using the computer within this broad perspective.

Human information processing and the computer

Individuals develop markedly different ways of mathematically thinking. Whilst some professional mathematicians will accept only those things which they can deduce logically step by step from carefully specified axioms, others demand an overall framework in which they can see a network of interrelationships between the concepts. The former viewpoint is a necessary pre-requisite for the *formalization* of mathematical concepts, the latter is invaluable for their *development*, both in mathematical research and in mathematical education. Yet, despite the need for both modes of thought, traditional mathematics teaching – especially at the higher levels – is usually more concerned with the sequential, deductive processes of the former rather than the holistic, predictive ones of the latter.

Published in *The Laboratory Approach to Teaching Calculus* ,
M.A.A. Notes Vol. 20, 15–25 (1991).

Mankind’s success as a species is enhanced through the invention and use of tools to extend human capabilities. Many of these tools compensate for limitations in evolutionary design, be it spectacles to improve vision, the telephone to extend hearing and communication, or the supersonic jet to give the power of flight. In *The Psychology of Learning Mathematics* (1971), Skemp made the perceptive comment that humans have built-in loudspeakers (voices) but not built-in picture projectors, causing the fundamental mode of human communication to be verbal rather than pictorial. He demonstrated how a geometric proof could be presented in pictures which convey the same information as a verbal-algebraic proof, in a form that may be more insightful for many individuals. His purpose was not to show a preference of one form of communication over another, but to question the dominance achieved by algebraic symbolism and to examine the contribution made by visual symbols.

Not long ago the computer was a purely sequential, symbolic device, with the operator punching in a sequence of symbols and receiving the response in a similar form. But recently developed graphical interfaces now allow the user to communicate with pictures, providing a tool that compensates for the human lack of a built-in picture projector. Modern computer operating systems are replacing symbolic typing of commands by visual pointing at icons to simplify the interface with the computer. Furthermore the computer is able to accept input in a variety of ways, and translate it flexibly into other modes of representation, including verbal, symbolic, iconic, graphic, numeric, procedural. It therefore gives mathematical education the opportunity to adjust the balance between various modes of communication and thought that have previously been biased toward the symbolic and the sequential.

A style of learning that uses the complementary powers of sequential/linear thought processes on the one hand and global/holistic processes on the other is said to be *versatile* (Brumby 1982). The computer with suitable software is a powerful tool to encourage versatile learning (Tall & Thomas 1988b).

The implications of the computer in curriculum sequencing

Traditional mathematics usually introduces learners to sequential techniques and develops each one in a comprehensive way before introducing the learner to higher order concepts. Research has shown that the absence of a broad enough range of experience can lead to the abstraction of a false principle that later proves extremely difficult to eradicate, for instance “subtraction makes smaller” (an implicit property of counting numbers that causes conflict when negative numbers are encountered) or “multiplication makes bigger” (implicitly true for whole numbers but false for fractions). In these examples the hierarchy of concepts implies that one cannot avoid working in the limited context before broadening it (counting numbers before negatives, integers before fractions). But in the case of more advanced concepts it may be possible to reorganise the sequencing of concepts, using the computer to provide a rich environment in which *formally* complex concepts can be met *informally* at a much earlier stage.

A good example is the introduction of the derivative in the calculus. Formally this requires the notion of the limit of $\frac{f(x+h)-f(x)}{h}$ as h tends to zero, so logically the introduction of the derivative must be preceded by a discussion of the meaning of a limit. To make the notion of limit simpler, the limiting process is first carried out with x fixed and only later is x allowed to vary to give the derivative as a function, giving the following sequence of development:

(1) Notion of a limit, graphically, numerically and/or symbolically,

(2) For fixed x, consider the limit of $\frac{f(x+h)-f(x)}{h}$ as h tends to zero,

(3) Call this limit $f'(x)$ and allow x to vary to give the derivative as a function.

If this sequence is followed, however “intuitively” it is done, then there are cognitive obstacles at every stage, which are well-known to every perceptive classroom teacher and have deeper connotations which have been detailed elsewhere (e.g. Cornu 1981, Tall & Vinner 1981, Orton 1983a,b).

One may conjecture that the interposition of a long chain of sub-tasks in building up a concept may *impede* conceptualization, because properties that arise in the restricted contexts en route lead to serious cognitive obstacles. There are two ways to attempt to solve this dilemma: one is to research the cognitive obstacles so that they may be addressed appropriately at a later stage, the other is to use a “deep-end” approach (Dienes 1960) in which the whole concept is met early on in a rich, but more informal, context designed to offer a cognitive foundation for a more coherent concept image.

In the case of the derivative the “deep-end” approach can be done by first considering the informal idea of the gradient of a curved graph through magnification. It is based on the idea that a differentiable function is precisely one which looks “locally straight” when a tiny portion of the graph is highly magnified (Tall 1982). The limiting process is then an *implicit* idea used as a tool, rather than the *explicit* focus of study. “Local straightness” is the generative idea of this graphic approach to the calculus, which proves to be both a *cognitive* and a *mathematical* foundation for the theory.

It is instructive to note that in 1981-2, when the Second International Mathematics Study considered the way that calculus was taught, in the report emanating from Ontario (McLean *et al* 1984), not one respondent mentions an idea equivalent to a “locally straight” approach, the favoured methods being the “intuitive” geometrical idea of a “chord approaching a tangent” or numerical or algebraic limiting processes. The “locally straight” idea has some precedents, for example in curricula in Holland and Israel, but in each case the absence of appropriate computer software in the early eighties prevented it from being fully realized in practice. It also exists in a more visionary way in non-standard analysis (e.g. Keisler 1976) in which an “infinite magnification” is used to reveal a differentiable function as a infinitesimal straight line segment. The computer brings a practical (though inexact) model of this abstract theory.

Numerical, Symbolic & Graphical Representations of Calculus Concepts

The first computer applications in the calculus were numerical - using numerical algorithms to solve equations, calculate rates of change (differentiation), cumulative growth (integration and summation of series) and the solution of differential equations. All of these can be performed in a straightforward, but sometimes inaccurate, manner using simple algorithms, and then improved dramatically by using higher order methods. However, unless interpreted imaginatively, tables of numerical data may give little insight and the concentration on the calculations may tend to obscure the underlying pure mathematical theory.

One method to improve matters is to engage the student in appropriate programming activities so that the act of programming requires the student to think through the processes involved. This may be done in any one of a number of computer languages, but it is preferable in a language that encourages the use of higher level of mathematical thought. Thus unstructured BASIC may allow numerical data to be calculated, but a structured language which allows the development of functional concepts is likely to be more useful. It is one thing to be able to calculate the numerical area under a graph $y=f(x)$ from $x=a$ to $x=b$ using the mid-ordinate rule for strips of width h. But if the language allows the specification of the area as a function $area(f, a, b, h)$ of the function f, the endpoints a,b and the strip-width h, then many further constructions are possible. For instance, the area $s(n)$ under n equal width strips under $f(x)=x^2$ from $a=0$ to $b=1$ is

$s(n) = area\ (x\hat{}2, 0, 1, 1/n)$

and its value may be studied as *n* increases. Or the area-so-far function *asf*(*x*) under *f*(*x*)=1/*x* from *x*=1, taking strips width 0.1 is

$asf(x) = area(1/x, 1, x, 0.1)$

which may be studied as a function of *x*.

This may be performed in a structured form of BASIC (such as BBC BASIC), which allows such functions to be defined (in an appropriate syntax, such as *FNarea*(*f$*,*a*,*b*,*h*) for the area function under the graph represented by the string expression *f$* from *a* to *b* with step *h*). However, this language has frustrating technical limitations, such as the need to specify expressions as strings of characters (which may include the name of a function which itself can be given by a multi-line procedure) and the lack of mathematical data types other than numbers and character strings. TRUE BASIC has many marvellous facilities, but lacks even the EVALuation operator of BBC BASIC which allows a string such as "*x*^2-2" to be evaluated for a given value of *x*. On the other hand, the Interactive SET Language, ISETL (Schwarz *et al* 1986) is particularly conducive to such programming activities, being explicitly designed to allow manipulation of such mathematical constructs as sets, ordered sets, functions, relations, quantifiers, and so on.

Apart from programming, two quite distinct strands of development have taken place with computers. One concentrates on symbolic manipulation, using software to manipulate algebraic expressions and carry out symbolic differentiation and integration. This initially required the power of main-frames to cope with the recursive routines that were necessary and led to such symbolic manipulation systems such as MACSYMA, Maple, Reduce and SMP (Van Hulzen & Calmet, 1983). Implementations have subsequently appeared on micros, allowing various levels of sophistication, including Maple, Reduce, MuMath (Stoutemyer *et al* 1983), Derive (Stoutemyer & Rich, 1989). More specialized programs have been designed to trace through various calculus techniques (for example Maths Workshop's "Symbolic Calculus" for the BBC computer implements recursive techniques for differentiation of combinations of standard functions, as well as offering *ad hoc* calculation of integrals). Other symbolic systems are also beginning to incorporate such step-by-step tracing of algorithms such as the differentiation of a composite function. However, whilst this gives the user a greater chance of understanding how to carry out the algorithms of formal differentiation, it does not help give insight into what a derivative *is*.

A second approach uses high-resolution graphics on micro-computers to translate the numerical methods into graphical representations. During the nineteen eighties, the number of such software packages has grown extensively (e.g. Kemeny 1986, Tall 1986a, Bach 1988, Tall *et at* 1990). Some early packages simply programmed calculus ideas numerically on the computer, but the later ones take the level of sophistication of the learner into account and attempt to present the ideas in a meaningful way.

These two strands accentuate the two different ways of mathematical thinking described earlier. The symbolic manipulator offers sequential manipulation to lead to a symbolic result, the graphical programs represent a vast amount of numerical data pictorially, giving the user the opportunity to gain an overall grasp of the information. Some symbolic manipulators offer the facility to trace the symbolic methods step by step, to see how the computer software progresses through the solution. The graphical representation, on the other hand, allows the user to see the numerical solution build up in real time.

In practice these strands are *complementary*, and each has different strengths. For example, not all differential equations have symbolic solutions, so numerical methods are essential. And even where symbolic methods are available, they often need geometric interpretation. In Tall (1986c) I gave the following example of a differential equation which was set on a national mathematics examination paper in the U.K.:

$$y\frac{dy}{dx}\ sec\ 2x = 1\text{-}y^2.$$

It is easily “solved” by separating the variables to get

$$\frac{y}{1\text{-}y^2}\ dy = cos\ 2x\ dx \qquad (*)$$

and integrated to give the “general solution”

$$\text{-}\tfrac{1}{2}\ ln|1\text{-}y^2| = \tfrac{1}{2}\ sin\ 2x + c,$$

but what does this *mean*? By regarding the differential equation (*) as specifying the direction of the tangent vector (dx,dy) to the solution curve through any point (x,y) enables a “direction field” of short line segments to be drawn in the appropriate directions through an array of points in the plane (figure 1).

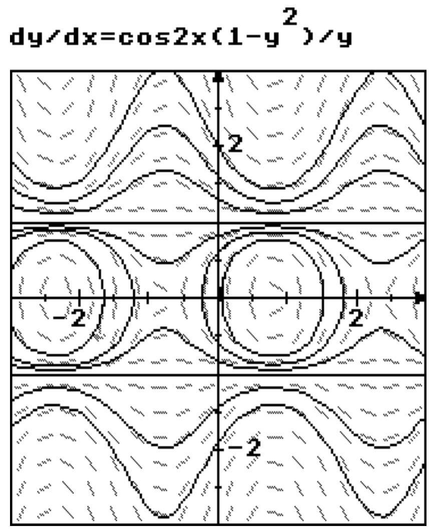

figure 1

It can be seen that some solutions are closed loops whilst others may be conceived as functions in the form $y=f(x)$. The symbolic solution in this case is of little value without a graphical interpretation of its meaning whilst the graphical interpretation alone lacks the precision of the symbolism.

The need for these complementary strands is consonant with empirical research into the success of students in coping with a first year university course in real analysis. Robert & Boschet (1984) hypothesized that there is a better prospect of successful mathematical learning for the student with knowledge (however imperfect) in many contexts than for a student with knowledge in one context. They found that the weakest performances over the year were by students having initial knowledge in very few contexts (usually numerical) whereas the more successful students also had initial knowledge in graphical and symbolic contexts. They found their hypothesis verified repeatedly in empirical experiments, and suggested that the crucial difference appeared to lie not in the mere existence of the prior knowledge but in the difference between two very different ways of thinking: the reductive effect of functioning in a single context as against the liberating effect of bringing several different ways of seeing the same problem from different viewpoints.

Research into the effectiveness of computer approaches

Much of the research and development of recent years has gone into the actual design and implementation of the software and there are only isolated reports of the testing of the materials in practice. Simons (1986) describes the use of hand-held computers programmed in BASIC to enhance the teaching of calculus. He reports that:

> ... the introduction of a personal computer into a course of this nature, whilst enhancing teaching and presentation in many areas, raises profound problems. (Simons 1986, page 552)

There were evident gains in the "immediate usefulness" of the work, but a substantial number of staff, "with long experience of teaching mathematics", but "little practical experience with a computer or numerical analysis" did not like the course. Simons suggests that "the aversion displayed by some members of staff" lies in the "feeling of uncertainty" in applying a numerical method.

> The traditional mathematician ... is clearly aware that for every numerical method a function exists for which the method produces a wrong answer. ... The statement that nothing is believed until it is proves is the starting point for teaching mathematics and introducing the computer forces the teacher away from this starting point. (*ibid.* p.552)

However, we should not infer from this that programming is of little value. On the contrary, though early use of programming did not always show enhancement of conceptual ideas in mathematics, more recent uses of programming environments to concentrate on specific constructs have shown very deep insights (Tall & Thomas 1988a). Programming in a language such as ISETL (which is designed to enhance the user's understanding of concepts through having to specify how processes are carried out) shows positive gains in conceptual understanding (Dubinsky, to appear).

The symbolic approach has powerful advocacy from several quarters. Lane *et al* (1986) suggests ways in which symbolic systems can be used to discover mathematical principles and Small *et al* (1986) reports the effects of using a computer algebra system in college mathematics. In the latter case the activities often consist of encouraging students to apply a technique, already understood in simple cases, to more complicated cases where the symbolic manipulator can cope with the difficult symbolic computations.

However, Hodgson (1987) observed:

> In spite of the fact that symbolic manipulation systems are now widely available, they seem to have had little effect on the actual teaching of mathematics in the classroom. (Hodgson 1987 p.59)

He quoted a report (Char *et al* 1986) of experiences using the symbolic system Maple in an undergraduate course where students were given free access to the symbolic manipulator to experiment on their own or to do voluntary symbolic problems which they could elect to count for credit. He noted a "somewhat limited acceptance of Maple by the students":

> While many explanations can be put forward for such a reaction (little free time, no immediate payoff, weaknesses of the symbolic calculator for certain types of problems, absence of numerical or graphical interface, lack of user-friendliness), it is clear that the crux of the problem concerns the full integration of the symbolic system to the course in such a way that it does not remain just an extra activity. This calls for a revision of the curriculum, identifying which topics should be emphasized, de-emphasized or even eliminated, and for the development of appropriate instruction materials. (*ibid.*)

Since this was written, the interface of Maple has been considerably improved (particularly on the Macintosh computer), and efforts have been made to enhance its user-friendliness, particularly for educational purposes. Just as the introduction of programming into mathematics courses received an initial mixed reaction only to show its greater value when used for explicit conceptual purposes, so symbolic manipulators may overcome their initial drawbacks as more imaginative *conceptual* uses are invented in teaching the calculus.

Heid (1984) reports her own research into an experimental calculus course which used the symbolic manipulator MuMath and appropriate graphical programs to introduce the concepts for twelve weeks, with practice of routine symbolic techniques only being studied in the final three weeks. She concluded that:

> Students showed deep and broad understanding of course concepts and performed almost as well on a final exam of routine skills as a group who had studied the skills for the entire fifteen weeks. (Heid 1984 p.2)

Based on the data from her experiment she formulated a number of conjectures, including the following:

> When concepts form the major emphasis in an introductory calculus course (assignments, class discussions, tests), and the computer is used to execute routine procedures:
> ... student understanding of course concepts will be broader and deeper ...
> ... student thinking will re-focus on the decision-making aspects of problem-solving ...
> ... students will remember concepts better, and be better able to apply them at a later time ...
> ... students will process information related to the concepts in larger "chunks"...
>
> A computer graphics approach to concept development in mathematics classes will result in better student performance on tests of "far transfer" of course concepts...
> Students will do more internal consistency checks when they work on conceptual problems if they can use the computer to process routine algorithms...
> Exposure to calculus skills through the use of symbol manipulation programs will not automatically improve the ability to perform these skills by hand or the ability to perform related algebraic manipulations...

> Use of symbol manipulation programs as tools will not eradicate algebra-based syntax errors... (*ibid* p.57 *et seq.*)

These conjectures, based on practical experience, are consonant with other related research.

Tall (1986b) reports the building and testing of a graphical approach to the calculus, using software designed to allow the user to play with examples of a concept, to enable the abstraction of the underlying principle embodied by the software. Such an environment is termed a "generic organizer". Generic organizers were designed for magnifying graphs (to see examples and non-examples of those that are "locally straight"), moving a chord along a (locally straight) graph to build up the gradient function, solving the reverse process of knowing the gradient and seeking the original function, numerical calculations of areas under graphs that emphasize the conceptual ideas, graphical solutions of first, second, and simultaneous first order differential equations (Tall 1986a, Tall *et al* 1990). In this research only the differentiation part was formally tested with pupils aged 16/17.

Using matched pairs selected on the pre-test, experimental students scored at a statistically significant higher level than the controls on global/holistic skills, including sketching derivatives, recognizing graphs of derivatives, relating the derivative of a function to its gradient function, and in explaining the notions of gradient, tangent and differentiation from first principles from a geometrical viewpoint. At the same time, more traditional logical/sequential tasks, such as explaining symbolic differentiation from first principles or routine differentiation of polynomials and powers, showed no significant difference. These results are very much in line with Heid's work.

Artigue (1983, 1987) and her colleagues developed a teaching programme for differential equations which first introduced qualitative ideas of families of solutions through studying pre-prepared computer drawn pictures. They then used these ideas to match pictures of the solution curves to the corresponding differential equations. For example, students were given different differential equations:

$$y'=\frac{y}{x^2-1},\ y'=y^2-1,\ \ y'=2x+y,\ y'=sin(xy),\ y'=\frac{sin(3x)}{1-x^2},\ y'=sinx.siny,\ y'=y+1$$

and the same number of corresponding pictures to match, of which two are shown in figure 2.

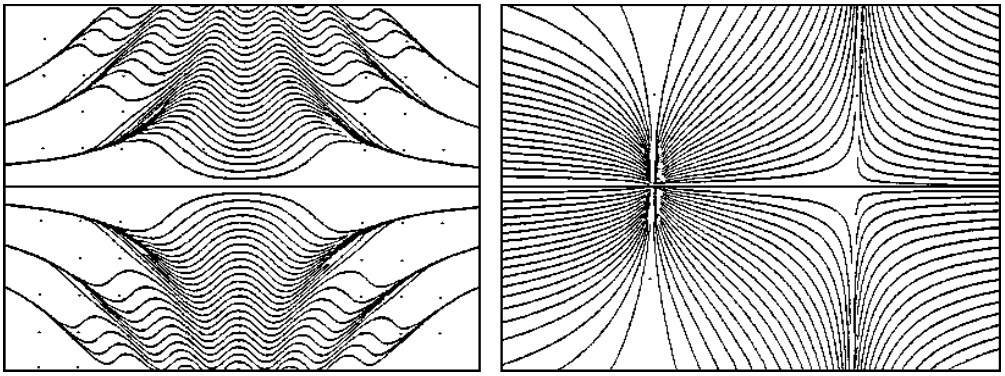

figure 2

They then related different methods available for solving specific differential equations using both qualitative and algebraic approaches and concluded by considering the qualitative theory of differential equations, with proofs of theorems based upon pictures of solution curves, dealing with barriers, trapping regions, funnels, attractors and so on. The students used both pre-prepared pictures and interactive computer programs, for instance to explore the phase portraits of differential equations depending on a parameter. At the end of the course the students showed themselves capable of giving meaning to the qualitative approach, to describe and draw solutions without symbolic differentiation and to coordinate algebraic and graphical representations.

Blackett (1987) introduced the relationship between the algebraic representation of a linear relation and its straight line graph to younger pupils (aged 14/15) using graph-plotting software. He taught three experimental classes of low, average and above average ability, who were matched with corresponding control classes. Those using the computer were able to tackle activities that might be too demanding using paper and pencil, for example, to instruct the computer to draw $y=x+20$ and $y=2x+10$ on the same axes to include their point of intersection (requiring some investigation to determine appropriate scales). The post-test showed a significant overall improvement in performance in all the groups, except for one control class taught by a teacher who used the pre-test to teach the pupils specific tasks likely to arise in the post-test. Detailed analysis of the responses revealed that these control students scored higher than the corresponding experimental students in questions that were almost identical to the pre-test, but they scored considerably lower on tasks with even tiny conceptual differences. Blackett reports that:

> Pupils who had been taught to answer specific questions rather than the underlying concepts experienced difficulty whenever new questions varied, even slightly, from those they had met previously. These results appear to highlight the effects of encouraging instrumental as opposed to relational understanding. (Blackett 1987, p.93)

He also noted substantial discrepancies between the children's performance on the post-test and traditional school tests. In particular there were a number of children who performed badly on traditional serialist/analytic questions, yet performed well on global/holistic tasks.

Blackett took these exceptional children from the lowest ability class and added them to the highest ability class for an introduction to the idea of the gradient of "locally linear" graph, using the "Graphic Calculus" software. Blackett found them well able to cope and concluded that:

> Students who had achieved a clear understanding of the straight line and its equation, particularly the significance of the gradient, can, with the aid of suitable computer graphics, develop an equally clear understanding of locally linear graphs and the curves associated with polynomial equations. ... There were pupils unable to handle number work successfully but were nevertheless able to demonstrate an understanding of advanced concepts presented in a visual form requiring either a visual interpretation or a drawing, rather than a calculation, as an answer. (Blackett 1987, pp. 127, 128)

His experiment indicated that these pupils, aged 14/15, were able to perform the task of sketching a global derivative at a level comparable with the 16/17 year old experimental students in Tall (1986b).

A theme running through all these pieces of research is the power of graphic computer software to improve the versatility of students thought processes, but with little significant change in their ability to cope with symbolic manipulation. On the other hand there is as yet little evidence that symbolic manipulators improve students manipulative ability, although there are hopeful indications that students may use them in suitably designed tasks.

A recurring observation is the difficulties experienced by teachers, both at university and in school, to come to terms with the new technology. Great experience of student problems in a pre-computer culture can sometimes be a hindrance in trying to predict what difficulties students may have when using the new technology. We are at present in the throes of a paradigmatic upheaval and cultural forces operate to preserve what is known and comfortable, and to resist new ideas until they are proven better beyond doubt.

In several countries the ability to implement a graphical approach on current microcomputers is leading to the production of such programs and the development of new curricula. For example, the School Mathematics Project in the U.K. is now designing a curriculum in which the first introduction to gradient is via "locally straight curves" and limiting processes are postponed to the second year of the course.

More recent developments in graphic approaches to calculus

Technology moves on apace. The new RISC (reduced instruction set chip) processors are becoming generally available in micro-computers at a lower price than primitive eight bit chips four or five years ago. The greater speed of these processors enable far more complex activities to be carried out on the computer in real time, including the plotting of models of nowhere differentiable curves. For example, it is easy to calculate the *blancmange function*, $y=bl(x)$ (figure 3) to any desired accuracy and to draw it extremely quickly, provided the software is sensibly programmed. (Some symbol manipulators such as *Maple* and *Derive* are self-defeating – as first published – being designed to allow any desired accuracy, and therefore failing to provide optimum, even adequate, speed.)

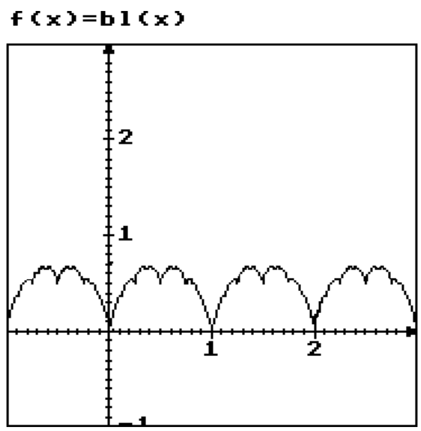

figure 3

This is calculated using the saw-tooth function $y=s(x)$ where y is calculated for given x as follows:

let $t=x-[x]$ be the fractional part of x

if $t<1/2$ then let $y=t$ else $y=1-t$.

The nth approximation to the blancmange is then found by adding together a sequence of sawteeth each a half the previous one.

$$bl(x,n)= s(x)+\frac{s(2x)}{2} +...+ \frac{s(2^{n-1}x)}{2^{n-1}} .$$

As n gets larger, the sawtooth $\frac{s(2^n x)}{2^n}$ gets very tiny and contributes little to the sum, so that $bl(x,n)$ stabilizes to look like the wrinkled blancmange $bl(x)$. This is easily seen graphically and the limiting process can be translated into a formal argument (Tall 1982).

More generally, the function which I call the Van der Waerden function, $van(x,m)$, can be calculated by adding together teeth to get

$$van(x,m,n) = s(x) + \frac{s(mx)}{m} + \ldots + \frac{s(m^{n-1}x)}{m^{n-1}}$$

and taking a sufficiently large value of n to see the picture stabilize to give (an approximation to) the non-differentiable function $van(x,m)$. These functions have the property that they *nowhere* magnify to look straight, so they are not differentiable anywhere.

In this way it is easy to build up a number of different functions which are continuous everywhere yet differentiable nowhere. The ready availability of such functions has interesting consequences for student's understanding of concepts in mathematical analysis.

Figure 4 shows two graphs superimposed: $y=\sin x$ and $y=\sin x+n(x)$ where the function $n(x)$ is a tiny Van der Waerden function. (It is calculated from $v(x)=van(x,3)$ as $n(x)=\frac{v(1000x)}{1000}$.)

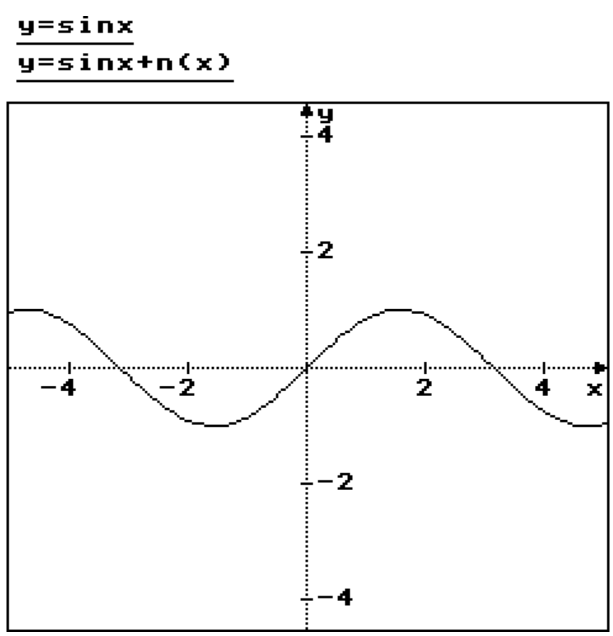

figure 4

Figure 5 shows the two graphs magnified to reveal the graph of $y=\sin x$ as locally straight, but $\sin x+n(x)$ as being nastily wrinkled. This informal idea represents the generative idea of the essential difference between a differentiable and a non-differentiable curve which are indistinguishable to a normal scale.

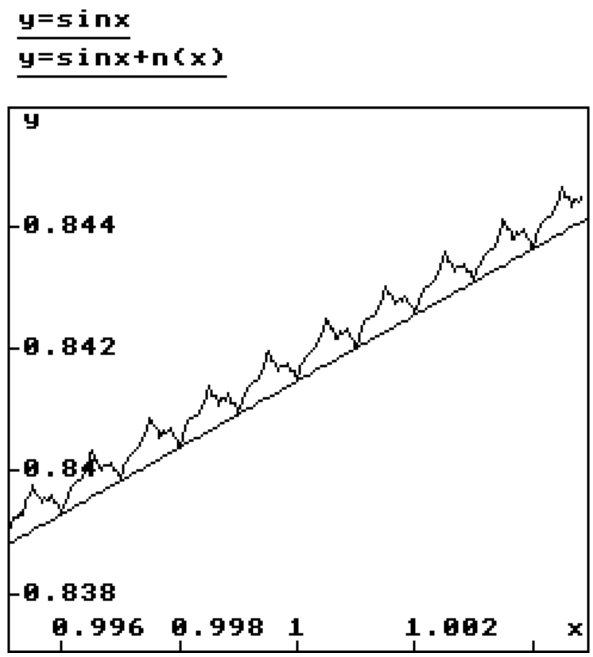

figure 5

Pictures drawn with today's technology have relatively large pixels, so the graphs look rough. This phenomenon can be turned to advantage, to underline that what is being drawn is a *model* of the graphs of the functions rather than the graphs themselves. What matters most is the quality of the picture in the student's mind, not on the screen!

Figure 6 shows the area function under the blancmange function being drawn in real time. A static picture in an article is completely inadequate to represent this potent dynamic idea. Here the line of dots represents the graph of the area function from 0 to the current point x. A straight line is drawn through the last two points, representing the gradient of the area function. As the area function grows, the gradient of the area function changes; and it visibly changes in a smooth way. The computer is giving an approximate model of an area function that is everywhere differentiable once, but not twice.

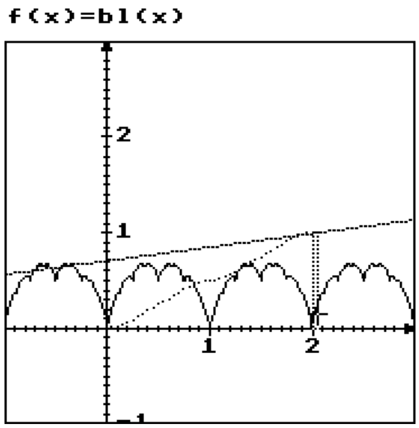

figure 6

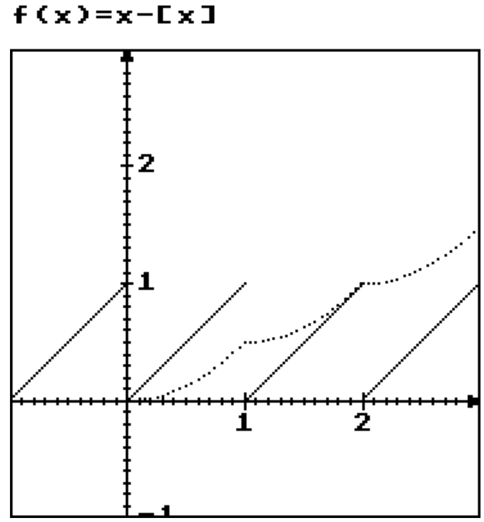

figure 7

Figure 7 shows the area function being calculated under the discontinuous function *y*=*x*-[*x*] (where [*x*] denotes the integer part of *x*). The dots are the "area so-far" under the curve from *x*=0. Clearly the area function has 'corners' at those points where the original function is discontinuous.

Figure 8 shows the curve *f*(*x*)=[*x*]+*bl*(*x*) where *bl*(*x*) is the blancmange. As *f*(*x*) is discontinuous at every integer and continuous, but not differentiable, everywhere else, what will the area function look like? Where will the area function be continuous and where will it be differentiable?

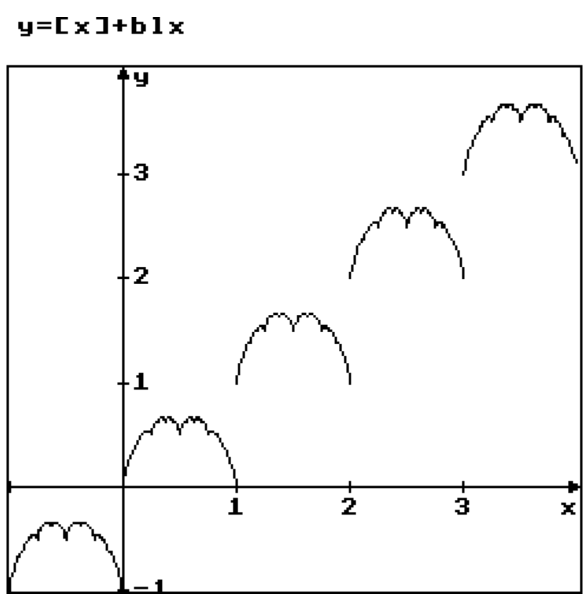

figure 8

Figure 9 shows the drawing of a solution to the differential equation $\frac{dy}{dx} = bl(x)$ where $bl(x)$ is the everywhere continuous, nowhere differentiable blancmange function of figure 3. Because $bl(x)$ has values between 0 and 1, a solution curve to the differential equation moves up and down (smoothly!) with gradient between 0 and 1.

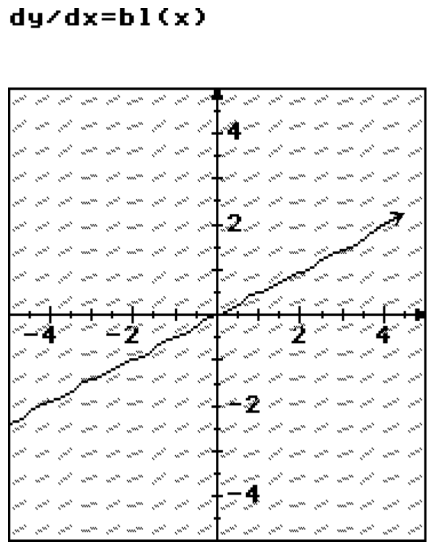

figure 9

Hubbard and West (1985) found that students had difficulty in understanding that a differential equation can have a solution when it is not possible to express the solution in "closed form" (made up of a combination of standard functions). To help students they developed interactive programs for the Macintosh computer that used the mouse interface to point at areas of interest to home in on singularities and to investigate the qualitative behaviour of solutions. The "existence" of a solution now depended on the ability to draw its graph, which could be done numerically without knowing a formula for it. Such experiences with a computer proved to give powerful insights into theorems of existence, uniqueness and behaviour of solutions of differential equations.

The way ahead

In the next few years, software of this nature can but get more and more powerful, with increasing use of more flexible input devices, such as using a mouse to point at a part of the picture that looks interesting, to zoom in and take a closer look.

Of greater importance in the software will be the development of flexible environments that unite numerical, symbolic and graphical facilities. Good software is not designed by encrusting the existing materials with more and more options. What is absolutely essential is to take into account all possible methods of information processing that can make the ideas simpler to handle. Such software is more likely to be successful if it uses the capacity of holistic/global thinking to complement the mathematician's traditional methods of deduction.

Back in 1971 Skemp made the pertinent observation that the teaching of mathematics by logical methods is good in that it shows that mathematics is a structured and not arbitrary science, but is weak in that it shows the *product of mathematical thought* rather than the *process of mathematical thinking*. Skemp subsequently emphasized this point by distinguishing between *building* and *testing* concepts. Testing involves subjecting the concepts to rigorous enquiry, to make sure that their construction and proof is founded on a firm logical base, but before they can be tested they must be built, and building is a cognitive act which is not served solely by demonstrating formal proofs. Using a logical deductive approach to the calculus is but one side of a two-sided coin. The other may be illuminated by graphical insight from well-designed computer software.

References

Artigue M. 1987: "Ingénierie didactique à propos d'équations differentielles", Cahier de Didactique No 15, Edition IREM, Paris 7.

Artigue M. & Szwed T. 1983: *Représentations Graphiques*, Edition IREM, Paris Sud.

Bach J.O. 1988: GrafMat-programmeme, Matematiklærerforeningen, Copenhagen.

Blackett N. 1987: *Computer Graphics and Children's Understanding of Linear and Locally Linear Graphs*, unpublished M.Sc. Thesis, The University of Warwick.

Brumby M.N. 1982: 'Consistent differences in Cognitive Styles Shown for Qualitative Biological Problem Solving', *British Journal of Educational Psychology*, 52, 244-257.

Cornu B. 1981: 'Apprentissage de la notion de limite: modèles spontanés et modèles propres', *Actes du Cinquiéme Colloque du Groupe Internationale P.M.E.*, Grenoble.

Char B.W., Fee G.J., Geddes K.O., Gonnet G.H., Marshman B.J. & Ponzo P. 1986: 'Computer Algebra in the Mathematics Classroom' in *Proc. 1986 Symposium on Symbolic and Algebraic Computation*, Assoc. Comput. Mach. 135-140.

Dienes Z.P. 1960 : *Building up Mathematics*, Hutchinson.

Dubinsky E. (to appear): 'Reflective Abstraction', in *Advanced Mathematical Thinking* (ed. Tall D.O.).

Heid K. 1984: *Resequencing skills and concepts in applied calculus through the use of the computer as a tool*, Ph.D. thesis, Pennsylvania State University.

Hodgson B. R. 1987: 'Symbolic and Numerical Computation: the computer as a tool in mathematics', in *Informatics and the Teaching of Mathematics* (ed. Johnson D.C. & Lovis F.) North-Holland, 55-60.

Hubbard J.H. & West B.H. 1985: 'Computer graphics revolutionize the teaching of differential equations', in *Supporting Papers for the ICMI Symposium*, IREM, Université Louis Pasteur, Strasbourg.

Keisler H.J. 1976: *Elementary Calculus*, Prindle Weber & Schmidt, Boston.

Kemeny J.G.1986: *Calculus*, True BASIC Inc.

Lane K.D., Ollongren A. & Stoutemyer D. 1986: 'Computer based Symbolic Mathematics for Discovery', in *The Influence of Computers and Informatics on Mathematics and its Teaching* (ed. Howson A.G. & Kahane J.-P.). Cambridge Univ. Press, 133-146.

Maths Workshop 1987: *Symbolic Calculus*, 45 Carson Rd, London E16 4BD.

McLean L.D., Raphael D. & Wahlstrohm M.W. 1984: 'The Teaching and Learning of Secondary School Calculus: Results from the Second International Mathematics Study in Ontario'.

Orton A. 1983a: 'Student's understanding of integration', *Ed. Studies in Math.*, 14, 1-18.

Orton A. 1983b: 'Student's understanding of differentiation', *Ed. Studies in Math.*, 14, 235-250.

Robert A. & Boschet F. 1984: 'L'acquisition des débuts de l'analyse sur **R** à la fin des études scientifiques secondaire françaises', *Cahier de didactique des mathématiques* 7, IREM, Paris VII.

Schwartz J.T., Dewar R.B.K., Dubinsky E. & Schonberg E. 1986: *Programming with Sets, An introduction to SETL*, Springer-Verlag.

Simons F.H. 1986: 'A course in calculus using a personal computer', *Int. J. Math. Ed. Sci., Techn.*, 17 5, 549-552.

Skemp R. R. 1971: *The Psychology of Learning Mathematics*, Penguin.

Small D., Hosack J., Lane K.D. 1986: 'Computer Algebra Systems in Undergraduate Instruction', *The College Mathematics Journal*, 17 5, 423-433.

Stoutemyer D. *et al* 1983: *MuMath*, The Soft Warehouse Honolulu, Hawaii.

Stoutmyer D. & Rich A. 1989: *Derive*, The Soft Warehouse, Honolulu, Hawaii.

Tall D.O. 1982: 'The blancmange function, continuous everywhere but differentiable nowhere', *Mathematical Gazette*, 66 11-22.

Tall D. O. 1986a: *Graphic Calculus*, (software for BBC compatible computers), Glentop Press, London.

Tall D.O. 1986b: *Building and Testing a Cognitive Approach to the Calculus Using Interactive Computer Graphics*, Ph.D. Thesis, The University of Warwick.

Tall D.O. 1986c: 'Lies, damn lies and differential equations', *Mathematics Teaching* 114 54-57

Tall D.O., Blokland P. and Kok D. 1990: *A Graphic Approach to the Calculus*, (software for IBM compatible computers), Sunburst Communications Inc.

Tall D.O. & Thomas M.O.J. 1988a: 'Longer Term Effects of the Use of the Computer in the Teaching of Algebra', *Proceedings of P.M.E.12. Hungary,* 601-608.

Tall D.O. & Thomas M.O.J. 1988b: 'Versatile learning and the computer', *Focus*, 11, 2 117-125.

Tall D.O. & Vinner 1981: 'Concept Image and Concept Definition in Mathematics, with particular reference to limits and continuity', *Ed.Studies in Mathematics*, 12 151-169.

Tall D.O. & West B.H. 1986: 'Graphic Insight into Calculus and Differential Equations', in *The Influence of Computers and Informatics on Mathematics and its Teaching* (ed. Howson A.G. & Kahane J-P.), Cambridge University Press, 107-119.

Van Hulzen J.A. & Calmet J. 1983: 'Computer Algebra Systems', in *Computer Algebra. Symbolic and Algebraic Computation* (2nd edition, ed. Buchberger B, Collins G. & Loos R.) Springer-Verlag.