Constructing Different Concept Images of Sequences & Limits by Programming

Lan Li

(formerly from)
Chengdu University
Chengdu
P. R. China

David Tall

Mathematics Education Research Centre
University of Warwick
Coventry, CV4 7AL
United Kingdom

As a transition between an informal paradigm in which a limit is seen as a never-ending process and the formal ε–N paradigm we introduce a programming environment in which a sequence can be defined as a function. The computer paradigm allows the symbol for the term of a sequence to behave either as a process or a mental object (with the computer invisibly carrying out the internal process) allowing it to be viewed as a flexible ***procept*** *(in the sense of Gray & Tall, 1991). The limit concept may be investigated by computing s(n) for large n to see if it stabilises to a fixed object. Experimental evidence shows that a sequence is conceived as a certain kind of procept, but the notion of limit remains more at the process level. Deep epistemological obstacles persist, but a platform is laid for a better discussion of formal topics such as cauchy limits and completeness.*

The difficulties faced by students in coming to terms with the limit concept are well-documented (e.g. Cornu, 1991), from the coercive effects of colloquial language (where words like “tends to”, “approaches” suggest a temporal quality in which the limit can never be attained) to difficulties coping with the formal definition and the quantifiers involved. Here we investigate the effects of introducing a computer environment allowing the student to construct some of the concepts through programming.

Following Dubinsky (1991), Sfard (1991), we formulate our observations using a theory whereby mathematical processes are subsequently conceived as objects of thought. We use the term *procept* (Gray & Tall, 1991) for the amalgam of process and concept where the same symbol is used both for the process and the output of the process. We hypothesise that different mental structures for the limit concept $\lim_{n\to\infty} s_n$ are produced by different environments, both for the term s_n as a computational process and a mental object, and also the limit itself, as process and object. These contain potential conflicts which require cognitive reconstruction to pass successfully from one paradigm to another.

We consider three separate paradigms:

(I) a (formula-bound) *dynamic* limit paradigm,

(II) a *functional/numeric computer* paradigm,

(III) the *formal ε–N* paradigm.

Paradigm (I) occurs in UK schools for students aged 16 to 18. In this curriculum the limit of a sequence (s_n) is studied only briefly in a dynamic sense as n increases, the main focus being on arithmetic and geometric progressions and convergence of the latter. Many students use the words “sequence” and “series” interchangeably in a colloquial manner. This approach emphasises the *potential* infinity of a process which cannot be completed in a finite time. The terms are seen as being given by *formulae* and the specific geometric (and arithmetic) progressions studied also have partial sums which can be

Published in *Proceedings of PME 17*, Japan, vol. **2**, pp. 41–48, 1992.

expressed as formulae. However, a general partial sum $s_n = a_1+...+a_n$ will rarely be given by a closed formula, so is again more likely to be seen as a process of addition rather than a limit object. The accent on process creates a "generic limit" concept in many students (Monaghan, 1986) in which the varying process is encapsulated as an indefinite "variable" object such as $0\cdot\dot{9}$ (which is "just less than 1") or $\frac{1}{\infty}$ which "just exceeds 0" (Cornu, 1991). This contains the seeds of conflict with the formal paradigm (III).

Paradigm (II) is the current focus of the paper. The individual may specify a procedural function taking a natural number as an input and outputting a real number. The computer language chosen allows such functions to be specified in a wide variety of ways – as a formula, or as a procedure involving logical decisions, loops, iteration, recursion. The symbol $s(n)$ may be considered either as representing the programmed procedure, or the output of the function and therefore behaves as a procept. The numerical computation is performed internally by the computer: we call such a procept a *cybernetic* procept. The computer language used has no built-in limiting process and the limit cannot be programmed in a proceptual way. Instead it may be investigated by computing the values of $s(n)$ for large values of n, say $s(1000)$ or $s(10000)$, to see if the value of the term stabilises. This gives a numerical value of the limit, allowing the limit to be studied as a (numerical) object. However, this produces the notion of a *cauchy limit* in which the terms become indistinguishable to a given level of accuracy rather than computing the exact numerical value of the limit. Paradigm (II) therefore allows the notion of sequence as a cybernetic procept with the limit being both process and object, yet differing subtly from the full proceptual structure of the formal paradigm.

Paradigm (III) is the eventual target paradigm, which will be studied later in the degree course. The notion of sequence will be defined formally as an *arbitrary function* $a:\mathbb{N}\rightarrow\mathbb{R}$ from the natural numbers to the real numbers, with the notion of limit given in terms of the ε–N definition. There are cognitive difficulties both with the notion of sequence as a function and with the limit. For instance, the definition of a sequence as a function from $\mathbb{N}$ to $\mathbb{R}$ includes the requirement that the function be specified simultaneously for *all* values on the infinite set $\mathbb{N}$, involving *actual infinity* rather than potential infinity. The definition of "limit" is formulated in terms of an unencapsulated *process* (given ε, an N can be found such that …) rather than being described explicitly as an object. It involves several layers of quantifiers which exceed the short-term memory processing capacity of many students. There is a severe problem of the *status* of the limit notion – can one *define* an object linguistically, or does it need to have an independent existence? For example, if a decimal such as $0\cdot\dot{9}$ (nought point nine recurring) is believed to exist as a number *less* than one, can it be *defined* to be something *equal* to one?

The plan of action is to use paradigm (II) as a transitional environment to provide students with experiences which will lay the cognitive foundations for the formal definitions and to study what cognitive changes occur and what obstacles prove resistant to change.

The experiment

The study took place at Warwick University in a 20 week (60 contact hours) course on programming and numerical methods using BBC structured BASIC. The students were first year trainee mathematics teachers with nominal minimum UK A-level grade C in mathematics and one other grade D. This places them in the top ten to fifteen percent of the total population but few have the required grades (A in mathematics, plus two other Bs) to study mathematics in the university mathematics department.

Each week one of the three contact hours was available for introduction of new topics and discussion of difficulties with the lecturer. Students were encouraged to work together from printed notes. Assessment was by four assignments, so there was no need for rote-learning for an examination.

The first term introduced fundamental programming constructs such as variables, FOR:NEXT loops, REPEAT:UNTIL loops, graphical commands, procedures, functions, and structured programming, including the development of a structured graph plotter and a project to write a computer game.

The second term concentrated on programming numerical methods and investigating their properties. Topics studied included solution of equations in the form $x=f(x)$ by iteration, $f(x)=0$ by bisection, decimal search, Newton-Raphson, calculating numerical gradients, areas, solutions of differential equations, the order of accuracy of various algorithms, sequences, series (and their possible limiting behaviour), calculating functions by procedural methods, including Taylor series.

To broaden the concept image of sequence beyond a formula, functions were defined such as:

```
1000 DEF FNiterate(n) : LOCAL x,k : x=1 : FOR k=1 TO n : x=COS(x) : NEXT k : =x
```

(The LOCAL command localises the variables so that their values are not affected elsewhere.)

`PRINT FNiterate(30), FNiterate(50), FNiterate(100), FNiterate(1000)` gives

```
0.739087043     0.739085135     0.739085135     0.739085135
```

so that the sequence stabilises to x=0.739085135 for which $x = \mathrm{COS}(x)$ within computer accuracy.

Other sequences defined in the course, to give more flexible concept imagery, included

```
1100  DEF FNo(n) : IF n MOD 2 = 0 THEN = 0 ELSE = 1/n

2000  DEF FNprime(n)
2010  LOCAL k,ok : ok=1
2020  IF n=1 THEN = 0 ELSE IF n=2 THEN = 1
2030  FOR k=2 TO SQR(n)
2040    IF n MOD k = 0 THEN ok=0 : k=n
2050  NEXT k
2060  =ok
```

`FNo(n)` returns 0 if n is odd and $1/n$ if n is even, `FNprime(n)` returns 1 if n is prime and 0 if composite.

A sequence of terms `FNa(n)` may be summed iteratively or recursively, as follows:

```
3000  DEF FNs(n) : LOCAL k,s : FOR k=1 TO n : s=s+FNa(k) : NEXT k : =s
4000  DEF FNS(n) : IF n=1 THEN = FNa(1) ELSE = FNS(n–1)+FNa(n)
```

The formal definition of sequence was given in the notes as a function from the natural numbers to the real numbers and the concept was further discussed in seminars. The students were invited to program various sequences and series and to investigate the behaviour for large values of n. When stabilisation occurred, it was emphasised that the greater the accuracy required, the larger the value of n necessary for the terms to be indistinguishable to this accuracy. This was used in a seminar to lead to the formal ε–N definition of a limit of a sequence.

The convergence of the sequence 0.9, 0.99, 0.999, … was discussed, with nth term $s_n = 1 - 1/10^n$ and its limit was demonstrated to be 1 using the definition. The meaning of an infinite decimal expansion such as π=3.14159… was re-defined to be the limit of the sequence (s_n) where s_1=3.1, s_2=3.14, ... and, in general, s_n is the decimal including the first n places of the decimal expansion. In particular, the sequence with $s_n = 0\cdot\underbrace{99\ldots9}_{n \text{ places}}$ has limit 1 and is written as $0\cdot999\ldots = 0\cdot\dot{9}$. Its value is therefore 1.

The effects of the experiment

Data was collected from the experiment in four ways: a pre-test with questions on limits of sequences and series, a post-test with essentially the same items (Li, 1992), interviews with selected students, and written work submitted for assessment.

In response to the question:

If you can, explain in your own words what is a sequence,

the pre-test revealed the overwhelming sense that a sequence needed a formula or pattern (Table 1).

What is a sequence?	pre-test (N=25)	post-test (N=23)
terms given by a formula or pattern	**17**	9
mentioning "series"	4	0
function from natural numbers to reals	0	**3**
function from reals to natural numbers	0	**4**
mention of *no formula* for terms	1	**3**
other	2	4
no response	1	0

Table 1: What is a sequence?

Responses revealed the colloquial interchangeability of "sequence" and "series", for instance,

A sequence is a series of numbers connected to the numbers before and after by a formula.

The change from pre-test to post-test showed a reduction in students mentioning a formula and an increase in those mentioning the function definition or denying the need for a formula. However, the definition was poorly remembered (it did not need to be rote-learnt for an exam) and four students reversed the order (from reals to natural numbers) which was subsequently explained verbally by one of them reading the term from left to right as "s_n is a real number which is related to the number n."

This confirms the often noted phenomenon that students in such a course do not rely on the concept definition to do mathematics, instead they evoke a concept image from their experience.

The notion of *series* changed substantially from pre-test to post-test. Table 2 shows that there was a strong move from terms being "given by a formula" to adding terms of a sequence. Specific mention of geometric and arithmetic sequences also diminish in the light of a wider variety of examples.

What is a series?	pre-test (N=25)	post-test (N=23)
adding terms of a *sequence*	0	**15**
adding numbers given by a *formula*	2	2
numbers related by a *formula*	**13**	4
geometric or arithmetic progression	**7**	0
other	7	2
no response	1	0

Table 2: What is a series?

This shows that, although students may have difficulty in expressing their knowledge and rarely evoke the definition, there is a general shift in understanding that a series is *technically* a sum of terms, even though the word "series" and "sequence" may continue to be used informally on occasion.

Two successive questions revealed interesting contrasts which changed little from pre-test to post-test:

(A) Can you add $0\cdot1+0\cdot01+0\cdot001+\ldots$ and go on forever and get an exact answer? (Y/?/N)

(B) $1/9=0\cdot\dot{1}$. Is 1/9 equal to $0\cdot1+0\cdot01+0\cdot001+\ldots$? (Y/?/N)

The favoured response on both pre-test and post test is *No* to (A) and *Yes* to (B) (Table 3).

Responses to (A)/(B)	Y / Y	Y / N	N / N	**N / Y**	N / ?	? / N	nr / Y
pre-test (N=25)	4	0	1	**18**	0	1	1
post-test (N=23)	**2**	2	2	**14**	1	0	0

Table 3: a recurring sum

How can the equation $0\cdot1+0\cdot01+0\cdot001+\ldots$=1/9 be *false* but 1/9=$0\cdot1+0\cdot01+0\cdot001+\ldots$ be *true*? One may hypothesise that each is read left to right and that the first represents a potentially infinite process which can never be completed but the second shows how 1/9 can be divided out to get as many terms as are required. Interviews suggested shades of meaning consistent with this but sometimes with a different emphasis. For instance, several students said that the initial statement "$1/9=0\cdot\dot{1}$" in (B) coloured their view and that they used this to equate $0\cdot\dot{1}$ and "$0\cdot1+0\cdot01+0\cdot001+\ldots$" One claimed not to know how to convert a fraction to a decimal, other than divide it out on a calculator; in this case 1/9 gives 0.11111111 and this would be sufficient in his estimation to show that the digit 1 was repeating.

Despite the experience of a sequence of terms becoming indistinguishable, and the seminar explanation of the definitions of limit and infinite decimals, there was little change in the response to the question "is $0\cdot\dot{9}=1$? (Y/?/N)." (Table 4).

Is $\boldsymbol{0\cdot\dot{9}=1}$ ***?***	Y	**N**	?	no response
pre-test (*N*=25)	2	**21**	1	1
post-test (*N*=23)	2	**21**	0	0

Table 4: Nought point nine recurring

Interviews revealed that students continued to conceive $0\cdot\dot{9}$ as "a sequence of numbers … getting closer and closer to 1", or not a fixed value "because you haven't specified how many places there are" or "it is the nearest possible decimal below 1". The programming experiences did not change this view, and it is important to note that one cannot compute the *exact* limit by programming in this environment, so the limit concept cannot be *constructed* through programming.

Another generic limit did, however, prove to change (Table 5), in response to:

Complete the following sentences: 1, 1/2, 1/4, 1/8, … tends to ____________
The limit of 1, 1/2, 1/4, 1/8, is ____________

"tends to" / "limit"	0 / 0	0 / $\frac{1}{\infty}$	$\frac{1}{\infty}$ / $\frac{1}{\infty}$	0 / ?	2 / 2	0 / 2	0 / 1
pre-test (*N*=25)	0	**11**	1	5	0	2	2
post-test (*N*=23)	**8**	3	3	0	4	0	2

Table 5: The evoked meanings of "tends to" and "limit"

The response "2" may indicate the sum of the *series* $1+\frac{1}{2}+\frac{1}{4}+\ldots$. An interview revealed the response "1" for the limit related to an interpretation of the "limit" of the sequence as the largest term. The most commonly occurring response changed from "tends to 0, limit $\frac{1}{\infty}$" to "tends to 0, limit 0" suggesting that the idea of $\frac{1}{\infty}$ as an indefinite number, arbitrarily small, is being replaced by the numeric limit 0.

There were considerable successes in programming. It is quite apparent that the students were able to program functions as procedures yet use the name of the procedure as an object in another piece of programming. For example, the *programming* in the following problem was successfully completed by 23 students out of 25:

Define a function FNb(*n*) which returns the value b_n where

$$b_n = \begin{cases} 1/n^2 \text{ if } n \text{ is prime} \\ 1/n^3 \text{ if } n \text{ is not prime and even} \\ 1/n! \text{ otherwise} \end{cases}$$

(Hint: it may help to define a function `FNprime(n)` which returns 1 if *n* is a prime and 0 if not, and another function `FNcalc(n)` which returns 1/*n*!)

Calculate the sum $b_1 + b_2 + ... + b_n$ for $n = 1000$. Does the series Σb_n converge?

The function `FNprime(n)` mentioned earlier was often used in a function of the following kind:

```
10000  DEF FNb(n)
10010  IF FNprime(n)=1 THEN =1/n^2 ELSE IF n MOD 2 = 0 THEN = 1/n^3
       ELSE = FNcalc(n)
```

Twenty three students used such a function to calculate $b_1 + b_2 + ... + b_n$ and seventeen of these programmed the partial sum as a function which added up the terms `FNb(k)` for k=1 to n. Thus there is considerable evidence to show success in using the function notion as procedure or object.

However, the calculation of a sum took longer for a larger number of terms. For instance it might take five seconds to compute a sum of 1000 terms, fifty seconds for 10 000 terms and five hundred seconds for 100 000 terms. This temporal aspect is illustrated by the following response to $\Sigma\, 1/n^2$:

> *$\Sigma 1/n^2$ converges – with count 1000 it appeared to converge towards 1.6440 (4 dps). However, on another occasion with count 7996, the sum was 1.6448 (4 dps). I think that the series does converge, as successive values get smaller and the difference between successive sums becomes smaller, but it takes a long time to converge – longer than I spent!*

It was also notable in class that, when students had programmed the partial sum as a formula, some became obsessed with the numerical values and no longer focused on the internal process of adding terms. For instance, it was possible for a student to program `FNs(n)` adding together $1/k^2$ for k=1 to n, and note that when n=1000 the sum is still changing in the 6th decimal place without explicitly noting that the thousandth term added on is $1/1000^2$. For many students, the computer laboratory work seemed to focus on the syntax of the programming and the use of the program to investigate numerical values rather than any reflection on the symbolic processes occurring.

The environment was able to provide a spectrum of phenomena from series such as $\Sigma 1/n!$ which stabilise in a few terms, to those such as Σn which clearly diverged, and in between, series like $\Sigma 1/n$, $\Sigma 1/n^2$ which grew by smaller and smaller amounts and were open to question. In the seminar proofs were given for convergence of $\Sigma 1/n^2$, and divergence of $\Sigma 1/n$. (In *theory* the latter diverges, but *in practice*, it grows so slowly that on today's computers it will not exceed 100 in a human life-span!)

The sum Σb_n proved to be very interesting. The procedures took considerably longer, so only a small number of terms could be computed (around 10 000 in a minute). But the nth term at various times could be $1/n^2$ or $1/n^3$ or $1/n!$, depending on the value of n, so the amount by which the sum increased as an extra term was added could vary considerably:

> *In places it looks as if it is not converging, but other parts of the series looks as if it is.*

Other students believed the series not to be converging:

> *The series does not converge, although it is increasing very slowly.*

This example of Σb_n proved to be very fruitful. Each term was less than or equal to $1/n^2$, so the nth partial sum was less than the nth partial sum of $\Sigma 1/n^2$ and the latter partial sums could be proved to be increasing and bounded above. There was a spirited dialectic argument in one seminar about whether an increasing sequence bounded above necessarily converged to a limit, or whether it could continue creeping up, never reaching the upper bound, and never actually converging. The *completeness axiom* therefore arose as part of a natural student conversation.

Reflections

The course provided an environment in which certain sequences were seen to stabilise after a certain number of terms, and the more accurate the required stabilised value, the further along the sequence one may have to go. Experience was provided for the definition of the limit, in the *cauchy* sense as well as the sense of tending to a specific value. In this context discussion of the completeness axiom occurred naturally.

But deeper epistemological questions remained. For many students, the meaning of an infinite decimal as a limit of a sequence was not established. It already had a different stable meaning and in the programming paradigm (II) such a limit cannot be *constructed*, only approached within reasonable practical accuracy, which fails to disturb the earlier meaning.

Some gains were made – the proceptual programming of a function as procedure and object, a clearer distinction between sequence and series, and some progress towards the perception of the limit object as a specific number rather than an indefinite generic limit. However, it is essential to examine the nature of computer-constructed objects with greater care. The programmed function is a *cybernetic* procept which auto-calculates the value and has subtle differences from the formal notion of function. The focus is taken away from the relationship between process and product which would be given by experiencing the calculation itself. The latter construction may therefore not be performed and necessary relationships may not be constructed. Deeper epistemological obstacles are likely to remain. Further cognitive reconstruction is necessary for transition to the formal ε–N paradigm, but at least experiences have been gained which may be fruitful as a basis of discussion.

References

Cornu, B. (1991): “Limits”. In D. Tall (Ed.) *Advanced Mathematical Thinking*, Dordrecht, Kluwer, 133–166.

Dubinsky, E. (1991): “Reflective abstraction in advanced mathematical thinking”. In D .Tall (Ed.) *Advanced Mathematical Thinking*, Dordrecht, Kluwer, 95–123.

Gray, E. M., & Tall, D. O. (1991): “Duality, ambiguity and flexibility in successful mathematical thinking”, *Proceedings of PME XIII, Assisi*, Vol. II, 72–79.

Li, L. (1992): *Studying functions and limits through programming,* unpublished M.Sc. thesis, Warwick University.

Monaghan, J. D. (1986): *Adolescents’ understanding of limits and infinity*, unpublished Ph.D. thesis, Warwick University.

Sfard, A. (1991): “On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin”, *Educational Studies in Mathematics*, ***22***, 1–36.