Albuquerque Conference 2001

New insight in mathematics by live CAS documents.

Ivan Cnop, Vrije Universiteit Brussels, Belgium,
icnop@vub.ac.be

Abstract
Traditional education in mathematics is mostly a matter of hindsight, and many mathematics texts offer little opportunity for students and learners to gain insight. Here we show how experiments in CAS can lead to new ways of handling problems, new conjectures, new visualisations, new proofs, new correspondences between theories and sometimes even new definitions. Experimentation will also have effect on the teaching of mathematics and its applications, together with the introduction of Computer Algebra Systems and symbolic programming. Many examples in this presentation involve randomness and simulations by CAS. The live documents are being offered in classroom lecturing using data projection, and also on CD−rom and local servers as a template for project work by the students. Next fall semester class work will be fully interactive in a special classroom equipped with laptop computers. The approach also offers opportunities for distance learning, and some of the material is already being used in such programs.
In this talk we want to present a selection of these documents, depending on the interest of the audience. All documents are written in Mathematica. Some of these were written under the "Exploot" project (experimental learning environment using education technology), funded by the Education Department of the Flemish regional Government in Belgium.

Introduction

Most modern mathematics texts are well structured, nicely formatted and polished, and offer little leeway both for the teacher and the learner. Results are presented only when authors feel they cannot immediately be improved, and the subsequent authors using these results or teachers mentioning them will present results in a simplified form or with alterations to accommodate the user groups they are addressing. It is a filtering process that in each step reduces the content, tries to trivialise the theory, and a process that moreover hides the creative effort from the initial authors. Important applications, which motivated the development, are often left out and the historic build−up is forgotten.
The final user is not aware that the build−up of mathematics has always been a creative process which most of the time involves lots of experimentation. We do not realise how much hard work was done by great mathematicians in the past. Students tend to think mathematical discovery only comes by stroke of genius. Students are convinced they will never have such luck, and this adds to the general uneasiness about mathematics and its teaching.
Many educators therefore want to replace the traditional paradigm using the sequence

Definition −> theorem −> proof −> corollary (−> application)

by an approach which is more historic using the discovery chain

Problem −> experiment −> conjecture and idea of proof.

A CAS allows lots of experimenting by the students, thus helping to find reasonable conjectures. It can offer new insight in how to prove these conjectures and may point to new mainstream developments in mathematics, and may even point to some topics that are now obsolete. Other mathematicians and teachers may regret that certain topics in the curriculum no longer deserve the attention they used to receive, but this is the way science develops.

All examples below are written in Mathematica, a general–purpose CAS that allows efficient symbolic programming closely following mathematical ideas [1]. In some of the examples the programming is part of the build–up of mathematical ideas. In a final example we show how list processing, rather than procedural or vector processing, should be included in algebraic reasoning.

1. Guessing the exactness of limits.

The well–known approximation of n! using logarithms is coarse for small values of n, but its relative error improves as n increases. The problem is to have an idea how good this approximation is, and what improvements are possible. Plotting values does not help much because of the huge scale of such plots. The quotient looks like some negative power of n, and we can ask which power it is.

In many mathematical problems the main question is not whether there is a limit, but rather how fast this limit is approached. This allows efficient approximate computation. The rule of L'Hopital is no longer sufficient, but has to be replaced by some careful analysis of the orders of growth involved. This approach follows ideas due to Landau and Hardy.

2. The burial of trigonometry.

Cosine and sine functions are merely projections of a point moving along the unit circle, also called harmonic movement. It is therefore not surprising that every trigonometric formula can be derived from a geometric feature in the complex number plane. A rapid introduction to the complex number system using polar co–ordinates in the plane, complex multiplication by adding polar angles and a straightforward representation of complex polynomials allows the user to work out problems involving roots of unity [2].

Powerful applications, such as encryption techniques and the discrete Fourier transform, can be offered at an early stage after this introduction.

3. Snowflakes, dragon curves and sets of measure zero in the plane.

Using complex numbers for positioning points in the plane, together with a turtle graphics description of the path, one obtains elegant descriptions of such fractals, and experimenting leads to insight in some sets of small or zero measure in the plane. We can approximate the Koch snowflake from inside, and from outside by iterating the design of the Mitsubishi logo. Programming such fractals is simple.

The reason why many fractal curves are not smooth becomes obvious from such constructions.

4. Looking at Fourier series from a distance.

Using the representation of polynomials one can introduce complex power series. By projecting their plots on horizontal and vertical planes in space, one obtains the cosine and sine Fourier series of the sequence of coefficients. These coefficients are determined by an interactive fitting process, mimicking mean square approximation. Delicate results concerning harmonic conjugates and a simple proof of Gibbs' phenomenon arise naturally.

5. The exponential as a transition from discrete to continuous processes.

It is important to shift from discrete to continuous phenomena and conversely whenever this is useful for the solution of a problem. Discrete solutions are given by powers of variables while the continuous solution often involves the exponential function. This duality should be learned from the very beginning. It is therefore unfortunate that curricula tend to partition mathematics into subheadings that allow only one of these approaches, either the discrete one (in linear algebra) or the continuous one (in calculus and analysis).

6. Perfect shuffle and combinatorial identities.

The exponential also arises in other problems related to probability theory such as the race horse problem. It is surprising that the solution of this problem is independent from the size of the problem. It is easy to program shuffles and students can use this technique to program betting games on their computer.

Here the approximation of combinatorial expressions by exponential functions allows simplification and fast computation of solutions. Using exponential functions is also the idea behind the theorem of large numbers and more generally in the use of probability densities.

7. Making quotient groups without checking normality of subgroups.

This is perhaps the most difficult topic in abstract algebra. Following the usual definitions one can program the construction of subgroups generated by subsets and cosets of these subgroups. We obtain colour diagrams that exhibit the structure of a quotient group without having to check the normality of the subgroup beforehand for non−commutative groups [3].

Isomorphic groups and certain decompositions are also discovered at a glance.

8. Continuous objects, and generalised functions.

The ε−δ condition is not well understood by many students, and difficult to check point by point. By looking at graphs as a whole, it is easy to see continuity is merely a property from dynamic geometry, described by shifting pictures in the plane. This holistic approach using allows definition of continuous objects that are no longer functions, such as inverses of functions, other multiple−valued relations and even geometric figures. The formulation of classic theorems, such as the fundamental theorem of calculus and the behaviour of the solutions of differential equations, is much easier in this setting.

The transition to mathematical transforms, defined as functions acting on other functions, becomes straightforward.

9. Symbolic programming

Symbolic programming follows the mathematical reasoning closely. Programming is writing functions for named arguments. Instead of handling results for specific numbers as is still done on elementary pocket calculators, it is much more interesting to write down general expressions depending on parameters and delay the replacement by the actual value as long as possible, thus handling all possible cases at once. A useful example is the dynamic system behaviour of models in economy and elsewhere. In many models it is the transition from one type of behaviour to another that is the most interesting part, and showing such a transition in an animation can only be done if all cases are handled at once.

10. Programming in list spaces.

Important mathematical objects such as polynomials and functions, sequences of outcomes, and geometric objects arising in the examples above are difficult to describe by vectors of fixed length. This is a problem for procedural programming where the size of an object has to be decided well in advance. The advent of list processing allows a concise and more flexible approach, together with a description of the objects that corresponds to mathematical intuition.

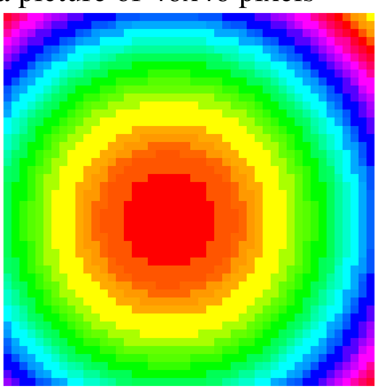
Conclusion.
The circulation of live documents in which experimenting by the student is allowed and encouraged has a tremendous impact on the teaching of mathematics and on the future curricula.

References.

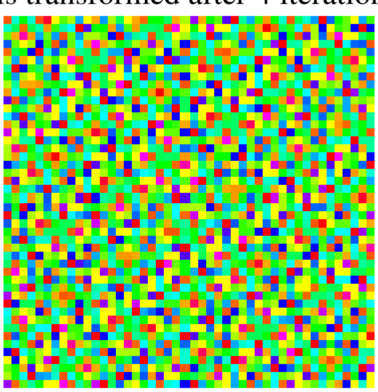
1. Stephen Wolfram: “Mathematica” version 4.0
2. Hania Uscka: “Complex Numbers” (in Dutch), on CD–rom, Exploot project, 2000
3. Ivan Cnop: “Implementing mathematical concepts in Mathematica”, ICTMT 3 Proceedings, Koblenz 1997.

Appendix: Some examples (same numbering as in the text)
Most examples involve animations, and it is not useful to reproduce these in a printed version. Therefore only some static examples are shown.

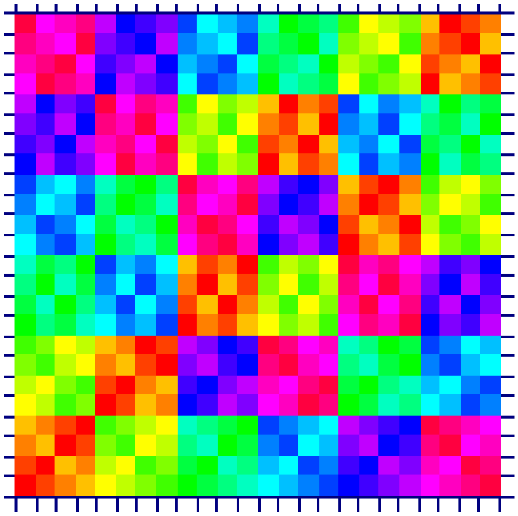
[5] **Effect of the discrete Arnold mapping**
a picture of 46x46 pixels



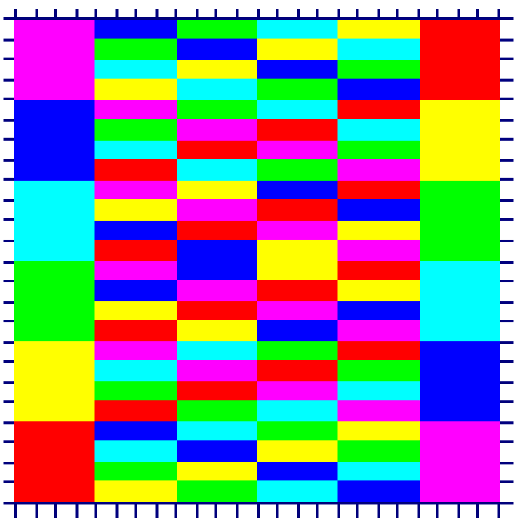
is transformed after 4 iterations into



[7] **Quotients in permutation groups**



A quotient in S4 by a normal subgroup



A quotient in S4 by a subgroup which is not normal

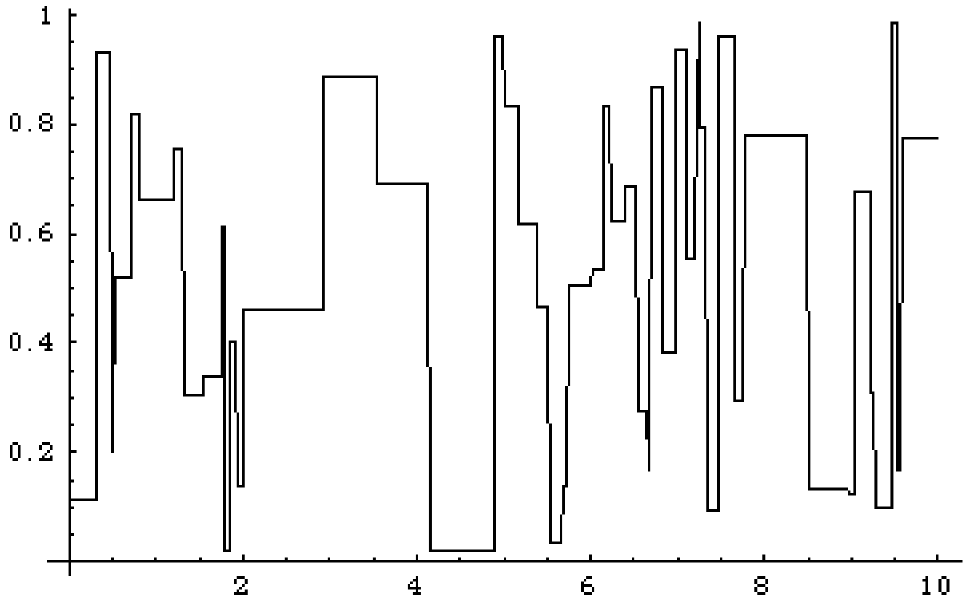
[9] **Random values on Random intervals: the Manhattan skyline**

Programming involves different object types: a list of values and a list of intervals symbolically defined by

slotFunction[{a_, b_}] := a < x < b

and a condition has to be applied giving Random values on intervals with Random a and b.
Here is a sample output of such program:

Plot[f[x], {x, left, right}, AxesOrigin –> {left, 0}]



[10] **A programming example: folding a strip of paper into a dragon curve**
Start by a recursive definition of a sequence consisting of entries 1 and −1

```
f[0] = {1};
f[n_] := Flatten[{f[n − 1], 1, −Reverse[f[n − 1]]}]Table[f[i], {i, 5}] // TableForm
```

Since we will have to add elements in a list of values we use FoldList . Next multiply some angle theta with these sums to obtain directions and fix the radius.

```
directions[n_] := theta FoldList[Plus, 0, f[n]]
points = FoldList[Plus, 0, Chop[.1 E^(I directions[14])]];
```

Now give a value to the angle theta and plot. Different values of theta will give completely different plots.

```
theta = /2 // N;
ListPlot[Transpose[{Re[points], Im[points]}], PlotJoined −> True,
  AspectRatio −> Automatic, Axes −> None]
```