A pedagogy-embedded Computer Algebra System as an instigator to learn more Mathematics [1]

Thierry Dana-Picard and Ivy Kidron
Department of Applied Mathematics
Jerusalem College of Technology
Havaad Haleumi Street, 21
Jerusalem 91160 – Israel
Email: dana@jct.ac.il – ivy@jct.ac.il

Abstract:
The constraints of a Computer Algebra System are generally classified as internal constraints, command constraints and organization constraints. In fact, a fourth kind of constraints exists, namely motivating constraints. These constraints consist in features or commands of the CAS whose understanding demands sometimes from the user to acquire more mathematical knowledge than what has been taught in a standard course. Theorems can appear which necessitate learning beyond the syllabus framework. Such "new" theorems appear generally in two situations, namely when using a pedagogy-embedded feature of the CAS (either a posteriori help, or a priori hints), or when using certain commands and trying to analyze the results. We describe a research frame in the first year Foundation Courses in Mathematics, in our Engineering College. With this research, we wish to understand more deeply the instrumentation processes at work with the students and to check motivations for a change in the institution's culture.

I. Levels of intervention of a Computer Algebra System.

As an assistant to mathematical learning, a Computer Algebra System (a CAS) offers three levels of help:

1) a technical tool performing technical tasks;
2) a tool whose performances help to develop more conceptual understanding;
3) a technological help to bypass a lack of conceptual knowledge, where such knowledge is out of reach, at least in ”the next future”.

The first level is the blackbox level and has no great pedagogical value. Maybe it allows the teacher to save time for reflexion and theoretical understanding, but a perverse effect is the loss of manual computation

[1] This research was supported by the Israel Science Foundation (grant No. 1340/05).

skills, as noted by (Herget et al., 2000). Integration techniques, techniques for solving equations, either linear or non-linear, are abilities which could disappear. The following claim has been heard in a professional meeting: "nowadays, there are computers who make the computations; thus there is no need anymore to teach integration techniques". We disagree with this claim, and wish to show that, on the contrary, Computer Packages enable to learn and to understand more Mathematics than expected.

We can distinguish a level 1½ , where the student uses the CAS for verifying results. There are at least two kinds of verifications:

- Verify either a numerical result or a "closed" algebraic expression. The mathematical correctness of the verification is not always evident. For example, two different CAS or even two different commands of the same CAS, or a CAS and hand-work, can provide different algebraic expressions, both valid. As inert expressions, they are different, but when defining functions, which are dynamical objects, different expressions can define the same function. The verification issue has been addressed by Lagrange (1999) and Pierce (2001).
- Perform the passage from n to $n+1$ in a recurrence, after the CAS enabled to conjecture a formula (see (Garry 2003) page 139).

Steiner and Dana-Picard (2004) commented aspects of level 2. Low-level commands are important for cognitive processes attempting to afford a good conceptual insight. A CAS command is called a *low-level* command if it performs a single operation, while a macro is a command programmed to perform a sequence of low-level commands. Low-level commands act as the atoms of every computerized process for solving a problem.

Because of syllabus limitations and of time limitations, level 3 is less commonly considered. It appears close to the frontier of the syllabus, either for exercises aimed to broadening knowledge beyond this frontier, or for problem solving when the necessary theorems have not been taught and will not been taught "in the next future". Technical use of the CAS fills the gap; see (Dana-Picard 2005b).

A fourth level exists: a CAS is a device whose performances may incite the user to acquire more mathematical knowledge. The reason can be one of the two following:

- Multiple commands are available for *seemingly* the same purpose. For the user to make an intelligent decision which command to use, he/she must have a good knowledge of the Mathematics implemented in the algorithms.

- There exist situations where a unique algorithm is available, either because of the theoretical state-of-the-art or because of the decisions of the developers. This limits the diversity offered by the CAS; this issue is studied by Artigue (2002), page 265. In such a case, the theorem transformed either into an algorithm or into a command is not always a standard theorem taught in a standard course; see the example with Derive in section II.

In every case, the implemented Mathematics has to be understood. In order to afford a real understanding of the process, the user has to learn new Mathematics. We called this occurrence a *motivating constraint* of the software (Dana-Picard 2006).

Generally, the word *constraint* evokes a limitation, an impossibility to go beyond a certain borderline. For a software package, this can be a limitation on the size of numbers, on the number of successive parentheses, etc. Among the most documented internal constraints are the finiteness of the screen for graphical applications, and the fact that the real numbers are always approximated by rational numbers. Following Balacheff (1994), Guin and Trouche (1999) distinguish three types of constraints of the artifact, called respectively *internal constraints* (linked to hardware), *command constraints* (linked to the existence and syntax of the commands), and *organization constraints* (linked to the interface artifact-user).

The constraint that we meet here is of a totally different nature: instead of limiting the user within the borders of a certain topic, the CAS demands from the user to go further, to learn a new theorem, a new technique. It is *a motivating constraint*, which leads to a broadening of the student's mathematical landscape. After its apparition, the mathematical knowledge is not supposed to be only shown anymore, the student is incited to learn the new theorem, and then becomes able to manipulate this knowledge, either with or without the help of the technology.

II. Pedagogy-embedded CAS.

Until recent times, the CAS did not give hints in order to find a pathway towards the solution of the given problem. This is not true anymore: pedagogical features have been implemented into Computer Algebra Systems. We call such systems *pedagogy embedded CAS.*

Derive 6 has a *step-by-step* feature, well developed for Calculus commands. Every step corresponds to one low-level command, as it implements one single theorem such as an integration formula. There exist surprising situations, e.g. the following formula is a central item:

$$(*)\quad \int_a^b f(x)\,dx = \frac{1}{2}\int_a^b \big(f(x)+f(a+b-x)\big)\,dx$$

As an example, look at the following integral: $I=\int_0^a \frac{x^p\,dx}{x^p+(a-x)^p}$, where p is a non-negative real parameter. For $p=0,1,2$, the computation is straightforward, but for larger integer values and for non integer values of the parameter, the work is non-illuminating. For given a, and for $p=1/3$, other CAS, where this formula seem not to be implemented, cannot generally compute the integral in a reasonable amount of time. Knowing the formula (*) enables to compute the integral with paper and pencil, within a few steps, and last but not least, for the general parameter:

$$\begin{aligned} I &= \int_0^a \frac{x^p\,dx}{x^p+(a-x)^p} \\ &= \frac{1}{2}\int_0^a \left(\frac{x^p}{x^p+(a-x)^p}+\frac{(a-x)^p}{(a-x)^p+(a-(a-x))^p}\right)dx \\ &= \frac{1}{2}\int_0^a 1\cdot dx = \frac{a}{2}. \end{aligned}$$

Formula (*) is not trivial; it is commented, and examples are given, in (Dana-Picard 2005b). An experienced lecturer, working in another institution, told to one of the authors: "I would not dare to ask my students to know such a theorem". We think that this implementation is a good opportunity to teach the theorem and some of its applications. As A. Rich says: “The transformation rules Derive displays are those *it* uses to simplify an expression. They may or may not be the same as those currently taught to students. However, if teachers see an advantage to an unfamiliar rule used by Derive, they may want to ask their students to verify the validity of the rule and then the students will have an additional tool in their arsenal” (Böhm et al., 2005, page 36).

This parametric integral has been proposed to an average student, named Ori, during the preparation to an oral examination. At first glance, as he thought that the parameter is a non negative integer, he proposed to decompose the integrand into a sum of partial fractions. The tutor showed him the Derive's step-by-step-solution.

Tutor: Do you recognize a known formula?
Ori shows Formula (*), then says: No, actually we have not been taught this.
Tutor: Can you apply the formula?
Ori: Yes. (works for a while); oh, I never saw this, you must teach this!

During another session, Ori is proposed the integral $I=\int_0^4 \frac{\sqrt{x}}{\sqrt{x}+\sqrt{4-x}}\,dx$. He says: this is the same case I saw last time; let us apply the formula.

Finally, at the end of the same tutorial session, he "receives" the following integral: $K = \int_0^{\pi/2} \frac{\cos x}{\cos x + \sin x} dx$.

Ori: It's not a power, but it must work the same way, as it's the same structure.

Tutor: And what about $L = \int_0^{\pi/2} \frac{\cos^p x}{\cos^p x + \sin^p x} dx$?

Ori: Surely the same thing.

He makes the work and claims: Oh yes! You must teach this in classroom!"

Maybe that without the *step-by-step*, the user would not have discovered the formula. Therefore, we consider this feature as part of the software's motivating constraints.

Maple is pedagogy embedded (via the Student package); here the conception is different from Derive's step-by-step philosophy, and the learning process induced by them develops otherwise. We present here an example in a different context.

Consider the following initial value problem: $\begin{cases} \frac{dy}{dx} - xy = x \\ y(0) = 1 \end{cases}$. Working with paper and pencil, a student is generally taught to use an integrating factor.

When using Maple's assistant for Ordinary Differential Equations, the student can choose the method: Lie methods, Classification methods, etc., but the usage of an integrating factor is not available for this exercise. A noticeable fact is that the pressing a button is accompanied by the (optional) translation of the command in Maple's language. The option "Laplace Transforms" leads to a much more complicated form. Therefore the student is incited to learn what these methods are, how they work and which benefit he/she can afford from their usage instead of what has been taught in regular class.

Tools *shape* the learning environment (Trouche 2004b), and the last influences the mathematical contents. The two embeddings of pedagogical features that we saw above, and the learning processes spanned by them are different. Note that each kind of software follows general algorithms, starting from pattern recognition, and whose sequential steps are based on the implementation of general theorems. The human brain works less sequentially, therefore intuition can lead to other pathways towards the solution of the exercise. With the integral of section II, we presented earlier an example of such a situation. This does not mean that technology has not been programmed properly: *a proper usage of technology does not require from the technology to mimic human actions*.

Let us compare briefly the two ways. On the one hand, Derive's step-by-step feature gives an indication on how the software works; if the student did not know how to solve the exercise, he/she has now an opportunity to understand by some kind of "post-mortem" analysis. Maybe an unknown theorem appears, as in our example, and the student can wish not only to discover it and to use it afterwards, but to try to have a more profound insight in its proof and its mathematical meaning. On the other hand, Maple's assistant lets some freedom of choice to the student, by offering different options before the computation is performed. This is still more evident when using the tutor for computing integrals. In this case, all the rules are presented as "buttons"; after a button has been pressed, an immediate indication is given whether the rule can be applied or not. If not, the student is invited to choose another rule, and so on.

Finally, we wish to note that even without a specific pedagogical feature, a CAS can be an instigator to further mathematical learning. This is the case in (Kidron 2003) for the conceptual understanding of the limit notion in the derivative, and deep learning of the theory is motivated by the usage of Mathematica.

III Instrumentation.

At the beginning, we saw Derive's step-by-step as providing the student with "a posteriori assistance", in order to understand what he/she would have been required to do. Actually, the usage of the step-by-step feature of the software can be considered as an "a priori" usage, in one of the following fashions:

- The user can discover a way of solving the problem either different from his/her way;
- Suppose that the student did not find how to solve the problem; he/she can ask for the first step (pressing the appropriate button) and the CAS opens a pathway. At every step, the student can abandon the step-by-step session. This is based on general theorems that the student does not automatically know.

When such a situation occurs in classroom, the teacher can build various activities, enriching by a large amount the mathematical knowledge and culture of the learners. If at the beginning, the student influenced the software's behavior in order to obtain the needed result, in the second scenario the software forces the educator to teach and the student to learn a new topic, a new theorem.

We have here elements of an *instrumentation process* (Chevallard 1992, Lagrange 2000, Artigue 2003 page 250, Trouche 2004a): "Les

potentialités et les affordances d'un artefact (en occurrence le CAS) favorisent le développement de nouveaux schèmes (ou font évoluer les schèmes antérieurs) de résolution d'un type de tâches (ici le calcul d'une intégrale définie)" (Trouche 2005; private e-mail). More briefly: "Instrumentation is precisely this part of the process where the artifact prints its mark on the subject" (Trouche 2004b, page 290).
Of course, this process is not reduced to the acquisition and internalization of one single theorem; the present examples are only one occurrence of the mechanisms involved.

IV. Contribution to the institution's culture.

We use the word "institution" in the sense of (Artigue 2002). Each institution has to decide whether to introduce the usage of a CAS in Mathematics courses or not to do so; not to deal with this issue is also a kind of decision. For example, the institution named JCT decided to teach MatLab and to use it in every engineering cursus.
Both authors act as coordinators of first year Foundation Courses in Mathematics, i.e. courses in which all Engineering students at JCT are involved. In a small subset of classes, which can also be viewed as an institution, the authors adopted other packages; for example, a course in Ordinary Differential Equations has been given last year together with practice sessions based on the usage of a CAS. The "institution culture" has already changed in certain classes, and is susceptible to change the institution's culture in a larger scale (e.g., all the first year Foundation Courses in Mathematics at JCT):

> "Tools are not passive, they are active elements of the culture into which they are inserted."(Noss and Hoyles 1996), page 58).

References

Artigue M. (2002). Learning Mathematics in a CAS Environment: The Genesis of a Reflection about Instrumentation and the Dialectics between Technical and Conceptual Work, *International Journal of Computers for Mathematical Learning* 7 (3), 245-274.

Balacheff N. (1994). La transposition informatique: note sur un nouveau problème sur la didactique, in M. Artigue et al. (edts), *Vingt ans de didactique en France,* La Pensée Sauvage, Grenoble, 364-370.

Böhm, J., Rich, A. and Dana-Picard, Th.(2005). About Stepwise Simplification, *Derive Newsletter* 57, 36-38.

Chevallard, Y. (1992). Concepts fondamentaux de la didactique: Perspectives apportées par une approche anthropologique, *Recherches en Didactique des Mathématiques* 12 (1), 77-111.

Dana-Picard, Th. (2005): Technology as a bypass for a lack of theoretical knowledge, *International Journal of Technology in Mathematics Education* 11 (3).

Dana-Picard, Th. (2005b): Parametric Integrals and Symmetries of Functions, *Mathematics and Computers Education*, Winter 2005, 5-12.

Dana-Picard, Th. (2006): Motivating constraints of a pedagogy-embedded Computer Algebra System, Preprint.

Garry, T. (2003): Computing, Conjecturing, and Confirming with a CAS Tool, in ***Computer Algebra Systems** in **Secondary School** Mathematics Education*, J. Fey et al. edts, NCTM, 137-150.

Guin, D. and Trouche, T. (1999). The complex process of converting tools into mathematical instruments: the case of calculators*, International Journal of Computers for Mathematical Learning* 3, 195-227.

Herget, W., Heugl, H., Kutzler, B. and Lehmann, E. (2000). *Indispensable Manual Calculation Skills in a CAS Environment*, http://b.kutzler.com/article/art_indi/art_indi.pdf.

Kidron, I. (2003). Is small small enough? Conceptualization of the continuous by means of the discrete. *Proceedings of the 5th International Mathematica Symposium*, 145-152, Imperial College Press, London, England (I.M.S 2003)

Lagrange, J.B. (1999): Techniques and concepts in pre-calculus using CAS: A two year classroom experiment with the TI-92*, International Journal of Computer Algebra in Mathematics Education*, 6 (2), 43-65.

Lagrange, J.B. (2000): L'intégration d'instruments informatiques dans l'enseignement: une approche par les techniques, *Educational Studies in Mathematics* 43, 1-30.

Noss, R. and Hoyles, C. (1996). *Windows on Mathematical Meanings: Learning Cultures and Computers*, Mathematics Education Library, Kluwer.

Pierce, R. (2001): Algebraic Insight for an Intelligent Partnership with CAS, *Proceedings of the 12th ICMI Conference*, Melbourne, Australia, 732-739.

Trouche, L. (2000): La parabole du gaucher et de la casserole à bec verseur: étude des processus d'apprentissages dans un environnement de calculatrices symboliques, *Educational Studies in Mathematics* 41 (2000), 239-264.

Trouche, L. (2004a): Environnements informatisés et Mathématiques: quels usages pour quels apprentissages?, *Educational Studies in Mathematics* 55 (2004), 181-197.

Trouche, L. (2004b): Managing the complexity of human/machine interactions in computerized learning environments: guiding students' command process through instrumental orchestrations, *International Journal of Computers for Mathematical Learning* 9, 281-307.