

# A computer program for the learning of algebra: description and first experiment.

Jean-François Nicaud, Denis Bouhineau, Hamid Chaachoua, Thomas Huguet  
MTAH, laboratoire Leibniz-IMAG, Grenoble, France  
46 Avenue Felix Viallet 38031 Grenoble Cedex FRANCE

Email: { Jean-Francois.Nicaud ; Denis.Bouhineau ; Hamid.Chaachoua ; Thomas.Huguet }@imag.fr

Alain Bronner

ERES, LIRDEF, IUFM de Montpellier, France  
2 Place Godechot, BP 4152, 34092 Montpellier Cedex FRANCE  
Email: alain.bronner@montpellier.iufm.fr

**Abstract:** We present APLUSIX, a computer system that helps students to learn algebra, available at <http://aplusix.imag.fr>. APLUSIX contains an advanced 2D editor of algebraic expressions that allows students to make the calculations (s)he wants, like in a paper-pencil context. The system verifies the student's calculations, by calculating the equivalence between two consecutive expressions, and shows the result to the student. It provides information concerning the progress of the resolution. Furthermore, the students can use commands like *Reduce*, *Expand* to ask the system to make some particular calculations. So APLUSIX combines features of a Microworld and of a CAS (Computer Algebra System) for education. This system has a lot of parameters allowing the teacher to choose the way it works. In particular, the teacher may active or not the verification of the equivalence between calculation steps, the progress indicators and the commands, to suit its own teaching strategy.

We present also an experiment of the APLUSIX system realised in September 2002 with ten classes of 15 to 17 years old students (grade 9, 10, 11) on the following tasks: expand or factor polynomial expressions, solve equations, inequalities or systems of linear equations. We describe the influence of some parameters on the students' behaviour and statistics concerning the work of the ten classes. We have observed and analysed the resolutions of the students. The analysis was made by replaying their actions (they are recorded by the system). We present some results of this analysis. Last, we summarize an inquiry among the teachers who participated to the experiment.

**Keywords:** microworld, CAS, verification, equivalence, equation

## 1. Introduction

Since the mid-eighties, several computer systems aiming at helping students to learn formal algebra have been realised. Almost all of them are menu-based systems that provide transformations rules in menus or buttons. With these systems, the student has to apply a rule to a sub-expression to perform an algebraic step, e.g., ALGEBRALAND [Foss 1987] for elementary equations, DISSOLVE [Oliver & Zukerman 1990] for equations, APLUSIX-TUTOR [Nicaud et al. 1990] for factorisations, MATHXPRT [Beeson 1990] for algebra and calculus, PAT [Koedinger et al. 1997] for the modelling of word problems and the resolution of elementary equations, l'ALGEBRISTA [Cerruli & Mariotti 2002] for the learning of a theory of algebra.

Menu-based systems have benefits, among them the possibility to have a shared work between the student and the system, e.g., the student chooses the rule and the system applies it, doing the calculations. They have also drawbacks: (1) some rules do not have a natural formulation because they are usually applied without being named; (2) when the field is not elementary, there are many menu items, sometimes more than 100; (3) these systems do not allow the student to say "this expression is my next step", obliging the student to find a rule in a menu to get it.

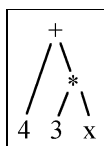
In 2000, we decided to build a new sort of system for the learning of formal algebra and we followed three major principles: (1) the student will present his/her calculation steps by providing the results; (2) non-intrusive feedback will be given to the student, including aspects linked to the correctness and to the progression of the calculations; (3) a few commands will be available to have some calculations made by the system, these commands depending of the level of the student. We have designed and implemented a system called APLUSIX with these goals. With points (1) and (2), this system has microworld features: the student is free to build algebraic expressions and reasoning steps. With point (3), this system has CAS (Computer Algebra System) features: commands to perform certain algebraic tasks. As far as we know, point (1) has never been realised for a large field; point (2) has only been implemented in the McArthur's system [McArthur & Hotta 1987] for elementary expressions.

In this paper, we present the APLUSIX system in section 2, 3, 4, 5 and 6. In section 7 and 8, we present an experiment realised in late 2002.

## 2. The editor of algebraic expressions

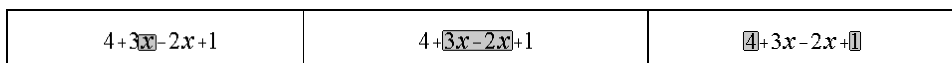
The usual representation of algebraic expressions is a 2D (two dimensions) representation in which one finds text-like parts, e.g.,  $2x+3y$  and box-like parts, e.g., the numerator and the denominator of a fraction. At the present time, there are a few editors of algebraic expressions that take into account this structure, e.g., MATHTYPE [MathType], which is the equation editor of Microsoft Word, and the editor of MATHEMATICA [Mathematica]. However, this text&box structure of algebraic expressions is not the right one. Having just this structure, one can select  $x+3$  in  $2x+3y$  which is algebraically incorrect because  $x+3$  is not a sub-expression of  $2x+3y$ . So we consider that a good editor of algebraic expressions must take into account the fundamental structure of the expressions resulting from the definition, having so an algebraic behaviour instead of just a text&box behaviour. We have replaced the explicit box found in text&box environment which are not pertinent from a mathematical point of view by implicit associations of elements driven by the algebraic structure of the expressions, associations that become explicit during selection, cut, copy, paste and drag&drop.

Definition (from the rewrite rules theory [Dershowitz & Jouannaud 1989]): *Algebraic expressions are recursively defined as: (1) constant symbols; (2) variable symbols; (3) operators applied to algebraic expressions.* For example, an expression can be built by applying the operator  $+$  to two arguments, the first one being the digit 4 and the second one the application of the operator  $*$  to the digit 3 and the variable  $x$ . The fundamental structure is a tree structure; the 2D form is the usual representation, see fig. 1.

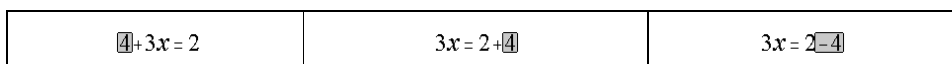


**Figure 1.** The fundamental representation of an expression having the usual representation  $4+3x$

The algebraic behaviour is particularly pertinent for selection, cut, copy, paste and drag&drop. In the APLUSIX system, the selection mechanism selects the sub-expressions and only them, see fig. 2; a paste over a selection is an algebraic substitution of a sub-expression by an expression; a drop at a cursor position is a combination of the dropped expression with the sub-expression near the insertion point with an operator, see fig. 3. In those activities, the algebraic structures of the expressions are explicitly shown.



**Figure 2.** On the left: we select  $x$  by dragging over  $x$ . On the middle: we continue to drag over  $2$ , the smallest sub-expression including  $x$  and  $2$  is selected. On the right: it is also possible to select multi-parts correct sub-expressions.



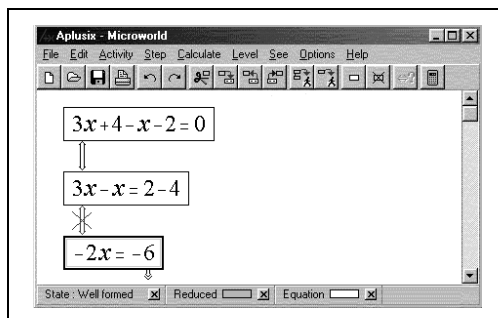
**Figure 3.** A typical algebraic transformation realised with a drag&drop of  $x$ .  $x$  is dropped (in the middle) with the  $+$  operator (the  $+$  operator is chosen by the system). To achieve the algebraic transformation, one hits the “-” key that changes the sign of the selection.

## 3. The environment of the student

One mode of the APLUSIX environment is called “The exercise activity”. In this activity, the student has to solve exercises stored in a file prepared by the teacher. The student develops his/her own calculation steps presented as frames by the system. The “duplicate” action allows to create a new frame with a copy of the previous expression; the editor allows to modify this expression to get the expression the student is thinking of.

When two steps are equivalent, a black equivalence arrow links the two frames. When they are not, a red crossed equivalence arrow links them.

When the student has finished to solve the exercise, (s)he tell the system. For a classical problem type (reduce, expand, factor, solve), the system analyses the resolution and agree or disagree. In the last case, the student is invited to continue to solve the exercises, but (s)he may also leave the exercises as it is.



**Figure 4.** The student made two steps. The first one is correct, the second one is not. The cross over the arrow is red and indicates that the equations of the two boxes are not equivalent. The status bar contains indicators of the state of the current step. Here, they say that  $-2x = -6$  is a well formed expression, is reduced and is not a solved form of an equation.

Besides the “exercise activity”, there is a “microworld activity” in which no exercise is provided. The student inputs an exercise and makes the calculations (s)he wants. In that mode, the end of the resolution is not analysed by the system.

## 4. The feedbacks

The first feedback of APLUSIX concerns the structure of the expressions. It relates to the syntactical level of algebra. Well-formed expressions are shown in black. Expressions like  $3x+(2x=0)$  are coloured in blue and said ‘out of domain’. Expressions like  $1/0$  are coloured in red and said ‘undefined’.

The second feedback of APLUSIX concerns the verification of the calculations. It relates to the semantic level. It uses the denotation (fundamental semantics) of the expressions. The domain of APLUSIX and the denotation are described in table 1.

**Table 1.** Domain of APLUSIX and denotations used.

Sub-domain	Constraints	Denotation	Verification process
Polynomial expressions	Any number of variables, any degree	The polynomial	Comparison of the coefficients of the canonical forms
Rational expressions	Any number of variables, any degree	Fraction of two polynomials	For A/B and C/D, comparison the polynomials AD and BC
Equations and inequalities of one variable	Equivalent to polynomials equations with degree $\leq 4$	Set of solutions	Comparison of the sets of solutions in a canonical form
Systems of linear equations	Maximum: 10 equations, 10 unknown	Set of solutions	Comparison of the solutions or of canonical forms when there is an infinity of solutions

The third feedback of APLUSIX concerns the progression of the calculations. It relates to the strategic level of algebra. Six indicators are available at the present time: *Sorted* is an indicator relating to the form of the expression. *N-factored* relates to arithmetical aspects, indicating if there are numerical factors that can be factored out from an expression. *Reduced* is an indicator relating to arithmetical and algebraic aspects, e.g. additive reductions  $ax+bx \rightarrow (a+b)x$ , multiplicative reductions  $x^n x^p \rightarrow x^{n+p}$ , and simplifications of fractions. *P-factored* and *Expand* are

indicators relating to algebraic aspects, *P-factored* indicates if there are non constant polynomials which can be factored out from an expression. *Equation* is an indicator relating to didactical aspects, it indicates, for equations, inequalities and systems of equations, if the expression corresponds to a solved form. Each indicator is given as a progress bar evolving from empty to full in few steps.

In “The exercise activity”, a fourth feedback concerns the analysis of the end. The APLUSIX system verifies that the given expression is a solved form, and, when the ‘verification of equivalence’ parameter is enabled, the system verifies that the path from the problem statement to the proposed solution consists of equivalences only.

## 5. The commands

The students can use commands to ask the system to make some particular calculations. This introduces ed-CAS (Computer Algebra System for education) compartment. The set of commands depends of a parameter relating to the level of the student. The level of the student is chosen by the teacher. The level 0, 1, 2 and 3 are arithmetical levels for the following sets of numbers: integers, rational numbers, decimals, and real numbers. Level 4 is an algebraic level. The commands and levels of APLUSIX are described in table 2.

**Table 2.** Commands and levels of the APLUSIX system.

Level	Calculate	Expand	Sort	Conversion
0	not available	not available	not available	not available
1	for integers	not available	not available	not available
2	for decimals	not available	not available	not available
3	for rational numbers	not available	not available	rational nb. ↔ decimals
4	calculations for real numbers combined with some reductions	for all expressions	for all expressions	rational nb. ↔ decimals

The available commands act on the selection made by the student. As a consequence of the selection mechanism, it means that the calculations are made about sub-expressions, the result of a calculation replaces the selection and if necessary parentheses are added to ensure to have an algebraic substitution.

## 6. The parameters

First of all, it seems worth to recall that within the system four activities are provided. The first two ones are the microworld activity and the exercise activity, and concern the student. The list of exercises used by a student can be prepared by the teacher. The third and fourth activities, i.e., the replay system and the environment dedicated for the construction of lists of exercises are devoted to the teacher. The difference between microworld activity and exercise activity have already been evocated, it remains to give precisions about the parameters given to the teacher to shape the environment to his students.

The most important parameters concern the feedbacks and commands available to the students. Concerning the calculus of the equivalence, the system can be parameterised to show the equivalence (1) every time, (2) only when the student asks for, or (3) never. Feedback on the end of exercise fully apply only when (2) or (3). The level of the command and the indicators are also important feedbacks and corresponds to parameters fixed by the teacher.

There are less important parameters. For example, at the syntactical level, there are parameters to set the way to input operators, i.e., are unbalanced parentheses allowed or not? Does a ‘(’ struck on the keyboard introduce ‘(?)’ or just ‘(’ ? Does a new step in the tree of the resolution introduce an empty expression ‘?’ or duplicate the current expression? Etc. At the level of semantics, for example, does new steps are allowed when the current step is not equivalent to the previous one? Etc. At the didactical level, for example, does the system ask for a commentary about the steps realised by the students? Should these commentaries be written a priori, or a posteriori? Have they to be chosen in a predefined list? If so, what is this list? Etc.

In the current system, there are 15 parameters or family of parameters, they introduce  $840 \cdot 32 \cdot 4 \cdot 2^{96} = 1\,321\,205\,760$  different systems. Each system differs from the other about significant behaviours relevant to the algebraic field or to the didactical field. Some can be seen as details, some not, from a general point of view, but for each, at one moment of the learning process, it seems reasonable to say that they are important.

## 7. The experiment

In September 2002, we conducted an experiment in France, which aimed at establishing a state of knowledge of the students (aged from 15 to 17) in algebra. This experiment was composed, for each grade, of three sequences of exercises of nearly the same difficulties (each sequence contains thirty exercises about polynomials to develop, reduce, factor, and about equations, inequalities, systems of equations to solve). This experiment last for about one month with three levels: 91 students of grade 9, 108 students of grade 10, and 74 students of grade 11. Each sequence took about one hour. The teachers were asked to give no mathematical help to the students. Students were allowed to use calculators, paper and pencil.

In the first situation, the system was parameterised so that no feedback about the equivalence of two steps of the resolution was given. In the second situation, the system showed the equivalence between two steps if and only if the student asks for it. In the third situation, the system was parameterised to always show the equivalence between two steps of the resolution. The exercises given to the students concerned about all the notions of the curriculum of the past year.

The system recorded the student's interactions. This gave us an important amount of data useful for the analysis. This permits also to replay the student's resolutions.

For each class, we have realised different analyses with several levels and various points of view.

### 7.1. Global study

Our first level of analysis is a comparison about the percentage of success between the three situations for each class. The results show a significant progression of this percentage in the third situation, in every class. We are aware that this experiment leads to some learning even if it was not intended to. We hypothesize that the crucial factors that explain this progression are the initial contact with the software and the feedback about the equivalence of the steps of the resolution.

### 7.2. Analysis of the results with respect to the problem type

An analysis of the results of two classes of 1<sup>re</sup> S (grade 11, scientific track) with respect to the problem type have shown that students still have difficulties with exercises of the *solve equation* and *solve inequalities* types (see table 3). This was surprising for these classes because of their expected level. Periods of teaching about those problem types are still necessary for these students.

Table 3. Percentage of success with respect to the problem types.

Problem types	1 <sup>re</sup> S(a)	1 <sup>re</sup> S(b)
Calculate	86	82
Factor out	83	81
Develop	66	73
Solve equation	60	60
Solve inequalities	45	48

### 7.3. Comparison of the results with respect to the exercises and study about variables

This experiment gives us the opportunity to point out the exercises with a bad percentage of success, and to identify possible didactical variables [Brousseau 1981] which could influence the strategies and procedures, for example:

- The nature of the numbers to be calculated and the coefficients;
- For the *factor* problem type: the existence of implicit numerical factor 1;
- For the *expand* and *factor* problem types: the number of terms, the number of symbols, the complexity of the terms (first degree polynomials, square or product of first degree polynomials), the number of rules to apply, the existence of a numerical factor  $-1$ .

### 7.4. Study of the student's knowledge and strategies

The replay system of APLUSIX allows accurate analysis of the student's behaviour, it permits to identify most of algebraic rewriting rules (correct or incorrect) used by the students and to try to discover the strategies used in the resolution of the algebraic problems. It gives us information about the use of the interface and about the student's instrumentation of the functionalities of the system.

Several students' rules are erroneous and some are steady for some students. For example, a lot of students factored the expression  $(4x-8) + (4x-8)(6x-9)$  and gave  $(4x-8)((4x-8)+(6x-9))$  as a result. We represent this transformation with the following rewriting rule:  $A + AB \rightarrow A(A+B)$ .

We have currently identified about one hundred of erroneous rewriting rules.

The feedbacks given by the system about the equivalence between steps forces the students to modify their strategies and make them disqualify the erroneous rewriting rules. Consequently, the system constitutes a milieu that favours the learning of algebra [Brousseau 1990].

## 8. The enquiry

We have asked 6 teachers, who have participated to the experiment, to answer a questionnaire about the integration of the APLUSIX system in their classroom. They find that the initial contact with the system does not introduce any difficulty for the students. On the one side, the teachers think that the APLUSIX system is a good tool for the learning of algebra, allowing the reinforcement of the acquired knowledge and the correction of invalid knowledge. On the other side, the APLUSIX system is not considered as an environment for the introduction of algebraic notions and techniques.

Very often, the autonomy gained by students with the system was emphasized as a positive aspect. Actually, students do not need the teacher to validate their solution. This relates to the importance given to the verification of the equivalence that we have already seen. Thanks to this form of autonomy, the teachers have proposed to students to use the system freely at school or at home as a continuation of the work conducted in the classroom.

## 9. Conclusion

APLUSIX is a new sort of computer system for the learning of algebra. Some of its main features have already proved their interest: the microworld aspect, the basic functions of the editor, and the verification of the calculations. Other features need adapted experiments to be evaluated, in particular the higher functions of the editor (cut, copy, paste, drag&drop), because they were little used (they were not taught) during the first experiment, and the commands, because we chose to provide no command to the students in this experiment.

The first experiment showed that the students and the teachers like APLUSIX, and that learning occurs with this system, although the goal was just to collect data to model the students.

The APLUSIX system can be currently used freely. It can be downloaded at: <http://aplusix.imag.fr>. It is available in French and English. It will be soon available in Portuguese and Japanese. It will become a commercial product.

The APLUSIX system is also a good tool to study the learning of students, as it records the interactions, allowing to analyse the files by hand or by program, and to replay the students' work. We are currently analysing a lot of students' data, building students' rules currently and students' conceptions soon. Some of the computer programs that we develop for this goal will be transformed later into tools for researchers and teachers.

## References

- Beeson, M. (1990). Mathpert, a computerized learning environment for Algebra, Trigonometry and Calculus, Journal of Artificial Intelligence in Education, p. 65-76.
- Brousseau, G. (1981). Problèmes de didactique des décimaux, RDM Vol 2.1, La Pensée Sauvage, Grenoble
- Brousseau, G. (1990). Le contrat didactique : le milieu, RDM Vol. 9/3, La pensée sauvage, Grenoble.
- Cerruli, M., Mariotti, M.A. (2002). L'ALGEBRISTA : un micromonde pour l'enseignement et l'apprentissage de l'algèbre. Sciences et techniques éducatives, vol. 9, No 1-2, Hermès, Paris.
- Dershowitz, N., Jouannaud, J.P. (1989). Rewrite Systems. In Handbook of Theoretical Computer Science, Vol B, Chap 15. North-Holland.

Foss, C.L. (1987). Learning from errors in algebra. IRL report No IRL87-0003

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *Int. Journal of Artificial Intelligence in Education*, 8.

Mathematica [www.wolfram.com](http://www.wolfram.com)

MathType [www.mathtype.com/msee](http://www.mathtype.com/msee)

McArthur D., Hotta Y. (1987). Learning problem-solving skills in algebra. *Journal of education technology systems*, 15.

Nicaud, J.F., Aubertin, C., Nguyen-Xuan, A., Saïdi, M., Wach, P. (1990). APLUSIX: a learning environment for student acquisition of strategic knowledge in algebra. *Proceedings of PRICAI'90*. Nagoya.

Oliver, J., Zukerman I. (1990). DISSOLVE: An Algebra Expert for an Intelligent Tutoring System. *Proceeding of ARCE*, Tokyo.