Variables in Computer Algebra, Mathematics, and Science

by André Heck
AMSTEL Institute, University of Amsterdam, Kruislaan 404, 1098 SM Amsterdam, The Netherlands. heck@science.uva.nl

We discuss the notion of variable in computer algebra, mathematics and science. Central are the questions how variables are actually used in mathematics and science, how computer algebra supports the various uses of variable, what conceptual differences exist, and what consequences follow from this for teaching, learning, and doing mathematics with a computer algebra system.

1. Introduction

In the educational research literature, little thought has been for the differences between the concept of variable in computer algebra and the notion of variable in mathematics and science. Yet one is immediately confronted with these differences when using a computer algebra system or a symbolic calculator. The syntax of the input and the interpretation of the results given by the computer algebra system form a barrier to many pupils at higher secondary school level and first year students in higher education. For example, see (Wain, 1994), (Guin and Trouche, 1998), and (Drijvers, 2000). Some of the difficulties in using a computer algebra system could have been avoided by developers of the systems; in many cases, lowering the level of mathematical sophistication or allowing customisation by the teacher would already help. Other difficulties are more fundamental or related to shortcomings in the students' knowledge of mathematics. Only developing good teaching strategies for learning to work with a computer algebra system can level this barrier out.

In this paper we shall address the following three topics: (i) the many possible uses of letters in mathematics and science; (ii) the way letters are used in computer science, in 'ordinary graphing calculators', and in symbolic systems; (iii) consequences for educational use of computer algebra. We shall list differences between the algebraic representations in a symbolic system and in traditional mathematics and science. This helps us to identify and understand some obstacles that teachers and students encounter while working with a symbolic system. It also shows how complex and subtle the relationship between computer algebra use, pencil-and-paper work and algebraic thinking really is. Our study of the consistency and discrepancy between the use of variables in mathematics and science is also motivated by the current development at the AMSTEL Institute of an integrated computer learning environment for mathematics and science education at secondary school level. This work builds on earlier research work that resulted into the Coach environment for science education (Heck, 1999; Mulder, to appear). The main problem in the design of the integrated environment, already present at the level of building a computer model of variable and algebraic expressions, is how to comply with the requirements that both mathematics and science make for the educational tool.

This paper is organised as follows. First we describe the notion of variable in graphing calculators. The reason for doing this is that these tools, which are widely accepted for numeric and graphical computations, already use variables in a way that differs from standard algebra. These differences are reinforced when symbolic computation comes into play, as we shall see in the next section on the use of variables in computer algebra. In the following sections we juxtapose these computer variables and the notion of variable in mathematics and science. We conclude with a discussion about consequences for teaching and learning mathematics and science. This is done in the form of recommendations to users of existent computer algebra systems.

2. Variables in Graphing Calculators

In a graphing calculator, a variable can be considered as a lettered box for storing and retrieving numerical values. To use a variable you need to know two things about it: its name and its type. The type of the variable specifies what values can be stored or how the contents must be interpreted. We speak about a computer variable because it also occurs in conventional programming languages such as FORTRAN and C. This concept of variable differs essentially from the notion of variable in mathematics:

- A variable in a graphing calculator always has a value.
- A variable in a graphing calculator can play more than one role in a single statement.
- The rules for manipulating computer variables differ from mathematical manipulation rules.
- Some computer variables do not exist in mathematics and have a special meaning.
- Arithmetic with numbers on a graphing calculator has its own rules.
- Some computer variables can be used in manners that standard mathematics does not allow.

Below, we work out these differences.

A computer variable always has a value and this value will often change during a computation, for example, by assignments like $x = 3$ and $x = x + 1$. In the last assignment, the following task is specified: take the current value of *x*, add 1, and store the result in the variable *x*. Here, the symbol *x* plays two roles: on the left-hand side it can be pictured as a lettered box for storing a value, and on the right-hand side it denotes a value retrieved from a lettered box. In mathematics however, a variable need not have a value and its role, when it appears in an expression or definition, is always fixed. The last assignment would probably be represented in mathematics by a recursive definition like $x_{n+1} = x_n + 1$.

Manipulation of computer variables differs from the algebraic computation in mathematics. For example, ordering of statements is of more importance in computer software and calculators than in mathematics. In many programming languages and systems, the following snippet of computer code, in which assignments are separated by semicolons, will exchange the values of the variables x and y: t = x; x = y; y = t; The ordering of statements cannot be changed without ruining this meaning. In mathematics, they are just three equations in three unknowns. Their solution can be expressed in any of the three variables, e.g., the solution is $x = t$ and $y = t$, with t free to choose. Here, the ordering of the equations does not matter.

Some variables have special meaning in a graphing calculator: for example, in a TI-83, the variables ENTRY and ANS are used for storing and retrieving the last entered input and the last computed result, respectively. An expression like $\mathrm{ANS}+1$ is interpreted as the command "add one to the previous result", which can be repeated by pressing the Enter key. $\mathrm{ANS}+1$ corresponds with the arrow $\xrightarrow{+1}$ in an operation diagram. In general, any expression that contains the variable ANS is in fact a value recipe, i.e., it is a procedure for doing something with a value in order to produce a new value. In conventional mathematics, the use of an expression as a recipe does not often occur in such an explicit way: only during the process of manipulating algebraic expressions, equations such as $x^2 + 2xy + y^2 = (x+y)^2$ and $\sin^2 x + \cos^2 x = 1$ are interpreted in this way. Note that the ordering of writing down the equation plays a delicate role here: reading from left to write, the first formula $x^2 + 2xy + y^2 = (x+y)^2$ evokes the idea of simplification, whereas the mathematically equivalent equation $(x+y)^2 = x^2 + 2xy + y^2$ brings more to mind a process of computation.

Computing with numbers on a graphing calculator has its own rules. Many calculators offer a mixed mode of computing and display of results: irrational numbers are first approximated by rational numbers before further processing, and large magnitude rational numbers as well as small magnitude rational numbers are displayed in decimal notation or in scientific notation. Overflow may cause surprising results: an extreme case is the computation of 12^{3456} on a TI-92, which results in the answer ∞, together with a warning message that overflow was replaced by ∞ or -∞. To put it mildly, a well-known mathematical symbol is used in an unusual way.

Although the earlier examples suggest that variables have numbers as values, a graphing calculator like the TI-83 provides more types of variables, e.g., matrices, lists, number sequences, and function variables. List variables allow the user to do arithmetic with lists of numbers. For example, adding 10 to the list variable L_1 with current contents $\{1,2,3\}$ via the statement $L_1 + 10$ results in the list $\{11,12,13\}$. You can use more complicated formulas with list variables: if $L_1 = \{1,2,3\}$, then ${L_1}^2 + 1$ will give $\{4,13,28\}$. Here, an algebraic formula is used as a procedure for doing something with lists of numbers in order to produce a new list of numbers. However, this arithmetic of list variables has two side effects: (i) it extends the domains of definitions of well-known mathematical functions; (ii) it leads to new objects. For example, the sine function is not only a real or complex function anymore, but it can also be applied to lists of numbers: entering $\sin(\{1,2,3\})$ will give the result $\{0.84, 0.91, 0.14\}$ to 2 significant digits. The second side effect can be illustrated with the definition of $y_1 = x + \{0,1,2\}$ on a TI-83. On the one hand, it is an object symbolised by y_1 that, like a function, maps a single number to a three-tuple, but on the other hand it displays in the graph screen as three lines with corresponding equations $y = x$, $y = x + 1$, and $y = x + 2$, respectively. So, the variable y_1 symbolises at the same time one function that maps on a three-dimensional space and a three-tuple of real functions. This kind of overloading of notation obscures the well-defined mathematical notion of function.

Variables denoting number sequences and function variables can be seen as the implementations in the graphing calculator of variables that symbolise changing magnitudes. For example, after defining the function $y_1 = 2x + 1$, one can define a function variable y_2 in terms of y_1, say $y_2 = y_1^2$. In a table of x, y_1, and y_2 one sees, by default, values of y_1 and y_2 tabulated for integer values of x. By changing the table settings, step values other than 1 can be chosen. If values of x change, then values of y_1 and y_2 change accordingly. Simply by pressing upward and downward arrow keys one can see more values being tabulated. The three variables are at any time related to each other.

Occasionally, linking variables has surprising side effects: for example, after plotting a graph of a function variable, say y_1, this variable will have a value determined by the last value of x used to draw the plot. In the default setting of a TI-83, after entering the function variable y_1 as $y_1 = 2x + 1$ and plotting its graph, the result of entering y_1 in the home screen will be the number 21. After pressing the `Trace` key and moving the cursor to another point on the graph, the values of x and y will be the horizontal and vertical coordinates of the chosen point on the graph. However, because of the link between the variables x and y_1, the value of y_1 will also be the vertical coordinate of the chosen point. What we conclude for this brand of graphing calculator is that y_1 is used in two ways, viz., as a normal variable with a numeric value and as a name of a function, which can be used to define other function variables or to compute function values. The screen that is active determines in which ways the variable can be used.

Despite the remarkable differences between variables in a graphing calculator and variables in mathematics, many research papers and reports describe how graphing calculators can achieve significant improvements in pupils' understanding of the concept of variable. For example, in (Graham and Thomas, 2000) and (Gage, 2001) is reported how the calculator model of a variable as a lettered store for numbers can help pupils to gain good understanding of 'letter as placeholder' and to overcome obstacles common in the early stages of the transition from arithmetic to algebra. In (Cedillo, 1997), the results of a classroom research study indicate that the use the calculator's symbolic language can help pupils to understand the use of literal symbols to generalise number patterns and that it facilitates pupils' investigation of algebraically equivalent expressions.

3. Variables in Computer Algebra

Computer algebra introduces another kind of computer variable, viz., one that does not serve as a store for numerical data, but one that points just to an object. This object, which is also called the value of a variable, can be almost anything: a number, a symbol (mostly letters or nouns), a polynomial, an equation or inequality, a differential equation, a function definition, a list of numbers, a graph, a table, a previously computed result, and so on. In many cases, a variable points to an algebraic expression. During the so-called evaluation of an expression it is transformed from one representation to another by replacing the variables that occur in the expression by their values. This process can be carried out recursively, in which case one speaks about complete or full evaluation.

When one applies a command to an expression, this expression suddenly gets its meaning. For example, the differentiate command implies that the expression is considered as a function in the variable(s) with respect to which one differentiates. The solve command implies that the symbols in an expression stand for yet unknown numbers that must be computed or be expressed in terms of parameters. And more examples could be given. Errors occur when the form of an expression is improper for carrying out a particular task. Source may be a faulty translation of a mathematical task into a computer algebra task or a syntactically correct translation that is however semantically wrong (re-ordering of statements may already have this effect).

In most computer algebra systems such as Derive, Maple, *Mathematica*, MuPAD, and Axiom, it is not necessary to declare variables and other objects. This would make these systems difficult to use because during a computation objects often change of type. However, this does not imply that an object has no type: in MuPAD and Axiom, which are strong type systems, it always has, but it is automatically determined during runtime or it is specified explicitly by the user. Other systems like Derive, Maple, and *Mathematica* are type-less systems: although objects have in strict sense no type information connected to them, one can ask in these systems whether a particular expression can be interpreted as one of certain type. For example, one can ask whether x^2+x+1 is a polynomial over the integers or not. One can also associate properties with a variable so that one can be more specific about the set of values that a variable can possibly take. For example, one can declare a variable as a positive real number and the computer algebra system will then take this into account during computations.

Although the usage of variables in computer algebra systems gives the impression that they have more in common with variables in mathematics than variables in a graphing calculator or in a conventional programming language have, computer algebra variables are in fact very different in quite some aspects. The differences between computer variables and variables in mathematics listed in the previous section remain or are even amplified. Below, we list the properties of computer algebra variables that make them essentially different from variables in mathematics:

- A computer algebra variable always points to a value.
- Manipulation of computer algebra variables has its own rules, in which internal storage of expressions, automatic simplification, and evaluation scheme play a role.
- An expression can represent a mathematical object as well as describe a particular process to be carried out. Both notions are frequently used.
- Some variables have special meaning distinct from standard mathematics.
- Although modern computer algebra systems try to mimic mathematical notation as much as possible, their users still have to translate to and from standard notation on many occasions.
- In computer algebra, there is a strong focus on solving generic problems, i.e., special cases such as special values of parameters are not taken into account.

Below we work out these differences.

Although a computer algebra system seems to allow variables just as symbols, they are behind the scenes still pointers to values. In *Mathematica*, for example, a variable used as a symbol is in fact internally a pointer to an object with header `Symbol` and argument the string `"x"`. One may think that it is in practice unimportant to know exactly how variables and expressions are stored inside a computer algebra system and how the program internally manipulates them. But this is not true: knowledge about internal representation of objects is needed to fully understand which mathematical subexpressions are recognised in a given expression. For example in Maple, replacing $1+x$ by y in $1+x+1/(1+x)$ via the `subs` command gives the result $1+x+1/y$ instead of $y+1/y$, as one may have expected. The reason is that Maple stores the expression $1+x+1/(1+x)$ internally as a sum of three sub-expressions, viz., 1, *x*, and $(1+x)^{-1}$. Maple does not recognize at the sum-level the subexpression $1+x$; at the power-level it does find this subexpression. Because the `subs` command only does a syntactic substitution it fails to give $y+1/y$; another command, viz., `algsubs` is required to do the semantic substitution. One can also apply simplification with respect to side relations, based on Gröbner basis theory. Related to this technique, an interesting remark has made in (Recio, 1998), viz., that the computer troubles for deriving conclusions from equalities through rewrite rules may actually make the students' difficulties with such tasks more explicit and better to understand.

A user of a computer algebra system must also develop some feeling for the algebraic manipulations and sorting of terms that are automatically carried out by the program to bring an expression into some "standard" form. Otherwise s/he may get frustrated that some obvious and correct manipulations cannot be done. For example, in Maple, $2(x+3)$ is automatically expanded into $2x+6$ and there is nothing to do about it. In *Mathematica*, when you input `Exp[x]^2`, the output is automatically converted into e^{2x}. This makes the reverse operation practically impossible in this system. On a TI-92, the terms of a univariate polynomial are in the expanded form always sorted in descending order with respect to the degree and no other orderings are available. Whether one is happy with such automatic simplifications depends very much on the reasons for using a computer algebra system. When it is used as an educational tool to learn and practice manipulation of algebraic expressions, one does not always want these simplifications automatically done. This explains why Derive, which has been designed primarily for education, refrains from automatic simplification. But when one uses a computer algebra system as a computational tool, which automates mathematics as much as possible, then one appreciates more its computational power and automatic simplification.

Anyway, there are plenty of cases in which there is little or no doubt about which expression is simpler: $x+0$ should simplify to x, $\sin(\pi)$ should simplify in computations to 0, and $5x$ is simpler and more readable than $x+x+x+x+x$. For reasons of usability and efficiency such 'obvious' simplifications are carried out automatically on most symbolic systems. Any other simplification is left to the user's control; the computer algebra system only provides the tools for doing jobs like factoring expressions, working out brackets, and collecting terms. However, one should realise that there is always some balance between rigorous mathematical correctness, usability and efficiency. For example, the automatic simplification of $0\times f(1)$ to 0, how obvious it seems, is not always correct. An exception is the case that $f(1)$ is undefined or infinity. In fact, the automatic simplification is only wanted if $f(1)$ is (known to be) finite, but difficult to compute.

Ordering of statements is in most computer algebra systems relevant as the following snippet of computer code, in which assignments are separated by semicolons, illustrates: `a = 2; y = a*x; a = 3;` In *Mathematica*, the value of `y` after executing the above instructions will be `2x`, i.e., the last assignment has no effect on the evaluation of `y`. In fact, every computer algebra system in which assignments are by default not delayed will evaluate in this way. In mathematics, however, the definition of y via the relationship $y=a\,x$ means that y is considered in the definition as the product of a and x, regardless of the current value of a. Only the evaluation of y changes during computations. If the value of a changes from 2 to 3, then the evaluation of y changes evaluation from $2x$ to $3x$. Derive, which uses by default delayed assignment, works in the mathematical sense. If the first two statements in the above computer code are interchanged, then *Mathematica* will also react in the mathematical way. The difficulty with the computer code of the example is that the variable a is actually used in two ways. In the first assignment as a normal programming variable, which is not evaluated but whose pointer is set to some value, and in the second assignment as a symbol representing a mathematical object, which is immediately evaluated.

The basic mode of working with a computer algebra system consists of dividing a given task into subtasks that can be carried out sequentially. Starting with the first subtask, one repeats this process of subdividing a task into smaller pieces until one arrives at a point where one is confident that one can carry out the first subtask without great difficulty or where one wants to give it a try. This subtask, which is in general much smaller than the original one, is then carried out by performing cyclically the following chain of actions:

1. On the basis of the previous result, decide what computational step comes next.
2. Figure out what symbolic input corresponds with the chosen action and enter it.
3. Monitor the internal processing of the computer algebra system and interrupt it because of time and memory constraints or because of change of mind.
4. Read and interpret the symbolic, numeric, graphical, or textual output. Again, differences in notation have to be faced.

The input may be an expression to be evaluated or a process to be carried out. For example, the input `x^2 + 1` may be used to introduce a polynomial or to add 1 to the square of x. Note that one symbolism is used for two purposes, viz., specification of a computational process and notation for a mathematical object, which may be the result of a process. On the input side, a user is also confronted with differences in mathematical notation and computer notation. Computer algebra distinguishes between the equal symbol in equations and in assignments, it forces the user to use brackets and operators for specification of intentions, it reserves names for various purposes (e.g., a symbol for referring to previous results and special names for mathematical constants), commands replace standard mathematical notation for integration, differentiation, and summation, etc.

All steps are of rather high level of abstraction: finding a strategy to solve a problem, translating mathematical concepts into symbolic input to be processed, using algebraic expressions both as objects to be manipulated and as processes to be carried out, interpreting intermediate results that may contain unfamiliar mathematical ingredients, making a decision on what to do next, and so on. Tall (Tall, 1992) identifies in a study of prerequisite knowledge that students need in order to work meaningfully with symbolic manipulators as the Achilles heel of these systems in education:

- The strong focus on symbolic input.
- The need for the individual to construct a meaning for the symbolism as flexible process and object.
- The internal processing of input in a manner that may not be transparent to the user.

We would like to add to this list:

- Computer algebra systems have been designed to solve generic problems only.
- Computer algebra systems are not always able to use all mathematical rules or assumptions they know in the right places.
- Internal processing of input cannot be adjusted to one’s needs. In particular, it is difficult for a user to control automatic computational steps like collecting and reordering of terms and applying rules for mathematical functions.

The focus on solving generic problem causes errors when first symbolic computations are carried out and afterwards concrete values are substituted in the result. For example, most computer algebra systems compute the antiderivative of x^n as $x^{n+1}/(n+1)$, which is wrong (even in the limit case) for $n=-1$. All computer algebra systems give two solutions when asked to solve the quadratic equation $ax^2+x+1=0$, which is wrong for the special case $a=0$. All systems compute the reduced row-echelon form of the 2×2 matrix with rows (1 1) and (*x* 1), respectively, as the identity matrix, ignoring the special case $x=1$. In mathematics however, distinguishing between special cases is an important aspect of solving problems and much attention is paid to it in education.

The inability of symbolic manipulators to use all knowledge properly can have side effects like finding too many solutions of a problem or obtaining erroneous results. For example, solving the cubic equation $x^3+x+1=0$ under the assumption that one works over the real numbers still yields complex solutions in *Mathematica*, Maple, and other systems. Not recognising zero in an intermediate result in combination with automatic simplifications involving 0 can be source of wrong answers, too: some systems compute $\lim_{\varepsilon\to 0}\dfrac{\varepsilon}{(\sin^2\phi+(1+\varepsilon)\cos^2\phi-1)}$ as 0, whereas the expression after application of the simplification rule $\sin^2\phi+\cos^2\phi=1$ does not depend ε on anymore, but equals $\dfrac{1}{\cos^2\phi}$.

4. Variables in Mathematics

In mathematics, variables are used in so many different ways that hardly any attempt is made to define them rigorously, except in the field of formal logic and abstract algebra. See (Schoenfeld and Arcavi, 1988) and (Wagner et al, 1984) for discussion about the notion of variable, its richness, the multiplicity of meanings, and why it is such a difficult concept to learn. The following examples (Etten, 1980) illustrate in how many different ways variables already appear in school mathematics.

$f: x \mapsto 2x+1$	$A = l \times w$	$a+b=7$
$a^2-9=(a-3)(a+3)$	$x \in \mathrm{N}$	$x+3=x+3$
n is a divisor of 24	$\sqrt{x^2}=x$	$x+3=2x+8$
p is a prime number	$x=y$	$A=2\pi r$
$a+b=b+a$	$a^2=9$	$S \subset \mathrm{Q}$
$\cos x+\sqrt{3}\sin x=1$	$x<9$	$x^2+y^2=z^2$
k is parallel to the *x*-axis	*P* lies on *l*	

Even if letters are used for numbers only, different roles of letters in the algebraic context can be distinguished (Kücheman, 1981; Usiskin, 1988). It may be

- an *indeterminate*, in statements like $a^2 - 9 = (a-3)(a+3)$.
- an *unknown,* in equations such as $a + b = 7$.
- a *known number* like π.
- a *variable (generalised) number*, e.g., in $x \in \mathrm{N}$, in declaring p a prime number, and in differences like $f(a+1) - f(a)$.
- a *computable number* like *A* in the formula $A = 2\pi r$.
- a *placeholder*, e.g., in function definitions $f : x \mapsto 2x+1$ or $f(x) = 2x+1$.
- a *parameter,* e.g., as a label in the function definition $f_p(x) = p\,x$ to distinguish several cases.
- an *abbreviation* like $V = \{1,2,3\}$.

Mathematical notation has evolved into a powerful and flexible symbolism. The simple expression $a(x+e)$ can already be used in various ways: as a generalised number $a \times (x+e)$ in which the symbol *e* may or may not stand for the base of the natural logarithm, as a function *a* applied to $x+e$ (in this case, *a* is usually not in italics), as a function in *x* with parameters *a* and *e*, as a function in two or more indeterminates, and as the instruction *a* applied to the argument $x+e$. The mathematical context or the wording used about the expression often gives a clue. But it is no big surprise that a secondary school pupil not always understands mathematical statements and expressions, and that s/he often makes a mess of notations. It takes time and practice to get used to the fact that a variable actually gets meaning in mathematics through its use (as indeterminate, as unknown, as parameter, etc.), through its domain of values, through its associated truth set if used in an open statement, and through the context in which it is used.

It is almost impossible to rigorously define the concept of variable, but this does not mean that one cannot classify the various appearances of variables in mathematics. Freudenthal (1983) distinguishes the following three uses of variable:

1. as a *placeholder*, which denotes the places in an expression where the same object is meant.

Variables as placeholders mostly occur in function definitions: for example, the definitions $f(x) = x^2$ and $f(y) = y^2$ both define one and the same function, viz., the square function. One also refers to these placeholders as dummy variables: they do not indicate objects anymore, but rather the locations for replacements with certain kind of objects. If other variables are present in function definition, e.g., in $f(x) = a\,x^2 + b\,x + c$, they are distinguished from the dummy variables and they are called parameters. At first sight this is an easy distinction, but the use of parameters is in practice more complicated and more difficult to master (Furinghetti and Paola, 1994; Drijvers and van Herwaarden, 2000).

2. as a *polyvalent name*, i.e., a name for an object that can take a multitude of values.

If n is a divisor of 24, the letter stands for any of the numbers 1, 2, 3, 4, 6, 8, 12, and 24. In the statement that we have a real number x such that $x^2 + x - 1 = 0$, the letter x refers to a number that is yet unknown, but can be computed, and that has the property that the sum of this number, its square, and −1 is equal to 0. Without knowing its exact value, one can deduce that for this number holds $x = x^3 + x^2$. Solving the equation means finding the x for which it is true. A priori x is indeterminate, a posteriori x can take two values.

3. as a *variable object*, i.e., a symbol for an object with varying value.

In mathematics, the object to be thought of can be a number whose value may change like in the locutions "2^n for n from 1 upward" and "a_n converges to 0 as n goes to infinity". Of course, one can replace these locutions with 2^n for $n \in \mathrm{N}$ and $\lim_{n\to\infty} a_n = 0$, but then one looses the kinematic aspect of the variable. However, this kind of variable very often occurs in mathematical models that are constructed to study real-world phenomena. The object can be a physical quantity such as time, position, and temperature, or an economic quantity such as price, capital, and income. Clearly quantities with varying values.

The numerical value of a quantity depends on the unit in which it is expressed. This affects formulas in which the quantity occurs. For example, the Quetelet index is defined as the quotient of body weight (in kilogram) divided by the square of the length (in meters). If units are changed from kilogram and meter to gram and centimetre, respectively, a scale factor of size 10 should be included in the definition to maintain the meaning of the Quetelet index.

A variable object may be related with others. One speaks of independent variables, whose values one is free to choose, and of dependent variables, whose values one can compute given the values of the independent variables. Many applications can be given: the position of a moving object depending on time, the room temperature depending on the time of the day, the temperature of a long rod as a function of place, the number of articles sold in a shop as a variable that depends on the price per unit, the income of a person as a function of his or her education, and so on. In many cases, a variable may depend on more variables than explicitly stated. For example, income of a person may depend on education, professional experience, age, number of hours at work, sex, job status, and so on. But in an economic or social study not all independent variables are taken into account; some of them are neglected, others appear as parameters in a model description. The roles of independent and dependent variables are often not fixed during a computation. For example, studying the motion of a sprinter, one may on the one hand consider acceleration as a function of time, but on the other hand describe it as a function of the velocity of the sprinter. One of the big ideas in calculus, and in mathematics in general, is the freedom of choosing independent and dependent variables. In (Hall, 1999) one can read a lively account of how in design-oriented work by a multidisciplinary team considerable time is actually spent on identifying the quantities of importance and formulating symbolic relationships between them.

The above distinction of three uses of variable can be applied to parameter as well: in the role of a placeholder, the parameter has one value at a time. For example, in the formula $T = 2\pi\sqrt{\frac{l}{g}}$ for the period of the mathematical pendulum, the letter g stands for the gravitational constant. You may study the motion of the pendulum on earth and as well as on moon, but always stands g for one value only. The given formula expresses the relationship between the period of the pendulum and the length of the pendulum. So the letter l plays the role of a variable object. Thirdly, as a polyvalent name, a parameter allows you to write general formulas. For example, the letters A (amplitude), ω (angular frequency), and ϕ (phase) are used in the formula $u = A\sin(\omega t + \phi)$ to describe harmonic motion mathematically. Typically, one keeps values of all parameters except one fixed and draws graphs for several values of the free parameter in one picture, i.e., the parameter in its generalising role is used to distinguish several cases.

5. Variables in School Algebra

In school mathematics, much attention is paid to the introduction of variables. Educators search for ways to familiarise pupils with variables (e.g., see Kieran, 1992; Kieran, 1997). The content of secondary school algebra has more or less evolved into a standard approach to these topics.

At the beginning of learning algebra, the link to arithmetic is emphasised: letters are used in algebraic expressions to represent quantities, numbers, and numerical relationships. At this stage, an algebraic expression is still considered as an arithmetic operation upon numbers and algebraic manipulation is more or less a change of the recipe to compute a result: $x + x$ and $2x$ are algebraically equivalent because the numerical results when x is replaced by a concrete value are the same, ignoring that the expressions are different from a computational point of view. Pupils start with studying the connection between a table, a simple computation, and a simple word formula. Labelled arrows represent simple computations like “multiply by 2”, “add 3”, and so on. Words are used to describe input and output. In situated contexts, pupils must find regular patterns in number sequences and express these in terms of word formulas. The relationships are almost always linear. Typical tasks are:

- Given a formula, construct the corresponding computational steps.
- Given the computational steps, fill out a table.
- Given a table of numbers, discover a pattern and find the corresponding formula.
- Given a situated context, create a formula and check correctness.

Next comes the study of the connection between a word formula, a table, and a graph. Typical exercises are:

- Given a formula, fill out a table.
- Given a table, draw the points in a graph.

- Given a graph, read off coordinates of points on the graph.

After this, the movement from word formulas to letter formulas and the learning of basic rules of simplification such as $a+a+a=3a$ takes place. Letters are used as abbreviations of quantities and pupils learn the usual rules of computing with letters in situated contexts.

Working with operation diagrams and function machines that have formulas as labels completes the introduction to algebra as a procedural tool. First chains of labelled arrows are studied to set up an operation diagram. Later on, such a chain is replaced with a single function machine labelled with a formula. The word "formula" has a special meaning in school mathematics and the role of the letters in the *formula* $y=x+1$ is not the same as in the *equation* $y-x=1$. The words "formula" and "equation" are used to distinguish between the case of a functional relationship between the isolated variable y that depends on the other variable and the case of a more general relationship between unknowns. For pupils it is important to make a clear distinction between these different notions. A mathematician, however, is much used to applying the same algebraic symbolism for many purposes: $y=x+1$ may stand for an equation, for an abbreviation of the expression $x+1$, as well as for the process of computing the value of y from the value of x.

After the introduction of variables, by working with simple algebraic expressions containing familiar arithmetic operations like addition, subtraction, multiplication, and division, pupils start learning to view an expression not only as a process of computing, but also as a result of this process. An expression also becomes a mathematical object on its own, which can be manipulated. In (Gray and Tall, 1994) the word "procept" is introduced for the combination of symbol, process, and concept, to make clear that a mathematical object never completely looses its process nature. The alternative conceptions are also referred to as "process and object" (Sfard, 1991), where the word "reification" is used for the objectivation of a process, and "procedure and structure" (Kieran, 1992). Gradually, pupils learn to see variables as a replacement not only for numbers, but also for expressions. They learn and practice the various ways in which algebraic expressions can be manipulated: combining literal terms, replacing subexpressions, factoring, completing the square in a quadratic polynomial, rationalising the denominator, subtracting the same term from both sides of an equation, solving systems of linear equations, and differentiating are examples of activities that are present in all mathematics curricula.

At this point, the status of school algebra has become generalised pure arithmetic, based on reference-free numbers, where variables are mostly used as polyvalent names and placeholders. In many research studies (e.g., see Kieran, 1989; Sfard at el, 1994; MacGregor and Stacey, 1997; Stacey and MacGregor, 2000; Tall et al, 2001) is reported that understanding generalised pure arithmetic is not easy: literal symbols are like numerals and words, yet they are different, and pupils have to learn to deal with these differences. They must understand the underlying structure of algebra and become familiar with the dual character of algebraic expressions (operational/structural, process/object) to gain competence in mathematics.

This is also the point where mathematics has completed the move into a direction different from the natural sciences, which use generalised quantity arithmetic, i.e., the arithmetic of modelling, in which variables are mainly used as variable objects. Pupils are often expected to make for themselves the link between the generalised pure arithmetic and the generalised quantity arithmetic.

6. Variables in Science

In (Vredenduin, 1979), the author discusses some of the differences in terminology between physics and mathematics, but most of his conclusions hold for any science field. Here, a variable is most often used as a name for a quantity that can vary (often with respect to time) and that in many cases can be measured. Physical quantities can be magnitudes such as temperature, mass, and length, or combinations of magnitude and direction such as velocity, acceleration, and force. Function, sample of function values, and single function value are mixed up easily in science and apparently without much harm: a physical quantity is sometimes a function of time, in other occasions a finite sample of values measured at different times, and sometimes a function value at a certain fixed time. As a concrete example may serve Boyle's law $\mathrm{p\,V} = \mathrm{constant}$, for pressure p and volume V. The variables are in fact functions of time, viz., $\mathrm{p} : t \mapsto \mathrm{p}(t)$ and $\mathrm{V} : t \mapsto \mathrm{V}(t)$. Boyle's law says that the product of these two functions is a constant function, i.e., $\mathrm{p}(t) \times \mathrm{V}(t) = \mathrm{constant}$, at every time t. In an experiment, one verifies or rediscovers the relationship by measuring p and V under different conditions. In a problem like "suppose that V=10 ml when p=2 bar, how much is V when p=4 bar?", p and V do not represent functions anymore but function values, viz., the pressure and volume at a certain fixed time.

In mathematics, this ambiguous use of notation for function, sample of function values, and function value rarely occurs. One exception: in statistics, a random variable X is in fact a function on the set of all possible results in a chance experiment. As an example, think of the experiment of throwing two fair dies and let X_1 and X_2 be the scores of the first and second die, respectively. The set of all possible results equals the set of 36 pairs $[i,j]$ with $i,j=1,2,\ldots,6$. The random variables X_1 and X_2 are functions defined by $X_1([i,j])=i$ and $X_2([i,j])=j$, respectively. Other random variables like $X=X_1+X_2$ can be constructed. Although random variables are functions, a statistician has no problem asking for the probability that X is in a certain interval. He or she may talk for example about the chance that $X>6$. In such a question, the variable represents function values. Efficiency of the mathematical language is the keyword here. Mathematical purism does not lead to any deeper insight in these matters and it will even be a burden in mathematical reasoning and understanding of variables and functions.

In mathematics it is often allowed to change names in an expression without changing the meaning: for example, $\{(x,y) \mid xy=1\}$ is the same set of ordered pairs as $\{(a,b) \mid ab=1\}$. But in science, such replacements are almost always forbidden: replacing traditional names for energy E, mass m, and the velocity of light c in

Einstein's law $E=mc^2$ will ruin it. The reason is simple: most variables in physics are not meaningless but deal with concepts in physics that have the property of quantity. Therefore, one often chooses the first character of the name of the concept as the name of the variable: F for force, m for mass, and a for acceleration in Newton's law $F = m\,a$ is one example of many. This strong sense of notational conventions in science, compared to the freedom in mathematics, pops up on many occasions. For example, in (Wijers et al, 2000) the use of ratio tables in secondary school mathematics and science are compared: in mathematics there is no preference which to choose as upper or lower row in a ratio table, but in science there is a preference, in particular in the compound-quantity case like speed (distance over time) or density (mass over volume), to let the ratio table labels reflect the 'formula' of the compound quantity involved.

Physical quantities can often be measured, but the measured values are always natural numbers (in counting processes) or floating-point numbers, possibly with margins of error. Irrational numbers like $\sqrt{2}$ or π do not occur in measurements. Nevertheless, scientists mostly compute with quantities as if they take real values. They also treat floating-point numbers differently than most mathematicians do. For many a mathematician, the number 1.23000 is the same as the number 1.23 without the trailing zeros. However, in numerical analysis and in science, the notation 1.23000 implies that the number is known more accurately than 1.23. A value of a physical quantity actually consists of three parts, viz., the numerical value (a number), the precision (the number of significant decimals or the margins of error), and the unit that is used to measure the quantity. This makes quantity arithmetic more difficult to learn and to use than reference-free number arithmetic.

In science, words like "big", "small", "relatively small", "negligible" can be used while talking about quantities. A small change of a quantity Q is also given a name, such as ΔQ, and one manipulates it as any other variable, except that one often ignores higher order terms like $(\Delta Q)^2$ to get a simpler model description. Going from calculus of small changes to infinitesimal change and calculus with differentials is then a natural step. The following example shows how it works. We look at the formula $s = a\,t^2$ for a moving body with s being the distance travelled as function of time t. Suppose that one is interested at the speed of the object during its fall. The change in distance travelled $\Delta s(t_1)$ during a small time interval $[t_1, t_1 + \Delta t]$ is given by $\Delta s(t_1) = a\,(t_1 + \Delta t)^2 - a t_1{}^2 = 2 a t_1 \Delta t + a(\Delta t)^2$. A physicist will say that, when Δt is small, the term with $(\Delta t)^2$ can be neglected. So, for the rate of change one has $\Delta s(t_1)/\Delta t \approx 2 a t_1$. In the limit case of infinitesimal changes, one has $ds/dt\,(t_1) = 2 a t_1$. So, the speed at time t_1 is equal to $2 a t_1$. The mathematician, on the other hand, is not happy with the sentence "when Δt is small, you can neglect the term with $(\Delta t)^2$". He or she will say that $\Delta s(t_1)/\Delta t = 2at_1 + a\,\Delta t$ and that therefore $\lim_{\Delta t \to 0} \Delta s(t_1)/\Delta t = 2at_1$. The same conclusion, but a different style of working.

It is clear that when a variable is mostly used as a placeholder, in which only a number or an expression that evaluates to a number can be substituted, a pupil gets the impression that a variable is a rather static mathematical entity: it makes no sense to him or her to consider change of the variable. In contrast, when a variable is used for a variable object, change becomes an important issue and calculus of small changes, ultimately leading to differentials, becomes a natural and useful technique in mathematical modelling. One can defend the standpoint that this does not only hold for the study of physical phenomena, but also for problems that are more looked upon as mathematical ones. For example, the relationship between the circumference and the area of a circle can be derived as follows: consider the area *A* as a variable dependent on the radius *r*. Let the radius increase with a small amount of Δr. This leads to a small increase of the area ΔA. These small changes are related by $\Delta A \approx C \times \Delta r$, where C is the circumference of a circle with radius *r*. This approximation can be explained by transforming the small rim smoothly into a rectangle of length *C* and width Δr. Choosing very small increases Δr and ultimately taking infinitesimal quantities leads to the following equation with differentials: $dA = C\,dr$. This corresponds with $dA/dr = C$. Knowing the formula $C = 2\pi r$, one finds $A = \pi r^2$. In mathematics education, there is a recurrent debate about whether the concept of differentials or instead the concept of derivative should be used in the calculus curriculum.

The above differences add a good deal to the understanding of the difficulties that pupils have in relating the mathematical methods and techniques that are used in natural sciences to what is learned in mathematics lessons. In mathematics, they use variables mostly as placeholders and polyvalent names. Emphasis is on generalised pure arithmetic, with reference-free numbers, and on the concept of function defined as a special kind of correspondence between nonempty sets *A* and *B* which assigns to each element in *A* one and only one element in *B*, i.e., on the Dirichlet approach to the concept of function. In science, the third kind of variable, viz., the variable object, comes into play. One is involved with functional relationships between varying quantities, in which one distinguishes between dependent and independent variables. In working with variable objects and relationships between them one uses mainly the theory and practice of solving equations in known and unknown quantities and one uses the calculus of change.

7. Computer Algebra in Education

Use of computer algebra systems and symbolic calculators is new in school mathematics and educational research is still in its childhood. Early in the nineties, pioneering educators had high expectations of computer algebra (e.g., see Karian, 1992). This resulted in many papers entitled “Using Computer Algebra …” in conference proceedings and system-related periodicals. The envisioned scope of computer algebra usage was broad and was in line with the original design goal of most symbolic systems, viz., providing a general purpose tool for technical computing in various areas. The overview of (Mayes, 1997) indicates that studies in this period try to find rationales for the use of computer algebra and investigate potentialities of symbolic systems such as improving conceptual understanding, overcoming limitations imposed by poor algebraic skills, and bringing applications of mathematics within students' reach in the expectation that the mathematics curriculum is more attractive in this way. A good overview of the state of computer algebra in mathematics education at that time is given by (Berry et al, 1997). Another source of information is (Heugl et al, 1996), which describes practical examples and experiences from a large Austrian Derive-project. Consensus in the pioneering work seems to exist on the most important advantage of using computer algebra in mathematics education: computer algebra has the potential of making mathematics more enjoyable for both teachers and pupils because it turns mathematical entities into concrete objects, which can be directly investigated, validated, manipulated, illustrated, and otherwise explored. Beautiful curves and graphs, interesting numerical and symbolic results, meaningful constructions and derivations with immediate feedback, all this becomes accessible. Abstraction, exact reasoning, and careful use of symbolism are not solely a hobby of the mathematics teacher anymore, but are immediately rewarded when using computer algebra.

In the second half of the nineties, classroom experiments have shown that the integration of computer algebra in mathematics education is more complex than expected. The optimism in constructivist approaches that the computer environment can help to create adequately problematic settings in which the learner meets insufficiency or inconsistency of his/her knowledge and starts to develop him/herself new knowledge is damped in the school practice. The fact is, the constructivist view assumes that the computer environment will provide the means for a predicable and meaningful interaction. However, results from a large-scale Derive-experiment in France (Artigue, 1997) indicate that it is only a matter of 'pseudo-transparency': in the mediation of the computer algebra system, the instrument may be transparent for the teacher who is an expert both in mathematics and in the operation of the symbolic system, but not for the students who perceive the mathematics through the symbolic system and are unaware of properties of the instrument and of differences between internal computer representations and (mathematical) representations returned by the user interface. Students may build unexpected or even incorrect mathematical meanings from the system's feedback.

In the same research study (Artigue, 1997), it is reported that students may interpret tasks different from teachers or authors of textbooks. They illustrate this by the following investigative task: "Propose, in the form of conjectures, factorisations of polynomials of the form $x^n - 1$". One interpretation of the task, and probably the one that its creator had in mind, is to use the computer algebra system as a computational tool to find theorems about factorisation of polynomials of this kind. Students however interpret this task often as "find conjectures about factorisations in the computer algebra system". This task may even turn out to be more complicated than the one of the first interpretation. Artigue refers to this phenomenon as “double reference”. Although the computer algebra system is a driving force in the activity, it is in general not the symbolic system itself that is of interest to the teacher; s/he wants to use the computer algebra system only as a tool for doing mathematics. Pupils may look upon this with different eyes: for them, working with the computer algebra system may be the mathematical activity.

Artigue also identifies four broad reactions to a new tool amongst students in the experimental classes: refusal, blind confidence, tool for computing and checking results, and tool for learning. Similarly, in (Guin and Trouche, 1998) and (Trouche, 2000) a great variety of student’s behaviour is observed. They identify the following styles of working, characterized by the frequency and style of tool use: random, mechanical, resourceful, rational, and theoretical. They illustrate that according to their profile characteristics, students develop different problem solving techniques and different relationships with their symbolic calculators. Another report of this kind is (Weigand and Weller, 2001), which describes the working styles of 11th graders at a German high school studying quadratic and trigonometric functions in a computer algebra environment. One conclusion of this empirical work is that it is not easy to bring pupils from working at a level of experimental heuristics to mathematical reflections about the problem and problem solving activities. Activation of basic mathematical knowledge while working with the computer is needed for developing mathematical thinking and the teacher plays an important role in this.

The aforementioned research studies make clear that efficient and successful use of symbolic systems is not self-evident: it takes time and effort to acquire the appropriate skills and knowledge for proficient use of such systems, and this process may differ from student to student. Obstacles that students encounter while working with a symbolic system have often both a technical and mathematical character. For example, Drijvers (Drijvers, 2000) identifies the following general obstacles for students using a TI-92 to study optimisation problems:

- The difference between the algebraic representations provided by the CAS and those students expect and conceive as ‘simple’.
- The difference between numerical and algebraic calculations and the implicit way the CAS deals with this difference.
- The limitations of the CAS and the difficulty in providing algebraic strategies to help the CAS to overcome these limitations.
- The inability to decide when and how computer algebra can be useful.
- The flexible conception of variables and parameters that using a CAS requires.

This list of obstacles is in line with results of other research studies that show the complexity of the so- called instrumentation process. For example, (Guin and Trouche, 1998) also point out that the confusion between mathematical objects and their representation by calculators is an obstacle in the process of instrumentation. The complexity of transforming the computer algebra tool into a mathematical instrument for students is probably one of the reasons why teachers in school mathematics have not embraced symbolic calculators immediately.

As symbolic systems have not fully penetrated in school mathematics (yet), it may be no surprise that they are not much used in school science either. Main reason is actually that the science lessons are principally descriptive, qualitative, and experimental in nature. For example, in secondary school chemistry, mathematics is almost limited to unit conversions, drawing and interpreting titration curves, balancing of chemical equations, and to computations related with a chemical equilibrium. Apart from solving equations, not much symbolic computation is present, and where mathematics is needed in chemistry, it is expected that one can build upon basic skills learned in mathematics lessons. In secondary school physics, more mathematics is used. For example, trigonometry, plotting and analysing measured data, working with vectors, algebra of small quantities, and coordinate transformations are required. But again, not overwhelmingly much symbolic computation is present. It is mostly limited to rewriting of 'simple' formulas, mathematical manipulation of measured data, solving equations, and to working with trigonometric expressions. Only in a few experiments, the measured data lead to a clear mathematical function and formula. For example, discharge of an electric conductor shows a nice exponential decay. Analytical solution of such an example problem already pays off, because it can be applied to many other physical processes that follow exponential growth or decay. Think of cooling of an object, radioactive decay, first-order kinetics of a chemical reaction, and so on.

In science, mathematics is anyway more a means to an end. Plotting graphs helps to interpret data; simulation of science experiments is done to understand real-world phenomena; algebraic manipulation often aims at presenting experimental results in a more convenient way; and so on. Computer programs and graphing calculators are mainly used for acquisition of experimental data, visualisation and analysis of experimental results, and for simulation of experiments. The need for symbolic computation is more felt at higher educational level, when mathematical modelling becomes more important. At this point the mathematical theory is extended with differential equations, integration, special functions, etc., and formula manipulation becomes more elaborate. Then, a computer algebra system comes into play as a personal educational assistant that allows a student to concentrate on science concepts instead of wasting time in algebraic drudgery.

Reports in the last ten years on the use of computer algebra in mathematics and science lessons during the first two years of university and higher vocational education plus personal observations over the last decade in courses that introduce Maple and *Mathematica* to novice users lead us to the following general findings about algebraic manipulation with computer algebra systems:

- With a powerful repertory of tools at hand, it becomes even more important for the user to realise what goals s/he has, what results already have been obtained in this direction, and what steps can come next. Stated in a one-liner: computer algebra is not hands-on, but brains-on computing.

- Basic manipulation skills are required for effective use of a computer algebra system.
- Computer algebra systems force their users to express their intentions in an explicit manner and in a syntax that is rather distinct from the flexible symbolism normally used in mathematics and science.
- Symbolic input is in most cases the driving force in obtaining results with a computer algebra system.
- Use of variables as variable objects is not much supported in a computer algebra system.

Especially the last two points have great impact on the usability of computer algebra in science education. Computer algebra systems provide little support for real-time acquisition and analysis of data from experiments, whereas this often is the starting point in science work. The symbolic calculator connected to a CBL can only be considered as a poor alternative for the data acquisition and analysis tools currently used in science education. In the use of a variable as variable object, it is the finiteness of any representation on a computer that imposes restrictions. Possible finite representations are:

- with a finite indexed listing of values.
- via an algorithm expressed in finitely many terms.

In (Spunde and Neidinger, 1999), the authors advocate an implementation of a variable object as a finite list of values and show that calculus with samples of values is possible with existent computer algebra systems. However, these systems lack the direct manipulation of lists and immediate linking of lists, as is offered by graphing calculators. The finite representation of a variable object via an algorithm rarely occurs: in Maple, formal power series are the only data structures of this kind.

8. Recommendations to Teachers

The list of general findings about algebraic manipulation with computer algebra systems at the end of the previous section would just form a collection of more or less obvious statements if they would not lead to recommendations to teachers who use existing computer algebra systems in their lessons. Based upon our experiences with first year university students who start learning to use computer algebra systems, but who are supposed to have reached already a sufficient level of algebraic thinking, we make some recommendations that may help avoid foreseeable pitfalls. They are in line with suggestions of others (e.g., see Lagrange, 1999; Guin and Trouche, 1998) for strategies to turn symbolic systems into efficient mathematical instruments. Our recommendations are:

1. *Let students develop a good understanding of when, why, and how computer algebra can be helpful.*

It is for pupils and teachers good practice first to explore a problem situation, to reformulate a problem in a seemingly appropriate mathematical way, to do some easy calculations or algebraic manipulation by pencil and paper, and to think about what kinds of answers are actually wanted. In this exploratory phase, small computer algebra calculations may assist in unravelling the initial problem and in coming to a solution strategy. Only after these introductory steps one can make right decisions on how to proceed, what computational tools to use, and what notation fits best. Two examples of much used solution strategies are:

- If the zeros of some unfamiliar function must be approximated, one first makes a plot of the function to get an idea where solutions can be found. Then one decides whether a graphical approximation of zeros suffices, or that this is used as a starting point in a numerical solution method.
- If one is asked to solve some non-linear system of equations, it may be wise first to reformulate the problem in terms of a system of polynomial equations, then apply standard techniques for solving such systems, and finally transfer the solutions back in terms of the original equations. A concrete example of this strategy is given in (González-López and Recio, 1993) for the inverse geometric model of a robot manipulator and in (Heck, 1993) for relating geodetic and geocentric coordinates.

Describing how a particular problem has been solved, discussing alternative solutions and methods, and so on, is supported by the documentation facilities of modern computer algebra systems. This does not mean that reporting is only by way of computer algebra worksheets or notebooks. Copying results from computer screen in a workbook, in standard mathematical notation, remains good practice for various reasons: it forces pupils to distinguish important results from intermediate results and to report about their work in standard mathematical notation, which can be understood by persons not familiar with the tools used. Only by practice, pupils learn to speak and write the language of mathematics and science.

2. *Ensure that pupils maintain basic algebraic skills and basic knowledge of mathematical properties.*

It is a misconception to believe that with the advent of computer algebra systems pupils need less training in manipulation of algebraic expressions because the computer programs will carry out the manipulative tasks. On the contrary, for effective use of a computer algebra system a good insight in formulas is required, must be developed, and must be maintained. A simple example: recognising the common term $2x+1$ in $(2x+1)^2 - x(2x+1)$ one easily gets the equivalent expression $(2x+1)(x+1)$ and the factor command in any computer algebra system would give this result, too. But the simplify command would in most systems just work out the expression into $2x^2+3x+1$. If a student relies on this command only, in the expectation that it will always produce the simplest and most suitable expression, then s/he will not acquire the various skills needed for formula manipulation. Elementary manipulation skills are also required for formulating a problem in such a way that the particular computer algebra system in use understands it or can handle it, and for recognising structure in symbolic output. Think of a science problem, say

$dp = a\sqrt{1+p^2}\,dx$ that is formulated in terms of differentials, but that must be entered in the computer program in terms of derivatives, for example in Maple as diff(p(x), x) = a*sqrt(1+p(x)^2). Another example is a complicated expression in which a pupil has to recognise and judge applicability of simple manipulation rules to achieve simplification. Solving a problem may drastically change when some simplification is done before carrying out a standard technique. For example, computing the derivative of $(x+1)/x$ by the quotient rule gives a rather complicated answer, and a computer algebra system will do this when asked, whereas after simplification to $1+1/x$ it can readily be seen that the derivative equals $-1/x^2$. Proving convergence of the definite integral $\int_1^\infty 1/(x^{73}+x+1)\,dx$ is easy once one realises that $1/(x^{73}+x+1) < 1/x^{73}$ on the domain of integration, but computing the definite integral with a computer algebra system takes a lot of computing time and computer memory. Transcribing a relatively simple algebraic formula from a textbook into the computer algebra language may already turn out a complicated task, in which the pupil must not mix notations and quite often needs to add brackets and operators at proper places in the input. Here too, basic knowledge about mathematical notation and about elementary manipulations of symbolic expressions is of key importance. Do not expect that pupils maintain these skills and knowledge of properties all by themselves or simply by using a computer algebra system.

3. *Prepare your pupils to working with mathematically sophisticated systems that may produce unfamiliar results.*

Altering the traditional choice and sequence of topics of the mathematics curriculum is necessary when computer algebra comes into play. Instead of treating topics in a compartmental way, in which a couple of lessons are spent on one main topic before the next one is treated, pupils must learn many basic things in short time. Not in great depth, but enough to understand or be able to handle computer output that contains unfamiliar ingredients. Advice: lead your pupils from "knowing little about much" to "knowing more about much" in mathematics. As a side effect, this prepares pupils for later professional life, where they may have to use tools of which they do not know the precise internal workings, but with which they are expected to study phenomena or analyse data in a sensible way. Future decision makers have to be able to read reports, judge information, interpret data and graphs, etc., without knowing all details of how data are collected and processed, and they must be able to ask the right questions to the persons who actually make the reports.

4. *Make a virtue of necessity: use the explicit symbolism of computer algebra to make pupils conscious of the versatile use of variables and use surprising results or bugs as opportunities to discuss mathematical topics further.*

Use of computer algebra demands meaningful choice and careful use of names and notation. Traditional notation and locutions that one sees in textbooks may not be appropriate. For example, a sloppy notation like $F(x) = \int f(x)\,dx$ brings problems when calling the function F with a numeric argument. This could easily be avoided by a more careful formulation like $F(x) = \int_0^x f(\xi)\,d\xi$. Tasks must be carefully formulated avoiding erroneous use of words that have already a different meaning in the tool that one uses. For example, do not ask to "solve an integral" because this may cause a pupil to use the solve command in the computer algebra system, which most probably can only be used for solving (systems of) equations. Do not ask to differentiate or plot a function $f(x) = \ldots$ because this may cause a pupil to enter a command like differentiate(f(*x*)=…) and plot(f(*x*)=…), respectively. The resulting error messages may make them angry and frustrated and most likely they will blame the computer algebra system, whereas in fact the problem lies somewhere else. It is clearer and actually not much more work to ask for the derivative and graph of the function f defined by …

Surprising results of a computer algebra system can be used as a good source for discussing mathematics. The integration of $10x(x^2+1)^4$ gives in Maple, *Mathematica*, and other systems a lengthy answer that does not factor to the obvious answer $(x^2+1)^5$; an opportunity to discuss the notion of integration constant! After discussing divergence of sums, the fact that Maple computes a standard example of a divergent sum, viz., $\sum_{n=1}^{\infty}(-1)^n$ as −1/2 gives food for discussion. Plot aliasing, as it appears in *Mathematica* and Maple when one asks to plot the surface $z = y\,\sin(2\pi\,x)$ for (x, y) in [0,75]×[0,1], invites for a discussion of what expectations one can have with regards to the graph of a function, knowing the constituents of the formula that defines the function.

5. *Teach pupils convenient styles of working with a computer algebra system and use them yourself.*

In doing mathematics and science with computer algebra software, the preferred ordering in definitions is often from the general to the more specific. An advantage of specifying the general formula first, is that it allows an easy visual check whether the appropriate expression has been entered via the keyboard. Note that this ordering of commands is opposite to the normal didactic order, in which specific cases are studied before the general case is considered. Other rules of thumb with regards to expressions are:

- Start each exercise or example with a clean slate so that previously assigned variables do not interfere.
- Try to validate symbolic results as much as possible. In many cases, exact results can be verified numerically, graphically, or by doing the same computation again using a different method.

- Resist the temptation to call every unknown *x* and *y*, but use meaningful names so that you and other persons later can still understand what it was all about.
- Always make a clear distinction between algebraic expressions, equations, and functions in a computer algebra system.

This last remark may seem obvious, but one of the most important sources of many user problems is in fact mixing these two different notions. The following small Maple example shows some of the subtleties. Consider the differential equation $dy/dx = 2y$. To enter this equation in Maple, one must use $y(x)$ instead of y and enter diff(y(x),x)=2*y(x) to make clear that one thinks of *y* as a function in *x*. When Maple is asked to solve the differential equation with initial conditions y(0)=1, it returns the answer $y(x)$ = exp(2*x*). Although it looks as a definition of the function *y*, it is in fact an equation in which $y(x)$ is used as an ordinary variable. To turn the equation into a function definition, one can make use of the special instruction unapply that transforms an expression into a function definition. Once *y* is the name of a function, one cannot use $y(x)$ as an ordinary variable anymore.

Especially when computer algebra is used for studying science problems one often has to change from expression to function and vice versa. In mechanics, for example, one likes to work with explicit coordinate functions x(t) and y(t) for the position of a moving body so that one can plot the trajectory of the object or compute positions at certain times. On the other hand, it is often convenient that x and y can be manipulated as ordinary variables. A good example is the algebra one often has to do before equations of motion are in suitable format.

The above list of recommendations may give you the impression that you constantly have to run the gauntlet with regards to symbolic input and output. This is partly true because, in computer algebra, the mathematical context is not implicitly used, so that computer algebra notation must indeed be less flexible than standard mathematical notation. But at the same time, the necessity to carefully specify in explicit form what one wants makes one very much aware of the many roles that variables and expressions play in mathematics. It makes a pupil (and the teacher!) conscious of the differences between mathematical entities that in standard notation may be expressed via one and the same formula. One can actually benefit from the fact that computer algebra syntax is less flexible than standard mathematical notation. It forces the pupil not to make a mess of mathematical notions and notations. And the reward is great in the form of interesting results, which one would not have easily obtained otherwise.

References

Artigue, M. (1997). Le logiciel ‘Derive’ comme révélateur de phénomènes didactiques lies a l’utilisation d’environnements informatiques pour l’apprentissage. *Educational Studies in Mathematics* **33 No 2**, 133-169.

Berry, J., Monaghan, J., Kronfellner, M. and Kutzler, B. (eds.). (1997*). The state of computer algebra in mathematics education*. Chartwell-Bratt, Bromley.

Cedillo, T. (1997) Algebra as a language in use: a study with 11-12 year olds using graphic calculators. In Pehkonen, E. (ed.), *Proceedings of the 21st international conference for the psychology of mathematics education*, vol 2. University of Lahti, Lahti.

Drijvers, P. (2000). Students encountering obstacles using a CAS. *International Journal of Computers for Mathematical Learning*, **5**, 189-209.

Drijvers, P. and Herwaarden, O. van. (2000). Instrumentation of ICT-tools: the case of algebra in a computer algebra environment. *International Journal of Computers in Mathematics Education*, **7 No 4**, 255-275.

Etten, B. van (1980). Variabelen in de schoolwiskunde. [Variables in school mathematics] *Nieuwe Wiskrant*, **23**, 15-18.

Freudenthal, H. (1983). Didactical phenomenology of mathematical structures. Reidel, Dordrecht.

Furinghetti, F. and Paola, D. (1994). Parameters, unknowns, and variables: a little difference? In Da Ponte, J.P. and Matos, J.F. (eds.), *Proceedings of the 18th International Conference for the Psychology of Mathematics Education,* vol 2. University of Lisbon, Lisbon

Gage, J. (2001). The Role of the Graphic Calculator in Early Algebra Lessons. In Borovcnik, M. et al (eds.). *Proceedings of the 5th international conference on technology in mathematics education*. Klagenfurt, Austria

González-López, M.J. and Recio, T. (1993). The ROMIN inverse geometric model and the dynamical evaluation method. In Cohen, A.M. (ed.). *Computer algebra in industry: problem solving in practice*. Wiley, Chichester.

Gray, E.M. and Tall, D.O. (1994). Duality, ambiguity and flexibility. A proceptual view of simple arithmetic. *Journal for Research in Mathematics Education*, **26 No 2**, 115-141.

Graham, A. and Thonas, M. (2000). Building a versatile understanding of algebraic variables with a graphic calculator. *Educational Studies in Mathematics*, **41 No 3**, 265-282.

Guin, D. and Trouche, L. (1998). The complex process of converting tools into mathematical instruments: the case of calculators. *International Journal of Computers for Mathematical Learning*, **3 No 3**, 195-227.

Hall, R. (1999). Following mathematical practices in design-oriented work. In Hoyles, C. et al (eds.). *Rethinking the mathematics curriculum*. Studies in Mathematics Education Series, Vol. 10, Falmer Press, London.

Heck, A. (1993). Transformation between geocentric and geodetic coordinates. In Cohen, A.M. (ed.). *Computer algebra in industry: problem solving in practice*. John Wiley & Sons Ltd, Chichester.

Heck, A. (1999). Coach: an environment where mathematics meets science and technology. In Maull, W. et al (eds.) *Proceedings of the 4th international conference on technology in mathematics education*. CD-ROM published by the University of Plymouth, ISBN 1-84102-056-7 UK.

Heugl, H., Klinger, W. and Lechner, J. (1996). *Mathematikunterricht mit Computeralgebra-Systemen*. Addison-Wesley, Bonn.

Karian, Z.A. (ed.) (1992). *Symbolic computation in undergraduate mathematics education*. MAA Notes 24, Mathematical Association of America, Washington, D.C.

Kieran, C. (1989). The early learning of algebra: a structural perspective. In Wagner, S. et al (eds.). *Research issues in the learning and teaching of algebra*. NCTM/Lawrence Erlbaum, Reston, VA.

Kieran, C. (1992). The learning and teaching of school algebra. In Grouws, D.A. (ed.). *Handbook of research on mathematics teaching and learning*. MacMillan, New York.

Kieran, C. (1997). Mathematical concepts at the secondary school level: the learning and teaching of algebra and functions. In Nunes, T. et al (eds.). *Learning and teaching mathematics*. Psychology Press, Hove.

Kücheman, D.E. (1881). Algebra. In Hart, K.M. (ed.). *Children's understanding of mathematics: 11-16,* John Murray, London.

Lagrange, J.-L. (1999). Complex calculators in the classroom: theoretical and practical reflections on teaching pre-calculus. *International Journal of Computers for Mathematical Learning*, **4 No 1**, 195-227.

MacGregor, M. and Stacey, K. (1997). Students' understanding of algebraic notation: 11-15. *Educational Studies in Mathematics,* **33 No 1**, 1-19.

Mayes, R. (1997). Current state of research into CAS in mathematics education. In Berry, J. et al (eds.). *The state of computer algebra in mathematics education*. Chartwell-Bratt, Bromley.

Mulder, C. (to appear). *Computer-based Investigations in Physics.* PhD Thesis, University of Amsterdam.

Recio, T. (1998). Didactical Relevance Of Meaningless Mathematics. *International Journal of Computers in Mathematics Education*, **5 No 1**, 15-27.

Schoenfeld, A.H. and Arcavi, A. (1988). On the meaning of variable. *Mathematics Teacher*, **81 No 6**, 420-427.

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 No 1**, 1-36.

Sfard, A. and Linchevsky, L. (1994). The gains and pitfalls of reification - the case of algebra. *Educational Studies in Mathematics*, **26**, 191-228.

Spunde, W.G. and Neidinger, R.D. (1999). Sample Calculus. *Mathematics Magazine*, **72 No 3**, 171-182.

Stacey, K. and MacGregor, K. (2000). Learning the Algebraic Method of Solving Problems. *Journal of Mathematical Behavior*, **18 No 2**, 149-167.

Tall, D.O. (1992). Mathematical processes and symbols in the mind. In Karian, Z.A. (Ed.). *Symbolic computation in undergraduate mathematics education*, MAA Notes 24, Mathematical Association of America, Washington, D.C.

Tall, D.O. and Thomas, M.O.J. (1991). Encouraging versatile thinking in algebra using the computer. *Educational Studies in Mathematics*, **22 No 2**, 125-147.

Tall, D.O. et al (2001). Symbols and the bifurcation between procedural and conceptual thinking. *Canadian Journal of Science, Mathematics and Technology Education*, **1 No 1**, 81-104.

Trouche, L. (2000). La parabole du gaucher et de la casserole à bec verseur: etude des procesus d'aprentissage dans un environnement de calculatrices symboliques. *Educational Studies in Mathematics*, **41 No 3**, 239-264.

Usiskin, Z. (1988). Conceptions of school algebra and uses of variable. In Coxford, A.F. et al (eds.). *The ideas of algebra, K-12 . NCTM 1988 Yearbook.* NCTM, Reston, VA.

Vredenduin, P. (1979). Terminologie in natuurkunde en wiskunde. [Terminology in physics and mathematics] *Euclides,* **55**, 81-94.

Wagner, S., Rachlin, S.L. and Jensen, R.J. (1984). *Algebra Learning Project – Final Report*. University of Georgia, Department of Mathematics Education.

Wain, G. (1994). Some Technical Problems in the use of Derive with school Pupils. *International Derive Journal* **1 No 1**, 49-55.

Weigand, H-G. and Weller, H. (2001) Changes of working styles in a computer algebra environment – the case of functions. *International Journal of Computers in Mathematics Education,* **6 No 1**, 87-11.

Wijers, M. et al (2000). Using ratio tables from mathematics in secondary school science. Paper presented at the *25th annual ATEE conference*, Barcelona.

Acknowledgments

The author would like to thank Jan van de Craats, Ton Ellermeier and Leendert van Gastel for careful reading of earlier drafts and for helpful remarks.