Math.Comput.Sci. (2023) 17:1
https://doi.org/10.1007/s11786-022-00551-6

Mathematics in Computer Science

Can I Bring My Calculator to the Exam? Some Reflections on the Abstraction Level of Computer Algebra Systems

Eugenio Roanes-Lozano

Received: 8 February 2022 / Revised: 7 November 2022 / Accepted: 15 November 2022 / Published online: 21 December 2022
© The Author(s) 2022

Abstract The author began working with computer algebra systems (CAS) in the 80s to perform effective computations for his Ph.D Thesis in algebra. He thought at that moment that there would be an explosion in the use of CAS for research and teaching (at all levels of education). Surprisingly, its use in secondary education is still scarce. This article details some personal reflections on elementary mathematics questions (from both the mathematical and the computational points of view) and proposes a classification of such questions, illustrated with several examples. It is focused on some of the present impressive capabilities of CAS, underlining their abstraction levels in some eye-catching examples. The article is mainly aimed at mathematics teachers who are not experts in CA. Nevertheless, it may also be of interest to CAS experts, as it includes reflections on a topic not usually treated: the abstraction level achieved by CAS and its impact in teaching and assessment.

Keywords Computer algebra · Abstraction level · Mathematics teaching with technology · Assessment

Mathematics Subject Classification 68W30 · 97P40 · 97P70 · 68N15

1 Introduction

For many years, the standard answer of many secondary education mathematics teachers in Spain to the question:

Can I bring my calculator to the exam?

regarding the college entrance exams (denoted EvAU [1]) was:

Yes, but it will be useless...

(many times accompanied by a big smile).

E. Roanes-Lozano (✉)
Instituto de Matemática Interdisciplinar & Departamento de Didáctica de las Ciencias Experimentales, Sociales y Matemáticas, Facultad de Educación, Universidad Complutense de Madrid, c/ Rector Royo Villanova s/n, 28040 Madrid, Spain
e-mail: eroanes@ucm.es

But this has completely changed by two facts:

- the release of symbolic calculators (*TI-92*[1] in 1995 [2–4], followed by several others),
- that some computer algebra systems (CAS) [5] have been ported to operating systems for smartphones (*Maxima*, *REDUCE*, *xCAS*, etc.).

Surprisingly, despite the success of *DERIVE* in the late 90s [6–8], the use of CAS in secondary education is still scarce.

This contrasts with the widely spread use of the dynamic geometry system (DGS) *GeoGebra* [9]. Nevertheless, this DGS (in fact a DGS with CAS capabilities) deserves a separate study, as it has changed many maths teachers' attitude against using technology (probably because it is easy to use, free and doesn't even requires of an installation—as it can be run remotely using a web browser). Unfortunately, many times the use of *GeoGebra* in teaching neither includes a meaningful use of its CAS capabilities nor a fruitful use of its DGS capabilities (for instance a certain Spanish secondary education textbook proposes to use *GeoGebra* for plotting functions but not for dealing with Thales's basic proportionality theorem!).

This article is mainly aimed at mathematics teachers who aren't experts in computer algebra. However, it may also be of interest to CAS experts as it reflects on the abstraction level achieved by CAS and its impact in teaching and assessment.

Sadly, this article could have been written 10 or 20 years ago, as the situation hasn't changed much. Memorizing and solving long tedious problems cope much of the time of secondary students' math practice in many countries. This article tries to convince and encourage mathematics teachers to allow the use of CAS by the students and to incorporate CAS into their curricula (adapting their assessments accordingly).

2 A Suggested Classification Scheme for Elementary Mathematics Questions

From the maths teachers' point of view, we could classify mathematical problems and the questions of maths exams in four classes:

1. Numerical computations and substitutions in known formulae,
2. Symbolic computations (simplifications, expansions, mechanical concatenation of algebraic calculations, etc.),
3. Theoretical questions regarding mathematical formulae,
4. Theorem proving

(let us observe that we are considering here the problem already mathematized, that is, in the language of algebra or calculus). We will discuss them next.

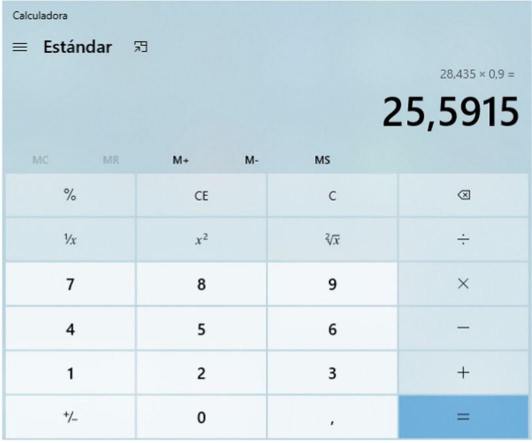
2.1 Numerical Computations and Substitutions in Formulae

They arise in all fields of maths (algebra, geometry, analysis, astronomy, etc.). They were solved in the past with pen and pencil and sometimes using logarithm tables, slide rules, etc. They can nowadays be solved with a (classic) calculator. They are well known.

Example 1 Calculate the final price of a skirt tagged a price of 23.5 Euros (VAT excluded) if the VAT is 21% and it has a 10% discount.
We can easily compute with a calculator: `23.5 * (1+21/100) * (1-10/100)` or, if we perform a previous simple mental calculation: `23.5 * 1.21 * 0.9` instead (Fig. 1).

[1] All product names, trademarks and registered trademarks are property of their respective owners.

Fig. 1 A numerical computation problem



28,435 × 0,9 =

25,5915

2.2 Symbolic Computations

In the past exercises about simplifications, expansions, concatenation of algebraic calculations, etc. could only be solved by human thought. Now these tasks can be completed by CAS, that are available for computers, tablets and smartphones.

The following examples are illustrated with the inputs corresponding to *Maple 2021* [10–14] and the outputs computed by this CAS. Obviously other CAS could be used instead [15–18].

Example 2 Simplify the algebraic expression: $(x+y)^2-(x-y)^2$. *Maple*'s `simplify` command does it:

```
> simplify((x+y)^2-(x-y)^2);
```

$$4\,x\,y$$

(inputs are preceded here by a ">" symbol while outputs are centered).

Remark 1 Regarding these examples, we don't mean that students should perform all these calculations with the computer. We just want to show some of the possibilities of CAS to those who don't know them or don't know some of them.

Let us consider a not so simple example.

Example 3 Analyse when a certain parameter-depending linear system has solutions and find them in such case. Consider, for instance, the system:

$$\begin{cases} ax + y = 0 \\ x - y = 2 \end{cases}$$

where x and y are the variables and a is a parameter. We can compute it with *Maple*:

```
> solve({a*x+y=0,x-y=2},{x,y});
```

$$\left\{x=\frac{2}{a+1},\, y=-\frac{2a}{a+1}\right\}$$

Now the different possibilities are clear:

- the denominators vanish when $a=-1$, so there is no solution in this case

- if $a \neq -1$ the system has a unique solution (the explicit solution is returned by *Maple*).

Moreover, the process isn't different if the linear equation system involves more equations and variables.

Remark 2 Keep in mind that in many CAS (such as *Maple*) the domain by default is $\mathbb{C}$. For example, the *Maple* input and output for $(ln(-1))^2$ are:

```
> ln(-1)^2;
```

$$-\pi^2$$

because:

```
> ln(-1);
```

$$I\pi$$

Of course, we can ask *Maple* to do the calculations in $\mathbb{R}$, for example using the package `Real Domain` (preceding the input with: `RealDomain:-`), obtaining this way the expected output:

```
> RealDomain:-ln(-1)^2;
```

$$undefined$$

Example 4 Compute $\int x \cdot ln(x)\, dx$ (calculating this indefinite integral is straightforward using the integration by parts method; however CAS can compute much more complex ones [19], as well as differential equations, even explaining them step by step [20]).
In this case *Maple*'s `int` command directly returns a primitive (if the CAS can't compute a primitive it either returns the input expression or an empty line):

```
> int(x*ln(x),x);
```

$$\frac{x^2 ln(x)}{2} - \frac{x^2}{4}$$

Example 5 Draw a function by computing its zeroes, maxima and minima, asymptotes, inflection points, etc. In the EvAU these functions are normally trickily chosen so that they have, for instance, two very close zeroes (in order a graphic calculator to be useless), like happens with:

$$\frac{9999}{100} \cdot x^2 - 200 \cdot x + 100$$

(this example is inspired by an example in [21]). If we plot this function between -1 and 2 we obtain the plot of Fig. 2, that looks like a second degree polynomial function with a single root. Meanwhile, a closer plot reveals two different roots (the plot between 0.98 and 1.02 is shown in Fig. 3).

Of course there are different algebraic ways to check this example. In this particular case the equation is a second degree one, so any student should know its solution. The discriminant isn't zero:

$$(-200)^2 - 4 \cdot \frac{9999}{100} \cdot 100 = 4$$

so the equation has two different roots.

Nevertheless the CAS can perform the whole process:

```
> p:=9999/100*x^2-200*x+100:
> solve(p,x);
```

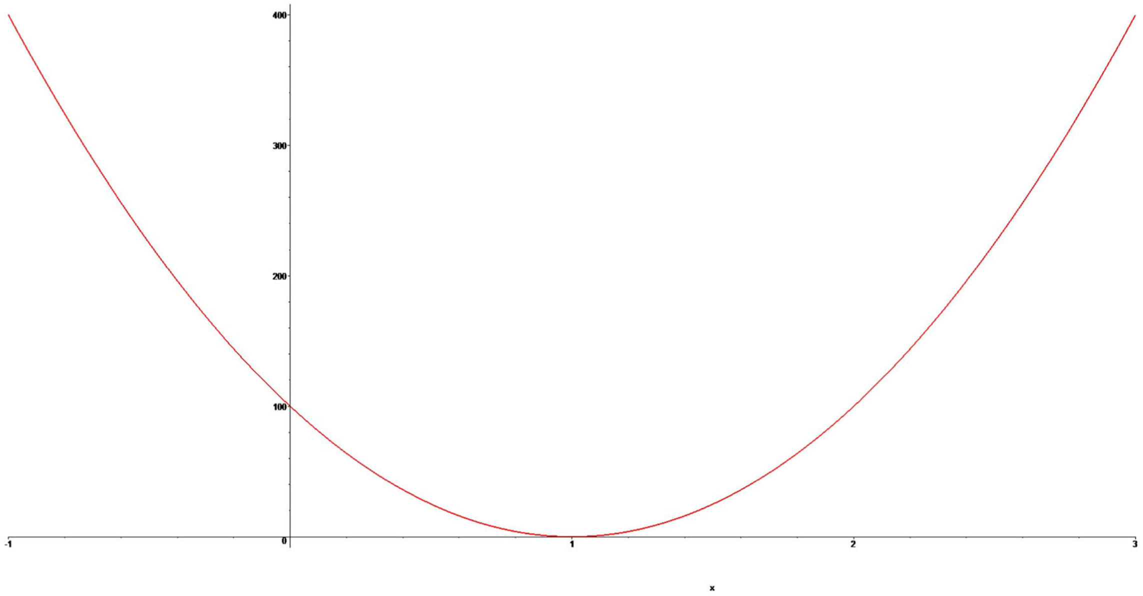


Fig. 2 A plot of a second degree function with (apparently) a single root

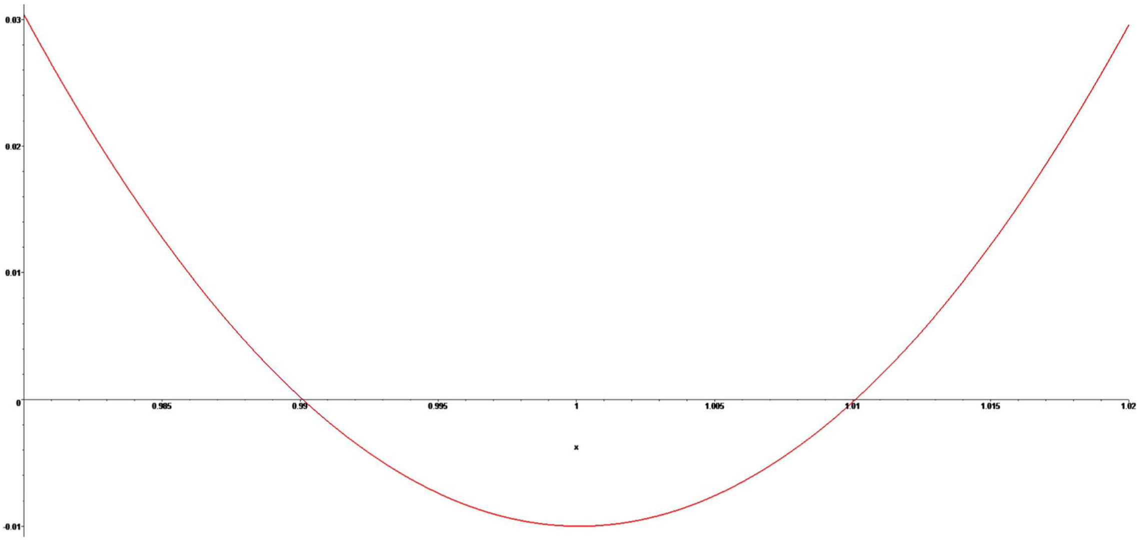


Fig. 3 A closer look to the plot of Fig. 2 revealing the existence of two different roots

$$\frac{100}{99}, \frac{100}{101}$$

```
> evalf(%);  #Approximate the previous values
```

$$1.010101010, 0.9900990099$$

```
> evalf(solve(diff(p,x)=0));  #Approximate the values where the
```

```
                                      #derivative of the function vanishes
                                      #(local extrema)
```

$$1.000100010$$

```
> evalf(diff(p,[x,x]));    #Compute the second derivative of the
                           #function (in order to decide whether it
                           #is a maximum or a minimum)
```

$$199.9800000$$

```
> diff(p,x);
```

$$\frac{9999x}{50} - 200$$

```
> limit(p,x=+infinity);
```

$$+\infty$$

```
> limit(p,x=-infinity);
```

$$-\infty$$

So the student can realize, just looking at the output of the CAS, that:

- the equation has two different roots,
- there is only a local extrema,
- it is a minimum, as the second derivative is positive,
- there is only one sign change in the slope of the curve (as the derivative is a first degree polynomial),
- there are no asymptotes at infinity.

These lines of code can be stored in a file, so that only the input function has to be introduced by the student. Moreover, a simple procedure that returns the answers with no external help or knowledge can be prepared in advance.

2.3 Theoretical Questions Regarding Mathematical Formulae

CAS have made possible to avoid using paper-based cheat sheets (and even to prepare them), as they can produce many general theoretical results on the fly.

Example 6 What is the cube of a binomial?

```
> expand((x+y)^3);
```

$$x^3 + 3x^2y + 3xy^2 + y$$

Example 7 What is $sin(2x)$ equal to?

```
> expand(sin(2*x));
```

$$2sin(x)cos(x)$$

But there are far more interesting examples, like:

Example 8 What is the determinant of a 3 × 3 matrix (Rule of Sarrus)?

```
> with(LinearAlgebra):   #To load a specific Maple package
> A := Matrix([[a11,a12,a13],[a21,a22,a23],[a31,a32,a33]]);
```

$$A := \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{bmatrix}$$

```
> Determinant(A);
```

$$a11\ a22\ a33 - a11\ a23\ a32 - a12\ a21\ a33 + a12\ a23\ a31 + a13\ a21\ a32 - a13\ a22\ a31$$

Example 9 Can you simplify the expression $(g \circ f)(x) + g(f(x))$?

```
> (g@f)(x)+g(f(x));
```

$$2\ g(f(x))$$

Example 10 What is the derivative of the product of two general functions?

```
> diff(f(x)*g(x),x);
```

$$\left(\frac{d}{dx} f(x)\right) g(x) + f(x) \left(\frac{d}{dx} g(x)\right)$$

And what is the derivative of the quotient of two general functions?

```
> simplify(diff(f(x)/g(x),x));
```

$$\frac{\left(\frac{d}{dx} f(x)\right) g(x) - f(x) \left(\frac{d}{dx} g(x)\right)}{g(x)^2}$$

Example 11 What is the indefinite integral of the derivative of a general function?

```
> int(diff(f(x),x),x);
```

$$f(x)$$

And, a bit more difficult (although immediate) calculation: $\int (f^2(x))' \cdot f(x)\ dx$

```
> int(f(x)*diff(f(x)^2,x),x);
```

$$\frac{2\,f(x)^3}{3}$$

Example 12 What is the Jacobian matrix of three general functions in two variables?

```
> with(VectorCalculus):
> Jacobian([f(x, y), g(x, y), h(x, y)], [x, y]);
```

$$A := \begin{pmatrix} \frac{\partial}{\partial x} f(x,y) & \frac{\partial}{\partial y} f(x,y) \\ \frac{\partial}{\partial x} g(x,y) & \frac{\partial}{\partial y} g(x,y) \\ \frac{\partial}{\partial x} h(x,y) & \frac{\partial}{\partial y} h(x,y) \end{pmatrix}.$$

2.4 Theorem Proving

There are two main lines regarding research in automated theorem proving:

- logical deduction from the axioms (using deduction rules), applicable to all fields of mathematics [22], and
- automated theorem proving in geometry (using algebraic methods, like Gröbner bases or Wu's method [23–29]), also denoted mechanical theorem proving. CAS are the key tool for this line of research.

The main problem is the readability of the proofs produced by the two lines of research aforementioned and their lack of elegance (synthetic proofs and proofs based on brilliant ideas can't be developed these ways).

Let us compare two well-known proofs (classic and mechanical) of the concurrency the three segment bisectors of triangle.

Example 13 The classic proof of the concurrency of the three segment bisectors of triangle ABC is brief and elegant: the points of the segment bisector of $\overline{AB}$ are at the same distance from A and from B, the points of the segment bisector of $\overline{BC}$ and at the same distance from B and from C, and therefore their intersection point is at the same distance from A and from C, so it lies on the segment bisector of $\overline{AC}$.

Example 14 Meanwhile, the (short) mechanical proof checks that the linear system corresponding to the equations of the three segment bisectors obtained allocating a general triangle conveniently is compatible. For instance, we can do it the following way in *Maple*: we can start defining a function that returns the equation of the perpendicular bisector to a segment (given its endpoints):

```
> restart;    #To reset Maple
> perp_bis:=(m1,m2,n1,n2)->
              (m1-n1)*(x-((m1+n1)/2))=(n2-m2)*(y-((m2+n2)/2)):
```

and without loss of generality we can assume that the vertices of the triangle are $A = (0, 0)$, $B = (b1, 0)$ and $C = (c1, c2)$ (Fig. 4):

```
> a1:=0: a2:=0: b2:=0:
> perp_bis(a1,a2,b1,b2),perp_bis(b1,b2,c1,c2),perp_bis(c1,c2,a1,a2);
```

$$-b1\left(x-\frac{b1}{2}\right)=0,\ (b1-c1)\left(x-\frac{b1}{2}-\frac{c1}{2}\right)=c2\left(y-\frac{c2}{2}\right),\ c1\left(x-\frac{c1}{2}\right)=-c2\left(y-\frac{c2}{2}\right)$$

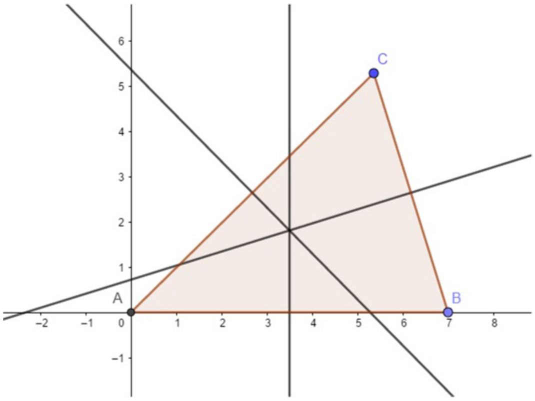


Fig. 4 Concurrency of the three segment bisectors of a general triangle

```
> solve({%},{x,y});
```

$$\left\{x = \frac{b1}{2}, y = -\frac{b1c1 - c1^2 - c2^2}{2c2}\right\}$$

(in *Maple* `%` represents the last output). That is, the three perpendicular bisectors of the triangle are concurrent and we have obtained the coordinates of the intersection point (the only degenerate case is $c2 = 0$, that corresponds to the degeneracy of the triangle into a segment—the three vertices are aligned in such case).

There are very interesting steps in this line of research that start from a dynamic geometry system, like in the earlier [30], even automatizing the whole process [31]. But automated theorem proving is not the key topic of this talk.

3 Revisiting the Second and Third Classes

From the CAS point of view (unlike what happens from the teacher's point of view) there is no difference between the questions in the second and third classes considered in Sect. 2:

1. symbolic computations (simplifications, expansions, concatenation of algebraic calculations, etc.),
2. theoretical questions regarding mathematical formulae,

as they can be completed by the CAS just performing symbolic computations. But they are of an abstraction level[2] higher than those of the first class (as they deal with non-assigned variables—variables in the mathematical sense, not in the computational sense).

But the CAS goes really further. Let us focus on the examples in Sect. 5:

- Example 8 (determinant of a 3 × 3 matrix) is interesting, as the possibility to handle matrices which elements are algebraic expressions allow to generate a well known formula.
- The case of Example 9 (function composition) is specially interesting, as it shows that the CAS reaches an even higher abstraction level: it deals not only with with unassigned variables but with general functions (functions not declared), themselves depending on non-assigned variables.
- Examples 10 and 11 are similar to Example 9, but incorporating differentiation and integration.
- Finally, Example 12 (Jacobian matrix) applies all the possibilities just mentioned in order to obtain the Jacobian matrix of three functions in two variables.

Possibly many mathematics teachers and many CAS users are unaware of some of these possibilities of (some) CAS and their potential impact on teaching and assessment.

[2] This term is used in a similar context in [32] regarding the abstraction level of the tool (the CAS), unlike other authors that focus on the level of abstraction of the student when using the tool, like [33].

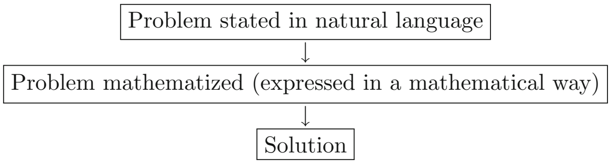

Problem stated in natural language
Problem mathematized (expressed in a mathematical way)
Solution

Fig. 5 Problem solving process

4 Conclusions

Summarising, CAS have reached an unprecedented abstraction level with many possibilities in different fields.

As a consequence, assessment in maths education must clarify the technological tools that can be used at classes and exams (see for example the detailed [34]), because a device running a CAS (computer, smartphone, calculator) can be used to solve symbolic problems and also as a technological live cheat sheet (in what a classic maths teacher would consider “theoretical questions”, where a classic calculator is useless).

In our opinion CAS should be used the same way as classic calculators: they don’t substitute, but complement mental as well as pen and pencil calculations when the calculations are long or tedious.

Figure 5 summarizes the steps of the problem solving process. Moving from the first step to the second requires human thought (at least until AI develops much more). Meanwhile, passing from the second to the third one can be carried out by the technological tool. Consider, for instance understanding a high school problem stated in natural language about two trains moving in opposite directions on a double track line at constant speed or a similar problem about two taps filling a swimming pool at constant flow. Once mathematized, they can be easily solved by a CAS.

From our point of view the solution is not to deny the existence of tools with CAS capabilities (a kind of “ostrich like attitude”). What has to be transmitted to the students are the underlying ideas (and also subsequent algorithms and procedures), but not algorithms or procedures isolated from their origin. And students should be trained in using the technological tools for appropriate uses (CAS in this case). If you think it twice it is the same attitude usually accepted nowadays regarding arithmetic and classic calculators.

Acknowledgements Partially supported by the research project PGC2018-096509-B-I00 (Government of Spain). I would like to thank the anonymous reviewers for their most valuable comments, that have helped to improve this paper. I would also like to thank Mag. Josef Böhm (founder of the *International DERIVE User Group* [7]) for his comments and suggestions after the presentation of this talk at the *Computer Algebra in Education* session at the *26th International Conference on Applications of Computer Algebra (ACA2021)*, that have also helped to improve this paper.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Munat Hervás, I.: Problemas de Selectividad de Matemáticas II. Comunidad de Madrid. Por examen y resueltos (2000-2021). –, Madrid (2021)
2. TI-92 series, https://en.wikipedia.org/wiki/TI-92_series. Last accessed 30 Oct (2021)
3. Camacho Machín, M., González Martín, A.S.: Una aproximación a los problemas de optimización en libros de Bachillerato y su resolución con la TI-92. Aula **10**, 137–152 (1998)
4. Beaudin, M.: Supporting engineering mathematics with the TI-92. The International Journal of Computer Algebra in Mathematics Education **7**(2), 143–155 (2000)
5. Wester, M.J. (ed.): Computer Algebra Systems: A Practical Guide. John Wiley & Sons, Chichester, UK (1999)

6. Böhm, J.: Dimensional analysis with DERIVE. Mathematics and Computers in Simulation **45**(1–2), 197–205 (1998)
7. International DERIVE User Group + CAS-TI., http://www.austromath.at/dug/index.htm. Last accessed 30 Oct (2021)
8. Roanes-Lozano, E., Galán-García, J.L., Solano-Macías, C.: Some Reflections About the Success and Impact of the Computer Algebra System DERIVE with a 10-Year Time Perspective. Mathematics in Computer Science **13**(3), 417–431 (2019)
9. GeoGebra. GeoGebra for Teaching and Learning Math, https://www.geogebra.org/. Last accessed 30 Oct (2021)
10. Bernardin, L., Chin, P., DeMarco, P., Geddes, K.O., Hare, D.E.G., Heal, K.M., Labahn, G., May, J.P., McCarron, J., Monagan, M.B., Ohashi, D., Vorkoetter, S.M.: Maple Programming Guide. Waterloo Maple Inc., Waterloo, Canada, Maplesoft (2020)
11. Corless, R.: Essential Maple. An Introduction for Scientific Programmers. Springer, New York (1995)
12. Heck, A.: Introduction to Maple. Springer, New York (2003)
13. Maplesoft: Maple User Manual. Maplesoft, Waterloo Maple Inc., Waterloo, Canada (2021)
14. Roanes-Macías, E., Roanes-Lozano, E.: Cálculos Matemáticos por Ordenador con Maple V.5. Editorial Rubiños-1890, Madrid (1999)
15. García López, A., García Mazario, F., Villa Cuencala, A.: Could it be possible to replace DERIVE with MAXIMA? The International Journal for Technology in Mathematics Education **18**(3), 137–142 (2011)
16. Karjanto, N., Husain, H.S.: Not Another Computer Algebra System: Highlighting wxMaxima in Calculus. Mathematics **9**, 1317 (2021)
17. Roanes Lozano, E., Cabezas Corchero, J., Ortega Pulido, P., Roanes Macías, Romo Santos, M. C., Vara Ganuza, M. V.: Un proyecto para el uso del sistema de cómputo algebraico Maxima en Educación Secundaria. Boletín de la Sociedad Puig Adam de Profesores de matemáticas **76**, 68–77 (2007)
18. Roanes Lozano, E., Cabezas Corchero, J., Vara Ganuza, M.V., Roanes Macías, E., Ortega Pulido, P., Romo Santos, M.C.: A Proposal for Filing the Gap Between the Knowledge of a CAS and its Application in the Classroom. In: Wester, M.J., Beaudin, M. (eds.) Computer Algebra in Education, pp. 7–17. Aulonna Press, El Dorado, White Lake, MI, USA (2008)
19. Jeffrey, D. J., Rich, A. D.: Recent Developments in the RUBI Integration Project. In: Heras, J., Romero, A. (eds.), EACA 2016. XV Encuentro Álgebra Computacional y Aplicaciones, pp. 109–112. Universidad de La Rioja, Logroño (2016)
20. Galán-García, J.L., Aguilera-Venegas, G., Rodríguez-Cielos, P., Padilla-Domínguez, Y., Galán-García, M.Á.: SFOPDES: A Stepwise First Order Partial Differential Equations Solver with a Computer Algebra System. Computers & Mathematics with Applications **78**(9), 3152–3164 (2019)
21. Koepf, W.: Numeric versus symbolic computation. In: Gilbert, R.P., Kajiwara, J., Xu, Y.S. (eds.) Recent developments in complex analysis and computer algebra, pp. 179–203. Kluwer, London-Dordrecht-Boston (1999)
22. Aransay Azofra, J.M., Domínguez Pérez, C.: Demostración asistida por ordenador. La Gaceta de la RSME **15**(1), 75–104 (2012)
23. Buchberger, B.: Bruno Buchberger's PhD Thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. Journal of Symbolic Computation **41**((3–4)), 475–511 (2006)
24. Buchberger, B.: Applications of Gröbner Bases in Non-Linear Computational Geometry. In: Rice, J.R. (ed.) Mathematical Aspects of Scientific Software, pp. 59–87. Springer, New York (1987)
25. Recio, T., Vélez, M.P.: Automatic Discovery of Theorems in Elementary Geometry. Journal of Automated Reasoning **23**, 63–82 (1999)
26. Roanes-Macías, E., Roanes-Lozano, E., Fernandez-Biarge, J.: Obtaining a 3D extension of Pascal theorem for non-degenerated quadrics and its complete configuration with the aid of a computer algebra system. Revista de la Real Academia de Ciencias. Serie A. Matemáticas **103**(1), 93–109 (2009)
27. Botana, F., Hohenwarter, M., Janičić, P., Kovács, Z., Petrović, I., Recio, T., Weitzhofer, S.: Automated Theorem Proving in GeoGebra: Current Achievements. Journal of Automated Reasoning **55**, 39–59 (2015)
28. Wu, W.-T.: On the Decision Problem and the Mechanization of Theorem Proving in Elementary Geometry. Scientia Sinica **21**, 157–179 (1978)
29. Chou, S.C.: Mechanical Geometry Theorem Proving. D. Reidel Publishing Company, Dordrecht (1988)
30. Roanes-Lozano, E., Roanes-Macias, E., Villar-Mena, M.: A bridge between dynamic geometry and computer algebra. Mathematical and Computer Modelling **37**(9–10), 1005–1028 (2003)
31. Kovács, Z., Recio, T., Tabera, L.F., Vélez, M.P.: Dealing with Degeneracies in Automated Theorem Proving in Geometry. Mathematics **9**, 1964 (2021)
32. Watt, S. M.: On the Future of Computer Algebra Systems at the Threshold of 2010. In: Proc. Joint Conference of ASCM 2009 and MACIS 2009: Asian Symposium of Computer Mathematics and Mathematical Aspects of Computer and Information Sciences, (MACIS 2009), December 14-17 2009, Fukuoka, Japan, pp. 422–430. COE Lecture Note Vol. 22, Kyushu University, Fukuoka, Japan (2009)
33. Kent, P.: Expressiveness and abstraction with computer algebra software. In: Lagrange, J.-B., Lenne, D. (eds.) Calcul formel et apprentissage des mathématiques: actes des journées d'étude: Environnements Informatiques de Calcul Symbolique et Apprentissage des Mathématiques, 15–16 juin 2000, Renne, pp. 49–58. Institut National de Recherche Pédagogique, Paris (2001)
34. University of Arizona Math Dept. 1.5. Which Calculators Can We Use in UA Math Classes or when taking the ALEKS PPL Assessment?, https://ua-math-dept.helpspot.com/academics/index.php?pg=kb.page&id=206 Last accessed 30 Oct (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.