Industrial Applications of Computer Algebra: Climbing Up a Mountain, Going Down a Hill

Laureano Gonzalez-Vega and Tomas Recio

Abstract. In this paper we present some personal experiences with Computer Algebra applications to industrial problems. In many cases the involved Computer Algebra problems seem as challenging as climbing up a difficult peak. Then one finds out that the trail leads up to a quite rugged hill ... This point of view will be illustrated with "real" examples coming from robot kinematics and path planning, parametric CAD and shape design in automotive industry.

1. Considerations about the Nature of Computer Algebra

Computer Algebra bounces back and forth, from Computer Science to Mathematics. As a scientific discipline Computer Algebra arises, in the 50's, as a response to the difficulties posed by the first attempts to implement computer programs for analytical integration or differentiation (see the historical chapter in [10]). These programs were, in many cases, application oriented. The addressed difficulties were of various sorts, ranging from very mathematical —for instance, purely algebraic— to very computational, such as the need of a specific memory management policy.

No matter how far from the implementation step could happen to be our research in Computer Algebra, we should not forget this: the success of so many symbolic computation programs that have been (and that are being) used to solve so many different problems, is the ultimate responsible for the existence, today, of Computer Algebra as a scientific discipline. There is not a "purely mathematical" Computer Algebra subfield. There are different levels of applicability, i.e. of proximity to an externally given goal ... Thus, its achievement should measure the success of the application. This applies, in particular, to Computer Algebra industrial applications.

Whether we agree totally, or just in part, or whether we disagree openly with the above statements, we will probably agree that the interest of Computer Algebra industrial applications should be primarily evaluated from the industrial partners' side of the picture. But they tend to be, for good or for bad, rather silent. They rarely spend time and energy writing papers, and they usually do not attend Computer Algebra conferences. A negative consequence of this state of things is that we do not have an objective way to provide a solid overview of

industrial applications, and that we do not readily learn about succesful/failed cases. A positive consequence is that we can claim whatever we aim to claim, without much precaution.

It was not always like this. In the earlier times, a handful of succesful Computer Algebra stories made their way to the communication media. We can recall articles published in the Scientific News [18], New York Times [23], Nature [22], Scientific American [17], etc. One could argue that it was, perhaps, just because of the scientific novelty of symbolic computation —performing in a few minutes or seconds some computations that required, previously, a titanic effort for humans— and that some of these were news on non-industrial aspects of Computer Algebra. Anyway, it has been quite long ago since Computer Algebra methods and tools have been echoed for the last time in the scientific news for the general public (with few exceptions such as [7]). And we all realize that, today, in many scientific fields, sucess stories with relevant consequences for industry, make their way, rather easily, into the newspapers. We could ask, then, what is the matter today with Computer Algebra industrial applications?

The goal of this paper is to present some personal considerations around this question. Next section will describe some of the external circumstances that face academic/industrial cooperation. In section 3, we will summarily describe the more intrinsic cooperation problems posed by the in days gone by promising field of Computer Algebra applications in Robotics: certainly, we were trying to climb up a high mountain ... Section 4 is devoted to present the more modest aims of an ongoing cooperation project which is effectively changing the practice of a concrete enterprise: we can say that we are now exploring the challenging top of a rugged hill.

2. Working with Industrial Partners

Working with industrial partners is quite unconfortable for us, academics[1]. They obstinately care about solving a problem, but just in most cases, or at least in some cases, or even in "this" particular case, instead of caring about solving it in general. Some times the solution they search for is conceptually rather simple, but tiresome to execute in practice. Other times they do not care about the problem they just have posed, if they see that you do not progress fast enough to solve it; and then they merely switch their mathematical model to a different and simpler one, asking you to forget about the interesting question you have just started to think about ... It is difficult to get a paper properly done under these changing conditions, it seems impossible to bind in a Ph.D. thesis based on such "rush hour" solutions ... As a result, only very few and very obstinate scholars persist working in this near-zero atmosphere (i.e. extremely poor in academic oxygen: publications and dissertations).

[1]The abstract of this paper explicitly states that we are telling here some **personal** experiences: the reader must have this in mind, when reading what follows.

Even if we assume this environement and if we try our best to achieve "costumer satisfaction", we can run into other kind of problems. The following example comes from the *FRISCO* ("A FRamework for Integrated Symbolic/Numeric COmputation", ESPRIT LTR Program, 1996–1999) project. This project arised as an industrial application of another ESPRIT project (the *POSSO* project: "Polynomial System Solving"). The FRISCO consortium was leaded by a software company, NAG Ltd (Oxford, UK) and included the universities of Cantabria (Spain), Pisa (Italy), Rennes I and the research institute INRIA at Sophia-Antipolis (both in France).

After the POSSO experience, shared by some of the FRISCO teams, the FRISCO partners carefully started addressing the problem of industrial cooperation through an opinion poll to R&D responsibles in different industries. The conclusions of this survey, published under the name "The Needs of Industry on Polynomial System Solving" (see [11]), were discussed at a Workshop (Barcelona, fall 96). The questionnaire attempted to identify symbolic polynomial system solving problems appearing in each of the addressed industries, and it included:

1. Information about the corporation itself.
2. A specific item on: how did the corresponding polynomial system appear in the industrial context?
3. What kind of system was it (number of variables, coefficient type, etc.)?
4. What kind of solutions were looked for?
5. When had it to be solved (real time, off-line)?

More than 60 enterprises were contacted, in several european countries, representing very different fields of economic activity: EDF, Alcatel, FIAT, REIS, Daimler Benz, VW, PEGOP, TECNATOM, CANDEMAT, SIMULOG, CASA, APIA XXI, LABEIN, ... From this large group, only about a dozen replied (mostly Spanish, also EDF, CCETT, PEGOP, REIS ROBOTICS, etc.). It must be said that the coordinator of this activity was one of the coauthors of this paper, and this could explain the greater success with Spanish firms. Finally, just half dozen of these industries made the effort to attend the Workshop (invited by FRISCO). A first conclusion we can extract from these data is that it is also very hard for engineers working in industries, to distract some time and energy from daily occupations to participate in long-term cooperation projects with Academia.

On the positive side, the information collected by the enquire was quite interesting. The polynomial system was, in most cases, generated by means of standard Symbolic/Numeric software. Usually the system had an equal number of equations and of unknowns (from 16 to several thousands). The system was quite sparse, but this fact was due, perhaps, to the way it was generated. Its degree was quite low, two or three, in general. The usual coefficients were real numbers, a few cases had coefficients in $\mathbb{Z}_3$, or they were exact rational numbers. If the system contained parameters, they were first specialized to numerical values, then the system was solved. The sought solutions were always real numbers, with a 10^{-7} accuracy requirement, both for solutions found on or off-line. If problems arised in the solving

process (for Newton-like methods) the mathematical model was then simplified. Usually, parametric solutions had not been regarded (but they would have been wellcome, in principle).

We can also extract some general conclusions. One, is that it is hard to communicate with industries, because of the lack of common language and because of the divergent short-term interests from both sides. Moreover, technically speaking, we can say that cooperation with industry requires,

1. to develop Symbolic/Numerical integrated packages,
2. to adapt symbolic solvers to the kind of systems appearing in industry (which are not of general type), and
3. to disseminate the existence of symbolic solvers that deal with parameters.

Moreover we observed that many industrially relevant situations are not so (mathematically speaking) challenging. Sometimes, to explore the capabilities of Computer Algebra into a particular problem involves changing the mathematical model, not simply solving it in the given model. Finally, we can remark that interesting contexts for cooperation seem to arise mainly with those industries which are in the academic vicinity.

3. Robot Kinematics and Motion Planning

There are, of course, other, more intrinsic, reasons, that explain the difficulties of Computer Algebra for industrial cooperation.

Few subjects have attracted more interest as an application domain for Computer Algebra than robot kinematics and motion planning problems. We can say that Robotics entered in Computer Algebra through the seminal paper [21]. The motion planning problems were, subsequently, reduced to a kind of quantifier elimination problem, in as many variables as the robot's degrees of freedom. Luckily, Computer Algebra experts have had successfully developed and implemented a Quantifier Elimination algorithm by that time (see [4] and [2]). In the next years, a number of prestigious scientists contributed to this problem. For instance J. Canny obtained the Young Scientist Presidential Award in 1987 for his dissertation ([3]) on this topic. R. Alami and J.-P. Laumond went even further (see [1]), providing a general solution for the task planning problem (i.e. algorithmically describing how a robot should deal with a given task, including non self-movable objects and fixed obstacles).

Yet, we can apply to this happy picture some statements taken from the Foreword to the chapter "Motion Planning", in the book "Autonomous Robot Vehicles" (see [5])[2]:

> "*Although the theoretical approach has the* **drawback that the algorithms produced tend to be far too complicated to implement**, *the heuristic approach suffers from the problem that there*

[2]The underlining is ours.

> *is no guarantee that the methods will work in all cases. However, the hope is that the theoretical results can reveal the fundamental structure of the problem, thus providing a framework to assist in making decisions as to what compromises should be made when developing a heuristic approach.*"

And, more clearly, in the chapter on "Prospects" from his book [14], Latombe states:

> "*Although* **these results have still made few inroads in industrial applications**, *this should change soon. Several graphic simulators already include collision checking capabilities and it will not take long before they also offer path planning tools.*"

The obvious reason is that the Quantifier Elimination algorithms of general purpose offered by Computer Algebra are still too far from being able to deal with real size motion problems, for the incredible amounts of time and space they currently require. One can hope that "this should change soon", of course ..., although complexity results [6] apparently say that it could take some time to become reality.

A similar analysis applies to the long standing problem of solving robot kinematic equations (those giving the position and orientation of each robot arm for a desired position and orientation of its hand) [16]. As McCarthy [15] mentions, the interesting case is that in which the equations include parameters, each corresponding to the possible geometric data for the different parts of a robot:

> "*At the core of both the analysis of existing machines and the design of new ones is the solution of sets of nonlinear algebraic equations with parameters obtained form either the dimensions of the machine or the designed set of output positions* .... *the designer is less often interested in a particular solution than in ranges of solutions and their relation to the prescribed features of the design.*"

We can track this problem for the 6R manipulator (with six degrees of freedom) from 1969 till the final computational solution in 1989 [20]. Essentially, the problem deals with solving symbolically a collection of non-linear equations and Computer Algebra provides with many different algorithms to cope with this situation. Yet, after 20 years of research in this issue, one of the persons who had contributed more to its solution, Professor Roth, from U. Stanford, confessed (in a personal communication) that he did not recall any immediate industrial reaction to his discovery. The reason, again, is that the obtained characteristic polynomials are by far too large to be handled with current computers.

Summarizing, we can say that the field who has attracted more interest as a challenging application for Computer Algebra in the past decades, seems to have been made little industrially oriented progress in such a high-tech field as Robotics. For a more down to earth and promising approach see [13].

4. CAD and Shape Design in Automotive Industry

To be less negative, we will like to present some ongoing work on an issue of modest theoretical challenge that is producing a concrete impact in the output of the industrial partner. The activity of the project "*Integrating new algebraic-numerical techniques in Computer Aided Geometric Design (CAGD): Developing Problem Solving Environments and Implementation into an industrial CAD/CAM framework*" funded by the spanish Ministery of Education (through European Union support for regional development) is directed to the resolution, in a more efficient way, of the mathematical problems arising when manipulating curves and surfaces into the CSIS software. This software is developped and maintained by the company CANDEMAT S.A. in order to be used for design purposes, production control and quality verification of the final product (shape design and cast construction in automotive industry).

In order to check the possibility of using Computer Algebra techniques to improve the CSIS software, firstly three concrete problems were isolated:

- the computation of the implicit equation of a surface in $\mathbb{R}^3$ presented by a rational parametrization,
- the problem of conversion between the VDA and IGES formats (see 4.2), and
- the tracing, algebraically guided, of algebraic curves implicitely defined.

The reasons justifying the choice of the first problem rely on the fact that, when available, the implicit equation of a parametric surface is very useful to deal with point-surface position problems, surface intersections, sectioning, trimmed surfaces and sculptured solids, etc. For the second problem, to remark that the manipulation of trimmed surfaces (surfaces with holes) involves the exact topology computation of several real algebraic plane curves defined in the parameter space.

4.1. Generic implicititation applied to sectioning and offsetting

Two main difficulties were encountered when trying to use inside CSIS the usual elimination technics to deal with implicitation problems: first, it is usually a very costly algebraic operation and, second, the coefficients of the parametrization are usually floating-point numbers.

The second difficulty was overcome by taking into account that, in general, a concrete object to model (and then to construct) is made by several hundreds (or thousands) of small patches, all of them sharing the same algebraic structure. For such an object a database has been constructed containing the implicit equation of every class of patches appearing in its definition. This database must also contains the inversion formulae (giving the parameters in terms of the cartesian coordinates) and must be pruned to avoid specialization problems. Moreover the database for a specific object is kept into a bigger and general database for a further use.

In the particular case of the objects provided by CANDEMAT S.A., the implicitation process was made by using Sylvester resultants, Grobner Basis computations and ad-hoc techniques for specific cases. Since the implicitation procedure

was a preprocessing step, no time is spent when the CAD/CAM user is working. Moreover the solution to the floating-point coefficients problem is rather simple under this approach, requiring just the specialization in the precomputed implicit equation.

Example 4.1. *In the database, the implicit equation of the parametric patch*

$$x = x_{00}\frac{t_2 - t}{t_2 - t_1} + x_{01}\frac{t - t_1}{t_2 - t_1}$$

$$y = y_{00}\frac{s_2 - s}{s_2 - s_1} + y_{11}\frac{s - s_1}{s_2 - s_1}$$

$$z = \left(z_{00}\frac{s_2 - s}{s_2 - s_1} + z_{10}\frac{s - s_1}{s_2 - s_1}\right)\frac{t_2 - t}{t_2 - t_1} + \left(z_{10}\frac{s_2 - s}{s_2 - s_1} + z_{11}\frac{s - s_1}{s_2 - s_1}\right)\frac{t - t_1}{t_2 - t_1}$$

appears as:

$$\begin{aligned}&(z_{00} - 2z_{10} + z_{11})xy + (y_{00}z_{10} + y_{11}z_{10} - y_{00}z_{11} - y_{11}z_{00})x\\ &+ (x_{00}z_{10} + x_{01}z_{10} - x_{00}z_{11} - x_{01}z_{00})y + (x_{00}y_{11} - x_{00}y_{00} + x_{01}y_{00} - x_{01}y_{11})z\\ &+ x_{01}y_{11}z_{00} - x_{00}y_{11}z_{10} + x_{00}y_{00}z_{11} - x_{01}y_{00}z_{10}\,.\end{aligned}$$

The sectioning by the plane $X = k$ of a B-spline proceeds, with this new strategy, by performing the following steps:

- First, we consider all the implicit equations $H_i(X, Y, Z)$ of the different patches integrating the considered B-spline surface.
- Second, the starting points in the parametric domain are determined and these are used to determine the interesting part of every curve
$$H_i(k, Y, Z) = 0$$
through a convenient discretisation.
- The obtained points are interpolated by a spline curve which represents the desired section.

We should mention, finally, that for specific objects, the time for sectioning, by using this strategy, has already been divided by a factor of 3 when compared with the previous approach followed in CSIS.

As a future work we regard the resolution of some problems arising in the implicitation problem:

- Some algebraic structures arising in the database construction are very complicated and the implicit equation has not been generated. Namely:
$$x = \frac{f_1(s,t)}{f_0(s,t)}, \quad y = \frac{f_2(s,t)}{f_0(s,t)}, \quad z = \frac{f_3(s,t)}{f_0(s,t)}$$
with ($i \in \{0, 1, 2, 3\}$):
$$\begin{aligned}f_i = {} & s^3(\alpha_3^{(i)}t^2 + \beta_3^{(i)}t + \gamma_3^{(i)}) + s^2(\alpha_2^{(i)}t^2 + \beta_2^{(i)}t + \gamma_2^{(i)})\\ & + s(\alpha_1^{(i)}t^2 + \beta_1^{(i)}t + \gamma_1^{(i)}) + \alpha_0^{(i)}t^2 + \beta_0^{(i)}t + \gamma_0^{(i)}\,.\end{aligned}$$

- We will like to deal in advance with the specialization problems: up to this moment these are detected "a posteriori" when substituting the parametrization into the candidate to be the implicit equation.

If the sectioning by using the implicitation procedure has been a first successful application, this will be much more important when dealing with the offsetting by taking into account that the offset of a B-spline is no longer a B-spline or, in general, a parametric surface (but an algebraic surface) and thus the use of implicit equations is a direct method not currently used in any CAD/CAM software.

4.2. Format conversion: from rational to polynomial parameterizations

`VGA` and `IGES` are two of the formats most commonly used in CAD/CAM systems to represent and deal with the data required to describe and communicate the essential engineering characteristics of physical objects such as manufactured products. The `VDA` (Verband der Automobilundustrie) format was created in 1982 by the `VDA` Committee founded mainly by several German automobile and automotive supply companies. The `IGES` (Initial Graphics Exchange Specification) was created by the IGES/PDES Organisation (USA).

The main difference between these two formats lies in the fact that `VDA` only accepts surfaces defined by polynomial parameterizations while `IGES` accepts both rational and polynomial. This is a very important problem in many companies in the automobile industry since they get the information in the `IGES` format but the specific CAD/CAM software they use only allows the use of the `VDA` format. The only way of solving this problem up to now has been by means of a method based upon an uniform subdivision of the parameter domain plus a polynomial interpolation procedure whose efficiency depends strongly on the degree of the polynomials to appear in the required polynomial parameterization (see [19]). The Computer Algebra behind this problem is still not completely well understood and new algorithms for solving this problem have been developed and initially checked by using new Hermite-Birkhoff interpolation schemes for multivariate polynomials (see [9]). These techniques also allow the reduction of the degree of the considered surface which produce a simplification of the implicitation procedure described in 4.1 since the degree of the parametric equations are smaller (see [8]).

4.3. Algebraically guided tracing of implicitly defined algebraic curves and surfaces

Many important problems in Computer Aided Geometric Design are reduced to the computation of the graph of a planar algebraic curve implicitely presented. For example if we want to section the surface

$$x = \frac{X(s,t)}{W(s,t)}, \quad y = \frac{Y(s,t)}{W(s,t)}, \quad z = \frac{Z(s,t)}{W(s,t)}; \qquad s,t \in [0,1]$$

with respect to the plane $X = x_0$ then we have two possibilities: either we draw "into the square unit" $[0,1] \times [0,1]$ the planar algebraic curve defined by

$$x_0 W(s,t) - X(s,t) = 0$$

and then this picture is lifted to the considered surface or, if the implicit equation $H(x, y, z)$ of the considered surface is available, the lifting procedure can be avoided by merely computing the graph of the planar curve $H(x_0, y, z) = 0$ as it was shown in section 4.1.

The problem of computing the graph (even topologically) of a planar algebraic curve defined implicitly has received special attention from Computer Algebra since it has been responsible for many advances regarding sub-resultants, real root counting, infinitesimal computations, etc. These algorithms have been successfully implemented in several Computer Algebra Systems (AXIOM and Maple) and now we are involved in the process of integrating the algorithms into the CSIS software (but some new practical problems have arisen since CSIS has just moved to Visual Basic under Windows 98 ...).

The main reason for including these algorithms in CSIS, apart from its use in the sectioning procedure, is found into the manipulation of trimmed surfaces: parametric surfaces with holes defined as parametric curves in the parametric domain of the surface. For example, the sectioning of this kind of surfaces is not currently available in the CSIS software.

5. Conclusions

It is clear at this moment that the practical algebraic resolution of nonlinear systems with thousands of equations and unknowns is far from the capabilities of Computer Algebra but probably, in many cases, Computer Algebra could probably help to reformulate the mathematical problem in such a way that the system to solve gets much smaller (and simpler to solve).

For example the following areas have already been identified as cornerstones when trying to apply algebraic techniques into an industrial practice:

- To make explicit the capabilities of Computer Algebra when dealing with the resolution of polynomial systems involving parameters: in many cases final users did not imagine that such a possibility existed and that is already available for some non trivial problems.
- To determine some specific needs in the CAD area with respect to polynomial system solving.
- To create/investigate links between Computer Algebra software and nonlinear optimization packages, or between Computer Algebra techniques and the theory of linear systems and control.
- To develop Computer Algebra facilities to deal with the nonlinear systems of equations which are produced by discretization schemes or finite elements.
- To establish a good collection of test suites, not only with a statement of the problem and results, but also containing calling sequences allowing the user to experiment with every system on a WWW server providing

access to Computer Algebra software. A test suites collection is already available from

`http://www-sop.inria.fr/saga/POL/`

and a very interesting collection of highly specialised symbolic software is available at the following web addresses:

1. The PoSSo Library:
   `http://janet.dm.unipi.it/posso_demo.html`
2. FGb+RS Servers:
   `http://posso.lip6.fr/~jcf/` and `https://calfor.lip6.fr/`
3. Gb+RealSolving through MuPAD:
   `http://www.loria.fr/~rouillie/software.html`
4. The ALP Library:
   `http://www-sop.inria.fr/safir/WHOSWHO/Bernard.Mourrain/ALP/`.

A very easy to start introduction to the use of almost all the libraries before mentioned can be found in [12].

Acknowledgements

First author acknowledges partial support by the grant DGESIC PB 98–0713–C02–02 (Ministerio de Educación y Cultura).

Second author acknowledges partial support by the grant DGESIC PB 98–0756–C02–02 (Ministerio de Educación y Cultura).

References

[1] R. Alami and J. P. Laumond, *A geometrical approach to planning manipulation tasks in robotics,* First Canadian Conference on Computational Geometry (Montreal, 1989).

[2] B. Buchberger, G. Collins, M. Encarnación, H. Hong, J. Johnson, W. Krandick, R. Loos and A. Neubacher, *A SACLIB Primer,* Tech. Rep. 92–34, RISC-Linz, Johannes Kepler University (Linz, Austria), **1992**.

[3] J. F. Canny, *The complexity of robot motion planning,* ACM Doctoral Dissertation Series, MIT Press, Cambridge Mass., **1988**.

[4] G. E. Collins, *Quantifier elimination for real closed fields by Cylindrical Algebraic Decomposition,* Lecture Notes in Computer Science (Second GI Conference on Automata Theory and Formal Languages), **33** (1975), 134–183.

[5] I. J. Cox and G. T. Wilfong, *Motion Planning,* Autonomous Robot Vehicles (I. J. Cox and G. T. Wilfong eds), Springer, **1990**.

[6] J. H. Davenport and J. Heintz, *Real Quantifier Elimination is Doubly Exponential,* Journal of Symbolic Computation, **5** (1988), 29–36.

[7] D. Duval, *Calcul symbolique: automatisation en cours,* La Recherche, **291** (1996), 64–71.

[8] J. Espinola, L. Gonzalez-Vega and I. Necula, *Generic implicitation of low degree rational surfaces,* Preprint available at `http://www.frisco.matesco.unican.es` (1999).

[9] J. Espinola, L. Gonzalez-Vega and I. Necula, *Algebraic approximation in CAGD,* Preprint available at `http://www.frisco.matesco.unican.es` (2000).

[10] K. Geddes, S. R. Czapor and G. Labhan, *Algorithms for computer algebra,* Kluwer Academic Publishers, Boston, **1992**.

[11] L. Gonzalez-Vega, *The Needs of Industry for Polynomial System Solving,* The SAC Newsletter, **3** (1998), 21–46.

[12] L. Gonzalez-Vega, *FRISCO Software Overview,* Preprint available at `http://www.frisco.matesco.unican.es` (1999).

[13] P. Kovacs, *Rechnergesützte Symbolische Roboterkinematik,* Vieweg Verlag, **1993**.

[14] J. C. Latombe, *Robot Motion Planning,* The Kluwer International Series Series in Engineering and Computer Sceince, Kluwer Academic Publishers, **1991**.

[15] J. M. McCarthy, *Kinematics of Robot Manipulators,* International Journal of Robotics Research, **5(2)** (1986).

[16] R. P. Paul, *Robot manipulators: Mathematics, Programming and Control,* The MIT Press Series in Artificial Intelligence, **1981**.

[17] R. Pavelle, M. Rothstein and J. Fitch, *Algebra por ordenador,* Scientific American (Spanish edition), **1981**.

[18] L. Steen, *Computer calculus,* Science News, **119** (1981), 250–251.

[19] N. Patrikalakis, *Approximate Conversion of Rational B-spline Patches,* Computer Aided Geometric Design, **6** (1989), 189–204.

[20] M. Raghavan and B. Roth, *Kinematic analysis of the 6R manipulator of general geometry.* Proceedings of the International Symposium on Robotics Research (Tokio), 314–320 (1989).

[21] J. T. Schwartz and M. Sharir, *On the "piano movers" problem. II. General techniques for computing topological properties of real algebraic manifolds,* Advances in Applied Mathematics, **4(3)** (1983), 298–351.

[22] *Algebra Made Mechanical,* Nature, **290** (1981), 198–200.

[23] *Liberating the prose of math from its grammar,* New York Times, july 19 (1988), 21.

Departamento de Matemáticas, Estadística y Computación,
Facultad de Ciencias,
Universidad de Cantabria,
Santander, Spain
E-mail address: `gvega@matesco.unican.es`
E-mail address: `recio@matesco.unican.es`