

Computer Algebra: Implications and Perspectives

Benno Fuchssteiner¹

1 Introduction

In this paper I would like to address the issue of computer algebra's role in the future development of mathematics. To begin, consider the following two quotations:

Daß die niedrigste aller Tätigkeiten die arithmetische sei, wird dadurch belegt, daß sie die einzige ist, welche auch durch eine Maschine ausgeführt werden kann.

Schopenhauer, Parerga und Paralipomena II, §356, 1851²

Although to many people the electronic computer has come to symbolize the importance of mathematics in the modern World, few mathematicians are closely acquainted with the machine.

S. M. Ulam, Scientific American, September 1964³

Both highlight the ambivalent relationship between the computer and mathematics, both contain truths and falsehoods, but from similar premises arrive at opposite conclusions. The philosopher sees, if for the wrong reasons, the implications of machines for mathematics while the mathematician seems to overlook these, emphasizing an entirely different influence. Both assessments are open to question. However it seems to me that they sum up the attitudes of, on the one hand, the public at large, and on the

¹A lecture presented on the annual meeting of the DMV 1991 in Bielefeld.

²Here, Schopenhauer means not just arithmetic but all of mathematics. This is clear from what he goes on to say: *Nun läuft aber alle Analysis finitorum et infinitorum im Grunde doch auf Rechenerei zurück. Danach bemesse man den "Mathematischen Tief-sinn" über welchen schon Lichtenberg sich lustig macht.* Bense [1, p. 135] also uses this quotation in his "Antimathematika". Schopenhauer's opinion of mathematics is not quite so simplistic as this extract would lead one to believe. Compare for example paragraphs 25 and 256 in Parerga II.

³At that time the number of computers installed world-wide was ca. 16,500. It is therefore astonishing that with the dramatic rise in the importance of the "machine" that of mathematicians, at least in the eyes of the public, has undergone just as dramatic a reversal.

other, the mathematical community, to the subject of "computers and mathematics".

I will deal with this question under the following headings:

- What is Computer Algebra?
- The State of the Art
- Two Examples
- The Implications of Computer Algebra for Teaching and Research: a Challenge to Mathematics?
- A Glimpse of the Future: My preferred system for the year 2000.

2 What is Computer Algebra?

I myself would like an answer! Consulting "Duden Informatik, ein Sachlexikon für Studium und Praxis 1988" we find precisely .. nothing! No entry for this or related topics. A second attempt with the "Dictionary of Computing" from Oxford University Press 1990, likewise ends in failure. Neither "computer algebra" nor any technical synonym is to be found. So we have our first proposition:

Computer Algebra is of no Importance!

Convincing evidence for this thesis is available.

- In a report to the NSF in 1990 the observation was made that: *One person who regularly gives talks about symbolic computation to the science and engineering communities estimates that 80% to 90% of the attendees at his talks have never heard of symbolic computation. Furthermore those who do use symbolic computation systems have difficulties of using them effectively.* [5, p. 35]

And again,

Most people doing research in mathematics, science and engineering are unfamiliar with even

the best-known symbolic computation systems. [5, p. 31]

- The system *Reduce*, one of the classic, easily available computer algebra systems and at that time probably the most widely used, between 1985 and 1988 was installed at only 1130 sites world-wide. A miserable 30% of them were in Europe, while Japan alone had 40% [5, p. 16]. In considering these figures it should also be borne in mind that *Reduce* was one of the few systems that could be run on the most popular PC.⁴

If computer algebra is unimportant in the rest of the world, in Germany it seems to be regarded as completely insignificant. Here are two instances:⁵

- To contrast with the poor support for computer algebra in the USA, the examples of Austria, Canada, France, Italy, Japan and the UK are given in [5, p. 31] as countries where computer algebra *is* being developed. Germany is not included in this list.
- At the DMV meeting in Berlin a computer algebra section was set up for the first time. The first official business was to move to a larger lecture hall because of overcrowding. However, three years on, this section seems to have been lost in transit, with few voices raised in protest.

Such instances show that in Germany computer algebra is held in low esteem. This is not to overlook the fact that the sparing use of computer algebra and the small number of installed systems in Germany (as in Europe) is in part due to the badly equipped state of most mathematical institutes. Computer investment programs have passed them over, leaving scarcely a trace. In general, the average grocery is better equipped with high-tech devices than a medium-sized mathematics faculty. If you consider this to be an exaggeration, here is a case in point (from a neighboring country):

In 1990 the research funding body of a (western) central European country presented a project, involving the development of a complicated algorithmic procedure, to an expert in the field for his evaluation. He was asked to pay particular attention to the question of whether the request for computing equipment, consisting of a (much too small) machine with a 386 processor and a 60 MB disk, was not overdoing it a bit;

⁴This situation has changed with the introduction of new systems and their clever and aggressive marketing campaigns.

⁵These examples say nothing about the standard of theoretical work done in Germany on this subject, which is indeed of value.

*this because the entire faculty of which the applicant was a member had two AT's (with quite inadequate peripherals) at its disposal.*⁶

Again, is computer algebra really insignificant, or even disreputable since it may have a bad name coming from its frequent association with those buzzwords "artificial intelligence". Actually I hope to convince you of the contrary.

To return to the original question of what computer algebra is all about. Here's a little story. In early 1991 several clever people got together in Oberwolfach to write a paper on the meaning of computer algebra. The first short section was to contain a definition of what computer algebra actually is. All of the first morning was taken up with a concerted effort to resolve this question, and after coming up with a working definition the group decided to spend no more time on it but to move on to weightier matters. However, on the second morning one of the participants cunningly managed, in passing, to reintroduce the question, thus ensuring that the entire second day was spent in fruitless discussion. On the third morning things didn't go any better and the group broke up with only an unsatisfactory definition and some fragments of the "weightier matters" to show for three days' work. However, a feature common to many of the proposed definitions was that they began with the words

Computer algebra is a **method for...**

I believe that in this "method for" we have the essential (if not quite definitive) part of computer algebra. It is a *method* - one *does* it, one doesn't waste time trying to define it.

Therefore I would like to close the discussion of what computer algebra is, even though I have been unable to give a generally acceptable definition, and turn instead to examine its current capabilities.

3 The State of the Art

Computer algebra systems permit interactive, formal computation with mathematical objects; mathematical objects of a complexity like those arising in the

⁶Mathematicians are in the habit of pointing with some pride to their lack of resources. In my opinion this is where modesty becomes irresponsibility. How should the leaders of tomorrow's technology be educated if mathematicians, those at the center of all scientific progress, overlook and are overlooked by today's technical advances? One must admit also that most mathematical institutes are not up to building and maintaining powerful computer networks.

daily work of an engineer or physicist. In contrast to purely numerical calculation however, computer algebra manipulates signs and symbols. That computers can work in this way should not be surprising: indeed it is real numbers which require a complicated representation, whereas a symbol like π is a very simple object. That computer algebra itself should be possible has long been surmised. For example Ada Augusta, Countess of Lovelace (who appears to have inherited more of her mother's sharp analytical mind than of her father's romantic nature) wrote in 1842 (cited in [5, p. 10]) that

Many persons ... imagine that the business of the engine [Babbage's engine] is to give results in numerical notation, the nature of its processes must consequently be arithmetical and numerical rather than algebraical and analytical. This is an error. The engine can arrange and combine its numerical quantities exactly as if they were letters or other general symbols; and in fact it might bring out its results in algebraical notation were provisions made accordingly.

Computer algebra⁷ behaves as a high-level declarative language⁸ in processing sophisticated mathematical statements. In combination with a multitude of elegant and ingenious algorithms, these systems become excellent tools for *rapid prototyping* thus speeding up greatly the twin engines of research and development.

As I've said, one shouldn't define computer algebra. To get better acquainted with it, a look at its present capacities is more useful. The following few, and extremely simple examples may serve to give a vague idea of the powers of such systems⁹:

- The calculation of the 800-digit number $341!$ requires just that literal command and close to zero seconds computer time.
- The decomposition of the 34-digit number $(31! + 1)$ into its four prime factors needs the command `ifactor(31! + 1)` and less than two seconds.

- On inputting

$$dsolve(diff(\varphi(t), t, t) + sin(\varphi(t)) = 0, \varphi(t))$$

after about a second the explicit solution for a pendulum with non-linear force (the physical

⁷For an overview of existing computer algebra systems see [6] which gives a description of all the systems introduced at the 1991 DMV meeting.

⁸Declarative programming languages specify the goal to be attained and not the means of getting there. Imperative languages take the opposite approach.

⁹Computed using MAPLE on a SUN 3/50

pendulum) is returned. If one has a little bit of patience, then one can wait for a printout of the graph.

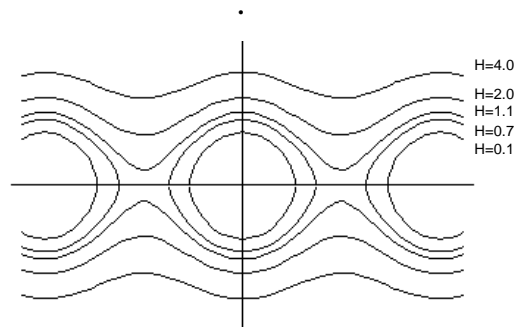


Fig. 1: Orbits of the pendulum in phase space

- A reasonable program for the calculation of the Fibonacci sequence, or any other recursively defined sequence, occupies no more than one line.
- The exact computation of the first fifty Taylor coefficients of

$$\sin(3x^2 + e^{\sqrt{1+6x^2}})$$

requires a half-line command and close to zero seconds.

- Given a half-line command, the integral of

$$\frac{\sin(x) - 2 \cos(x)}{\cos(x) - 3 \sin(x)}$$

is returned in about two seconds.

Bear in mind that these are simple, indeed trivial examples. As such they do have a role to play at an introductory stage. The mathematician's interest lies with problems of a more general and theoretical nature¹⁰, but also these are facilitated by use of computer algebra.

We have seen mathematical propositions, apparently complicated for the outsider, converted to forms capable of solution by machine. Does this mean that we are forced into agreeing with Schopenhauer's pointed remarks? On the contrary: the foregoing simple examples can serve to demonstrate the basic strength and vitality of mathematics. After all it was mathematical analysis which remolded these concepts into a

¹⁰Since in this paper I'm primarily concerned with the implications for teaching in other areas, I prefer to stick to examples of a simple analytic nature.

machine-tractable form. Mathematics itself removed the mathematical mystery from these problems.

It might be argued from this that mathematics will one day simplify itself out of existence,¹¹ but this is to take a superficial view. The automation of routine problems is an ever-present process in the evolution of mathematics, one which leaves us free to break new ground, to tackle new and exciting ideas. This process not always follows the same pace, sometimes there are leaps and dramatic changes. For example when differential calculus was invented, solving problems which required deep thoughts from Archimedes up to Fermat, became so simple that almost no mathematical understanding was required any more. Such leaps, making mathematics more simple, are highly desired because after some time lag they always lead to a new blossoming of our science.

It is not yet clear whether with the emergence of computer algebra we are now facing such a new giant leap, one opinion on that: *For many fields computer algebra significantly expanded the frontiers* [7, p. 103].

If it is true that in this century mathematical research has been characterized by *"ein weg vom Quantitativen und hin zum Qualitativen"* [10, p. 1], we now have the opportunity to include the "quantitative" in our efforts to reach a new qualitative understanding in many areas.

It is now possible effortlessly to handle mathematical problems and computations of a new order of magnitude. This ability has given rise to the discipline of *Experimental Mathematics*, new not for the concept but for the scale of what it sets out for itself: mathematicians have always experimented, but it was never like this. Michael Hazewinkel in [4] and [3] presents an impressive variety of the things experimental mathematics can do.

Even if one does not agree with all that has been said, one must admit that computer algebra has implications for how mathematical formulae and techniques will be used in future applications of our science:

*These programs do in a few brief minutes virtually all mathematics that most engineers and scientists know*¹².

The gains provided by speed of calculation alone should not be underestimated, this because:

Mathematics is a basis of technological progress, and technological progress is a key for international competitiveness. Automating an important part of the mathematical problem-solving progress is a key technology for a nation that wishes to control, structure

¹¹Public opinion often follows this line of thought.

¹²Cited in [5] from L. A. Steen, Computer Calculus, SIAM News, p. 198-200 (1981).

and accelerate technological progress. The automation of the solution of mathematical problems is a powerful lever by which human productivity and expertise can be amplified many times [5],

and:

Considering the human life span, a reduction in the time needed for an operation from 20 years to 20 hours is a qualitative gain[7, p. 104].

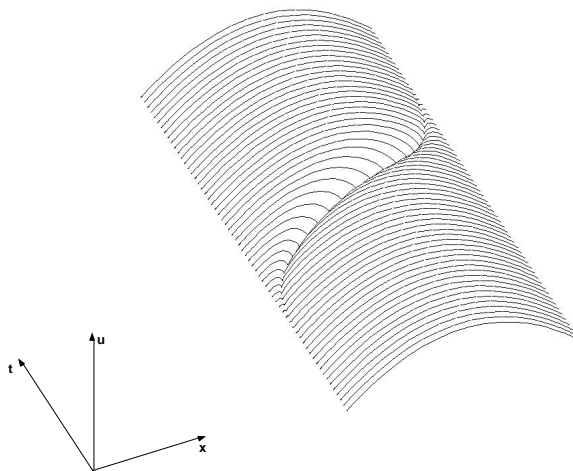
However I do not wish to end this section on a note of self-satisfaction. We must not allow good and useful mathematics to be overwhelmed by the 'son et lumiere' of a new approach. As it often happens with progress, valuable thoughts and methods unfortunately will pass into oblivion and a new superficial attitude will take place instead.¹³ On the other hand, one has to admit that computer algebra still has to go a long way until it is a satisfactory appliance for the use by the sophisticated mathematician, helping him to open up and to speed the development of new avenues of research.

4 Two Examples

I would like to give two examples from my own area of mathematical interest to illustrate the power and the limitations of present-day computer algebra.

4.1 The Power

Here's a solution

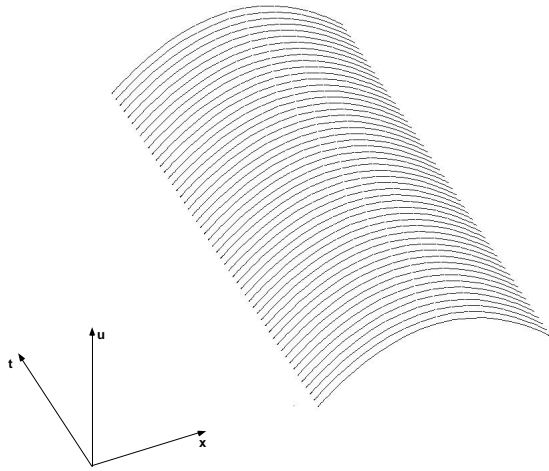


to the so-called Harry Dym equation

$$u_t = u^3 u_{xxx} .$$

¹³This is so in all areas of progress. Think of the possibilities presented to the art of building design by a material such as reinforced steel concrete, and then of the use to which it was actually put.

The solution itself is of interest to physicists because of the characteristic furrow. When we come to consider a second solution this peculiarity no longer appears.

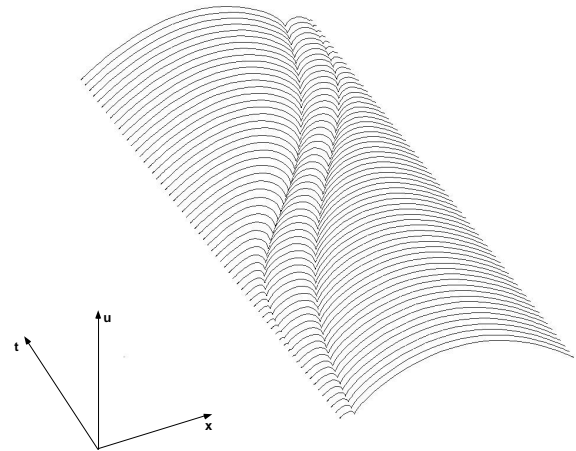


By comparing the plots it becomes apparent that the furrow arises and decays at least exponentially fast. Consequently very small perturbations, practically infinitesimal, of the initial condition can cause the characteristic phenomenon to appear. Because of this instability it is impossible to find a numerical solution, even though the system is not chaotic. Therefore one can study the problem only through the examination of explicit, particular solutions. In principle, there is a very simple recipe for finding explicit solutions:

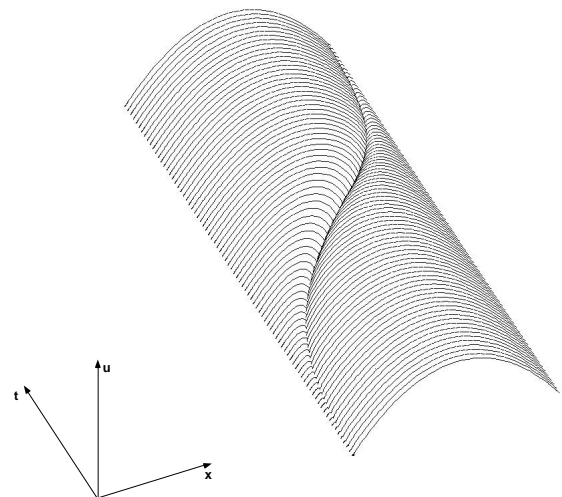
1. Find a reflection-free potential of the classical one-dimensional Schrödinger operator.
2. Determine explicitly the square of the eigenvector of the operator with this potential.
3. Use the function so determined as an initial condition for a given non-linear integrable infinite-dimensional system (the so-called singularity-manifold equation of the Korteweg de Vries equation).
4. Subject the solution obtained to a kind of hodograph transformation i.e. a transformation which exchanges dependent and independent variables.
5. Then as a result one has the solution as pictured.

The difficulty here is the explicit solution of the eigenvalue problem for the Schrödinger operator; but we can bring a little theory to bear. Because the given

potential is reflection-free one can find a potential-dependent integro-differential operator which maps the given potential to the squares of its eigenvectors. The order of this operator increases linearly with the number of eigenvalues, which gives rise to a slight drawback to this method of solution: though it is surprisingly easy to describe, the solution provided by this method is 50 A4 pages long. Working with this by hand would cost a creative mathematician a great deal of his valuable time, whereas a computer algebra system would quickly and happily digest these fifty pages, delivering a result in a matter of seconds. But this was a modest example. One can be interested in additional features, for example increasing the number of furrows



or the order of the original non-linear differential equation. Consider for example an explicit solution



to the so-called Kawamoto equation

$$u_t = 10u^4 u_{xx} u_{xxx} + 5u^4 u_x u_{xxxx} + u^5 u_{xxxxx}$$

Using similar methods (which must first be justified

mathematically) one obtains a formula several hundreds of pages long.

These problems far exceed the limits of what is possible using pen-and-paper methods. However, at these computational extremes we are getting not so much numerical as *structural* insight into the nature of the given non-linear system. We find ourselves doing *pure mathematics*.

4.2 The Limitations

For some time we at Paderborn have been interested in the question of whether non-linear partial differential equations like the Korteweg de Vries equation

$$u_t = u_{xxx} + 6uu_x ,$$

the Harry Dym equation or the Kawamoto equation are linearizable, via some kind of magical manifold transformation. A deciding factor in this question is the existence of a simple co- and simple contravariant tensor Φ , which, with respect to all vector fields a and b , satisfies the following identity (hereditary condition [2])

$$\frac{\partial}{\partial \varepsilon} \{ \Phi(u + \varepsilon \Phi(u)a)b - \Phi(u)\Phi(u + \varepsilon a)b \}_{\varepsilon=0} = \text{symmetric}(a, b).$$

In the case of the Korteweg de Vries equation, this condition leads to the well-known operator

$$\Phi_{KdV} = D^2 + 2DuD^{-1} + 2Du ,$$

which is understood as a map on the tangent bundle. Here D denotes differentiation with respect to x and D^{-1} integration from $-\infty$ to x .

The identity above has a deep geometric meaning and can be rewritten in a differential-geometric invariant form. The existence of such an operator ensures the complete integrability of the associated equation.

Consider now the Kawamoto equation. With a modest amount of structure-mathematics one arrives at the guess that for this equation the following operator Φ has the required property.

$$\Phi(u) = u^2 DJ(\varphi)\Theta(\varphi)D^{-1}u^{-2} .$$

Here,

$$\varphi = uu_{xx} - \frac{1}{2}(u_x)^2$$

and J and Θ are defined via

$$\Theta(\varphi) = uDuDu\varphi_x + 3u\varphi\varphi_x$$

$$J(\varphi) = uDuDu\varphi_x + 3(u\varphi\varphi_x + uD\varphi^2) + 2[uDuD\varphi D^{-1}\varphi u^{-1} + D^{-1}\varphi D\varphi Du\varphi] + 8[\varphi^2 D^{-1}\varphi u^{-1} + D^{-1}\varphi^3 u^{-1}] .$$

So, only the modest identity defining the hereditary condition has to be checked. In this checking process

however, the fly in the ointment is that the given operator behaves as an integro-differential operator, so that only by skillful partial integration can we get that terms do cancel out. Therefore we cannot hope to avoid first solving a normal form problem for integration, within the rule for partial integration. This being accomplished, one must then expand the previous expression so that the corresponding summands may be converted to the normal form.

Unfortunately, the appearance of variational derivative leads to a first estimate of about four billion for the number of terms in the integro-differential operator. An eighteen kilometer high stack of A4 sheets would be needed to write this out in full - that's a one thousand cubic meter mountain of paper. If a degree were to be awarded for this work, it would surely be a posthumous one.

Undeterred, with our previous success in mind, we turn to the computer. Here we are dealing with a datum many gigabytes in size, and the reaction of conventional computer algebra systems to something this big is to move it for further treatment in the main memory. This would require a workstation with the capacity of several Crays.

Although the structural nature of the problem is not too difficult, conventional computer algebra systems cannot handle such problems efficiently.

5 A Challenge to Mathematicians

In the medium term computer algebra will change utterly the way we and others do mathematics. Within the next ten years lasting change will come about in the approach of mathematicians to teaching.

Computer algebra systems which in the recent past could be run only on a mainframe and which today run on a middle-sized workstation, in five years will be found in the notebook of every scientist and engineer.

Very many mathematicians earn their research time and resources by teaching scientists and engineers: for this reason, if research freedom for us is to be protected, there will have to be changes to teaching practices, this because many of those applying mathematics and who have to deal frequently with complicated mathematical formulae will try to escape their difficulties using computer algebra. This is a convenient escape for a number of reasons: computer literacy is already widespread (for reasons nothing to do with computer algebra); computers can lend a patina of authority to otherwise trivial work; and as com-

puters get more and more user-friendly there will be less and less reason to fear them.¹⁴ The users of future computer algebra systems will need barely any mathematical instruction in their use.

If mathematicians do not equip themselves with these essential tools then tomorrow's scientists will get instruction in their use from those who are already none too open to the subtleties of mathematics. A new level of intellectual superficiality, along with restrictions on research freedoms would be the result. The present marked tendency for the reputation of mathematicians to decrease with the increased use of mathematics would be reinforced.

Certainly new curricula must be found: the question is, how much are we prepared to suffer in the search? The danger is that our initial difficulties with these developments indicate a complacency or worse, the first signs of a serious downturn in the fortunes of mathematical research. All the more unfortunate that many young talents would be lost to the subject, having found it in a stagnant condition.

However I do not believe that we should prepare ourselves for the future merely from a sense of panic or helplessness, rather from a sense of inquiry, an interest in new ideas, a critical open-mindedness. There is certainly a great deal that is satisfying in this work. Besides, there is a duty to be discharged, that of not allowing content to be overwhelmed by outward appearance and of preserving an awareness that the use of these new technologies must be meaningful. It should be a great concern for us to foster this awareness in our own area, the discipline which lies at the heart of all technical development.

Mathematicians are called upon to bend to a different perspective the naive belief in the omnipotence of computers, through intelligent criticism allied to an informed openness. Even if it is not possible to alter the popular conception, at least that of technicians and scientists.

That the future has already begun can be seen by looking at the case of our neighbor, Austria, where the Ministry of Education has bought the computer algebra system *Derive* for all secondary schools in Austria [6, p. 21]. From the Autumn of 1991 *Derive* (whose marketing slogan is "*2000 Jahre Mathematik auf einem Disk*") is to be the standard mathematical tool for all schools.¹⁵ In Germany too, the situation is beginning to change.

¹⁴Just how much good interfaces matter to the acceptance and usage of computer algebra systems is amply demonstrated by the case of *Mathematica*.

¹⁵Slogans of this kind are one reason for the critical attitude of many mathematicians.

Significant changes will come about in many areas of research, in particular because

- the introduction of declarative, high-level languages, syntactically very close to everyday mathematical formulae, standard access to mathematical expertise. Rapid Prototyping (absolutely not quick-and-dirty) will make routine the most complex mathematical processes for mathematicians, engineers and scientists.
- these systems will have almost unrestricted access to good (as well as bad) algorithms.
- the freedom from the need to perform tedious calculations means that problems at present inaccessible will be laid open to attack.

I've already mentioned briefly the role of *experimental mathematics*. It is obvious that easily performed mathematical experiments can provide us with new views and intuitions and make possible progress in quite new directions. A multitude of easy-to-manage examples will lead to new discoveries and new structures.

Everyone knows that the borders between disciplines such as algebra and analysis are fluid and we use them only as a first crude classification. Nevertheless these very delineations have given rise to much inertia over the last fifty years, from time to time obstructing our view of the wider mathematical landscape. These borders will be weakened and the lodestone of mathematics exposed once more. What is quite clear is that new methods and tools will alter our outlook on mathematics. Computer algebra ensures that what was seen yesterday as analysis will today be seen as algebra. Algebra and discrete mathematics, because they are closer to the algorithmic nature of mathematical reality, will flourish.

Our attitude towards what we consider *simple* mathematical facts will change. Mathematicians know of course that *simplicity* is a structural or aesthetic property and not wholly a question of size. Nevertheless in day-to-day work we are tempted to take the complexity of the descriptions we use as a measure of underlying complexity. When an interface separates us from the basic mathematical formulae our view will clear dramatically. Think of the solution to the partial differential equation problem we saw earlier. Just examining the images will give one the insight that one has to do with simple structural properties. That this would not be discovered is only the fault of our view obscured by the jumble of necessary formulae, which, in time to come, will not trouble our aesthetically corrected view.

Which challenges do we face with regard to science? A story may help to clarify things. Some time ago I heard a lecture given by a mathematician from the research department of a big computer firm. He expressed the opinion that the barycenter of mathematics in the future should move to the area of *scientific computing*. And he did prophesy, that if we could not manage the necessary change of emphasis mathematics would come to an end.

So will there be a time when every mathematician is happy to play around in computer algebra or scientific computing or algorithm theory or complexity theory or discrete mathematics? No, I don't believe that. I am even convinced that some of these areas will break away from the mainstream of mathematics, each developing its own impetus and importance. And that might not be a bad thing! At the periphery of an active field, there is always a deal of turbulence: what in my youth was quite new to mathematicians is commonplace in today's engineering faculties. I believe that the fractioning of new disciplines from the core is evidence for the vitality of mathematics. What is more, since we cannot impede this breakaway we should at least take care that that these new areas are not leaving from mathematics as neglected children. Mutual concern, critical and open-minded, is necessary if both old and new are to prosper.

To repeat: I don't believe that it is sensible for us to switch our mathematical interests because of technical progress. However a questioning attitude should accompany such progress.

In this case, what are the challenges we must face? Certainly, a drastic improvement in the efficiency of computer algebra can only be achieved with the help of mathematicians. Mathematicians are called upon to

- develop further the mathematical background, libraries and such, to these systems.
- consider a standardization mathematical syntax and formulae. The question is how to write unambiguous, syntactically correct mathematics so that a computer can work on the meaning of what's been written, rather than on just the symbols. What's needed is an abstract, mathematically well-developed high-level programming language which is close to our everyday work with mathematical formulae.¹⁶

The creation of a mathematical high-level language is important for two reasons. Firstly to enable communications between different computer algebra system

¹⁶This idea is not new. It has been around for years, and was one of the inspirations for the *Euromath project*.

(and word-processing systems), and secondly to handle new developments in *Electronic Publishing*. In 20 years time a mathematical or technical article will have very little to do with the paper it's printed on¹⁷. Behind its surface there will be a hierarchical structure which enables its reader to take its formulae, make some conjectures, verify them and paste the results in his own publication.¹⁸ A consequence of this scenario is that the difference between computer algebra systems and word processing systems will be lost; indications that this is taking place are already to hand. It now depends not so much on hardware developments but more on advances in mathematics. To point out another challenge, computer algebra systems are *algebra* systems and not *analysis* systems, let alone *mathematics* systems: for example up to now it has been the case that inequalities could not be dealt with in an intelligent way. I am convinced that in the next ten years we will see a new discipline emerge, the aim of which is to understand the way in which *we* think about such objects. Success here will mean that we can forget about computer algebra systems and move on to *computer mathematics systems*.

I have not said if I believe these changes to be desirable. However I am sure that despite their many drawbacks these changes will afford us the opportunity to make great strides in mathematical research, freed from the shackles of our (present) toils and troubles.

6 A Glimpse of the Future

Now for my second proposition:

Computer algebra is important, but still in an unsatisfactory state.

With daily use one comes to appreciate the inadequacies of computer algebra:

- The design, specifications and implementation of computer algebra are subject to commercial interest. European mathematicians had rather little influence on the development of general-purpose systems.
- New developments are difficult to put in place, however often the contrary is held to be true.

¹⁷However, for reading the printed page will predominate for some time to come.

¹⁸Certainly I am not saying that this is a desirable state of affairs.

- The source code of these systems is not fully accessible. This makes unreasonable demands of the serious user since many fundamental algorithms are implemented, at least partly, in the system core. Knowing the source code of an essential tool on which our own work is based is not only a matter of our scientific ethos but also a matter of practicability.

Had these lines been written by my colleague Neubüser from Aachen, you would be reading a diatribe against the commercialization of computer algebra and of mathematics.

However I do recognize that it takes heavy investment of people and resources to develop these systems. For this reason I cannot endorse fully his opinion, though the problem of the commercialization of the intellectual property on which we will be dependent should not be overlooked. More systems *should* come from the non-commercial research sector. This assumes however that those who work there make great efforts to develop the very necessary expertise in the areas of interfaces and documentation, to name but two. Because of the mutual interdependence between commercial developers and the scientific community it would be a wise policy to grant the non-commercial scientific community unrestricted access to all electronic tools they need in areas like mathematics. The benefit would go both ways: critical assessment and exchange is essential to stimulate new ideas.

In the area of computer algebra present levels of cooperation between commercial developers and scientific community are pitiful, but much more critical is the inadequate capacity of computer algebra systems.

Existing systems represent wonderful tools which permit the efficient handling of data structures up to a megabyte in size, or a little larger. They ease the daily grind of the mathematician. However the real point is not to simplify present methods but to extend into new orders of magnitude in calculation, the gigabyte realm and beyond. Solving new practical problems demands this, and existing systems are insufficient to that task.

I am sure that they will appear toy-like when measured against those of the future. That this view is not entirely wrong is demonstrated by our concrete work in the area of simple nonlinear systems, where algebraic data in the region of several gigabytes big is the norm. I believe that we are not alone in this experience:

- The popular belief that the performance of computers would improve so rapidly that considerations of storage space and running time would

no longer be relevant is false, fundamentally so. For real problems, just as for mathematical problems, economic use of storage capacity and run time will play a crucial role. As always, the battle between technical advances and people's greed to abuse these will be lost for the side of technical progress. Technical progress always is too slow as soon as its potential is realized.

- Even with the introduction of good computer algebra systems we are still some way from solving satisfactorily real mathematical problems. A lot of effort will have to be invested in the development of more powerful software. In this respect we are only at the beginning.
- Most computer algebra systems are geared towards a rather simple-minded use. These days one is readily impressed when a computer does in a trice what would take a professional two days to complete but, as I've said, our notion of what is simple will change drastically. What's needed are "heavy-duty" systems with small but efficient kernels accessing a moderate number of libraries, and with basic routines and elementary operators realized at processor level.
- Continued development of parallel computers and suitable programming environments is the order of the day. Here, computer algebra is an excellent example, where efficient use of parallel computing would bring about great advances.

6.1 My Preferred System for the year 2000

From this scenario follows that the system of the future should fit the following description:

- It is a general-purpose system, consisting of reusable program modules. The source codes are open to all.
- The system permits parallel computation according to a user defined profile. It can work in parallel automatically when dealing with large algebraic objects. The parallel communication structure is independent of the hardware in use.
- Standard routines are available with which the system can plan a parallel approach to a sequential problem defined by the user. In this planning, not only the syntactic but also the logical structure of the problem is taken into account.

- The system has compilers and interpreters using a flexible programming language of which the structure is based on a consensus in the mathematical community. The system will have cross compilers to and from all relevant computer algebra systems.
 - The system provides links both to machine-independent user interfaces and to word-processing systems which work on the mathematical syntax of formulae. The differences between computer algebra systems and word-processing systems will have disappeared to a great extent.
 - The system has available to it the basics for automatic program generation. This is not just a question of practicality, but should reflect a similarity to the way mathematicians work with algebra.
 - This can no longer be described as a computer algebra system. It is now a computer *mathematics* system.
 - The system is capable of learning: that is to say it uses its program-generation abilities to redesign its operating strategy as it works through a problem. We are now in the area described by that disreputable phrase *artificial intelligence*.
 - Further development of the system is the job of the mathematical scientific community as a whole.
 - Basic parts of the system are realized at processor level.
- [5] A. C. Hearn, Ann Boyle and B.F. Caviness: *Future Directions for Research in Symbolic Computation*, Siam Reports on Issues in the Mathematical Sciences, Philadelphia, 1990
- [6] *MathPAD: Eine nutzerorientierte Information aus der MathPAD-Gruppe an der Universität-Gesamthochschule Paderborn*, Vol. 1, Heft 3, September 1991
- [7] R. Pavelle, M. Rothstein und J. Fitch: *Computer algebra*, Scientific American, 245, p.102-113, December 1981
- [8] Arthur Schopenhauer: *Sämtliche Werke Band 5: Parerga und Paralipomena*, Reclam, Leipzig,
- [9] S. M. Ulam: *Computers*, Scientific American, in: *Computers and Computation*, Readings from the Scientific American, Reprint of an article from 1964, p.151-161, 1971
- [10] F. Winkler, B. Kutzler and F. Lichtenberger: *Computer-Algebra Systems*, Risc-Linz series no 88-10.0, p.33 pages, 1988

References

- [1] M. Bense: *Konturen einer Geistesgeschichte der Mathematik*, Claassen und Goverts, Hamburg, 1946
- [2] B. Fuchssteiner: *Linear aspects in the theory of Solitons and nonlinear integrable equations*, Journal of the Phys. Soc. Japan, 60, p.1473-1496, 1991
- [3] M. Hazewinkel: *Experimental Mathematics*, Math. Modelling, 6, p.175-211, 1985
- [4] M. Hazewinkel: *Experimental Mathematics*, in: *Mathematics and Computer Science* (J.W. de Bakker, M. Hazewinkel, J.K. Lenstra, eds.) CWI/North Holland Publ., p.193-234, 1986